

Het handboek van KatePart

**Thad McGinnis
Anne-Marie Mahfouf
Anders Lund
T.C. Hollingsworth
Christoph Cullmann
Lauri Watts**

**Vertaling van het handboek: Niels Reedijk
Vertaler/Nalezzer: Tom Albers
Vertaler/Nalezzer: Hannie Lafeber-Dumoleyn**



Het handboek van KatePart

Inhoudsopgave

1	Inleiding	8
2	Enige basisbeginselen	9
2.1	Slepen en loslaten	9
2.2	Sneltoetsen	9
3	Werken met de editor van KatePart	13
3.1	Overzicht	13
3.2	Door de tekst navigeren	14
3.3	Werken met de selectie	14
3.3.1	Blokselectie gebruiken	15
3.3.2	Selectie overschrijven	15
3.3.3	Blijvende selectie	15
3.4	Tekst kopiëren en plakken	16
3.5	Tekst zoeken en vervangen	16
3.5.1	De balken voor zoeken en vervangen	16
3.5.2	Tekst vinden	17
3.5.3	Tekst vervangen	17
3.6	Bladwijzers gebruiken	18
3.7	Automatische regelafbreking	18
3.8	Automatisch inspringen gebruiken	19
3.9	Markeringen voor wijzigingen in regels	20
3.10	De schuifbalk met mini-map	20
3.11	Meerdere cursors	21
3.11.1	Meerdere cursors worden aangemaakt	21
3.11.2	Werken met meerdere cursors	21
4	Het menu Items	22
4.1	Het menu Bestand	22
4.2	Het menu Bewerken	23
4.3	Het menu Selectie	25
4.4	Het menu Beeld	27
4.5	Het menu Ga naar	28
4.6	Het menu Hulpmiddelen	29
4.7	De menu's Instellingen en Help	33

5	Geavanceerde hulpmiddelen voor bewerking	34
5.1	Commentaar toevoegen/verwijderen	34
5.2	De commandoregel in de editor	34
5.2.1	Standaard commandoregel opdrachten	35
5.2.1.1	commando's om de editor in te stellen	35
5.2.1.2	Bewerkcommando's	37
5.2.1.3	Navigatiecommando's	42
5.2.1.4	Opdrachten voor basis functies van de Editor (Deze hangen af van de toepassing waarin de editorcomponent gebruikt wordt)	42
5.3	Code-invouwing gebruiken	43
6	KatePart uitbreiden	44
6.1	Inleiding	44
6.2	Werken met syntaxisaccentuering	44
6.2.1	Overzicht	44
6.2.2	Het systeem voor syntaxisaccentuering van KatePart	45
6.2.2.1	Hoe het werkt	45
6.2.2.2	Regels	46
6.2.2.3	Contextstijlen en sleutelwoorden	46
6.2.2.4	Standaard stijlen	46
6.2.3	Het XML-formaat voor definitie van accentuering	47
6.2.3.1	Overzicht	47
6.2.3.2	De secties in detail	50
6.2.3.3	Beschikbare standaard stijlen	52
6.2.4	Detectieregels accentueren	53
6.2.4.1	De regels in detail	55
6.2.4.2	Tips & trucs	59
6.3	Met kleuren thema's werken	60
6.3.1	Overzicht	60
6.3.2	De kleurenthema's KSyntaxHighlighting	61
6.3.3	De kleurenthema's in JSON-formaat	62
6.3.3.1	Overzicht	62
6.3.3.2	De JSON-structuur	63
6.3.3.3	Hoofdsecties van bestanden voor de JSON-kleurenthema's	63
6.3.3.4	Metagegevens	65
6.3.4	Kleuren in detail	66
6.3.4.1	Kleuren van bewerker	66
6.3.4.2	Standaard tekststijlen	72
6.3.4.3	Aangepaste tekststijlen voor accentuering	75
6.3.5	De GUI voor kleurenthema's	76
6.3.5.1	Een nieuw thema aanmaken	77
6.3.5.2	JSON themabestanden im- of exporteren	77

6.3.5.3	Kleurenthema's bewerken	77
6.3.5.3.1	Kleuren	77
6.3.5.3.2	Standaard tekststijlen	78
6.3.5.3.3	Stijlen voor accentuering van tekst	78
6.3.6	Tips & trucs	78
6.3.6.1	Contrast van tekstkleuren	78
6.3.6.2	Suggesties over consistentie met syntaxisaccentuering	78
6.4	Scripting met JavaScript	79
6.4.1	Inspringscripts	79
6.4.1.1	De kop van het inspringscript	79
6.4.1.2	De broncode van de indenteerder	80
6.4.2	Commandoregelscripts	81
6.4.2.1	De kop van het commandoregelscript	81
6.4.2.2	De broncode van het script	82
6.4.2.2.1	Koppelen van sneltoetsen	83
6.4.3	API voor scripts maken	84
6.4.3.1	Cursors en bereiken	84
6.4.3.1.1	Het prototype van cursor	84
6.4.3.1.2	Het prototype Range	85
6.4.3.2	Globale functies	86
6.4.3.2.1	Lezen & invoegen van bestanden	86
6.4.3.2.2	Debugging	87
6.4.3.2.3	Vertaling	87
6.4.3.3	De API voor weergave	87
6.4.3.4	De API van Document	89
6.4.3.5	De API voor bewerken	95
7	KatePart configureren	96
7.1	De instellingen voor de tekstbewerkercomponent	96
7.1.1	Uiterlijk	96
7.1.1.1	Algemeen	96
7.1.1.2	Randen	98
7.1.2	Kleurenthema's	99
7.1.3	Bewerking	99
7.1.3.1	Algemeen	99
7.1.3.2	Tekstnavigatie	100
7.1.3.3	Inspringmethode	101
7.1.3.4	Automatische aanvulling	102
7.1.3.5	Spellingcontrole	103
7.1.3.6	VI-invoermethode	103
7.1.4	Openen/opslaan	103

7.1.4.1	Algemeen	103
7.1.4.2	Geavanceerd	105
7.1.4.3	Modi & bestandstypen	105
7.2	Configureren met documentvariabelen	107
7.2.1	Hoe KatePart variabelen gebruikt	107
7.2.2	Beschikbare variabelen	108
7.2.3	Uitgebreide opties in <code>.kateconfig</code> bestanden	110
8	Dankbetuigingen en licentie	112
9	De VI-invoermethode	114
9.1	VI-invoermethode	114
9.1.1	Incompatibiliteiten met Vim	114
9.1.2	Modi omschakelen	115
9.1.3	Integratie met wat Kate kan	115
9.1.4	Ondersteunde commando's in normale/visuele modus	116
9.1.5	Ondersteunde verplaatsingen	117
9.1.6	Ondersteunde tekstobjecten	119
9.1.7	Ondersteunde commando's in invoegmodus	119
9.1.8	Het kommatekstobject	120
9.1.9	Ontbrekende functies	120
A	Reguliere expressies	121
A.1	Inleiding	121
A.2	Patronen	122
A.2.1	Escape-teken	122
A.2.2	Tekenklassen en afkortingen	122
A.2.2.1	Tekens met een speciale betekenis in tekenklassen	124
A.2.3	Alternatieven: overeenkomen met 'één van'	124
A.2.4	Subpatronen	124
A.2.4.1	Alternatieven specificeren	124
A.2.4.2	Overeenkomende tekst vangen (achterwaartse verwijzingen) . . .	124
A.2.4.3	Vooruitblik-toekenningen	125
A.2.4.4	Achteruitblik-toekenningen	125
A.2.5	Tekens met een speciale betekenis in patronen	125
A.3	Aantallen	126
A.3.1	Happig	126
A.3.2	In voorbeelden van context	127
A.4	Toekenningen	127
B	Index	129

Samenvatting

KatePart is een editorcomponent vol met functies door Qt™ voor KDE toepassingen.

Hoofdstuk 1

Inleiding

KatePart is een tekstverwerkercomponent gebruikt door vele Qt™ en KDE toepassingen. KatePart is meer dan een tekstverwerker; het is bedoeld als editor voor programmeurs en zou kunnen worden beschouwd als op zijn minst deels een alternatief voor krachtigere editors. Eén van de hoofdfuncties van KatePart is de gekleurde syntaxis, aangepast voor veel verschillende programmeertalen zoals: C/C++, Java™, Python, Perl, Bash, Modula 2, HTML en Ada.

KWrite is een eenvoudige tekstverwerker gebaseerd op KatePart. Het heeft een interface voor een enkel document (SDI) waarmee u één bestand per keer per venster kunt bewerken. Omdat KWrite een erg eenvoudige implementatie is van KatePart, vereist het geen eigen documentatie. Als u weet hoe KWrite te gebruiken, dan kunt u KatePart overal gebruiken!

Hoofdstuk 2

Enige basisbeginselen

KWrite en vele andere gebruikers van KatePart zijn erg eenvoudig te gebruiken. Iedereen die ooit een tekstverwerker heeft gebruikt, zal geen problemen ervaren.

2.1 Slepen en loslaten

KatePart gebruikt het protocol voor slepen en neerzetten van KDE. Bestanden kunnen vanuit het bureaublad, de bestandsbeheerder Dolphin of vanaf een externe FTP-site, geopend in een venster van Dolphin, naar KatePart gesleept worden.

2.2 Sneltoetsen

De meeste sneltoetsen zijn in te stellen via het menu [Instellingen](#). Standaard biedt KatePart de volgende sneltoetsen:

Ins	Wissel tussen de modi Invoegen en Overschrijven. Wanneer de modus Invoegen geactiveerd is, zullen getypte tekens ingevoegd worden op de plaats waar u deze typt. Wanneer de modus Overschrijven gekozen is, zal elk teken dat u typt het teken direct rechts van de tekstcursor overschrijven.
Left	Verplaats de cursor één positie naar links.
Right	Verplaats de cursor één positie naar rechts.
Up	Verplaats de cursor één regel naar boven.
Down	Verplaats de cursor één regel naar beneden.
Ctrl+E	Naar vorige bewerkingslocatie in het document.
Ctrl+Shift+E	Naar volgende bewerkingslocatie in het document.
Alt+Shift+Up	Cursor verplaatsen naar vorig overeenkomend inspringen.
Alt+Shift+Down	Cursor verplaatsen naar vorig overeenkomend inspringen.

Ctrl+6	Ga naar bijhorend haakje.
PgUp	Verplaats de cursor één pagina naar boven.
PgDn	Verplaats de cursor één pagina naar beneden.
Home	Ga naar het begin van de regel.
End	Ga naar het einde van de regel.
Ctrl+Home	Ga naar begin van document.
Ctrl+End	Ga naar einde van document.
Ctrl+Up	Een regel naar boven.
Ctrl+Down	Een regel naar beneden.
Ctrl+Right	Woord naar rechts.
Ctrl+Left	Woord naar links.
Ctrl+Shift+Up	Regels naar boven verplaatsen.
Ctrl+Shift+Down	Regels naar beneden verplaatsen.
Ctrl+.	Geselecteerde regels naar beneden dupliceren.
Ctrl+B	Bladwijzer instellen.
Alt+PgUp	Vorige bladwijzer.
Alt+PgDn	Volgende bladwijzer.
Del	Verwijder het teken rechts van de cursor (of de geselecteerde tekst).
Backspace	Verwijder het teken links van de cursor.
Ctrl+Del	Rechter woord verwijderen.
Ctrl+Backspace	Linker woord verwijderen.
Ctrl+K	Regel verwijderen.
Shift+Enter	Voeg een nieuwe regel in, inclusief tekens uit de huidige zin anders dan letters of cijfers. Het is bijv. nuttig bij het schrijven van commentaar in de code: druk op de sneltoets aan het einde van de regel '//' mijn tekst' en de volgende regel begint al met '// '. U hoeft dus niet meer de commentaar-tekens in te typen aan het begin van elke nieuwe regel met commentaar.
Ctrl+Shift+Enter	Een nieuwe regel aanmaken onder de huidige regel.
Ctrl+Alt+Enter	Een nieuwe regel aanmaken boven de huidige regel.
Shift+Left	Selecteer het teken links van de cursor.
Shift+Right	Selecteer het teken rechts van de cursor.
Ctrl+F	Zoeken.
F3	Volgende zoeken.
Shift+F3	Vorige zoeken.
Ctrl+H	Selectie zoeken.
Ctrl+Shift+H	Selectie achterwaarts zoeken.
Ctrl+Shift+Right	Woord rechts selecteren.
Ctrl+Shift+Left	Woord links selecteren.
Shift+Home	Tot begin van regel selecteren.
Shift+End	Tot einde van regel selecteren.
Shift+Up	Tot vorige regel selecteren.
Shift+Down	Tot volgende regel selecteren.
Ctrl+Shift+6	Selecteren tot bijhorend haakje.

Ctrl+Shift+PgUp	Selecteren tot bovenzijde van weergave.
Ctrl+Shift+PgDn	Selecteren tot onderzijde van weergave.
Shift+PgUp	Een pagina omhoog selecteren.
Shift+PgDn	Pagina omlaag selecteren.
Ctrl+Shift+Home	Tot begin van document selecteren.
Ctrl+Shift+End	Tot einde document selecteren.
Ctrl+Home	Alles selecteren.
Ctrl+Shift+A	Selectie opheffen.
Ctrl+Shift+B	Blokselectiemodus.
Ctrl+C / Ctrl+Ins	Kopieer de geselecteerde tekst naar het klembord.
Ctrl+D	Commentaar.
Ctrl+Shift+D	Commentaar verwijderen.
Ctrl+G	Ga naar regel...
Ctrl+I	Selectie inspringen.
Ctrl+Shift+I	Inspringing van selectie verwijderen.
Ctrl+J	Regels samenvoegen.
Ctrl+P	Afdrukken.
Ctrl+R	Vervangen.
Ctrl+S	Roept de opdracht Opslaan aan.
Ctrl+Shift+S	Opslaan als.
Ctrl+U	Omzetten naar hoofdletters.
Ctrl+Shift+U	Omzetten naar kleine letters.
Ctrl+Alt+U	Elk woord met hoofdletter beginnen.
Ctrl+V / Shift+Ins	Plak de tekst van het klembord in de regel.
Ctrl+X / Shift+Ins	Kopieer de geselecteerde tekst naar het klembord en verwijder deze uit de tekst.
Ctrl+Z	Ongedaan maken.
Ctrl+Shift+Z	Opnieuw.
Ctrl+-	Lettertekens verkleinen.
Ctrl++Ctrl+=	Lettertekens vergroten.
Ctrl+Shift+-	Bovenste niveau van nodes invouwen.
Ctrl+Shift++	Bovenste niveau van nodes uitvouwen.
Ctrl+Space	Code-aanvulling aanroepen.
F5	Herladen.
F6	Pictogramrand tonen/verbergen.
F7	Overschakelen naar commandoregel.
F9	Vouwmarkeringen tonen/verbergen.
F10	Dynamische regelafbreking.
F11	Regelnummering tonen/verbergen.
Ctrl+T	Lettertekens transponeren.
Ctrl+Shift+O	Automatische spellingcontrole.
Ctrl+Shift+V	Naar volgende invoermodus omschakelen.
Ctrl+8	Bovenstaand woord hergebruiken.
Ctrl+9	Onderstaand woord hergebruiken.
Ctrl+Alt+#	Afkortingen uitbreiden.
Ctrl+Alt+Up	Voeg een cursor toe boven de huidige cursor.
Ctrl+Alt+Down	Voeg een cursor toe onder de huidige cursor.
Shift+Alt+I	Een cursor aan het eind van elke regel in de selectie aanmaken.
Alt+J	Volgende overeenkomst van het woord onder de cursor zoeken en het selecteren.

Het handboek van KatePart

Ctrl+Alt+Shift+J	Alle overeenkomsten van het woord onder de cursor zoeken en deze selecteren.
-------------------------	--

Hoofdstuk 3

Werken met de editor van KatePart

Anders Lund
Dominik Haumann
Vertaler/Nalezer: Natalie Koning
Vertaler/Nalezer: Freek de Kruijf

3.1 Overzicht

De editor van KatePart is het editor-gedeelte in het venster van KatePart. Deze editor wordt gedeeld tussen Kate en KWrite en kan ook in Konqueror gebruikt worden om tekstbestanden, vanaf de lokale computer of vanaf het netwerk, weer te geven.

De editor bestaat uit de volgende componenten:

Het bewerkgebied

Hier bevindt zich de tekst van uw document.

De schuifbalken

De schuifbalken tonen de positie van het zichtbare gedeelte in het document en kunnen gebruikt worden om door het document te navigeren. De tekstcursor wordt niet verplaatst als de tekst door middel van de schuifbalken verplaatst wordt.

De schuifbalken worden alleen getoond als dit noodzakelijk is, als dit niet noodzakelijk is blijven ze verborgen.

De pictogramrand

De pictogramrand is een smalle rand aan de linkerkant van de editor, hierop worden naast gemarkeerde regels (regels met een bladwijzer) kleine pictogrammen geplaatst.

U kunt [bladwijzers](#) bij zichtbare regels plaatsen of verwijderen door met de linkermuisknop op de pictogramrand naast die regel te klikken.

De pictogramrand kan in- of uitgeschakeld worden in het menu **Beeld** → **Pictogramrand tonen**.

De regelnummering

De regelnummers van alle zichtbare regels worden in een rand aan de linkerkant van de editor geplaatst.

Het tonen van regelnummers kan in- en uitgeschakeld worden in het menu **Beeld** → **Regelnummering tonen**.

Het invouwpaneel

Het opvouwpaneel stelt u in staat om opvouwbare blokken regels in of uit te vouwen. De berekening van de opvouwbare gebieden wordt gedaan volgens de regels in de syntaxis voor definities voor accentuering in het document.

ANDERE ONDERWERPEN IN DIT HOOFDSTUK:

- [Door de tekst navigeren](#)
- [Werken met de selectie](#)
- [Tekst kopiëren en plakken](#)
- [Tekst zoeken en vervangen](#)
- [Bladwijzers gebruiken](#)
- [Automatisch regels afbreken](#)
- [Automatisch inspringen gebruiken](#)

3.2 Door de tekst navigeren

Door de tekst lopen in KatePart is vergelijkbaar met de meeste grafische tekstbewerkers. U beweegt de cursor door de pijltoetsen en de toetsen **PgUp**, **PgDn**, **Home** en **End** in combinatie met de **Ctrl** and **Shift** om de functie te wijzigen. De **Shift**-toets wordt altijd gebruikt om een selectie te maken, terwijl de **Ctrl**-toets verschillende effecten heeft op verschillende toetsen:

- Voor de toetsen **Up** en **Down** betekent het schuiven van het beeld in plaats van verplaatsen van de cursor.
- Voor de toetsen **Left** en **Right** betekent het sla woorden over in plaats van tekens.
- Voor de toetsen **PgUp** en **PgDn** betekent het ga naar de zichtbare rand van het beeld in plaats van bladeren.
- Voor de toetsen **Home** en **End** betekent het verplaatsen naar het begin of eind van het document in plaats van het begin of eind van de regel.

KatePart biedt u ook een manier om snel naar een overeenkomstige accolade of haakje te gaan: plaats de cursor aan de binnenkant van een haakje of accolade, en druk op **Ctrl+6** om naar het overeenkomstige haakje of de accolade te springen.

U kunt ook [bladwijzers](#) plaatsen om snel naar posities te springen die u zelf definieert.

3.3 Werken met de selectie

Er zijn twee manieren om tekst in KatePart te selecteren: met de muis en met het toetsenbord.

Om met de muis te selecteren houdt u de linkermuisknop ingedrukt terwijl u de muiswijzer van het begin van de te selecteren tekst tot het einde ervan sleept. De tekst wordt geselecteerd terwijl u sleept.

Een dubbelklik op een woord selecteert dat woord.

Een drievoudige klik in een regel selecteert de gehele regel.

Als u de **Shift**-toets ingedrukt houdt terwijl u klikt, wordt tekst op de volgende manier geselecteerd:

- Als er nog niets geselecteerd is, vanaf de positie van de tekstcursor tot de positie van de muiswijzer.
- Als er al tekst geselecteerd is, vanaf het begin van die selectie tot de positie van de muiswijzer.

OPMERKING

Wanneer u tekst selecteert door met de muis te slepen, wordt de geselecteerde tekst naar het klembord gekopieerd. De selectie kan geplakt worden door met de middelste muisknop in de editor of in een andere toepassing te klikken.

Om met het toetsenbord te selecteren houdt u de **Shift**-toets ingedrukt terwijl u de navigatietoetsen (de toetsen, **PgUp**, **PgDn**, **Home** en **End**) gebruikt, eventueel in combinatie met de **Ctrl**-toets om het verplaatsen van de cursor uit te breiden.

Zie ook [Door de tekst navigeren](#) in dit hoofdstuk.

Om de huidige selectie te Kopiëren kunt u het menu **Bewerken** → **Kopiëren** of de sneltoetsen **Ctrl+C** gebruiken.

Om de selectie te Deselecteren kunt u het menu **Bewerken** → **Deselecteren**, of de sneltoetsen **Ctrl+Shift+A** gebruiken, of met de linkermuisknop in de editor klikken.

3.3.1 Blokselectie gebruiken

Wanneer "Blokselectie" ingeschakeld is, kunt u 'verticale selecties' in de tekst maken, dit betekent dat u een aantal kolommen uit meerdere regels kunt selecteren. Dit kan handig zijn als u met tekst werkt die opgemaakt is in kolommen.

Blokselectie kan in-/uitgeschakeld worden door het menu **Bewerken** → **Blokselectiemodus** te gebruiken. De standaard sneltoets is **Ctrl+Shift+B**.

3.3.2 Selectie overschrijven

Als "Selectie overschrijven" ingeschakeld is, wordt de geselecteerde tekst vervangen door de tekst die u intypt of plakt. Als het niet ingeschakeld is, wordt nieuwe tekst op de cursorpositie ingevoegd.

Selectie overschrijven is standaard ingeschakeld.

De instellingen voor deze optie kunt u wijzigen door gebruik van de pagina [Cursor & selectie](#) in het dialoogvenster [Instellingendialoog](#).

3.3.3 Blijvende selectie

Wanneer "Blijvende selectie" ingeschakeld is, wordt de selectie niet ongedaan gemaakt wanneer u tekst typt of de cursor verplaatst. Dit betekent dat u de cursor ergens anders heen kunt verplaatsen en daar tekst kunt intypen.

Blijvende selectie is standaard uitgeschakeld.

Blijvende selectie kan ingesteld worden op de pagina [Cursor & selectie](#) in [Instellingendialoog](#).

WAARSCHUWING

Als "Blijvende selectie" en "Selectie overschrijven" beide ingeschakeld zijn, wordt de selectie gede-selecteerd en vervangen door getypte of geplakte tekst wanneer de tekstcursor zich binnen de selectie bevindt.

3.4 Tekst kopiëren en plakken

Om tekst te kopiëren selecteert u de tekst en daarna gebruikt u het menu **Bewerken** → **Kopiëren**. Bovendien wordt met de muis geselecteerde tekst naar de X-selectie gekopieerd.

Om tekst van het klembord te plakken gebruikt u het menu **Bewerken** → **Plakken**.

Bovendien kan tekst die met de muis is geselecteerd, geplakt worden door met de middelste muisknop op de gewenste positie te klikken.

TIP

Als u het KDE-bureaublad gebruikt, kunt u tekst die al eerder vanuit andere toepassingen gekopieerd is, ophalen door het Klipper-pictogram in het systeemvak te gebruiken.

3.5 Tekst zoeken en vervangen

3.5.1 De balken voor zoeken en vervangen

KatePart heeft een incrementele zoekbalk en een krachtige zoek- en vervangbalk, die de mogelijkheid biedt om vervangende tekst in te voeren samen met enkele extra opties.

De balken bieden de volgende gezamenlijke opties:

Zoeken

Hier typt u de tekenreeks in die gezocht moet worden. De interpretatie ervan is afhankelijk van enkele opties die hieronder beschreven worden.

Hoofdlettergevoelig

Indien geactiveerd, kunt u kiezen of bij het zoeken rekening gehouden moet worden met hoofdletters of kleine letters.

De krachtige zoek- en vervangbalk biedt enkele extra opties:

Platte tekst

De te zoeken tekenreeks moet letterlijk overeenkomen.

Hele woorden

Als deze optie geactiveerd is, wordt alleen een overeenkomst gevonden er een woordgrens aan beide kanten van de tekenreeks staat, wat betekent geen alfanumeriek teken of een ander zichtbaar teken of een einde regel.

Escape Sequences

Als deze optie geactiveerd is, komt er een menuitem **Toevoegen** aan het eind van het contextmenu van de tekstvakken beschikbaar, die u in staat stelt om uit een voorgedefinieerde lijst "escape-reeksen" toe te voegen.

Reguliere expressie

Als deze optie geactiveerd is, wordt de zoektekenreeks als een reguliere expressie geïnterpreteerd. Het menuitem **Toevoegen** komt aan het eind van het contextmenu van de tekstvakken beschikbaar, die u in staat stelt om uit een voorgedefinieerde lijst "reguliere expressies" toe te voegen aan het zoekpatroon.

Zie [Reguliere expressies](#) voor meer informatie hierover.

Alleen in de selectie zoeken

Indien geactiveerd, zal zoeken en vervangen alleen worden uitgevoerd binnen de geselecteerde tekst.

Allen zoeken

op deze knop drukken accentueert alle overeenkomsten in het document en toont het aantal gevonden overeenkomsten in een kleine popup.

3.5.2 Tekst vinden

Om tekst te vinden start u de incrementele zoekbalk met **Ctrl+F** of met het menuitem **Bewerken** → **Zoeken...**

Dit opent de zoekbalk onder in het venster van de editor. Links in de balk bevindt zich een knop met een pictogram om de balk te sluiten, gevolgd door een klein invoerveld voor het opgeven van het zoekpatroon.

Het zoeken begint onmiddellijk nadat u de eerste letters van uw zoekpatroon intypt. Als er een overeenkomst is in de tekst, wordt deze gemarkeerd en verandert de achtergrondkleur van de invoer in lichtgroen. Als er geen overeenkomst gevonden wordt in de tekst wordt dit aangegeven door een lichtrode achtergrondkleur van het invoerveld.

Gebruik de knop  of  om naar de volgende of vorige overeenkomst in het document te gaan.

Overeenkomsten in het document zijn geaccentueerd zelfs wanneer u de zoekbalk sluit. Om deze accentuering te wissen drukt u op de **Esc**-toets.

U kunt kiezen of bij het zoeken rekening gehouden moet worden met hoofdletters of kleine letters. Kiest u , dan krijgt u alleen de tekst die precies overeenkomt met de hoofd- en kleine letters in het zoekpatroon.

Klik op de knop  rechts van de incrementele zoekbalk om over te schakelen naar het balk geavanceerd zoeken en vervangen.

Om de laatste zoekbewerking te herhalen, indien aanwezig, zonder de incrementele zoekbalk op te roepen, gebruikt u **Bewerken** → **Volgende zoeken (F3)** of **Bewerken** → **Vorige zoeken (Shift+F3)**.

3.5.3 Tekst vervangen

Om tekst te vervangen, start u de krachtige zoek- en vervangbalk met het commando **Bewerken** → **vervangen** of de sneltoets **Ctrl+R**.

Links op de balk bevindt zich een knop met een pictogram om de balk te sluiten, gevolgd door een kleine keuzelijst om het zoekpatroon in te voeren. De lijst herinnert zich recent gebruikte patronen.

U kunt de zoekmodus aanpassen door uit het afrolvak te kiezen voor **Platte tekst**, **Hele woorden**, **Escape-sequences** of **Reguliere expressie**.

Als **Escape sequences** of **Reguliere expressie** geselecteerd is, zal het menu-item **Toevoegen...** onderaan het contextmenu van de tekstinvoervelden geactiveerd worden. Hierdoor kunt u escape sequences of reguliere expressie-items toevoegen aan het zoek- of vervangpatroon vanuit voorafgedefinieerde lijsten.

Gebruik de knop  of  om naar de volgende of vorige overeenkomst in het document te gaan.

Typ de vervangende tekst in in het tekstinvoerveld met de naam **Vervangen** en klik op de knop **Vervangen** om alleen de geaccentueerde tekst te vervangen, of de knop **Alles vervangen** om de gezochte tekst in het hele document te vervangen.

U kunt het gedrag van zoeken en vervangen wijzigen door onderin de balk verschillende opties te kiezen. Het kiezen van  resulteert in alleen die tekenreeksen waarvan elke letter overeenkomt met de hoofdletter of kleine letter zoals opgegeven in de zoekopdracht.  zal alleen binnen de huidige selectie zoeken en vervangen. De knop **Alles zoeken** accentueert alle overeenkomsten in het document en toont het aantal gevonden resultaten in een klein pop-upvenster.

Klik op de knop  rechts in de balk zoeken en vervangen om over te schakelen naar de incrementele zoekbalk.

TIP

Als u reguliere expressies gebruikt om de te vervangen tekst te zoeken, kunt u gebruik maken van terugverwijzingen om tekst die tussen haakjes in subpatronen staat opnieuw te gebruiken. Zie [Reguliere expressies](#) voor meer informatie hierover.

TIP

U kunt de commando's **find**, **replace** en **ifind** (incrementeel zoeken) vanaf de [commandoregel](#).

3.6 Bladwijzers gebruiken

Door middel van bladwijzers kunt u bepaalde regels markeren om ze snel terug te kunnen vinden.

U kunt op twee manieren bladwijzers instellen of verwijderen:

- Beweeg de invoercursor naar de regel en activeer het commando **Bladwijzers** → **Bladwijzer instellen** (Ctrl+B).
- Klik in de pictogramrand naast de regel.

Het menu **Bladwijzers** bevat een lijst met alle bladwijzers. Elke bladwijzer is herkenbaar aan het regelnummer en het begin van de tekst op de regel. Om de tekstcursor naar het begin van een gemarkeerde regel te verplaatsen selecteert u de gewenste bladwijzer in het menu.

Om snel van bladwijzer naar bladwijzer te bewegen of naar de volgende/vorige bladwijzer, gebruikt u de commando's **Bladwijzers** → **Volgende** (Alt+PgDn) of **Bladwijzers** → **Vorige** (Alt+PgUp).

3.7 Automatische regelafbreking

Met behulp van deze functie kan tekst op een eenvoudige manier opgemaakt worden: de regels worden afgebroken op de ingestelde regellengte. Dit gebeurt echter niet bij aaneengesloten tekenreeksen die langer zijn dan de ingestelde regellengte.

U kunt deze functie in- en uitschakelen door het keuzevakje **Statische regelafbreking** in de [bewerkingspagina](#) van de [configuratiedialoog](#) aan of uit te zetten.

Om de maximale regellengte in te stellen (het maximum aantal tekens per regel), gebruikt u de optie **Regels afbreken op** in de pagina **Bewerking** van de [configuratiedialoog](#).

Ingeschakeld heeft deze optie de volgende effecten:

- Terwijl u typt, zal de editor automatisch een regeleinde invoegen na de laatste witruimte (bijvoorbeeld een spatie of tabstop) vóór de maximale regellengte is bereikt.
- Terwijl een document geladen wordt, zal de editor de tekst ook afbreken, indien de witruimte in de regels dit mogelijk maakt, zodat de regels niet langer zijn dan de maximale regellengte.

OPMERKING

Het is momenteel niet mogelijk om regelafbreking per documenttype in te stellen; ook kan regelafbreking niet per document in- of uitgeschakeld worden. Dit zal wel mogelijk zijn in een toekomstige versie van KatePart.

3.8 Automatisch inspringen gebruiken

De editor-component van KatePart ondersteunt diverse modi om tekst automatisch in te laten springen, ontworpen voor verschillende soorten tekstbestanden. In het menu **Hulpmiddelen** → **Inspringen** kunt u kiezen uit de beschikbare modi. Met behulp van de functie **Hulpmiddelen** → **Inspringen formatteren** kunt u de inspringbreedte van de geselecteerde regels of van de huidige regel laten herberekenen. U kunt dus in uw gehele document het inspringen opnieuw doen door alle tekst te selecteren en deze actie te activeren.

Alle inspringmodi gebruiken het inspringen gerelateerd aan de instellingen in het actieve document.

TIP

U kunt allerlei soorten configuratievariabelen instellen, inclusief die zijn gerelateerd aan inspringen met [Documentvariabelen](#) en [Bestandstypen](#).

BESCHIKBARE MODI BIJ AUTOMATISCH INSPRINGEN

Geen

In deze modus wordt automatisch inspringen uitgeschakeld.

Normaal

Bij deze modus wordt de regel op dezelfde manier ingesprongen als de voorgaande regel. U kunt dit combineren met de opties "Inspringen" en "Inspringing verwijderen" om de tekst naar eigen inzicht in te laten springen.

C-stijl

Een programma voor indentering dat de regels laat inspringen op de manier zoals gebruikt wordt in C en gelijksoortige talen, C++, C#, Java™, JavaScript, enzovoort. Dit indenteren werkt niet voor scripttalen zoals Perl of PHP.

Haskell

Een programma voor indentering dat specifiek is voor de Haskell taal.

Lilypond

Een programma voor indentering dat specifiek is voor de Lilypond notatietaal voor muziek.

Lisp

Een programma voor indentering dat specifiek is voor de Lisp scripttaal en Lisp dialecten.

Python

Een programma voor indentering dat specifiek is voor de python scripttaal.

XML-stijl

Een programma voor indentering dat specifiek is voor talen die lijken op XML.

3.9 Markeringen voor wijzigingen in regels

De markeringen voor wijzigingen in regels van KatePart laten u gemakkelijk zien wat u recent in een bestand hebt gewijzigd. Opgeslagen wijzigingen worden standaard aangegeven door een groene balk links van een document, terwijl niet opgeslagen wijzigingen aangegeven worden door een oranje balk.

```
// new block for empty buffer I write new stuff HERE
andNewStuffHere = 1;
if (newLineChangedAgain)
    lala ();
more = 1;
afterSaveEdited ();

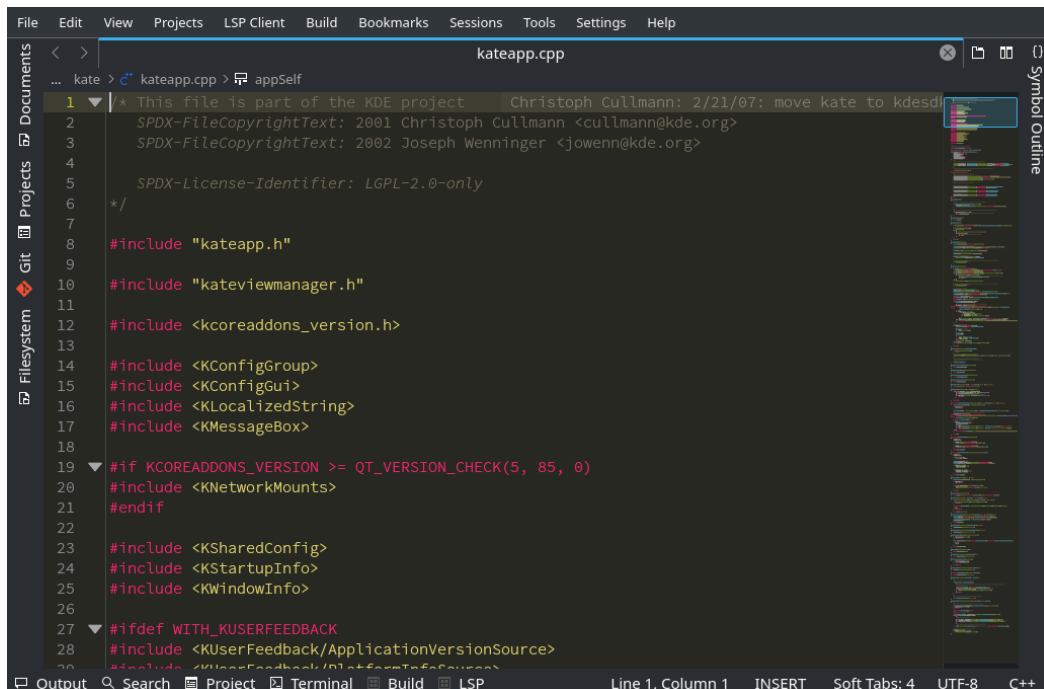
TextBlock *newBlock = new TextBlock (this, 0);
newBlock->appendLine (TextLine (new TextLineData()));
```

Markeringen voor wijzigingen in regels in actie.

U kunt de kleuren gebruikt in het [instellingenpaneel Lettertypes & kleuren](#) wijzigen of u kunt deze mogelijkheid volledig uitschakelen in het tabblad [Randen](#) van het [instellingenpaneel Uiterlijk](#).

3.10 De schuifbalk met mini-map

De schuifbalk met mini-map van KatePart toont voorbeelden van documenten in plaats van de schuifbalk. Het nu zichtbare deel van het document is geaccentueerd.



De schuifbalk met mini-map toont een voorbeeld van de broncode van Kate.

U kunt de mini-map tijdelijk in- of uitschakelen door **Beeld** → **Schuifbalk met mini-map tonen** te selecteren of permanent in [de sectie Uiterlijk van de instellingen van KatePart](#).

3.11 Meerdere cursors

Ondersteuning voor meerdere cursors is geïntroduceerd in versie 5.93 van katepart.

3.11.1 Meerdere cursors worden aangemaakt

- Om ze via de muis aan te maken, gebruik **Alt** + linkermuisknop. De modifier is te configureren, zie [Multicursor-modifier configureren](#)
- Om via het toetsenbord aan te maken, druk op **Ctrl+Alt+Up** om cursor boven de primaire cursor aan te maken en **Ctrl+Alt+Down** om de cursor eronder aan te maken. Deze sneltoetsen zijn ook te configureren
- Om cursors uit een selectie aan te maken, selecteer eerst enige tekst en druk daarna op **Shift+Alt+I**. Dit zal een cursor aan het eind van elke regel in de selectie aanmaken.
- **Alt+J** gebruiken om de het volgende voorkomen van het woord onder de cursor en het selecteren + een cursor aanmaken. Als u het huidige woord onder de cursor wilt overslaan, druk op **Alt+K** en het zal het nu geselecteerde woord als overgeslagen markeren. Wanneer u opnieuw op **Alt+J** drukt zal het de selectie van het huidige woord ongedaan maken en naar het volgende woord gaan.
- **Ctrl+Alt+Shift+J** gebruiken om alle overeenkomstige woorden van die onder de cursor te vinden en ze selecteren met een cursor aan het eind van elke selectie. U kunt **Alt+J** gebruiken om door de geselecteerde woorden te lopen en **Alt+K** te gebruiken om de selectie ongedaan te maken van elk woord genoemd in de vorige paragraaf.

3.11.2 Werken met meerdere cursors

Nadat u een paar cursors hebt gemaakt, kunt u de meeste bewerkingen op hen doen zoals u zou doen op een enkele cursor. Bijvoorbeeld, bij een letter typen zal het voor elke cursor doen. Evenzo kunt u teksttransformaties doen bijv., hoofdletter maken voor alle posities of selecties.

Soms wilt u cursors verwijderen. Om dat te doen kunt u **Alt** + linkermuisknop op de cursor doen die u wilt verwijderen. Als u alleen cursors wilt verwijderen op lege regels, dan is er een al gemaakte actie voor die dat voor u zal doen. Om de actie uit te voeren, open de Commandobalk met **Ctrl+Alt+I** en zoek naar **Cursors verwijderen uit lege regels** en druk op **Enter**. U kunt ook een sneltoets voor deze actie configureren.

Hoofdstuk 4

Het menu Items

4.1 Het menu Bestand

Bestand → Nieuw (Ctrl+N)

Hiermee opent u een nieuw document in een nieuw en onafhankelijk editorvenster.

Bestand → Nieuw venster

Maak een nieuw venster aan met het huidige document als inhoud. Alle wijzigingen in het document in het ene venster worden meegenomen in het andere en vice versa.

Bestand → Openen... (Ctrl+O)

Toont een standaard KDE **Bestand openen**-dialoogvenster. Gebruik het bestandsoverzicht om het te openen bestand te selecteren en klik op de knop **Openen** om het te openen.

Bestand → Recent geopend

Dit geeft een lijst van bestanden die u recent hebt opgeslagen. Door op een bepaald bestand te klikken wordt het geopend in KatePart. Dit kan uiteraard alleen als het bestand nog bestaat op de aangegeven locatie.

Bestand → Opslaan (Ctrl+S)

Slaat het huidige document op. Indien dit bestand al eerder is opgeslagen met deze naam, dan zal het bestand zonder verdere vragen worden opgeslagen. Indien dit de eerste keer is dat een nieuw document wordt opgeslagen zal het dialoogvenster "opslaan als" worden getoond. Dit venster wordt hieronder verder beschreven.

Bestand → Opslaan als... (Ctrl+Shift+S)

Dit maakt het mogelijk om een document op te slaan onder een nieuwe bestandsnaam. De nieuwe naam kunt u opgeven in het venster dat identiek is aan het venster zoals dat in de sectie [Openen](#) reeds is beschreven.

Bestand → Opslaan met codering

Een document opslaan met een nieuwe bestandsnaam in een andere codering.

Bestand → Kopie opslaan als...

Een kopie van een document opslaan met een nieuwe bestandsnaam en doorgaan met werken in het originele document.

Bestand → Herladen (Ctrl+F5)

Laadt het actieve bestand opnieuw van de schijf. Deze opdracht is nuttig wanneer een ander programma of proces het bestand heeft gewijzigd terwijl u het open hebt staan in KatePart.

Bestand → Afdrukken... (Ctrl+P)

Opent het afdrukvenster waarin u kunt aangeven wat, waar en hoe er afgedrukt moet worden.

Bestand → Als HTML exporteren

Sla het huidige open document op als een HTML-bestand, die opgemaakt wordt met de huidige accentuering van syntaxis en instellingen van het kleurschema.

Bestand → Sluiten (Ctrl+W)

Sluit het actieve venster met deze opdracht. Als u wijzigingen hebt aangebracht, zal u gevraagd worden het bestand op te slaan voordat KatePart het sluit.

Bestand → Afsluiten (Ctrl+Q)

Sluit het huidige venster. Indien u via de menu-items **Nieuw** of **Nieuw venster** meerdere KWrite-vensters geopend hebt, zullen die niet worden gesloten.

4.2 Het menu Bewerken

Bewerken → Ongedaan maken (Ctrl+Z)

Maakt de laatste bewerking ongedaan (typen, kopiëren, knippen, etc.)

OPMERKING

Dit kan verscheidene bewerkingen, zoals het intypen van tekst, tegelijk ongedaan maken.

Bewerken → Opnieuw (Ctrl+Shift+Z)

Dit draait de gevolgen van "Ongedaan maken" weer terug.

Bewerken → Knippen (Ctrl+X)

Deze opdracht verwijdert de huidige selectie en plaatst het op het klembord. Het klembord werkt onzichtbaar en maakt het mogelijk gegevens over te brengen tussen toepassingen.

Bewerken → Kopiëren (Ctrl+C)

Dit kopieert de huidige selectie naar het klembord zodat die elders kan worden geplakt. Het klembord werkt onzichtbaar en maakt het mogelijk gegevens over te brengen tussen toepassingen.

Bewerken → Als HTML kopiëren

Kopieer de selectie als HTML, opgemaakt volgens de huidige accentuering van de syntaxis en instellingen van het kleurschema.

Bewerken → Plakken (Ctrl+V)

Dit plakt het eerste item in het klembord op de plaats van de cursor. Het klembord werkt onzichtbaar en maakt het mogelijk gegevens over te brengen tussen toepassingen.

OPMERKING

Als de optie "Selectie overschrijven" ingeschakeld is, zal de geplakte tekst eventueel geselecteerde tekst overschrijven.

Bewerken → Selectie plakken (Ctrl+Shift+Ins)

Dit zal de inhoud van de [muissselectie](#) die eerder is gedaan plakken. Markeer enige tekst met de muisaanwijzer om het in het nu open bestand te plakken met dit menu-item.

Bewerken → Met inhoud van klembord omwisselen

Dit zal de geselecteerde tekst met de inhoud van het [klembord](#) omwisselen.

Bewerken → Klembordgeschiedenis plakken

Dit zal een dialoog openen om een item uit de klembordgeschiedenis te selecteren en te plakken.

Bewerken → Invoer-modi

Schakel tussen een normale en een vi-achtige modale bewerkingsmethode. De vi-invoermodus ondersteunt de meestgebruikte opdrachten en bewegingen van de normale en visuele modus van vim en heeft een optionele vi-modus statusbalk. Deze statusbalk toont opdrachten terwijl ze worden ingevoerd, uitvoer van opdrachten en de huidige modus. Het gedrag van deze modus kan aangepast worden op het tabblad [Vi-invoermethode](#) in het tabblad **Bewerking** van de instellingendialoog van KatePart.

Bewerken → Overschrijfmodus (Ins)

Schakelt de overschrijfmodus in of uit. Wanneer de modus **INS** is, worden tekens ingevoegd op de plaats van de cursor. Wanneer de modus **OVR** is, zullen de tekens die u typt de huidige tekens rechts van de cursor vervangen. De statusbalk toont de huidige status van de overschrijfmodus, of **INS** of **OVR**.

Bewerken → Modus alleen lezen

Stel het huidige document in op Alleen lezen. Dit voorkomt dat er tekst wordt toegevoegd of wijzigingen in de opmaak worden aangebracht.

Bewerken → Zoeken... (Ctrl+F)

Dit opent de zoekbalk onder in het venster van de editor. Links in de balk bevindt zich een knop met een pictogram om de balk te sluiten, gevolgd door een klein invoerveld voor het opgeven van het zoekpatroon.

Het zoeken begint onmiddellijk nadat u de eerste letters van uw zoekpatroon intypt. Als er een overeenkomst is in de tekst, wordt deze gemarkeerd en verandert de achtergrondkleur van de invoer in lichtgroen. Als er geen overeenkomst gevonden wordt in de tekst wordt dit aangegeven door een lichtrode achtergrondkleur van het invoerveld.

Gebruik de knop  of  om naar de volgende of vorige overeenkomst in het document te gaan.

Overeenkomsten in het document worden geaccentueerd zelfs wanneer u de zoekbalk sluit. Om deze accentuering te wissen, druk op de toets **Esc**.

U kunt kiezen of bij het zoeken rekening gehouden moet worden met hoofdletters of kleine

letters. Kiest u , dan krijgt u alleen de tekst die precies overeenkomt met de hoofd- en kleine letters in het zoekpatroon.

Klik op de knop  rechts van de incrementele zoekbalk om over te schakelen naar het balk geavanceerd zoeken en vervangen.

Bewerken → Varianten zoeken → Volgende zoeken (F3)

Hiermee herhaalt u de laatste zoekhandeling, indien aanwezig, zonder de incrementele zoekbalk te activeren, waarbij voorwaarts gezocht wordt in het document vanaf de positie van de cursor.

Bewerken → Varianten zoeken → Vorige zoeken (Shift+F3)

Hiermee herhaalt u de laatste zoekhandeling, indien aanwezig, zonder de zoekbalk te activeren, waarbij achterwaarts in plaats van voorwaarts gezocht wordt in het document.

Bewerken → Varianten zoeken → Selectie zoeken (Ctrl+H)

Vindt het volgende stuk geselecteerde tekst.

Bewerken → Varianten zoeken → Selectie achterwaarts zoeken (Ctrl+Shift+H)

Vindt het vorige stuk geselecteerde tekst.

Bewerken → Vervangen... (Ctrl+R)

Deze opdracht opent het venster "geavanceerd zoeken en vervangen". Links boven in het venster bevindt zich een knop met een pictogram om het venster te sluiten, gevolgd door een klein tekst invoerveld om het zoekpatroon in te voeren.

U kunt de zoekmodus aanpassen door uit de lijst te kiezen voor **Platte tekst**, **Hele woorden**, **Escape sequences** of **Reguliere expressie**.

Als **Escape sequences** of **Reguliere expressie** geselecteerd is, zal het menu-item **Toevoegen...** onderaan het contextmenu van de tekst invoervelden geactiveerd worden. Hierdoor kunt u escape sequences of reguliere expressie-items toevoegen aan het zoek- of vervangpatroon vanuit voorgedefinieerde lijsten.

Gebruik de knop  of  om naar de volgende of vorige overeenkomst in het document te gaan.

Typ de vervangende tekst in in het tekst invoerveld met de naam **Vervangen** en klik op de knop **Vervangen** om alleen de geaccentueerde tekst te vervangen, of de knop **Alles vervangen** om de gezochte tekst in het hele document te vervangen.

U kunt het gedrag van zoeken en vervangen wijzigen door onderin de balk verschillende opties te kiezen. Het kiezen van  resulteert in alleen die tekenreeksen waarvan elke letter overeenkomt met de hoofdletter of kleine letter zoals opgegeven in de zoekopdracht.

 zal alleen binnen de huidige selectie zoeken en vervangen. De knop **Alles zoeken** accentueert alle overeenkomsten in het document en toont het aantal gevonden resultaten in een klein popup-venster.

Klik op de knop  rechts in de balk zoeken en vervangen om over te schakelen naar de incrementele zoekbalk.

4.3 Het menu Selectie

Selectie → Alles selecteren (Ctrl+A)

Dit selecteert het hele document. Op deze manier kunt u op een handige manier alle tekst kopiëren naar een andere toepassing.

Selectie → Deselecteren (Ctrl+Shift+A)

Als er tekst in de editor geselecteerd is, wordt de selectie opgeheven.

Selectie → Blokselectiemodus (Ctrl+Shift+B)

Wisselt van selectiemodus. Wanneer de selectiemodus **BLOK** is, de statusbalk toont de tekenreeks [**BLOK**] en u kunt verticaal selecteren, bijv. selecteer kolom 5 tot 10 in regel 9 tot 15.

Selectie → Commentaar (Ctrl+D)

Dit voegt één spatie toe aan het begin van de regel waar de tekstcursor zich bevindt of aan het begin van geselecteerde regels.

Selectie → Commentaar verwijderen (Ctrl+Shift+D)

Dit verwijdert één spatie (indien aanwezig) aan het begin van de regel waar de tekstcursor zich bevindt of aan het begin van geselecteerde regels.

Selectie → Regels samenvoegen (Ctrl+J)

Voegt de geselecteerde regels samen, of de huidige regel en de regel eronder, met één spatie als scheidingsteken. Overtollige witruimte in de samengevoegde regels wordt verwijderd.

Selectie → Elk woord met hoofdletter laten beginnen (Ctrl+Alt+U)

Laat de geselecteerde tekst of het huidige woord met een hoofdletter beginnen.

Selectie → Omzetten naar hoofdletters (Ctrl+U)

Zet de geselecteerde tekst of de letter na de cursor om in hoofdletters.

Selectie → Omzetten naar kleine letters (Ctrl+Shift+U)

Zet de geselecteerde tekst of de letter na de cursor om in kleine letters.

Selectie → Inspringen opschonen

Dit schoont het inspringen voor de huidige selectie of voor de regel waarin zich de cursor bevindt op. Inspringen opschonen maakt dat alle geselecteerde tekst de inspringmethode van uw keuze krijgt.

Selectie → Inspringen formatteren

De huidige regel of geselecteerde regels worden opnieuw uitgelijnd, waarbij gebruik gemaakt wordt van de inspringmethode en inspring-instellingen in het document.

Selectie → Uitlijnen op...

Dit commando lijnt regels uit in het geselecteerde blok of het gehele document op de kolom gegeven door een reguliere expressie waarom u zal worden gevraagd.

Als u een leeg patroon geeft zal er standaard uitgelijnd worden op het eerste niet-blanke teken.

Als het patroon een vangst heeft zal het inspringen op de gevangen overeenkomst.

Voorbeelden:

Met `'-'` zal het spaties invoegen voor de eerste `'-'` van elke regel om ze allemaal uit te lijnen op dezelfde kolom.

Met `'alignon :\\s+(.)'` zal het spaties invoegen vóór het eerste niet-blanke teken dat verschijnt na een dubbele punt om ze allemaal uit te lijnen op dezelfde kolom.

Selectie → Regelafbreking toepassen

Pas statische regelafbreking toe voor het hele document. Dat betekent dat er automatisch een nieuwe tekstregel gestart wordt wanneer de huidige regel de lengte overschrijdt die opgegeven is bij de optie **Regels afbreken op:** op het tabblad Bewerken in het menu **Instellingen → Editor instellen...**

Selectie → Dakje boven cursor toevoegen (Ctrl+Alt+Up)

Voegt nog een dakje boven de huidige cursor toe. Het dakje wordt geplaatst in de regel er direct boven en in dezelfde kolom als de huidige cursor.

Selectie → Dakje onder cursor toevoegen (Ctrl+Shift+Down)

Voegt nog een dakje onder de huidige cursor toe. Het dakje wordt geplaatst in de regel er direct boven en in dezelfde kolom als de huidige cursor.

Selectie → Cursors aan regeleinden toevoegen (Ctrl+Shift+I)

Voegt een cursor toe aan elke nu geselecteerde regel.

Selectie → Zoek en selecteer volgende aanwezigheid (Alt+J)

Zoekt het volgende voorkomen van het woord nu onder de cursor, selecteert het en voegt een cursor toe.

Selectie → Zoek en selecteer alle voorkomens (Ctrl+Alt+Shift+J)

Zoekt alle voorkomens van het woord nu onder de cursor, selecteert ze en voegt een cursor toe aan elk van hen.

4.4 Het menu Beeld

Beeld → Lettertype vergroten (Ctrl++)

Dit vergroot de lettertekens die getoond worden.

Beeld → Lettertype verkleinen (Ctrl+-)

Dit verkleint de lettertekens die getoond worden.

Beeld → Regelafbreking → Dynamische regelafbreking (F10)

Schakelt dynamisch regelafbreking in/uit in de huidige weergave. Dynamische regelafbreking maakt alle tekst in een weergave zichtbaar zonder de noodzaak voor horizontaal schuiven door een actuele regel, indien nodig, op meer zichtbare regels af te beelden.

Beeld → Regelafbreking → Dynamische-regelafbreking-indicatoren

Kies wanneer en hoe de dynamische-regelafbreking-indicators moeten worden getoond. Dit is alleen beschikbaar wanneer de optie **Dynamische regelafbreking** is aangevinkt.

Beeld → Regelafbreking → Statische regelafbrekingmarkering tonen

Wanneer deze optie is aangevinkt, verschijnt er een verticale lijn in de regelafbrekingkolom zoals gedefinieerd op het tabblad **Bewerking** van de **Instellingen** → **Editor instellen**. Merk op dat de regelafbrekingmarkering alleen getoond wordt wanneer u een niet-proportioneel lettertype gebruikt.

Beeld → Randen → Pictogramrand tonen (F6)

Dit is een wissel-item. Als het is aangevinkt wordt de pictogramrand zichtbaar links van de actieve editor en omgekeerd. De pictogramrand geeft de positie weer van de gemarkeerde lijnen in de bewerker.

Beeld → Randen → Regelnummering tonen (F11)

Dit is een wisselitem. Als het is aangevinkt verschijnt er aan de linkerkant van de actieve editor een paneel waarin de regelnummers van het document worden getoond, en vice versa.

Beeld → Randen → Schuifbalkmarkeringen tonen

Wanneer deze optie geactiveerd is, worden in de huidige weergave markeringen getoond op de verticale schuifbalk. Deze markeringen zijn gelijk aan de markeringen op de [Pictogramrand](#).

Beeld → Randen → Schuifbalk met mini-map tonen

Dit vervangt de schuifbalk door een visualisatie van het huidige document. Voor meer informatie over de schuifbalk-minimap, zie [Section 3.10](#).

Beeld → Code vouwen

Deze opties behoren bij [code vouwen](#).

Beeld → Randen → Code invouwen → Invouwmarkeringen tonen

Schakelt het tonen van het paneel voor invouwen in/uit aan de linkerzijde van de weergave.

Beeld → Code vouwen → Huidige node invouwen

Vouw het gebied dat de cursor bevat in.

Beeld → Code vouwen → Huidige node uitvouwen

Vouw het gebied dat de cursor bevat uit.

Beeld → Code vouwen → Nodes op topniveau invouwen (Ctrl+Shift+-)

Vouw alle gebieden van het bovenste niveau in het document in. Klik op het naar rechts wijzende driehoekje om alle gebieden op het hoogste niveau uit te vouwen.

Beeld → Code vouwen → Nodes op topniveau uitvouwen (Ctrl+Shift++)

Vouw alle gebieden van het bovenste niveau in het document uit.

Niet-afdrukbare spaties tonen

Omringend vak rond niet-afdrukbare spaties tonen/verbergen.

4.5 Het menu Ga naar

Ga naar → Ga naar regel... (Ctrl+G)

Dit opent onderaan het venster de balk Ga naar regel, waarmee u de cursor een sprong kunt laten maken naar een bepaalde regel (door een getal op te geven) in het document. U kunt het regelnummer direct invoeren in het tekstinvoerveld of u kunt klikken op de pijltjes naar boven en beneden naast het tekstinvoerveld. Het pijltje naar boven verhoogt het regelnummer, en het pijltje naar beneden verlaagt het. U sluit de balk door te klikken op de knop met het pictogram links op de balk.

Ga naar → Ga naar eerder gewijzigde regel (Ctrl+E)

Deze actie springt naar de eerder gewijzigde regel in de [multicursorconfiguratie](#).

Ga naar → Ga naar volgende gewijzigde regel (Ctrl+Shift+E)

Deze actie springt naar de volgende gewijzigde regel in de [multicursorconfiguratie](#).

Ga naar → Ga naar de vorige gewijzigde regel

Regels die zijn gewijzigd na het openen van het bestand worden gewijzigde regels genoemd. Deze actie springt naar de vorige gewijzigde regel.

Ga naar → Ga naar de volgende gewijzigde regel

Regels die zijn gewijzigd na het openen van het bestand worden gewijzigde regels genoemd. Deze actie springt naar de volgende gewijzigde regel.

Ga naar → Ga naar het bijbehorende haakje (Ctrl+6)

Beweeg de cursor naar het bijbehorende openings- of sluithaakje.

Ga naar → Tot het overeenkomstige haakje selecteren (Ctrl+Shift+6)

Selecteert de tekst tussen het overeenkomstige openings- en sluithaakje.

Ga naar → Bladwijzers (Ctrl+Shift+6)

Onder de items die hier beschreven worden, wordt elke bladwijzer in het document vermeld. De tekst bestaat uit de eerste woorden van de gemarkeerde regel. Selecteer een bladwijzer om de cursor naar het begin van die regel te verplaatsen. De editor zal de tekst opschuiven om de regel zichtbaar te maken.

Ga naar → Bladwijzers → Bladwijzer instellen (Ctrl+B)

Plaats een bladwijzer op de huidige regel in het actieve document (als er al een bladwijzer is, wordt deze verwijderd).

Ga naar → Bladwijzers → Alle bladwijzers wissen

Deze opdracht verwijdert alle bladwijzers uit het document evenals de lijst met bladwijzers die onderaan dit menu-item hangt.

Ga naar → Bladwijzers → Vorige (Alt+PgUp)

Hiermee verplaatst u de cursor naar het begin van de regel met de eerste bovenliggende bladwijzer. In de tekst van het menu-item staat het regelnummer en het eerste stuk tekst op die regel. Dit item is alleen beschikbaar wanneer er een bladwijzer in een regel boven de cursor bestaat.

Ga naar → Bladwijzers → Volgende (Alt+PgDn)

Hiermee verplaatst u de cursor naar het begin van de volgende regel met een bladwijzer. In de tekst van het menu-item staat het regelnummer en het eerste stuk tekst op die regel. Dit item is alleen beschikbaar wanneer er een bladwijzer in een regel onder de cursor bestaat.

4.6 Het menu Hulpmiddelen

Hulpmiddelen → Modus

Kies het gewenste bestandstype voor het actieve document. Dit overschrijft alleen voor het huidige document het globale bestandstype ingesteld bij **Instellingen → Editor instellen** op het tabblad Bestandstypen in de sectie Openen/opslaan.

Hulpmiddelen → Accentuering

Kies het gewenste accentueringsschema voor het actieve document. Dit overschrijft alleen voor het huidige document de globale accentueringregels ingesteld bij **Instellingen → Editor instellen**.

Hulpmiddelen → Inspringen

Kies de gewenste manier van inspringen voor het actieve document. Dit overschrijft alleen voor het huidige document de globale inspringmodus ingesteld bij **Instellingen → Editor instellen**.

Hulpmiddelen → Codering

U kunt de standaardcodering, ingesteld bij **Instellingen → Editor instellen...** in de pagina **Openen/opslaan** om een andere codering in te stellen voor het huidige document. De codering die u hier instelt zal alleen gelden voor het huidige document.

Hulpmiddelen → Bytevolgordemarkering (BOM) toevoegen

Door deze actie aan te vinken kunt u expliciet een bytevolgordemarkering voor documenten gecodeerd in unicode toevoegen. De bytevolgordemarkering (BOM) is een Unicode-teken dat gebruikt wordt om de "endianness" (byte-volgorde) van een tekstbestand of stream aan te geven. Zie voor meer informatie: [Byte Order Mark](#).

Hulpmiddelen → Regeleinde

Kies de gewenste regeleindemodus voor het actieve document. Dit overschrijft alleen voor het huidige document de globale einderegelmodus ingesteld bij **Instellingen → Editor instellen...**

Hulpmiddelen → Scripts

Dit submenu bevat een lijst met alle acties in scripts. De lijst kan gemakkelijk worden gewijzigd door [uw eigen scripts te schrijven](#). Op deze manier kan KatePart worden uitgebreid met gebruikergedefinieerde hulpmiddelen.

Hulpmiddelen → Scripts → Navigatie

Hulpmiddelen → Scripts → Navigatie → Cursor verplaatsen naar vorig overeenkomend inspringen (Alt+Shift+Up)

Verplaatst de cursor naar de eerst regel boven de huidige regel die op hetzelfde niveau inspringt als de huidige regel.

Hulpmiddelen → Scripts → Navigatie → Cursor verplaatsen naar volgend overeenkomend inspringen (Alt+Shift+Down)

Verplaatst de cursor naar de eerst regel na de huidige regel die op hetzelfde niveau inspringt als de huidige regel.

Hulpmiddelen → Scripts → Bewerking

Hulpmiddelen → Scripts → Bewerking → Geselecteerde tekst sorteren

De geselecteerde tekst of het gehele document sorteren in oplopende volgorde.

Hulpmiddelen → Scripts → Bewerking → Regels omlaag verplaatsen (Ctrl+Shift+Down)

Geselecteerde regels omlaag verplaatsen.

Hulpmiddelen → Scripts → Bewerking → Regels omhoog verplaatsen (Ctrl+Shift+Up)

Geselecteerde regels omhoog verplaatsen.

Hulpmiddelen → Scripts → Bewerking → Geselecteerde regels omlaag dupliceren (Ctrl+Alt+Down)

De geselecteerde regels omlaag dupliceren.

Hulpmiddelen → Scripts → Bewerking → Geselecteerde regels omhoog dupliceren (Ctrl+Alt+Up)

De geselecteerde regels omhoog dupliceren.

Hulpmiddelen → Scripts → Bewerking → Geselecteerde tekst URI-coderen

Codeert de geselecteerde tekst zodat het gebruikt kan worden als onderdeel van een zoektekst in een URL, waarbij de selectie vervangen wordt door de gecodeerde tekst.

Hulpmiddelen → Scripts → Bewerking → Geselecteerde tekst URI-decoderen

Als een deel van de zoektekst van een URL is geselecteerd, dan zal dit het decoderen en de selectie vervangen door de originele ruwe tekst.

Hulpmiddelen → Scripts → Emmet

Hulpmiddelen → Scripts → Emmet → Afkortingen expanderen

Converteert de geselecteerde tekst naar een paar HTML of XML openings- en sluittags. Bijvoorbeeld, als **div** is geselecteerd, zal dit item dat vervangen door `<div></div>`.

Hulpmiddelen → Scripts → Emmet → Met tag omgeven

Omgeeft de geselecteerde tekst met de tag op de [opdrachtagregel](#).

Hulpmiddelen → Scripts → Emmet → Cursor naar overeenkomstige tag verplaatsen

Als de cursor zich binnen een HTML/XML-openings-tag bevindt, dan zal dit item deze naar de sluittag verplaatsen. Als de cursor zich binnen de sluittag bevindt, zal deze in plaats daarvan naar de openingstag verplaatsen.

Hulpmiddelen → Scripts → Emmet → Inhoud van HTML/XML-tag binnenwaarts selecteren

Wanneer de cursor zich binnen een paar HTML/XML-tags bevindt, zal deze optie de selectie wijzigen zodat de inhoud tussen die HTML/XML-tags, zonder de tags zelf, wordt geselecteerd.

Hulpmiddelen → Scripts → Emmet → Inhoud van HTML/XML-tag buitenwaarts selecteren

Wanneer de cursor zich binnen een paar HTML/XML-tags bevindt, zal dit item de selectie wijzigen zodat de inhoud tussen die HTML/XML-tags, inclusief de tags zelf, worden geselecteerd.

Hulpmiddelen → Scripts → Emmet → Commentaar omschakelen

Als het geselecteerde deel geen commentaar is, dan zal dit item dat deel in HTML/XML commentaar insluiten (bijv. `<!-- geselecteerde tekst -->`). Als het geselecteerde deel een commentaar is, zullen de commentaar-tags worden verwijderd.

ingesloten → Scripts → Emmet → Tag onder cursor verwijderen

Als de cursor nu binnen een HTML/XML-tag zich bevindt, dan zal dit item de gehele tag verwijderen.

Hulpmiddelen → Scripts → Emmet → Getal met 1 verminderen

Dit item zal één aftrekken van de nu geselecteerde tekst, als het een getal is. Als bijvoorbeeld **5** is geselecteerd, zal het **4** worden.

Hulpmiddelen → Scripts → Emmet → Getal met 10 verminderen

Dit item zal tien aftrekken van de nu geselecteerde tekst, als het een getal is. Als bijvoorbeeld **15** is geselecteerd, zal het **5** worden.

Hulpmiddelen → Scripts → Emmet → Getal met 0.1 verminderen

Dit item zal 0.1 aftrekken van de nu geselecteerde tekst, als het een getal is. Als bijvoorbeeld **4.5** is geselecteerd, zal het **4.4** worden.

Hulpmiddelen → Scripts → Emmet → Getal met 1 ophogen

Dit item zal één optellen bij de nu geselecteerde tekst, als het een getal is. Als bijvoorbeeld **5** is geselecteerd, zal het **6** worden.

Hulpmiddelen → Scripts → Emmet → Getal met 10 ophogen

Dit item zal tien optellen bij de nu geselecteerde tekst, als het een getal is. Als bijvoorbeeld **5** is geselecteerd, zal het **15** worden.

Hulpmiddelen → Scripts → Emmet → Getal met 0.1 ophogen

Dit item zal 0.1 optellen bij de nu geselecteerde tekst, als het een getal is. Als bijvoorbeeld **4.5** is geselecteerd, zal het **4.6** worden.

Hulpmiddelen → Overschakelen naar opdrachtregel (F7)

Toont de commandoregel van KatePart onderaan het venster. Typ **help** op de commandoregel om hulp te krijgen en **help list** voor een lijst met commando's. Voor meer informatie over de commandoregel, zie [De commandoregel van de bewerkingscomponent](#).

Hulpmiddelen → Code aanvullen oproepen (Ctrl+Space)

Roep handmatig commando-aanvulling aan, meestal door een sneltoets te gebruiken die verbonden is met deze actie.

Hulpmiddelen → Woordaanvulling

Woord eronder hergebruiken (Ctrl+9) en **Woord erboven hergebruiken (Ctrl+8)** vullen de huidige getypte tekst aan door vanaf de huidige cursorpositie vooruit en achteruit te zoeken naar woorden die overeenkomen. **Shell-aanvulling** geeft een lijst met overeenkomsten.

Hulpmiddelen → Spelling → Automatische spellingcontrole (Ctrl+Shift+O)

Wanneer **Automatische spellingcontrole** ingeschakeld is, wordt tijdens het typen fout gespelde tekst onderstreept.

Hulpmiddelen → Spelling → Spelling....

Dit initialiseert het programma Spellingcontrole - een programma gemaakt om de gebruiker te helpen bij het ondervangen en corrigeren van spelfouten. Door te klikken op dit item start u de spellingcontrole en opent u het dialoogvenster Spellingcontrole, waarmee u het proces kunt aansturen. Verticaal in het middel van het dialoogvenster treft u onder elkaar drie instellingen met links ervan hun overeenkomstige labels. Beginnend bovenin zijn dat:

Onbekend woord:

Hier toont de spellingcontrole het huidige woord dat nagekeken wordt. Dit gebeurt wanneer de spellingcontrole een woord tegenkomt dat niet voorkomt in het woordenboek - een bestand bestaande uit een lijst met correct gespelde woorden waarmee elk woord in de editor vergeleken wordt.

Vervangen door:

Als er in het woordenboek van de spellingcontrole vergelijkbare woorden staan, zal het eerste hier verschijnen. De gebruiker kan de suggestie accepteren, zijn of haar eigen correctie intypen, of een andere suggestie kiezen uit de keuzelijst.

Taal:

Indien u meerdere woordenboeken geïnstalleerd hebt, kunt u hier kiezen welk woordenboek/taal gebruikt moet worden.

Rechts van het dialoogvenster bevinden zich 6 knoppen waarmee de gebruiker het proces van de spellingcontrole kan aansturen. Deze zijn:

Aan woordenboek toevoegen

Door op deze knop te klikken wordt **Onbekend woord** toegevoegd aan het woordenboek van de spellingcontrole. Dit betekent dat de spellingcontrole dit woord in het vervolg altijd beschouwt als correct gespeld.

Suggestie

De spellingcontrole kan hier een aantal mogelijke woorden tonen ter vervanging van het woord dat nagekeken wordt. Door te klikken op één van de suggesties zal dat woord opgenomen worden in het invoerveld **Vervangen door**.

Vervangen

Met deze knop vervangt de spellingcontrole het woord dat nagekeken wordt in het document door het woord in het invoerveld **Vervangen door**.

Alles vervangen

Met deze knop vervangt de spellingcontrole niet alleen het huidige **Onbekend woord**; maar zorgt er automatisch voor dat ditzelfde woord overal in het document vervangen wordt.

Negeren

Door deze knop te activeren gaat de spellingcontrole verder zonder een verandering te hebben aangebracht.

Alles negeren

Deze knop zorgt ervoor dat de spellingcontrole niets doet met het huidige **Onbekend woord**; en dat geldt tevens voor ditzelfde woord elders in het document.

OPMERKING

Dit geldt alleen voor de spellingcontrole die op dat moment loopt. Wanneer de spellingcontrole later opnieuw gestart wordt, zal die weer bij ditzelfde woord stilstaan.

Horizontaal onderin het dialoogvenster van de spellingcontrole bevinden zich nog drie knoppen. Deze zijn:

Help

Dit roept het KDE helpstelsel op met de hulppagina van deze dialoog.

Beëindigd

Met deze knop beëindigt u het proces van de spellingcontrole en keert u terug naar het document.

Annuleren

Met deze knop annuleert u het proces van de spellingcontrole, alle wijzigingen worden teruggedraaid en u keert terug naar het document.

Hulpmiddelen → Spelling → Spelling (vanaf cursor)....

Hiermee start u het programma spellingcontrole, maar het begint bij de cursor in plaats van aan het begin van het document.

Hulpmiddelen → Spelling → Selectie op spelling controleren....

Controleert de spelling van de huidige selectie.

Hulpmiddelen → Spelling → Woordenboek wijzigen

Toont onderaan het venster een keuzelijst met alle beschikbare woordenboeken voor de spellingcontrole. Hierdoor kunt u eenvoudig wisselen van spellingcontrole-woordenboek bijv. voor automatische spellingcontrole van tekst in verschillende talen.

4.7 De menu's Instellingen en Help

Instellingen → Kleurenthema van bewerker

Dit menu toont de beschikbare kleurschema's. U kunt hier het schema wijzigen voor de huidige weergave, om het standaard schema te wijzigen moet u de pagina [Lettertypen & kleuren](#) van de instellingendialoog gebruiken.

KatePart heeft de standaard menu van KDE **Instellingen** en **Help**, lees voor meer informatie de secties over het [Menu Instellingen](#) en [Help Menu](#) van de KDE Fundamentals.

Hoofdstuk 5

Geavanceerde hulpmiddelen voor bewerking

Anders Lund
Dominik Haumann
Vertaler/Nalezer: Tom Albers
Vertaler/Nalezer: Natalie Koning
Vertaler/Nalezer: Freek de Kruijf

5.1 Commentaar toevoegen/verwijderen

Met de commando's **Commentaar** en **Commentaar verwijderen** in het menu **Hulpmiddelen** kunt u commentaarmarkering aan geselecteerde regels toevoegen of eruit verwijderen, mits commentaren door het bestandsformaat waar u aan werkt worden ondersteund. Als er geen tekst geselecteerd is wordt de huidige regel gebruikt.

De regels over de wijze waarop commentaar ingevoegd wordt, zijn vastgelegd in de syntaxisdefinities. Als syntaxisaccentuering niet gedefinieerd is, zal commentaar invoegen/verwijderen niet mogelijk zijn.

Sommige syntaxisdefinities definiëren commentaarmarkeringen per regel, sommige definiëren markeringen die over meerdere regels kunnen lopen, en sommige definiëren beide. Indien er per regel gemarkeerd wordt, is het niet mogelijk om een selectie te markeren waarvan de laatste regel niet volledig is geselecteerd.

Wanneer commentaarmarkering per regel mogelijk is, dan zal dit de voorkeur hebben. Het voorkomt namelijk problemen met genest commentaar (commentaar binnen commentaar).

Wanneer commentaar verwijderd wordt, mogen er geen regels geselecteerd zijn die geen commentaar bevatten. Wanneer er commentaar dat over meerdere regels loopt verwijderd wordt, zullen eventuele spaties aan het begin en einde genegeerd worden.

Om een gedeelte te markeren als commentaar kiest u het menu **Hulpmiddelen** → **Commentaar**. U kunt ook de sneltoets gebruiken, de standaard is **Ctrl+D**.

Om commentaar te verwijderen kiest u het menu **Hulpmiddelen** → **Commentaar verwijderen**. U kunt ook de sneltoets gebruiken, de standaard is **Ctrl+Shift+D**.

5.2 De commandoregel in de editor

De editor-component van KatePart beschikt over een interne commandoregel, zodat u diverse commando's direct, zonder uitgebreide GUI, kunt uitvoeren. De commandoregel bevindt zich

onder het editor-gedeelte. Om hiermee te werken gebruikt u het menu **Beeld** → **Overschakelen naar commandoregel** of drukt u op **F7**. De commando's van de editor worden hieronder beschreven, extra commando's kunnen door plugins geleverd worden.

Om een commando uit te voeren, typt u het commando en daarna drukt u op de Enter-toets. Op de commandoregel verschijnt een melding over het al dan niet slagen van de opdracht. Als u de commandoregel via **F7** geopend hebt, zal deze automatisch na een paar seconden verdwijnen. Om de melding te verwijderen en een nieuw commando te typen drukt u nogmaals op **F7**.

Typ **help** om de ingebouwde help van de commandoregel te zien. Typ **help list** om een lijst met beschikbare commando's te zien. Om de help voor een specifiek commando te zien, typt u **help commando**.

De commandoregel houdt de geschiedenis van ingetypte commando's bij in een lijst, zodat u eerder getypte commando's opnieuw kunt gebruiken. Gebruik de toetsen **Up** en **Down** om door de lijst te navigeren. Wanneer een commando uit de lijst getoond wordt, is het argument-gedeelte geselecteerd, zodat u dit gemakkelijk kunt overschrijven.

5.2.1 Standaard commandoregel opdrachten

ARGUMENTTYPEN

BOOLEAN

Dit wordt gebruikt met commando's die iets aan- of uitzetten. Toegestane waarden zijn **on**, **off**, **true**, **false**, **1** of **0**.

INTEGER

Een geheel getal.

STRING

Een tekenreeks, omgeven door quotes (') of aanhalingstekens (") wanneer deze spaties bevat.

5.2.1.1 commando's om de editor in te stellen

Deze commando's worden door de editor-component geleverd en maken het mogelijk om instellingen voor het actieve document en het beeld te maken. Dit kan nuttig zijn als u andere dan de standaardinstellingen wilt gebruiken, bijvoorbeeld voor het inspringen.

set-tab-width INTEGER width

Stelt de tablengthte in op **width**.

set-indent-width INTEGER width

Stelt de breedte voor het inspringen in op **width**. Gebruik dit alleen als u met spaties inspringt.

set-word-wrap-column INTEGER width

Stelt de regellengte in op **width**. Dit wordt gebruikt als u tekst automatisch laat afbreken.

set-icon-border BOOLEAN enable

Toont of verbergt de pictogramrand.

set-folding-markers BOOLEAN enable

Toont of verbergt de vouwmarkeringen.

set-line-numbers BOOLEAN enable

Toont of verbergt de regelnummering.

set-replace-tabs BOOLEAN enable

Als deze optie ingeschakeld is, worden tabs door spaties vervangen terwijl u typt.

set-remove-trailing-space BOOLEAN enable

Als deze optie ingeschakeld is, wordt witruimte aan het einde van de regel verwijderd als de cursor naar een andere regel gaat.

set-show-tabs BOOLEAN enable

Als deze optie ingeschakeld is, worden tabs en spaties aan het einde van de regel zichtbaar gemaakt door middel van een punt.

set-show-indent BOOLEAN enable

Als deze optie ingeschakeld is, wordt inspringen zichtbaar gemaakt door middel van een verticale lijn van puntjes.

set-indent-spaces BOOLEAN enable

Als deze optie ingeschakeld is, zal de editor elk inspringniveau met `indent-width` spaties laten inspringen in plaats van met tabs.

set-mixed-indent BOOLEAN enable

Als deze optie ingeschakeld is, zal KatePart tabs en spaties gebruiken voor het inspringen. Elk inspringniveau zal het aantal spaties hebben dat bij `indent-width` is ingesteld en meerdere inspringniveaus zullen geoptimaliseerd worden door zoveel mogelijk tabs te gebruiken.

Wanneer dit commando uitgevoerd wordt, zal ook het inspringen met spaties ingeschakeld worden. Als de inspringbreedte niet gespecificeerd is, zal deze op de helft van `tab-width` voor het document ingesteld worden.

set-word-wrap BOOLEAN enable

Schakelt dynamische regelafbreking in of uit, afhankelijk van **enable**.

set-replace-tabs-save BOOLEAN enable

Als deze optie ingeschakeld is, zullen tabs vervangen worden door spaties zodra het document opgeslagen wordt.

set-remove-trailing-space-save BOOLEAN enable

Als deze optie ingeschakeld is, zullen spaties aan het einde van elke regel verwijderd worden zodra het document opgeslagen wordt.

set-indent-mode STRING name

Stelt de automatische inspringmodus in op **naam**. Als **naam** onbekend is, wordt "none" (geen) gebruikt. Geldige modi zijn 'none', 'normal', 'cstyle', 'haskell', 'lilypond', 'lisp', 'python', 'ruby' en 'xml'.

set-auto-indent BOOLEAN script

Automatisch inspringen in- of uitschakelen.

set-highlight STRING highlight

Stelt de accentuering van syntaxis in voor het document. Het argument moet een geldige naam zijn, zoals die in het menu **Hulpmiddelen** → **Accentueringmodus** staat. Bij dit commando is automatische aanvulling beschikbaar voor het argument.

reload-scripts

Alle door Kate gebruikte [JavaScript-scripts](#) herladen, inclusief indenteerders en commandoregelscripts.

set-mode STRING modus

Het bestandstypeschema voor het huidige document kiezen.

nn[oremap] STRING original STRING mapped

Laat de **originele** toetsenreeks overeenkomen met de **overeenkomstige**.

5.2.1.2 Bewerkcommando's

Deze commando's wijzigen het huidige document.

indent

Laat de geselecteerde regels of de huidige regel inspringen.

unindent

Verwijdert de inspringing van de geselecteerde regels of van de huidige regel.

cleanindent

Schoont de inspringing van de geselecteerde regels of van de huidige regels op, in overeenstemming met de instellingen in het document.

comment

Voegt commentaarmarkeringen aan de geselecteerde regels of aan de huidige regel toe, in overeenstemming met de tekstopmaak zoals die gedefinieerd is voor de syntaxisaccentuering van het document.

uncomment

Verwijdert commentaarmarkeringen van de geselecteerde regels of van de huidige regel, in overeenstemming met de tekstopmaak zoals die gedefinieerd is voor de syntaxisaccentuering van het document.

kill-line

Verwijdert de huidige regel.

replace STRING pattern STRING replacement

Vervangt tekst die overeenkomt met **pattern** door **replacement**. Als u spaties in **pattern** wilt gebruiken, moet u om **pattern** en om **replacement** enkele of dubbele aanhalingstekens zetten. Als er geen aanhalingstekens om de argumenten staan, wordt het eerste woord als **pattern** gebruikt en de rest als **replacement**. Als **replacement** geen tekens bevat, wordt elk gevonden **pattern** verwijderd.

U kunt extra opties aan de zoekopdracht meegeven door een dubbele punt toe te voegen, gevolgd door één of meer letters die elk een optie vertegenwoordigen. De zoekopdracht krijgt de vorm **replace:options pattern replacement**. Beschikbare opties zijn:

b

Zoek achterwaarts.

c

Zoek vanaf de cursorpositie.

e

Zoek alleen in de geselecteerde tekst.

r

Gebruik een reguliere expressie in de zoekopdracht. In dit geval kunt u ook **N** gebruiken, N is een getal dat het aantal overeenkomsten in de vervangende tekenreeks vertegenwoordigt.

s

De zoekopdracht is hoofdlettergevoelig.

p

Vraag om toestemming om de volgende overeenkomst te vervangen.

w

Zoek alleen naar hele woorden.

date STRING format

Voegt een datum/tijd in volgens de gespecificeerde **format**, of in de opmaak 'yyyy-MM-dd hh:mm:ss' als er geen opmaak gespecificeerd is. De mogelijke vormen van **format** en de resulterende interpretatie:

d	De dag als een getal zonder voorloopnul (1-31).
dd	De dag als een getal met voorloopnul (01-31).
ddd	De verkorte gelokaliseerde dagnaam (bijv. 'Ma'..'Zo').
dddd	De lange gelokaliseerde dagnaam (bijv. 'Maandag'..'Zondag').
M	De maand als een getal zonder voorloopnul (1-12).
MM	De maand als een getal met voorloopnul (01-12).
MMMM	De lange gelokaliseerde maandnaam (bijv. 'januari'..'december').
MMM	De verkorte gelokaliseerde maandnaam (bijv. 'Jan'..'Dec').
YY	Het jaar als een getal van twee cijfers (00-99).
YYYY	Het jaar als een getal van vier cijfers (1752-8000).
h	Het uur zonder voorloopnul (0..23 of 1..12 als AM/PM gebruikt wordt).
hh	Het uur met voorloopnul (00..23 of 01..12 als AM/PM gebruikt wordt).
m	De minuut zonder voorloopnul (0..59).
mm	De minuut met voorloopnul (00..59).
s	De seconde zonder voorloopnul (0..59).
ss	De seconde met voorloopnul (00..59).
z	De milliseconden zonder voorloopnul (0..999).
zzz	De milliseconden met voorloopnul (000..999).
AP	Gebruik AM/PM. AP wordt vervangen door "AM" of "PM".
ap	Gebruik am/pm. ap wordt vervangen door "am" of "pm".

char STRING identifier

Met behulp van dit commando kunt u tekens invoegen door het intypen van een decimaal, octaal of hexadecimaal getal. Open het dialoogvenster "Bewerkcommando's" en typ **char** : **[number]** in het invoerveld, klik daarna op **OK**.

Example 5.1 char voorbeelden

Invoer: **char : 234**

Uitvoer: ê

Invoer: **char : 0x1234**

Uitvoer: jué

s///[ig] %s///[ig]

Dit commando voert een zoeken/vervangen-opdracht op de manier van sed uit op de huidige regel of in het gehele bestand (%s///).

Kort gezegd gebeurt er het volgende: In de tekst wordt naar de eerstvolgende overeenkomst met het *zoekpatroon* gezocht, de reguliere expressie tussen de eerste en de tweede slash, en wanneer een overeenkomst gevonden is, wordt het overeenkomende gedeelte van de tekst vervangen door de expressie tussen de tweede en de laatste slash. Haakjes in het zoekpatroon (search pattern) vertegenwoordigen *terugverwijzingen*, dit betekent dat het commando onthoudt welk gedeelte overeenkomt met het gedeelte tussen de haakjes; deze tekenreeksen kunnen opnieuw in de vervangende tekst gebruikt worden, \1 verwijst naar de eerste set haakjes, \2 naar de tweede set, enzovoort.

Om naar een (of) te zoeken moet u het teken vooraf laten gaan door een backslash: \ (\)

Als u een **i** aan het einde van de expressie toevoegt, maakt de zoekopdracht geen verschil tussen hoofd- en kleine letters. Als u een **g** aan het einde toevoegt, worden alle overeenkomsten met het zoekpatroon vervangen, anders wordt alleen de eerste overeenkomst in een regel vervangen.

Example 5.2 Tekst vervangen in de huidige regel

Uw aardige compiler is zojuist gestopt met de mededeling dat de class `myClass` in regel 3902 in uw bronbestand niet gedefinieerd is.

"Kreukels nog an toe!" denkt u, het is natuurlijk `MyClass`. U gaat naar regel 3902 en in plaats van te proberen het woord in de tekst te vinden, opent u het dialoogvenster "Bewerkcommando's", u typt **s/myclass/MyClass/i**, klikt op **OK**, slaat het bestand op en compileert – ditmaal zonder fouten.

Example 5.3 Tekst vervangen in het gehele bestand

Stel dat u aan een bestand werkt waarin de naam 'Anna Jensen' verscheidene keren voorkomt. Dan hoort u van iemand dat ze pasgeleden getrouwd is met 'Harry Jones'. U wilt natuurlijk overal in het bestand 'Anna Jensen' vervangen door 'Anna Jones-Jensen'.

Open de commandoregel en typ **%s/Anna Jensen/Anna Jones-Jensen/** en druk op de Enter-toets; dat was alles.

Example 5.4 Een iets ingewikkelder voorbeeld

In dit voorbeeld gebruiken we zowel *terugverwijzingen* als *teken-class* (als u niet weet wat dit is, lees dan de uitleg hieronder).

Stel dat u de volgende regel hebt:

```
void MyClass::DoStringOps( String      &foo, String &bar, String *p, int  & ←
    a, int &b )
```

U bedenkt dat dit geen nette code is en besluit dat u het sleutelwoord `const` wilt gebruiken voor alle ‘adressen van’ argumenten, herkenbaar aan de operator `&` voor de naam van het argument. U wilt ook de witruimte (de spaties of tabs) aanpassen, zodat er overal slechts één witruimteteken tussen de woorden staat.

Open het dialoogvenster “Bewerkcommando’s” en typ: **s/\s+(\w+)\s+(\&)/ const \1 \2/g**, klik daarna op **OK**. De **g** aan het einde van de expressie zorgt ervoor dat de reguliere expressie voor elke overeenkomst opnieuw opgebouwd wordt om de *terugverwijzingen* te behouden.

Uitvoer: `void MyClass::DoStringOps( const String &foo, const String &bar, String *p, const int &a, const int &b )`

Taak volbracht! Wat is er nu gebeurd? We hebben gezocht naar witruimte (`\s+`) gevolgd door één of meer lettertekens (`\w+`) gevolgd door nog meer witruimte (`\s+`) gevolgd door een ampersand, en tegelijkertijd hebben we de lettertekens en de ampersand opgeslagen voor hergebruik bij het vervangen. Daarna hebben we het overeenkomende gedeelte van de regel vervangen door één spatie gevolgd door ‘const’ gevolgd door één spatie gevolgd door de opgeslagen lettertekens (`\1`) gevolgd door één spatie gevolgd door de opgeslagen ampersand (`\2`).

Nu waren de lettertekens op sommige plaatsen ‘String’, op andere plaatsen ‘int’, het gebruik van de tekenklasse `\w` en het bepalingsteken `+` bleek een handig hulpmiddel.

sort

De geselecteerde tekst of het gehele document sorteren.

natsort

De geselecteerde regels of het gehele document op een natuurlijke manier sorteren.

Example 5.5 sort vs. natsort

sort(a10, a1, a2) resulteert in a1, a10, a2

natsort(a10, a1, a2) resulteert in a1, a2, a10

moveLinesDown

Geselecteerde regels omlaag verplaatsen.

moveLinesUp

Geselecteerde regels omhoog verplaatsen.

uniq

Dubbele regels uit de geselecteerde tekst of het gehele document verwijderen.

rtrim

Spaties aan het eind van de regel uit de geselecteerde tekst of het gehele document verwijderen.

ltrim

Spaties aan het begin van de regel uit de geselecteerde tekst of het gehele document verwijderen.

join [STRING separator]

Geselecteerde regels of het gehele document samenvoegen. Neemt optioneel een parameter die een scheidingsteken definieert, bijvoorbeeld: **join ‘, ’**

rmblank

Alle spaties uit de geselecteerde tekst of het gehele document verwijderen.

alignon

Dit commando lijnt regels uit in het geselecteerde blok of het gehele document op de kolom gegeven door een reguliere expressie gegeven als argument.

Als u een leeg patroon geeft zal er standaard uitgelijnd worden op het eerste niet-blanke teken.

Als het patroon een vangst heeft zal het inspringen op de gevangen overeenkomst.

Voorbeelden:

alignon - zal spaties invoegen voor de eerste '-' van elke regel om ze allen uit te lijnen op dezelfde kolom.

alignon :\\s+(.) zal spaties invoegen vóór de eerste niet-blanke teken dat verschijnt na een dubbelepunt om ze allemaal uit te lijnen op dezelfde kolom.

unwrap

Alle regelafbrekingen uit de geselecteerde tekst of het gehele document ongedaan maken.

each STRING script

Met een JavaScript-functie als argument, roep die aan voor de lijst met geselecteerde regels en vervang ze met de terugkeerwaarde van die aanroep.

Example 5.6 Geselecteerde regels samenvoegen

```
each 'function(lines){return lines.join(", ")}'
```

Of, korter:

```
each 'lines.join(", ").'
```

filter STRING script

Met een JavaScript-functie als argument, roep die aan voor de lijst met geselecteerde regels en verwijder ze als de terugkeerwaarde van die aanroep 'false' is.

Example 5.7 Blanco regels verwijderen

```
filter 'function(l){return l.length > 0;}'
```

Of, korter:

```
filter 'line.length > 0'
```

map STRING script

Met een JavaScript-functie als argument, roep die aan voor de lijst met geselecteerde regels en vervang de regel met de terugkeerwaarde van die aanroep.

Example 5.8 Blanco regels verwijderen

```
map 'function(line){return line.replace(/^s+/, "");}'
```

Of, korter:

```
map 'line.replace(/^s+/, "")'
```

duplicateLinesUp

De geselecteerde regels boven de huidige selectie dupliceren.

duplicateLinesDown

De geselecteerde regels onder de huidige selectie dupliceren.

5.2.1.3 Navigatiecommando's

goto INT line

Dit commando plaatst de cursor op de gespecificeerde regel.

grep STRING pattern

Doorzoek het document met de reguliere expressie **patroon**. Voor meer informatie, zie aanhangsel A.

find STRING pattern

Dit commando plaatst de cursor bij de eerstgevonden overeenkomst met **pattern**, afhankelijk van de eventuele opties. Volgende overeenkomsten kunnen gevonden worden door het menu **Bewerken** → **Volgende zoeken** te gebruiken of door op **F3** te drukken.

Het gedrag van de zoekopdracht kan ingesteld worden door een dubbele punt gevolgd door één of meer opties aan het commando toe te voegen: **find:options pattern**. De volgende opties kunnen gebruikt worden:

- b** Zoek achterwaarts.
- c** Zoek vanaf de cursorpositie.
- e** Zoek alleen in de geselecteerde tekst.
- r** Gebruik een reguliere expressie in de zoekopdracht. In dit geval kunt u ook **N** gebruiken, N is een getal dat het aantal overeenkomsten in de vervangende tekenreeks vertegenwoordigt.
- s** De zoekopdracht is hoofdlettergevoelig.
- w** Zoek alleen naar hele woorden.

ifind STRING pattern

Met behulp van dit commando kunt u zoeken 'terwijl-u-typet'. Het gedrag van de zoekopdracht kan ingesteld worden door een dubbele punt gevolgd door één of meer opties aan het commando toe te voegen: **ifind:options patroon**. De volgende opties kunnen gebruikt worden:

- b** Zoek achterwaarts.
- r** Gebruik een reguliere expressie in de zoekopdracht.
- s** De zoekopdracht is hoofdlettergevoelig.
- c** Zoek vanaf de cursorpositie.

5.2.1.4 Opdrachten voor basis functies van de Editor (Deze hangen af van de toepassing waarin de editorcomponent gebruikt wordt)

w Het huidige document opslaan.

wa Alle huidige geopende documenten opslaan.

q	Het huidige document sluiten.
qa	Sluit alle geopende documenten.
wq	Het huidige document opslaan en sluiten.
wqa	Alle huidige open documenten opslaan en sluiten.
x	Het huidige document alleen opslaan en sluiten als het is gewijzigd.
x	Alleen alle huidige geopende documenten opslaan en sluiten als ze zijn gewijzigd.
bp	Naar het vorige document in de documentenlijst gaan.
bn	Naar het volgende document in de documentenlijst gaan.
new	Een nieuw document openen in horizontaal gesplitste weergave.
vnew	Een nieuw document openen in verticaal gesplitste weergave.
e	Het huidige document herladen als het op de schijf is gewijzigd.
enew	Een nieuw document bewerken.
print	Het printerdialogvenster openen om het huidige document af te drukken.

5.3 Code-invouwing gebruiken

Code invouwen stelt u in staat om delen van een document in de editor te verbergen, waarmee het gemakkelijker is om overzicht te houden over grote documenten. In KatePart worden de invouwbare gebieden berekend met regels gedefinieerd in de definities van syntaxis accentuering en zijn daarom alleen beschikbaar in sommige formaten - typisch broncode van programma's, XML-markup en vergelijkbaar. De meeste definities voor accentuering met ondersteuning van code invouwen laten u ook handmatig invouwbare gebieden definiëren, typisch met de **BEGIN** en **END** sleutelwoorden.

Om de functie codeinvouwen te gebruiken, activeert u de invouwmarkering met het menu-item **Beeld → Invouwmarkeringen tonen** als ze nog niet zichtbaar zijn. Het paneel voor invouwmarkeringen links op het scherm toont een grafische weergave van de invouwgebieden, met driehoekssymbolen om de mogelijke bewerking op een gegeven gebied aan te geven: een naar beneden gerichte driehoek geeft aan dat het gebied niet is ingevouwen, erop klikken zal het gebied invouwen en een naar rechts wijzende driehoek zal in plaats daarvan worden getoond.

Er zijn drie commando's die de status van invouwgebieden veranderen, zie de [menudocumentatie](#).

De ingevouwen regels worden herinnerd wanneer een bestand wordt gesloten, dus bij het opnieuw openen van het bestand zullen de ingevouwen nodes nog steeds ingevouwen zijn. Dit is ook van toepassing bij herladen.

Als u geen gebruik wilt maken van code invouwen, dan kunt u de optie **Invouwmarkering tonen (indien beschikbaar)** in de pagina [Uiterlijk](#) van de editorconfiguratie.

Hoofdstuk 6

KatePart uitbreiden

T.C. Hollingsworth
Vertaler/Nalezer: Freek de Kruijf

6.1 Inleiding

Zoals elke geavanceerde tekstbewerkercomponent biedt KatePart een aantal manieren om zijn functionaliteit uit te breiden. U kunt [eenvoudige scripts schrijven om functionaliteit toe te voegen met JavaScript](#). Tenslotte, wanneer u Kate, hebt uitgebreid, bent u welkom om [met ons mee te doen](#) en uw verbeteringen te delen met de wereld!

6.2 Werken met syntaxisaccentuering

6.2.1 Overzicht

Accentuering van syntaxis zorgt er voor dat de tekstbewerker automatisch tekst in verschillende stijlen/kleuren laat zien, afhankelijk van de functie van de tekenreeks in relatie tot het doel van het bestand. In broncode voor programma's bijvoorbeeld, kunnen besturingsstatements vet worden weergegeven, terwijl typen gegevens en commentaar een andere kleur krijgen dan de rest van de tekst. Dit verbetert de leesbaarheid van de tekst enorm en helpt de auteur dus om efficiënter en productiever te zijn.

```
Theme Repository::defaultTheme(Repository::DefaultTheme t)
{
    if (t == DarkTheme)
        return theme(QLatin1String("Breeze Dark"));
    return theme(QLatin1String("Default"));
}
```

Een C++-functie, weergegeven met syntaxisaccentuering.

```
Theme Repository::defaultTheme(Repository::DefaultTheme t)
{
    if (t == DarkTheme)
        return theme(QLatin1String("Breeze Dark"));
    return theme(QLatin1String("Default"));
}
```

Dezelfde C++-functie, weergegeven zonder accentuering.

Welke van de twee voorbeelden is gemakkelijker te lezen?

KatePart komt met een flexibel, te configureren en capabel systeem voor accentuering van de syntaxis en de standaard distributie levert definities voor een brede reeks van programmeer-, script- en markup-talen en andere formaten van tekstbestanden. Bovendien kunt u uw eigen definities leveren in eenvoudige XML-bestanden.

KatePart zal automatisch de juiste syntaxisregels detecteren wanneer u een bestand opent, gebaseerd op het MIME-type van het bestand, bepaald door zijn extensie of, als deze er geen heeft, de inhoud. Ervaart u een foute keuze, dan kunt u handmatig de te gebruiken syntaxis instellen uit het menu **Hulpmiddelen** → **Accentuering**.

De stijlen en kleuren die door elke definitie voor syntaxisaccentuering kan worden ingesteld met het tabblad [Accentuering van tekststijlen](#) van de [Instellingendialoog](#), terwijl de MIME-types en bestandsextensies waarvoor ze gebruikt zouden moeten worden, worden behandeld door het tabblad [Modi & bestandstypen](#).

OPMERKING

Syntaxisaccentuering is er om de leesbaarheid van correcte tekst te verhogen, maar u kunt het niet vertrouwen om uw tekst te valideren. Tekst markeren voor syntaxis is moeilijk, afhankelijk van het formaat dat u gebruikt en in sommige gevallen zullen de auteurs van de syntaxisregels trots zijn als 98% van de tekst juist wordt weergegeven, hoewel u meestal een zeldzame stijl nodig hebt om de onjuiste 2% tegen te komen.

6.2.2 Het systeem voor syntaxisaccentuering van KatePart

Deze sectie zal het mechanisme voor syntaxisaccentuering van KatePart in meer detail laten zien. Het is er voor u als u er iets over wilt weten of als u syntaxis-definities wilt veranderen of aanmaken.

6.2.2.1 Hoe het werkt

Wanneer u een bestand opent, is een van de eerste dingen die de tekstbewerker KatePart doet, het detecteren van welke syntaxis-definitie voor het bestand moet worden gebruikt. Terwijl de tekst van het bestand wordt ingelezen en terwijl u het intypt, zal het systeem voor syntaxisaccentuering de tekst analyseren met de regels gedefinieerd door de syntaxis-definitie en het markeren waar verschillende contexten en stijlen beginnen en eindigen.

Tijdens het typen van het document wordt de nieuwe tekst geanalyseerd en direct gemarkeerd, zodat wanneer u een teken verwijdert die is gemarkeerd als het begin van einde van een context, de stijl van de tekst er omheen overeenkomstig wijzigt.

De syntaxis-definities die door het systeem voor syntaxisaccentuering van KatePart wordt gebruikt zijn XML-bestanden, bevattende

- Regels voor detecteren van de rol van tekst, georganiseerd in contextblokken

- Lijsten met sleutelwoorden
- Definities van stijl-items

Bij het analyseren van de tekst worden de detectieregels geëvalueerd in de volgorde waarin ze zijn gedefinieerd en, als het begin van de huidige tekenreeks overeenkomt met een regel, wordt de gerelateerde context gebruikt. Het beginpunt in de tekst wordt verplaatst naar het eindpunt waarop die regel overeenkomt en een nieuwe ronde van de regels begint, beginnend in de context ingesteld door de overeenkomende regel.

6.2.2.2 Regels

De detectieregels zijn het hart van het detectiesysteem voor accentuering. Een regel is een tekenreeks, teken of [reguliere expressie](#) waartegen de tekst die wordt geanalyseerd wordt vergeleken. Het bevat informatie over welke stijl te gebruiken voor het overeenkomende deel van de tekst. Het kan de in werking zijnde context van het systeem omschakelen ofwel naar een expliciet genoemde context of naar de vorige context in gebruik bij de tekst.

Regels zijn georganiseerd in contextgroepen. Een contextgroep wordt gebruikt voor hoofdtekstconcepten in het formaat, bijvoorbeeld tekenreeksen tussen aanhalingstekens of commentaarblokken in broncode van programma's. Dit zorgt ervoor dat het accentueringssysteem niet door alle regels hoeft te doorlopen wanneer het niet noodzakelijk is en dat sommige tekenreeksen in de tekst anders kan worden behandeld afhankelijk van de huidige context.

Contexts kunnen dynamisch worden gegenereerd om het gebruik van specifieke gegevens bij toestanden in regels toe te staan.

6.2.2.3 Contextstijlen en sleutelwoorden

In sommige programmeertalen worden door de compiler (het programma dat de broncode vertaalt in een binair uitvoerbaar programma) gehele getallen anders behandeld dan drijvende-kommagetallen en er kunnen tekens zijn met een speciale betekenis in een tekenreeks tussen aanhalingstekens. In zulke gevallen, is het zinvol om ze verschillend weer te geven in hun omgeving zodat ze gemakkelijk zijn te identificeren bij het lezen van de tekst. Dus zelfs als ze speciale contexts representeren, kunnen ze als zodanig door het systeem voor accentuering van syntaxis worden gezien, zodat ze gemarkeerd kunnen worden voor een anders weergeven.

Een syntaxis-definitie kan zoveel stijlen bevatten als nodig is om de concepten van het formaat te dekken waarvoor het wordt gebruikt.

In vele formaten zijn er lijsten woorden die een specifiek concept representeren. Bijvoorbeeld in programmeertalen, zijn controlstatements een concept, namen van gegevenstypen een andere en ingebouwde functies van de taal een derde. Het accentueringssysteem voor syntaxis van KatePart kan zulke lijsten gebruiken om woorden in de tekst te detecteren en te markeren om concepten te accentueren van de tekstformaten.

6.2.2.4 Standaard stijlen

Als u in KatePart een C++ bronbestand, een Java™ bronbestand en een HTML-document opent, dan zult u zien dat zelfs wanneer de formaten verschillend zijn en dus verschillende woorden voor een speciale behandeling worden gekozen, de gebruikte kleuren hetzelfde zijn. Dit komt omdat KatePart een voorgedefinieerde lijst met standaard stijlen heeft die worden gebruikt bij de individuele definities van syntaxis.

Dit maakt het gemakkelijk om gelijkvormige concepten in verschillende tekstformaten te herkennen. Bijvoorbeeld: commentaar is in bijna elk programma, script of markup-language aanwezig en wanneer ze in alle talen op dezelfde manier worden weergegeven, dan hoeft u niet meer na te denken om ze in de tekst te herkennen.

TIP

Alle stijlen in een syntaxisdefinitie gebruiken een van de standaard stijlen. Een paar syntaxisdefinities gebruiken meer stijlen dan er standaarden zijn, dus als u een format vaak gebruikt kan het het waard zijn om de instellingendialoog te starten om te zien of sommige concepten dezelfde stijl gebruiken. Er is bijvoorbeeld slechts één standaard stijl voor tekenreeksen, maar de Perl programmeertaal werkt met twee typen tekenreeksen, u kunt het accentueren verbeteren door ze enigszins verschillend in te stellen. Alle [beschikbare standaard stijlen](#) zullen later worden verklaard.

6.2.3 Het XML-formaat voor definitie van accentuering

6.2.3.1 Overzicht

KatePart gebruikt het framework voor accentuering van `sysntaxis` uit KDE Frameworks. De standaard XML-bestanden voor accentuering geleverd met KatePart zijn standaard gecompileerd in de bibliotheek voor accentuering van `sysntaxis`.

Deze sectie is een overzicht van het formaat van 'Highlight Definition XML'. Het zal de hoofdcomponenten, hun betekenis en gebruik beschrijven op basis van een klein voorbeeld. De volgende sectie zal in detail de accentueringdetectieregels uitleggen.

De formele definitie, ook bekend als XSD kunt u vinden in [Opslagruimte voor accentuering van `sysntaxis`](#) in het bestand `taal.xsd`

Zelf gemaakte `.xml` bestanden voor definitie van accentuering zijn geplaatst in `org.kde.syntax-highlighting/syntax/` in uw gebruikersmap te vinden met `qtpaths--paths GenericDataLocation` die gebruikelijk `$HOME /.local/share/` en `/usr/share/` zijn.

Voor pakketten uit Flatpak en Snap zal de bovenstaande map niet werken omdat de gegevenslocatie verschillend is voor elke toepassing. In een Flatpak toepassing is de locatie van aangepaste XML-bestanden gewoonlijk `$HOME /.var/app/ flatpak-pakketnaam /data/org.kde.syntax-highlighting/syntax/` en in een Snap toepassing is die locatie `$HOME /snap/ snap-pakketnaam /current/.local/share/org.kde.syntax-highlighting/syntax/`.

Op Windows[®] zijn deze bestanden gelokaliseerd in `%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\syntax`. `%USERPROFILE%` wordt gewoonlijk `C:\Users\gebruiker`.

Samenvattend, voor de meeste configuraties is de map met aangepaste XML-bestanden als volgt:

Voor lokale gebruiker	<code>\$HOME /.local/share/org.kde.syntax-highlighting/syntax/</code>
Voor alle gebruikers	<code>/usr/share/org.kde.syntax-highlighting/syntax/</code>
Voor Flatpak pakketten	<code>\$HOME /.var/app/ flatpak-package-name /data/org.kde.syntax-highlighting/syntax/</code>
Voor Snap pakketten	<code>\$HOME /snap/ snap-package-name /current/.local/share/org.kde.syntax-highlighting/syntax/</code>
Op Windows [®]	<code>%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\syntax</code>
Op macOS [®]	<code>\$HOME /Library/Application Support/org.kde.syntax-highlighting/syntax/</code>

Als meerdere bestanden bestaan voor dezelfde taal zal het bestand met het hoogste **versie**-attribuut in het **taal**-element worden geladen.

HOOFDSECTIES VAN BESTANDEN VOOR ACCENTUERINGSDEFINITIES VAN KATEPART

Een bestand voor accentuering bevat een kop die de XML-versie instelt:

```
<?xml version="1.0" encoding="UTF-8"?>
```

De basis van het definitiebestand is het element **language**. Beschikbare attributen zijn:

Vereiste attributen:

name stelt de naam van de taal in. Het verschijnt nadien in de menu's en dialogen.

section specificeert de categorie.

extensions definieert bestandsextensies, zoals `"*.cpp;*.h"`

versie specificeert de huidige revisie van het definitiebestand in termen van een geheel getal. Wanneer u een geaccentueerd definitiebestand wijzigt, verzeker u ervan dat dit getal wordt verhoogt.

kateversion specificeert de laatst ondersteunde versie van KatePart.

Optionele attributen:

mimetype associeert het MIME-type van bestanden.

casesensitive definieert of de sleutelwoorden gevoelig zijn voor hoofd- en kleine letters.

priority is nodig als een andere bestand met accentueringsdefinities dezelfde extensies gebruiken. De hogere prioriteit wint.

author bevat de naam van de auteur en zijn/haar e-mailadres.

license bevat de licentie, gewoonlijk de MIT licentie voor nieuwe bestanden met accentuering van syntaxis.

style bevat de geleverde taal en wordt gebruikt door de indenteerders voor het attribuut `required-syntax-style`.

indenter definieert welke indenteerder standaard gebruikt zal worden. Beschikbare indenterders zijn: *ada*, *normal*, *cstyle*, *cmake*, *haskell*, *latex*, *lilypond*, *lisp*, *lua*, *pascal*, *python*, *repl-code*, *ruby* en *xml*.

hidden definieert of de naam moet verschijnen in de menu's van KatePart.

De volgende regel kan er als volgt uitzien:

```
<language name="C++" version="1" kateversion="2.4" section="Sources"
  extensions="*.cpp;*.h" />
```

Vervolgens komt het element **highlighting**, die het optionele element **list** bevat en de vereiste elementen **contexts** en **itemDatas**.

Elementen **list** bevatten een lijst met sleutelwoorden. In dit geval zijn de sleutelwoorden *class* en *const*. U mag zoveel elementen 'list' toevoegen als nodig is.

Sinds KDE Frameworks 5.53 kan een lijst sleutelwoorden uit een andere lijst of taal/bestand invoegen met gebruik van het element **include**. **##** wordt gebruikt om de naam van de lijst en de naam van de taaldefinitie te scheiden, op dezelfde manier als in de regel **IncludeRules**. Dit is nuttig om duplicaten van lijsten met sleutelwoorden te vermijden, als u het nodig hebt om de sleutelwoorden van een andere taal/bestand in te voegen. De lijst *anderen naam* bevat het sleutelwoord *str* en alle sleutelwoorden van de lijst *types*, die behoren bij de taal ISO C++.

Het element **contexts** bevat alle contexten. De eerste context is per definitie het begin van de accentuering. Er zijn twee regels in de context *Normal Text*, die overeenkomt met de lijst sleutelwoorden met de naam *somenaam* en een regel die een aanhalingsteken detecteert en de context om schakelt naar *String*. Om meer over regels te leren, leest u het volgende hoofdstuk.

Het derde deel is het element **itemDatas**. Het bevat alle kleur- en lettertypestijlen nodig voor de contexten en regels. In dit voorbeeld worden **itemData** *Normal Text*, *String* en *Keyword* gebruikt.

```
<highlighting>
  <list name="eennaam">
    <item>class</item>
    <item>const</item>
  </list>
  <list name="othername">
    <item>str</item>
    <include>types##ISO C++</include>
  </list>
  <contexts>
    <context attribute="Normal Text" lineEndContext="#pop" name="↵
      Normal Text" >
      <keyword attribute="Keyword" context="#stay" String="eennaam" ↵
        />
      <keyword attribute="Keyword" context="#stay" String="anderrnaam ↵
        " />
      <DetectChar attribute="String" context="string" char="&quot;" ↵
        />
    </context>
    <context attribute="String" lineEndContext="#stay" name="string" ↵
      >
      <DetectChar attribute="String" context="#pop" char="&quot;" />
    </context>
  </contexts>
  <itemDatas>
    <itemData name="Normal Text" defStyleNum="dsNormal" />
    <itemData name="Keyword" defStyleNum="dsKeyword" />
    <itemData name="String" defStyleNum="dsString" />
  </itemDatas>
</highlighting>
```

Het laatste deel van een definitie voor accentuering is de optionele sectie **general**. Het mag informatie over sleutelwoorden, invouwen van code, commentaar, inspringen, lege regels en spellingcontrole bevatten.

De sectie **comment** definieert met welke tekenreeks een enkele regel commentaar wordt aangegeven. U kunt ook commentaar op meerdere regels definiëren met *multiLine* met het additionele attribuut *end*. Dit wordt gebruikt als de gebruiker op de bijbehorende sneltoets voor *comment/uncomment* drukt.

De sectie **keywords** definieert of lijsten met sleutelwoorden onderscheid maken tussen hoofd- en kleine letters. Andere attributen zullen later worden verklaard.

De andere secties, **folding**, **emptyLines** en **spellchecking** zijn gewoonlijk niet nodig en worden later uitgelegd.

```
<general>
  <comments>
    <comment name="singleLine" start="##"/>
    <comment name="multiLine" start="####" end="####" region="↵
      CommentFolding"/>
  </comments>
  <keywords casesensitive="1"/>
  <folding indentationsensitive="0"/>
  <emptyLines>
    <emptyLine regexpr="\s+"/>
    <emptyLine regexpr="\s*#.*/>
  </emptyLines>
  <spellchecking>
    <encoding char="á" string="\`a"/>
```

```
<encoding char="à" string="\`a"/>
</spellchecking>
</general>
</language>
```

6.2.3.2 De secties in detail

Dit deel zal alle beschikbare attributen voor contexten beschrijven, itemDatas, sleutelwoorden, commentaar, code opvouwen en inspringen.

Het element **context** behoort tot de groep **contexts**. Een context zelf definieert context specifieke regels zoals wat zou moeten gebeuren als het systeem voor accentuering het eind van een regel bereikt. Beschikbare attributen zijn:

name geeft de contextnaam aan. Regels zullen deze naam gebruiken om de context te specificeren om om te schakelen als de regel overeenkomt.

attribuut identificeert de te gebruiken stijl voor een teken wanneer geen regel overeenkomt of wanneer een regel geen attribuut specificeert. In het laatste geval zal het *attribuut* van de context gespecificeerd in de *context* van de regel worden gebruikt.

lineEndContext definieert de context waarnaar het systeem voor accentuering omschakelt als het einde van een regel wordt bereikt. Dit kan of de naam van een andere context zijn, **#stay** om de context niet om te schakelen (bijv. niets doen) of **#pop** wat de oorzaak is van het verlaten van deze context. Het is mogelijk om bijvoorbeeld **#pop#pop#pop** te gebruiken om drie keer te poppen of zelfs **#pop#pop!AndereContext** om twee keer te poppen en om te schakelen naar de context genaamd **AndereContext**. Het is ook mogelijk om om te schakelen naar een context die behoort bij een andere taaldefinitie, op dezelfde manier als in de regels **IncludeRules**, bijv. **SomeContext##JavaScript**. Omschakelen van context wordt ook beschreven in Section 6.2.4. Standaard: **#stay**.

lineEmptyContext definieert de context als een lege regel wordt bereikt. De nomenclatuur van omschakelen van context is hetzelfde als eerder beschreven in *lineEndContext*. Standaard: **#stay**.

fallthroughContext specificeert de volgende context waarnaar om te schakelen als er geen regel overeenkomt. De nomenclatuur van context omschakelen is hetzelfde als eerder beschreven in *lineEndContext*. Standaard: **#stay**.

fallthrough definieert of het accentueringssysteem naar de context schakelt gespecificeerd in **fallthroughContext** als geen regel overeenkomt. Merk op dat sinds KDE Frameworks 5.62 dit attribuut is afgekeurd met voorkeur voor **fallthroughContext**, omdat als het attribuut **fallthroughContext** aanwezig is het impliciet wordt verstaan dat de waarde van **fallthrough** *true* is. Standaard: *false*.

noIndentationBasedFolding schakelt op inspringen gebaseerd invouwen in de context uit. Als dit invouwen niet is geactiveerd, is dit attribuut nutteloos. Dit is gedefinieerd in het element *folding* van de groep *general*. Standaard: *false*.

Het element **itemData** is in de groep **itemDatas**. Het definieert de lettertypestijl en kleuren. Het is dus mogelijk om uw eigen stijlen en kleuren te definiëren. Wij bevelen echter aan om, indien mogelijk, bij de standaard stijlen te blijven zodat de gebruiker altijd dezelfde zal zien in verschillende talen. Soms is er echter geen andere manier en is het noodzakelijk de kleur- en lettertypeattributen te wijzigen. De attributen naam en defStyleNum zijn vereist, de anderen zijn optioneel. Beschikbare attributen zijn:

name stelt de naam van het itemData in. Contexten en regels zullen deze naam in hun attribuut *attribute* gebruiken om naar een itemData te verwijzen.

defStyleNum definieert welke standaard stijl te gebruiken. Beschikbare standaard stijlen worden later in detail verklaard.

color definieert een kleur. Geldige formaten zijn '#rrggbb' of '#rgb'.

selColor definieert de selectiekleur.

italic indien *true*, de tekst zal cursief worden.

bold indien *true*, de tekst zal vet worden.

underline indien *true*, de tekst zal onderstreept worden.

strikeOut indien *true*, door de tekst zal een streep worden gezet.

spellChecking indien *true*, de tekst zal gecontroleerd worden op spelling.

Het element **keywords** in de groep **general** definieert sleutelwoordeigenschappen. Beschikbare attributen zijn:

casesensitive mag *true* of *false* zijn. Indien *true*, worden alle sleutelwoorden met onderscheid tussen hoofd- en kleine letters met elkaar vergeleken

weakDelimiter is een lijst met tekens die niet als scheiding tussen woorden werken. Bijvoorbeeld: de punt ' .' is een scheider van woorden. Als een sleutelwoord in een **list** een punt bevat, dan zal die alleen overeenkomen als u de punt als een zwak scheidingsteken specificeert.

additionalDelimiter definieert extra scheidingstekens.

wordWrapDelimiter definieert tekens waarna een regel mag afbreken.

Standaard scheidingstekens en regelafbreektekens zijn de tekens . () : ! + , - < = > % & * / ; ? [] ^ { | } ~ \, spatie (' ') en tab ('\t').

Het element **comment** in de groep **comments** definieert eigenschappen van commentaar die gebruikt worden voor Hulpmiddelen → Commentaar, Hulpmiddelen → Commentaar omschakelen. Beschikbare attributen zijn:

name is ofwel *singleLine* of *multiLine*. Als u *multiLine* kiest zijn de attributen *end* en *region* vereist.

start definieert de gebruikte tekenreeks om commentaar te laten beginnen. In C++ zou dit *"/"* zijn. Dit attribute is vereist voor de typen *multiLine* en *singleLine*.

end definieert de gebruikte tekenreeks om commentaar te laten eindigen. In C++ zou dit *"*/"* zijn. Alleen dit attribuut is beschikbaar en is vereist voor commentaar van het type *multiLine*.

region moet de naam van het opvouwbare multiregel commentaar. U hebt *beginRegion= "Comment" ... endRegion= "Comment"* in uw regels, u zou *region= "Comment"* moeten gebruiken. Op deze manier werkt verwijderen zelfs als u niet alle tekst van het multiregel commentaar selecteert. De cursor hoeft alleen in het multiregel commentaar te staan. Dit attribuut is alleen beschikbaar voor het type *multiLine*.

positie definieert waar het commentaar op een enkele regel wordt ingevoegd. Standaard wordt het commentaar op een enkele regel geplaatst aan het begin van de regel op kolom 0, maar als u *position= "afterwhitespace"* gebruikt wordt het commentaar ingevoegd rechts na voorloopwitruimte, voor het eerste niet-witruimteteken. Dit is nuttig om commentaar juist in te voegen in talen waar inspringen belangrijk is, zoals Python of YAML. Dit attribuut is optioneel en de enig mogelijke waarde is *afterwhitespace*.

Het element **folding** in de groep **general** definieert de eigenschappen van het invouwen van broncode. Beschikbare attributen zijn:

De markerings voor code opvouwen, als **indentationsensitive** *true* is, zullen worden toegevoegd gebaseerd op inspringen, zoals in de scripttaal Python. Het is gewoonlijk niet nodig dit in te stellen omdat de standaard *false* is.

Het element **emptyLine** in de groep **emptyLines** definieert welke regels beschouwd moeten worden als lege regels. Dit biedt het wijzigen van het gedrag van het attribuut *lineEmptyContext* in de elementen **context**. Beschikbare attributen zijn:

regexpr definieert een reguliere expressie die behandeld zal worden als een lege regel. Standaard bevatten lege regels geen enkel teken, daarom voegt dit extra lege regels toe,

als u, bijvoorbeeld, regels met spaties ook wil beschouwen als lege regels. In de meeste syntaxisdefinities is er geen noodzaak om dit attribuut in te zetten.

Het element **encoding** in de groep **spellchecking** definieert een codering van tekens voor spellingcontrole. Beschikbare attributen:

char is een gecodeerd teken.

string is een serie tekens die gecodeerd zullen worden als het teken *char* in de spellingcontrole. In de taal LaTeX, bijvoorbeeld, representeert de tekenreeks `\~{A}` het teken **Ä**.

6.2.3.3 Beschikbare standaard stijlen

Standaard stijlen zijn [al verklaard](#), als een korte samenvatting: Standaard stijlen zijn voorgedefinieerde lettertypen- en kleurstijlen.

Algemene standaard stijlen:

dsNormal, wanneer geen speciale accentuering is vereist.

dsKeyword, ingebouwde sleutelwoorden van taal.

dsFunction, functie-aanroepen en definities.

dsVariable, indien van toepassing: namen van variabelen (bijv. `$eenVar` in PHP/Perl).

dsControlFlow, sleutelwoorden voor flowcontrol zoals `if`, `else`, `switch`, `break`, `return`, `yield`, ...

dsOperator, operatoren zoals `+`, `-`, `*`, `/`, `::`, `<`, `>`

dsFunction, ingebouwde functies, klassen en objecten.

dsExtension, gezamenlijke extensies, zoals Qt™-klassen en functies/macro's in C++ en Python.

dsPreprocessor, preprocessor statements of macro-definities.

dsAttribute, annotaties zoals `@override` en `__declspec(...)`.

Standaard stijlen gerelateerd aan tekenreeksen

dsChar, enkele tekens, zoals `'x'`.

dsSpecialChar, tekens met een speciale betekenis in tekenreeksen zoals escapes, substituties of regex-operatoren.

dsString, tekenreeksen zoals `"hello world"`.

dsVerbatimString, letterlijke of raw-tekenreeksen zoals `'raw \backslash'` in Perl, CoffeeScript en shells, evenals `r'\raw'` in Python.

dsSpecialString, SQL, regexes, HERE-docs, L^AT_EX math mode, ...

dsImport, `import`, `include`, `require` van modules.

Standaard stijlen gerelateerd aan getallen:

dsDataType, ingebouwde typen gegevens zoals `int`, `void`, `u64`.

dsDecVal, decimale waarden.

dsBaseN, waarden met een andere basis dan 10.

dsFloat, drijvende-komma waarden.

dsConstant, ingebouwde en gebruikergedefinieerde constanten zoals `pi`.

Standaard stijlen gerelateerd aan commentaar en documentatie:

dsComment, commentaar.

dsDocumentation, `/** Documentatie commentaar */` of `"""docstrings"""`.

dsAnnotation, documentatiecommando's zoals `@param`, `@brief`.

dsCommentVar, de in bovenstaande commando's gebruikte namen van variabelen, zoals `"foobar"` in `@param foobar`.

dsRegionMarker, gebiedsmarkeringen zoals `//BEGIN`, `//END` in commentaar.

Andere standaard stijlen:

dsInformation, notities en tips zoals `@note` in doxygen.

dsWarning, waarschuwingen zoals `@warning` in doxygen.

dsAlert, speciale woorden zoals `TODO`, `FIXME`, `XXXX`.

dsError, accentuering van fouten en foute syntaxis.

dsOthers, wanneer niets anders past.

6.2.4 Detectieregels accentueren

Deze paragraaf beschrijft de syntaxis van detectieregels.

Elke regel kan nul of meer tekens aan het begin van de tekenreeks bevatten, waartegen ze worden getest. Als de regel overeenkomt worden de overeenkomstige tekens toegekend aan de stijl of *attribuut* gedefinieerd door de regel en een regel kan vragen de huidige context om te schakelen.

Een regel ziet er als volgt uit:

```
<RuleName attribute="(identifier)" context="(identifier)" [regelspecifieke  
attributen] />
```

Het *attribute* identificeert de te gebruiken stijl voor overeenkomende tekens bij naam en de *context* identificeert de vanaf hier te gebruiken context.

De *context* kan worden geïdentificeerd door:

- Een *identifier*, wat de naam is van de andere context.
- Een *opdracht* die de engine vertelt om in de huidige context te blijven (**#stay**) of terug te gaan naar een eerdere context met in de tekenreeks (**#pop**). Een lege of afwezige context is gelijk aan **#stay**.
Om meer stappen terug te gaan kan het **#pop** commando worden herhaald: **#pop#pop#pop**
- Een *order* gevolgd door een uitroepteken (!) en een *identifier*, die er voor zorgt dat de engine eerst de order volgt en dan omschakelt naar de andere context, bijv. **#pop#pop!AndereContext**.
- Een *identifier*, die een contextnaam is, gevolgd door twee hashes (##) en een andere *identifier*, die de naam is van een taaldefinitie. Deze naamgeving is gelijk aan die gebruikt in **IncludeRules** regels en u toestaat om naar een context om te schakelen behorende bij een andere syntaxisaccentueringsdefinitie, bijv. **SomeContext##JavaScript**.

Specifieke attributen van regels variëren en zijn beschreven in de volgende secties.

GEZAMENLIJKE ATTRIBUTEN

- *attribute*: Een attribuut komt overeen met een gedefinieerd *itemData*. Standaard: *attribuut* uit de context gespecificeerd in attribuut *context*.

- *context*: Specificeer de context waarnaar het accentueringssysteem omschakelt als de regel overeenkomt. Standaard: #stay.
- *beginRegion*: Start een invouwblok voor code. Standaard: niet ingesteld.
- *endRegion*: Sluit een invouwblok voor code. Standaard: niet ingesteld.
- *lookAhead*: Indien *true*, zal het accentueringssysteem niet de overeenkomende lengte verwerken. Standaard: *false*.
- *firstNonSpace*: Komt alleen overeen als de tekenreeks de eerste niet-witruimte in de regel. Standaard: *false*.
- *column*: Komt alleen overeen als de kolom overeenkomt. Standaard: niet ingesteld.

DYNAMISCHE REGELS

- *dynamic*: kan (*true* | *false*) zijn.

Hoe werkt het:

In de [reguliere expressies](#) van de regels **RegExpr** wordt alle tekst binnen haakjes (**PATTERN**) gevangen en onthouden. Deze vangsten kunnen gebruikt worden in de context waarin het is gevangen, in de regels met het attribuut **dynamic true**, door *%N* (in *String*) of *N* (in *char*).

Het is belangrijk om te noemen dat een tekst gevangen in een **RegExpr** regel alleen wordt opgeslagen voor in deze context, gespecificeerd in zijn **context**-attribuut.

TIP

- Als de vangsten niet gebruikt zullen worden, zowel door dynamische regels als in dezelfde reguliere expressie, dan zullen **non-capturing groeperingen** moeten worden gebruikt: **(?:PATTERN)**
De *lookahead* or *lookbehind* groeperingen zoals **(?=PATROON)**, **(?!PATROON)** of **(?<PATROON)** worden niet gevangen. Zie voor meer informatie [Reguliere expressies](#).
- De gevangen groeperingen kunnen gebruikt worden binnen dezelfde reguliere expressie, met gebruik van respectievelijk *\N* in plaats van *%N*. Voor meer informatie, zie [Vangen van overeenkomende tekst \(achterwaartse referenties\)](#) in [Reguliere expressies](#).

Voorbeeld 1:

In dit eenvoudige voorbeeld wordt de tekst overeenkomend met de reguliere expressie **=*** gevangen en ingevoegd in *%1* in de dynamische regel. Dit biedt het commentaar om te eindigen met dezelfde hoeveelheid van **=** als aan het begin. Dit komt overeen met tekst als: **[[commentaar]]**, **[=[commentaar]=]** of **[====[commentaar]=====]**.

Bovendien zijn de vangsten alleen beschikbaar in deze context *Multi-line commentaar*.

```
<context name="Normaal" attribute="Normal Text" lineEndContext="#stay">
  <RegExpr context="Multi-line Comment" attribute="Comment" String="\[(=*) \<
    \[" beginRegion="RegionComment"/>
</context>
<context name="Multi-line Comment" attribute="Comment" lineEndContext="#<
  stay">
  <StringDetect context="#pop" attribute="Comment" String="]%1]" dynamic="<
    true" endRegion="RegionComment"/>
</context>
```

Voorbeeld 2:

In de dynamische regel komt %1 overeen met de vangst die overeenkomt met **#+** en %2 met **"+;**. Dit komt overeen met tekst als: **#label~~~~~inside the context~~~~~#**.

Deze vangsten zullen niet beschikbaar zijn in andere contexten, zoals *OtherContext*, *FindEscapes* of *SomeContext*.

```
<context name="SomeContext" attribute="Normal Text" lineEndContext="#stay">
  <RegExpr context="#pop!NamedString" attribute="String" String="(#+) (?:[\\w ←
    -]|[^[[:ascii:]])(&quot;+)" />
</context>
<context name="NamedString" attribute="String" lineEndContext="#stay">
  <RegExpr context="#pop!OtherContext" attribute="String" String="%2(?:%1) ←
    ?" dynamic="true" />
  <DetectChar context="FindEscapes" attribute="Escape" char="\" />
</context>
```

Voorbeeld 3:

Dit komt overeen met tekst zoals: **Class::function<T>(...)**.

```
<context name="Normal" attribute="Normal Text" lineEndContext="#stay">
  <RegExpr context="FunctionName" lookAhead="true"
    String="\b([a-zA-Z_][\\w-]*) (::) ([a-zA-Z_][\\w-]*) (?:&lt;[\\w-\\ ←
    s]*&gt;)?(\\()" />
</context>
<context name="FunctionName" attribute="Normal Text" lineEndContext="#pop">
  <StringDetect context="#stay" attribute="Class" String="%1" dynamic="true ←
    "/>
  <StringDetect context="#stay" attribute="Operator" String="%2" dynamic=" ←
    true"/>
  <StringDetect context="#stay" attribute="Function" String="%3" dynamic=" ←
    true"/>
  <DetectChar context="#pop" attribute="Normal Text" char="4" dynamic="true ←
    "/>
</context>
```

LOKALE SCHEIDINGSTEKENS

- *weakDelimiter*: lijst met tekens die niet werken als woordscheidingstekens.
- *additionalDelimiter* definieert extra scheidingstekens.

6.2.4.1 De regels in detail

DetectChar

Detecteert een enkele specifiek teken. Gewoonlijk gebruikt, bijvoorbeeld om het einde van een tekenreeks tussen accenten of aanhalingstekens.

```
<DetectChar char="(teken)" (common attributes) (dynamic) />
```

Het attribuut **char** definieert het teken dat moet overeenkomen.

Detect2Chars

Detecteert twee specifieke tekens in een gedefinieerde volgorde.

```
<Detect2Chars char="(teken)" char1="(teken)" (gezamenlijke attributen) ←
/>
```

Het attribuut **char** definieert het eerste teken dat moet overeenkomen, **char1** the tweede. Deze regel is aanwezig voor historische redenen en voor leesbaarheid heeft het voorkeur om **StringDetect** te gebruiken.

AnyChar

Detecteer één teken van een set van gespecificeerde tekens.

```
<AnyChar String="(tekenreeks)" (common attributes) />
```

Het attribuut **String** definieert de set tekens.

StringDetect

Detecteert een exacte tekenreeks.

```
<StringDetect String="(tekenreeks)" [insensitive="true|false"] (common ↵  
attributes) (dynamic) />
```

Het attribuut **String** definieert de overeen te komen tekenreeks. Het attribuut **insensitive** is standaard *false* en wordt doorgegeven aan de vergelijkingsfunctie van de tekenreeks. Als de waarde *true* is wordt 'insensitive' vergelijking gebruikt.

WordDetect

Detecteert een exacte tekenreeks maar vereist extra woordgrenzen zoals een punt '.' of witruimte aan het begin en eind van het woord. Denk aan **\b<tekenreeks>\b** in termen van een reguliere expressie, het is echter sneller dan de regel **RegExpr**.

```
<WordDetect String="(tekenreeks)" [insensitive="true|false"] ( ↵  
gezamenlijke attributen) (lokale scheidingstekens) />
```

Het attribuut **String** definieert de overeen te komen tekenreeks. Het attribuut **insensitive** is standaard *false* en wordt doorgegeven aan de vergelijkingsfunctie van de tekenreeks. Als de waarde *true* is wordt 'insensitive' vergelijking gebruikt.

Sinds: Kate 3.5 (KDE 4.5)

RegExpr

Komt overeen met een reguliere expressie.

```
<RegExpr String="(tekenreeks)" [insensitive="true|false"] [minimal=" ↵  
true|false"] (common attributes) (dynamic) />
```

Het attribuut **String** definieert de reguliere expressie.

insensitive is standaard *false* en wordt doorgegeven aan de engine voor de reguliere expressie.

minimal is standaard *false* en wordt doorgegeven aan de engine voor de reguliere expressie.

Omdat de regels altijd vergeleken worden met het begin van de huidige tekenreeks, zal een reguliere expressie die begint met een dakje (^) aangeven dat de regel alleen overeen moet komen met het begin van een regel.

Zie [Reguliere expressies](#) voor meer informatie hierover.

trefwoord

Detecteert een sleutelwoord uit een gespecificeerde lijst.

```
<keyword String="(lijstnaam)" (common attributes) (lokale ↵  
scheidingstekens) />
```

Het attribuut **String** identificeert de lijst met sleutelwoorden met een naam. Een lijst met die naam moet bestaan.

Het systeem voor accentuering bewerkt sleutelwoordregels op een zeer geoptimaliseerde manier. Dit maakt het absoluut noodzakelijk dat elk te vinden sleutelwoord omgeven wordt door gedefinieerde scheidingstekens, ofwel impliciet (de standaard scheidingstekens) of expliciet gespecificeerd binnen de eigenschap *additionalDelimiter* van de *sleutelwoord*-tag.

Als een te vinden sleutelwoord een scheidingsteken moet bevatten, dan moet dit teken toegevoegd worden aan de eigenschap *weakDelimiter* van de *sleutelwoord*-tag. Dit teken zal dan zijn eigenschap scheidingsteken in alle *sleutelwoord*-regels verliezen. Het is ook mogelijk om het attribuut *weakDelimiter* van *sleutelwoord* te gebruiken zo dat deze wijziging alleen op deze regel van toepassing is.

Int

Een geheel getal detecteren (als de reguliere expressie `\b[0-9]+`).

```
<Int (gezamenlijke attributen) (lokale scheidingstekens) />
```

Deze regel heeft geen specifieke attributen.

Zwevend

Een drijvende kommagetal detecteren (als de reguliere expressie: `(\b[0-9]+\.[0-9]*|\.[0-9]+)([eE][-+]?[0-9]+)?`).

```
<Drijvende komma (gezamenlijke attributen) (lokale scheidingstekens) />
```

Deze regel heeft geen specifieke attributen.

HlCOct

Een octale representatie van een getal detecteren (als de reguliere expressie: `\b0[0-7]+`).

```
<HlCOct (gezamenlijke attributen) (lokale scheidingstekens) />
```

Deze regel heeft geen specifieke attributen.

HlCHex

Een hexadecimale representatie van een hexadecimaal getal (als de reguliere expressie: `\b0[xX][0-9a-fA-F]+`).

```
<HlCHex (gezamenlijke attributen) (lokale scheidingstekens) />
```

Deze regel heeft geen specifieke attributen.

HlcStringChar

Detecteer een escaped teken.

```
<HlcStringChar (common attributes) />
```

Deze regel heeft geen specifieke attributen.

Komt overeen met de letterlijke representatie van tekens, gewoonlijk gebruikt in programmacode, bijvoorbeeld `\n` (nieuwe-regel) of `\t` (TAB).

De volgende tekens zullen overeenkomen als ze volgen op een backslash (`\`): `abefnrtv~' ?\`. Bovendien escaped hexadecimale getallen zoals bijvoorbeeld `\xff` en escaped octale getallen, zoals bijvoorbeeld `\033` zullen overeenkomen.

HlCChar

Detecteer een C-teken.

```
<HlCChar (gezamenlijke attributen) />
```

Deze regel heeft geen specifieke attributen.

Komt overeen met een C-teken tussen accenten (Bijvoorbeeld: 'c'). De accenten kunnen een eenvoudig teken zijn of een escaped teken. Zie HLCStringChar voor overeenkomende reeksen van escaped tekens.

RangeDetect

Detecteert een tekenreeks met gedefinieerde begin- en eindtekens.

```
<RangeDetect char="(teken)" char1="(teken)" (gezamenlijke attributen) ↵  
/>
```

char defines the character starting the range, **char1** the character ending the range.

Bruikbaar om bijvoorbeeld kleine tekenreeksen of zoiets te detecteren, maar merk op dat de accentiëringssysteem werkt op één regel tegelijk, hierdoor zullen tekenreeksen over meerdere regels niet gevonden worden.

LineContinue

Komt overeen met een gespecificeerd teken aan het eind van een regel.

```
<LineContinue (gezamenlijke attributen) [char="\"] />
```

char optioneel teken dat moet overeenkomen, standaard is de backslash ('\'). Nieuw sinds KDE 4.13.

Deze regel is bruikbaar voor het omschakelen van context aan het einde van een regel. Dit is bijvoorbeeld nodig in C/C++ om macro's of tekenreeksen door te laten lopen.

IncludeRules

Voeg regels in uit een ander context- of taalbestand.

```
<IncludeRules context="contextlink" [includeAttrib="true|false"] />
```

Het attribuut **context** definieert welke context in te voegen.

Als het een eenvoudige tekenreeks is dan zijn alle gedefinieerde regels in de huidige context ingevoegd, bijvoorbeeld:

```
<IncludeRules context="anotherContext" />
```

Als de tekenreeks **##** bevat zal het accentueringssysteem zoeken naar een context uit een andere taaldefinitie met de gegeven naam, bijvoorbeeld

```
<IncludeRules context="String##C++" />
```

zou de context *String* uit de accentueringsdefinitie van C++ invoegen.

Als attribuut **includeAttrib** *true* is, dan wijzigt het doelattribuut naar die van de bron. Dit is vereist om bijvoorbeeld het aanbrengen van commentaar te laten werken als tekst die overeenkomt met de ingevoegde context een verschillende accentuering is van de host-context.

DetectSpaces

Detecteert witruimte.

```
<DetectSpaces (gezamenlijke attributen) />
```

Deze regel heeft geen specifieke attributen.

Gebruik deze regel als u weet dat er verschillende witruimtes aankomen, bijvoorbeeld aan het begin van inspringende regels. Deze regel slaat alle witruimte in een keer over, in plaats van testen met meerdere regels en een voor een overslaan vanwege geen overeenkomst.

DetectIdentifier

Tekenreeksen voor een identifieer detecteren (als de reguliere expressie: **[a-zA-Z_][a-zA-Z0-9_]***).

```
<DetectIdentifier (gezamenlijke attributen) />
```

Deze regel heeft geen specifieke attributen.

Gebruik deze regel om een reeks woordtekens in een keer over te slaan, in plaats van deze te testen met meerdere regels en ze een voor een over te slaan omdat er geen overeenkomst is.

6.2.4.2 Tips & trucs

- Reguliere expressies zijn gemakkelijk te gebruiken maar vaak is er een andere veel snellere manier om hetzelfde resultaat te behalen. U wilt alleen een overeenkomst vinden met het teken '#' als het eerste teken is in de regel. Een oplossing gebaseerd op een reguliere expressie zou er zo uit zien:

```
<Regex attribute="Macro" context="macro" String="\s*#" />
```

. U kunt hetzelfde veel sneller bereiken door te gebruiken:

```
<DetectChar attribute="Macro" context="macro" char="#" firstNonSpace="↔ true" />
```

. Als u de reguliere expressie '^#' wilt laten vereen komen dan kunt u nog steeds **DetectChar** gebruiken met het attribuut **column="0"**. Het attribuut **column** telt tekens zodat een tab nog één teken is.

- Gebruik in regels **Regex** het attribuut **column="0"** als het patroon **^PATROON** gebruikt zal worden om met tekst overeen te komen aan het begin van een regel. Dit verbetert de prestaties, omdat het niet zal zoeken naar overeenkomsten in de rest van de kolommen.
- Gebruik in reguliere expressies non-capturing groeperingen (**?:PATROON**) in plaats van vangstgrouperingen (**(PATROON)**), als de vangsten niet gebruikt zullen worden in dezelfde reguliere expressie of in dynamische regels. Dit vermijdt het onnodig opslaan van vangsten.
- U kunt van contexts wisselen zonder tekens te bewerken. Neem aan dat u van context wilt wisselen wanneer u de tekenreeks ***/** tegenkomt, maar u moet die tekenreeks in de volgende context bewerken. De onderstaande regel zal overeenkomen en het attribuut **lookAhead** zal de overeenkomende tekenreeks voor accentuering bewaren voor de volgende context.

```
<StringDetect attribute="Comment" context="#pop" String="*/" lookAhead="↔ true" />
```

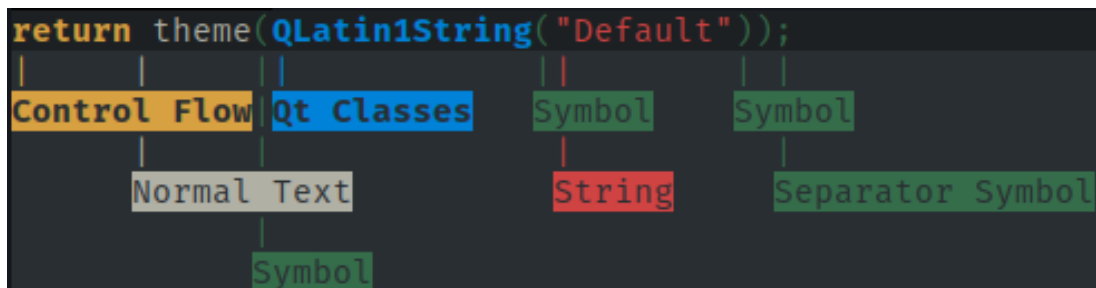
- Gebruik **DetectSpaces** als u weet dat er veel witruimtes zijn.
- Gebruik **DetectIdentifier** in plaats van de reguliere expressie **'[a-zA-Z_]\w*'**.
- Gebruik standaard stijlen wanneer dat kan. Op deze manier vindt de gebruiker een bekende omgeving.
- Kijk in andere XML-bestanden om te zien hoe andere mensen ingewikkelde regels implementeren.

- U kunt elk XML-bestand valideren met de opdracht **validatehl.sh mySyntax.xml**. Het bestand `validatehl.sh` gebruikt `language.xsd` die beiden beschikbaar zijn in [Opslagruimte voor accentuering van syntaxis](#).
- Als u complexe reguliere expressies erg vaak gebruikt dan kunt u *ENTITIES* gebruiken. Voorbeeld:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE language
[
    <!ENTITY myref      "[A-Za-z_:] [\w.:\_~]*">
]>
```

Nu kunt u *&myref;* gebruiken in plaats van de reguliere expressie.

- In de bewerkter Kate kunt u syntaxes herladen met de ingebouwde opdrachtregel (**F7** standaard sneltoets) en de opdracht **reload-highlighting**.
- U kunt het hulpmiddel op de opdrachtregel genaamd **ksyntaxhighlighter6** gebruiken (**kate-syntax-highlighter** op oudere versies) om een syntaxis te testen en de stijl en gebieden geassocieerd met elk deel van een tekst te tonen.



Resultaat van `ksyntaxhighlighter6 --output-format=ansi --syntax-trace=format test.cpp`.
Gebruik `ksyntaxhighlighter6 -h` voor meer opties.

6.3 Met kleuren thema's werken

6.3.1 Overzicht

Kleurenschema's definiëren de kleuren van het [tekstbewerkingsgebied](#) en de [syntaxisaccentuering](#). Een kleuren thema omvat het volgende:

- De tekststijl, gebruikt voor syntaxisaccentuering via de *standaard stijlen attributen*. Bijvoorbeeld, de tekstkleur en de geselecteerde tekstkleur.
- De achtergrond van het tekstbewerdersgebied, inclusief de tekstselectie en de huidige regel.
- De pictogramrand van het tekstgebied: zijn achtergrond, de scheidingslijn, de regelnummers, de regelafbreekmarkering, de markering van gewijzigde regels en de ingevouwen code.
- Tekstdecoratie zoals de zoekmarkeringen, de indentatie en markering van tab/spatie in regels, de bij elkaar behorende haakjes en de spellingcontrole.
- Bladwijzers en tekstfragmenten.

Om verwarring te voorkomen, het volgende valt buiten het onderwerp:

- Het lettertype en de lettergrootte.

- De kleuren van de tekstbewerkertoepassing, zoals de schuifbalkplaats, de menu's, de tabblad-balk, de vensterkleur, etc. In KDE toepassingen, zoals Kate of KDevelop, zijn deze kleuren gedefinieerd door het **globale kleurenschema van KDE Plasma**, die zijn gezet in de module [Kleuren in Systeeminstellingen](#) of uit de toepassing zelf in het menu **Instellingen** → **Kleurenschema**.

```

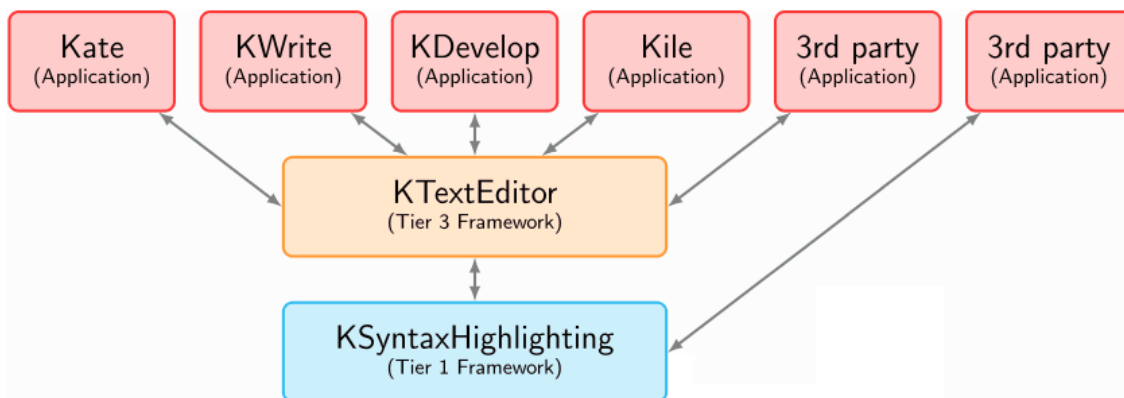
1  /**
2   * SPDX-FileCopyrightText: 2020 Christoph Cullmann <cullmann@kde.org>
3   * SPDX-License-Identifier: MIT
4   */
5
6  // BEGIN
7  #include <string>
8  #include <QString>
9  // END
10
11 /**
12  * TODO: improve documentation
13  * @param magicArgument some magic argument
14  * @return magic return value
15  */
16 int main(uint64_t magicArgument)
17 {
18     if (magicArgument > 1) {
19         const std::string string = "source file: \"__FILE__\"";
20         const QString qString(QStringLiteral("test"));
21         return qrand();
22     }
23
24     /* BUG: bogus integer constant inside next line */
25     const double g = 1.1e12 * 0b01'01'01'01 - 43a + 0x11234 * 0234ULL - 'c' * 42;
26     return g > 1.3f;
27 }

```

‘Breeze licht’ en ‘Breeze donker’ kleurenthema’s met de ‘C++’ syntaxisaccentuering.

6.3.2 De kleurenthema’s KSyntaxHighlighting

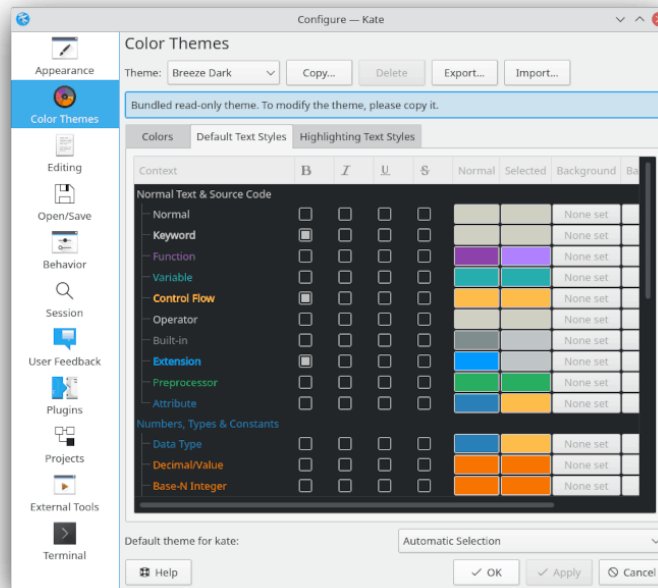
Het framework [KSyntaxHighlighting](#), wat de [syntaxisaccentuering](#) engine is, is de bibliotheek die de **kleurenthema’s levert en beheert**. Dit is onderdeel van KDE Frameworks en wordt in KDE tekstbewerkers gebruikt zoals Kate, KWrite, Kile en KDevelop. Deze afhankelijkheid ziet er uit als volgt:



Afhankelijk van KDE Frameworks bibliotheken op tekstbewerkers.

KSyntaxHighlighting omvat een variëteit van ingebouwde thema’s die getoond worden op de pagina [Kleurenthema’s van de website van de tekstbewerker Kate](#).

Het framework [KTextEditor](#), wat de tekstbewerkers-engine is, levert een gebruikersinterface voor aanmaken en bewerken van kleurenthema's, inclusief een hulpmiddel voor importeren en exporteren van thema's. Dit is de eenvoudigste manier om ze aan te maken en te bewerken, u kunt er toegang toe krijgen vanuit de dialoog 'Configureren' van de tekstbewerker. Meer details in [Section 6.3.5](#).



De GUI voor beheer van kleurenthema's in instellingen van Kate.

Het is belangrijk om te noemen dat in de KDE tekstbewerkers zoals Kate of KDevelop, de kleurenthema's van [KSyntaxHighlighting](#) gebruikt worden [sinds KDE Frameworks 5.75](#), uitgegeven op 10 oktober, 2020. Eerder werden de kleurenschema's van Kate (op KConfig gebaseerde schemaconfiguratie) gebruikt en dezen zijn nu afgekeurd. Het is echter mogelijk om de oude schema's van Kate te converteren naar de kleurenthema's van [KSyntaxHighlighting](#). The [KSyntaxHighlighting opslagruimte](#) bevat het script `utils/kateschema_to_theme_converter.py` en het programma `utils/schema-converter/` voor dat doel.

6.3.3 De kleurenthema's in JSON-formaat

6.3.3.1 Overzicht

Kleurenthema's worden opgeslagen in bestanden in JSON-formaat, met de extensie `.theme`.

In de [broncode van KSyntaxHighlighting](#) zijn de JSON bestanden van ingebouwde thema's gelokaliseerd in de map `data/themes/`. Merk op dat in tekstbewerkers, de ingebouwde thema's gecompileerd zijn in de bibliotheek van [KSyntaxHighlighting](#), de manier van toegang naar hen is via de broncode of door [ze exporteren uit de GUI om thema's van KTextEditor te beheren](#).

Het is ook mogelijk om gemakkelijk extra of aangepaste thema's toe te voegen, die geladen worden uit het bestandssysteem. Bestanden met door de gebruiker aangepaste themabestanden zijn gelokaliseerd in de map `org.kde.syntax-highlighting/themes/` in uw gebruikersmap, die u kunt vinden met het commando `qtpaths --paths GenericDataLocation` en zijn gewoonlijk `$HOME/.local/share/` en `/usr/share/`.

Voor pakketten uit Flatpak en Snap zal de bovenstaande map niet werken omdat de gegevenslocatie verschillend is voor elke toepassing. In een Flatpak toepassing is de locatie van aangepaste

themabestanden gewoonlijk `$HOME / .var/app/ flatpak-pakketnaam /data/org.kde.syntax-highlighting/themes/` en in een Snap toepassing is die locatie `$HOME /snap/ snap-pakketnaam /current/.local/share/org.kde.syntax-highlighting/themes/`.

In Windows[®] zijn deze bestanden gelokaliseerd in `%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\themes`. `%USERPROFILE%` wordt gewoonlijk `C:\Users\gebruikersnaam`.

Samenvattend, voor de meeste configuraties is de aangepaste map met thema's als volgt:

Voor lokale gebruiker	<code>\$HOME / .local/share/org.kde.syntax-highlighting/themes/</code>
Voor alle gebruikers	<code>/usr/share/org.kde.syntax-highlighting/themes/</code>
Voor Flatpak pakketten	<code>\$HOME / .var/app/ flatpak-package-name /data/org.kde.syntax-highlighting/themes/</code>
Voor Snap pakketten	<code>\$HOME /snap/ snap-package-name /current/.local/share/org.kde.syntax-highlighting/themes/</code>
Op Windows [®]	<code>%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\themes</code>
Op macOS [®]	<code>\$HOME /Library/Application Support/org.kde.syntax-highlighting/themes/</code>

Als meerdere themabestanden bestaan met dezelfde naam, zal het bestand met de hoogste **revisie** worden geladen.

6.3.3.2 De JSON-structuur

De structuur van een JSON-bestand wordt uitgelegd op [hun website](#). In de basis bestaat een JSON-bestandsformaat uit:

- Verzamelingen van paren sleutel/waarde, gescheiden door komma's en gegroepeerd in `{ }` die we 'objects' zullen noemen.
- Geordende lijsten met waarden, gescheiden door komma's en gegroepeerd in `[ ]` die we 'array' zullen noemen.

De naamgeving 'sleutel', 'waarde', 'object' en 'array' zullen in dit artikel gebruikt worden. Als dit uw eerste keer is van werken met JSON-bestanden, ze begrijpen is zo eenvoudig als naar de onderstaande voorbeelden kijken.

6.3.3.3 Hoofdsecties van bestanden voor de JSON-kleurenthema's

Het hoofdobject van het JSON-bestand met kleurenthema bevat de volgende schemasleutels:

- **metadata**: is verplicht. De waarde is een object met de metagegevens van het thema, zoals naam, revisie en licentie.
Dit is gedetailleerd in Section [6.3.3.4](#).
- **editor-colors**: het is verplicht. De waarde is een object met de kleuren van het tekstbewerkingsgebied, zoals de achtergrond, de pictogramrand en de tekstdecoratie.
Dit is gedetailleerd in Section [6.3.4.1](#).

- **text-styles**: het is verplicht. De waarde is een object met de attributen *standaard tekststijl* van de syntaxisaccentuering. Elk attribuut definieert zijn *tekstkleur*, zijn *geselecteerde tekstkleur* of of het bijvoorbeeld *vet* of *cursief* moet zijn. De tekststijlen kunnen gerefereerd worden uit [de attributen van de XML bestanden met syntaxisdefinities](#).

Dit is gedetailleerd in Section 6.3.4.2.

- **custom-styles**: het is optioneel. Definieert tekststijlen voor de attributen van specifieke definities voor syntaxisaccentuering. Bijvoorbeeld, in een definitie voor accentuering zoals **Python** of **Markdown** kunt u een andere tekststijl specificeren die de standaard gedefinieerde in **text-styles** overstijgt.

Dit is gedetailleerd in Section 6.3.4.3.

De taal JSON ondersteunt geen commentaar. U kunt echter de optionele sleutel **_comments** in het hoofdobject gebruiken om commentaar te schrijven, bijvoorbeeld, als u een bestaand thema aanpast kunt u de URL van de originele opslagruimte er in zetten. De meest praktische manier is om een array van tekens te gebruiken.

Hier onder is een voorbeeldbestand voor het thema 'Breeze Light'. U kunt zien dat, om te vermijden dat het voorbeeld te groot is, dat de **kleuren van de bewerk** en objecten met **tekststijlen** niet alle vereiste sleutels bevatten. U kunt het volledige archief van [het thema Breeze Light](#) zien in de opslagruimte [KSyntaxHighlighting](#).

```
{
  "_comments": [
    "Dit is commentaar.",
    "Als dit thema een aanpassing is van een ander, zet de koppeling ↔
      naar de originele opslagruimte hier."
  ],
  "metadata": {
    "name" : "Breeze Light",
    "revision" : 5,
    "copyright": [
      "SPDX-FileCopyrightText: 2016 Volker Krause <vkrause@kde.org>",
      "SPDX-FileCopyrightText: 2016 Dominik Haumann <dhaumann@kde.org ↔
        >"
    ],
    "license": "SPDX-License-Identifier: MIT"
  },
  "editor-colors": {
    "BackgroundColor" : "#ffffff",
    "CodeFolding" : "#94caef",
    "BracketMatching" : "#ffff00",
    "CurrentLine" : "#f8f7f6",
    "IconBorder" : "#f0f0f0",
    "IndentationLine" : "#d2d2d2",
    "LineNumbers" : "#a0a0a0",
    "CurrentLineNumber" : "#1e1e1e",
    "De andere kleursleutels van de bewerk..."
  },
  "text-styles": {
    "Normal": {
      "text-color" : "#1f1c1b",
      "selected-text-color" : "#ffffff",
      "bold" : false,
      "italic" : false,
      "underline" : false,
      "strike-through" : false
    }
  }
}
```

```
    "Keyword" : {
      "text-color" : "#1f1c1b",
      "selected-text-color" : "#ffffff",
      "bold" : true
    },
    "Function" : {
      "text-color" : "#644a9b",
      "selected-text-color" : "#452886"
    },
    "Variable" : {
      "text-color" : "#0057ae",
      "selected-text-color" : "#00316e"
    },
    De andere tekststijlsleutels...
  },
  "custom-styles": {
    "ISO C++": {
      "Data Type": {
        "bold": true,
        "selected-text-color": "#009183",
        "text-color": "#00b5cf"
      },
      "Keyword": {
        "text-color": "#6431b3"
      }
    },
    "YAML": {
      "Attribute": {
        "selected-text-color": "#00b5cf",
        "text-color": "#00b5cf"
      }
    }
  }
}
```

6.3.3.4 Metagegevens

Het JSON-object van de **metagegevens** sleutel bevat relevante informatie over het thema. Dit object heeft de volgende sleutels:

- **name:** Het is een *tekenreeks* die de naam van de taal instelt. Het verschijnt nadien in de menu's en dialogen. Het is verplicht.
- **revisie** Het is een *geheel getal* dat de huidige revisie specificeert van het themabestand. Wanneer u een kleurenthemabestand bijwerkt, verzeker u ervan dat dit getal wordt verhoogt. Het is verplicht.
- **license:** het is een *tekenreeks* die de licentie van het thema definieert, met gebruik van de identifier **SPDX-License-Identifier** uit de standaard [SPDX licentiecommunicatieformaat](#). Het is optioneel.

U kunt de volledige lijst met SPDX-licentie-identifiers [hier](#) zien.

- **copyright:** het is een *array* van *tekenreeksen* die de auteurs van het thema specificeren, met gebruik van de identifier **SPDX-FileCopyrightText** uit de standaard [SPDX licentiecommunicatieformaat](#). Het is optioneel.

```
"metadata": {
  "name" : "Breeze Light",
  "revision" : 5,
  "copyright": [
    "SPDX-FileCopyrightText: 2016 Volker Krause <vkrause@kde.org>",
    "SPDX-FileCopyrightText: 2016 Dominik Haumann <dhaumann@kde.org>"
  ],
  "license": "SPDX-License-Identifier: MIT"
}
```

6.3.4 Kleuren in detail

Deze sectie detailleert alle beschikbare kleurattributen en beschikbare kleurinstellingen.

6.3.4.1 Kleuren van bewerker

Komt overeen met de kleuren van het [tekstbewerkingsgebied](#).

In het [JSON themabestand](#), heeft de respectievelijke sleutel **editor-colors** als waarde een *object* waar elke sleutel refereert naar een kleurattribuut van de tekstbewerker. Hier zijn **alle beschikbare sleutels verplicht**, hun waarden zijn **tekenreeksen** met hexadecimale kleurcodes, zoals '#00B5CF'.

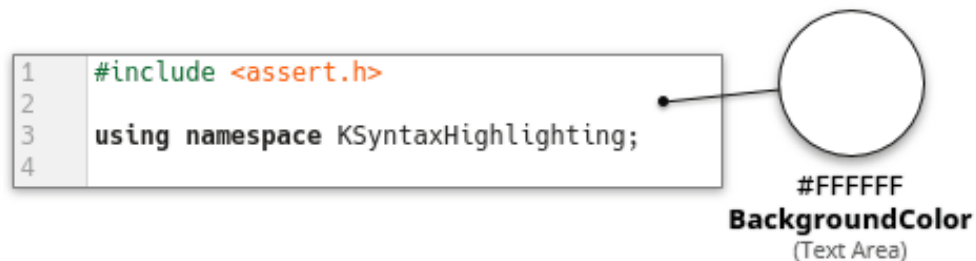
In de [GUI om thema's van KTextEditor te beheren](#) kunnen deze attributen gewijzigd worden in het tabblad **Kleuren**.

De beschikbare sleutels zijn als volgt; de sleutels gebruikt in het [JSON bestand](#) staan in de lijst *vet*, de namen gebruikt in de [GUI](#) worden getoond in ronde haakjes.

Achtergrondkleuren van bewerker

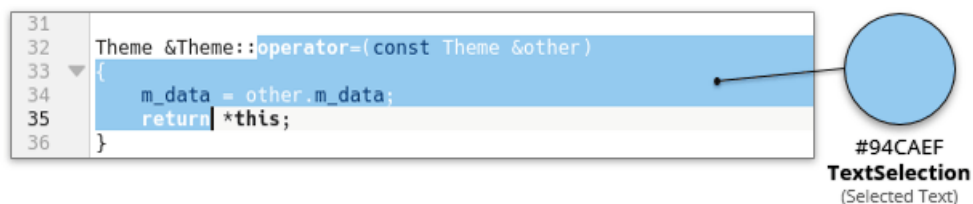
BackgroundColor (Tekstgebied)

Dit is de standaard achtergrond voor het tekstgebied. Het is de dominante kleur in de editor.



TextSelection (Geselecteerde tekst)

Dit is de achtergrond voor geselecteerde tekst.



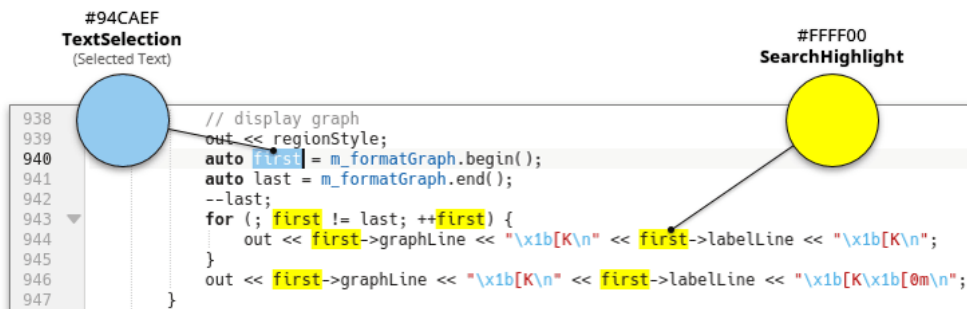
CurrentLine (Huidige regel)

Stel de kleur in voor de huidige regel. Door de kleur iets te laten afwijken van de normale tekst wordt de focus op deze regel gericht.



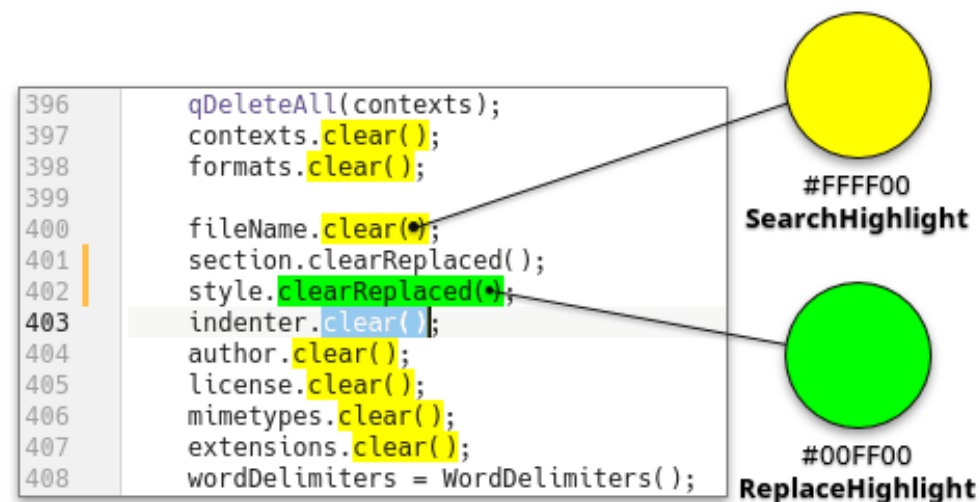
SearchHighlight (Accentuering van gezocht)

Stel de kleur in voor de tekst die overeenkomt met uw laatste zoekopdracht.



ReplaceHighlight (Accentuering van vervanging)

Stel de kleur in voor de tekst die overeenkomt met uw laatste vervanging.



IconBorder (Achtergrondgebied)

Deze kleur wordt gebruikt voor de randen van markeringen, regelnummers en in-vouwmarkeringen links van de editorweergave als die getoond worden.

LineNumbers (Regelnummers)

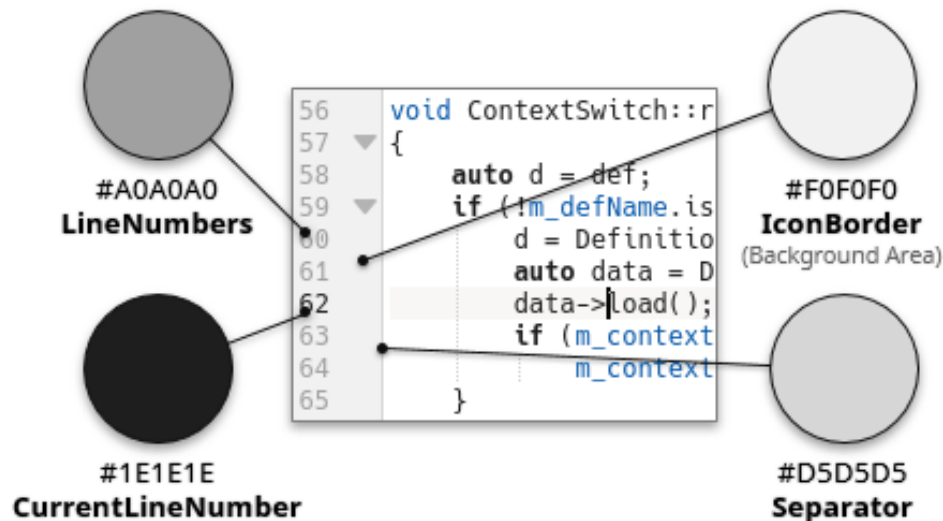
Deze kleur wordt gebruikt voor de regelnummers links van de weergave als die getoond worden.

CurrentLineNumber (Huidige regelnummers)

Deze kleur wordt gebruikt om het regelnummer van de huidige regel aan de linkerkant van de weergave te tekenen, indien getoond. Dit een beetje anders instellen dan 'LineNumbers' helpt bij het gericht blijven op de huidige regel.

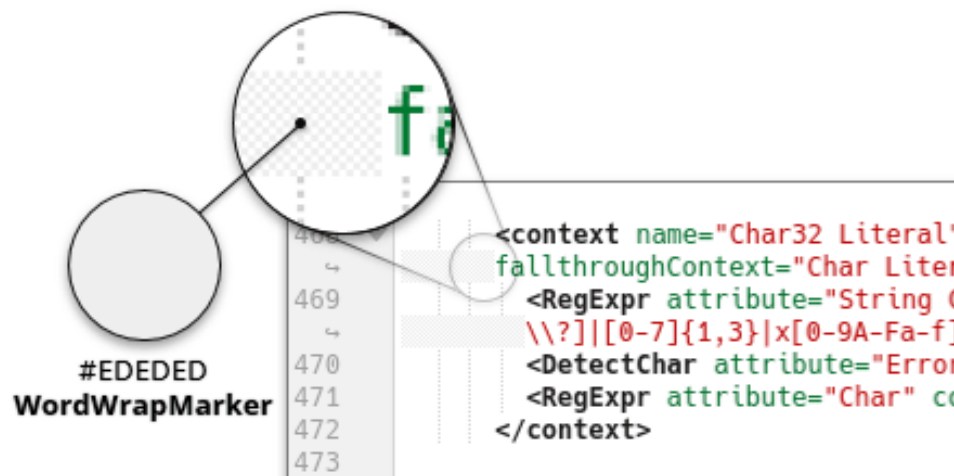
Scheidingsteken (Scheidingsteken)

Deze kleur wordt gebruikt om de verticale lijn te tekenen die de pictogramrand scheidt van de achtergrond van het tekstgebied.



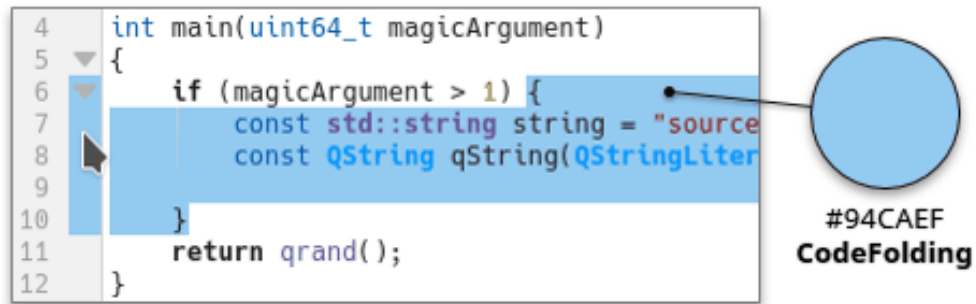
WordWrapMarker (Markering van regelafbreking)

Deze kleur wordt gebruikt om een patroon te tekenen links van dynamisch afgebroken regels wanneer die verticaal uitgelijnd zijn, evenals voor de statische regelafbrekingmarkering.



CodeFolding (Code invouwen)

Deze kleur wordt gebruikt om de sectie met code te accentueren die zou worden ingevouwen wanneer u met de muis klikt op de codeinvouwpijl links van een document. Voor meer informatie, zie [de documentatie over code invouwen](#).



ModifiedLines (Gewijzigde regels)

Deze kleur wordt gebruikt om links van een document regels te accentueren die gewijzigd zijn, maar nog niet opgeslagen. Voor meer informatie, zie Section 3.9.

SavedLines (Opgeslagen regels)

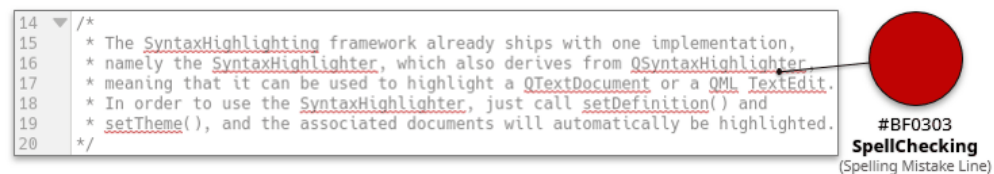
Deze kleur wordt gebruikt om links van een document regels te accentueren die in deze sessie gewijzigd en opgeslagen zijn. Voor meer informatie, zie Section 3.9.



Tekstdecoraties

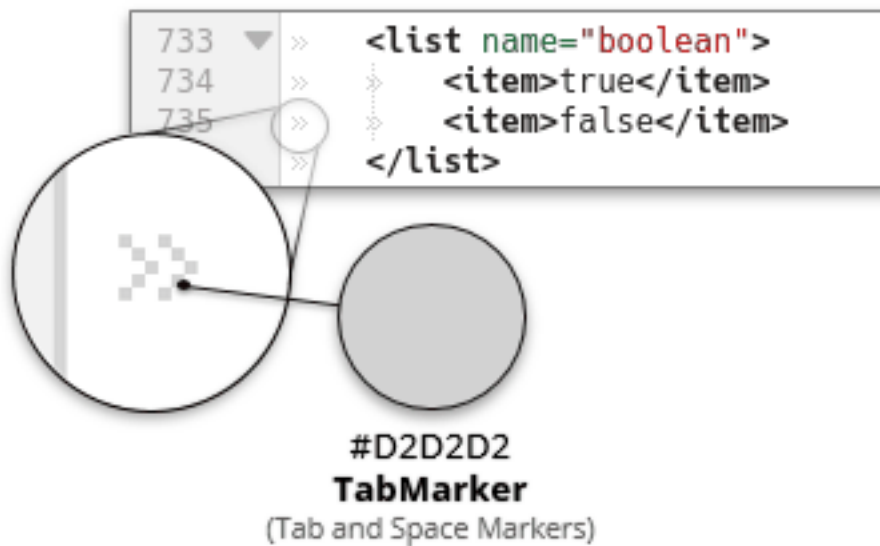
SpellChecking (Regel met verkeerde spelling)

Deze kleur wordt gebruikt om spelfouten aan te geven.



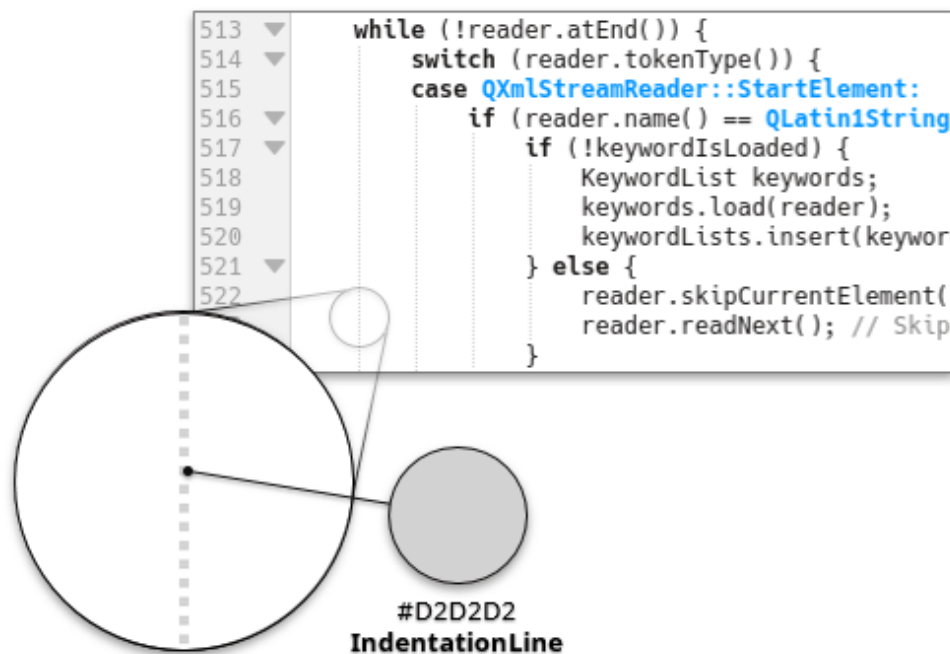
TabMarker (Tab en spatie markering)

Deze kleur wordt gebruikt om indicators in de witruimte te tekenen, als die ingeschakeld zijn.



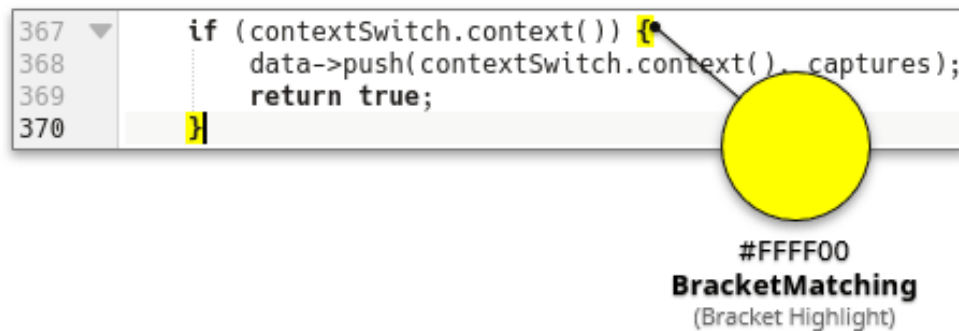
IndentationLine (Inspringlijn)

Deze kleur wordt gebruikt om een lijn links van de inspringende blokken te tekenen als die optie geactiveerd is.



BracketMatching (Accentuering van overeenkomende haakjes)

Deze kleur wordt gebruikt voor de achtergrond van bij elkaar horende haakjes.



Kleuren van markeringen

MarkBookmark (Bladwijzer)

Deze kleur wordt gebruikt om bladwijzers aan te geven. Merk op dat deze kleur een dekking heeft van 22% (en 33% voor de huidige regel) met betrekking tot de achtergrond. Voor meer informatie, zie [Section 3.6](#).



MarkBreakpointActive (Actief breekpunt)

Deze kleur wordt gebruikt door de GDB-plug-in om een actief breekpunt aan te geven. Merk op dat de kleur een mindere dekking heeft ten opzichte van de achtergrond. Voor meer informatie, zie [de documentatie van de GDB-plug-in](#).

MarkBreakpointReached (Bereikt breekpunt)

Deze kleur wordt gebruikt door de GDB-plug-in om een actief breekpunt aan te geven dat is bereikt tijdens debuggen. Merk op dat de kleur een mindere dekking heeft ten opzichte van de achtergrond. Voor meer informatie, zie [de documentatie van de GDB-plug-in](#).

MarkBreakpointDisabled (Uitgeschakeld breekpunt)

Deze kleur wordt gebruikt door de GDB-plug-in om een inactief breekpunt aan te geven. Merk op dat de kleur een mindere dekking heeft ten opzichte van de achtergrond. Voor meer informatie, zie [de documentatie van de GDB-plug-in](#).

MarkExecution (Uitvoering)

Deze kleur wordt gebruikt door de GDB-plug-in om een regel aan te geven die wordt uitgevoerd. Merk op dat de kleur een mindere dekking heeft ten opzichte van de achtergrond. Voor meer informatie, zie [de documentatie van de GDB-plug-in](#).

MarkeringsWaarschuwing (Waarschuwing)

Deze kleur wordt gebruikt door de build-plug-in om een regel aan te geven die een waarschuwing bij compilatie heeft veroorzaakt. Merk op dat de kleur een mindere dekking heeft ten opzichte van de achtergrond. Voor meer informatie, zie [de documentatie van de Build-plug-in](#).

MarkError (Fout)

Deze kleur wordt gebruikt door de build-plug-in om een regel aan te geven die een fout bij compilatie heeft veroorzaakt. Merk op dat de kleur een mindere dekking heeft ten opzichte van de achtergrond. Voor meer informatie, zie [de documentatie van de Build-plug-in](#).

Tekstjablonen & -fragmenten

TemplateBackground (Achtergrond)

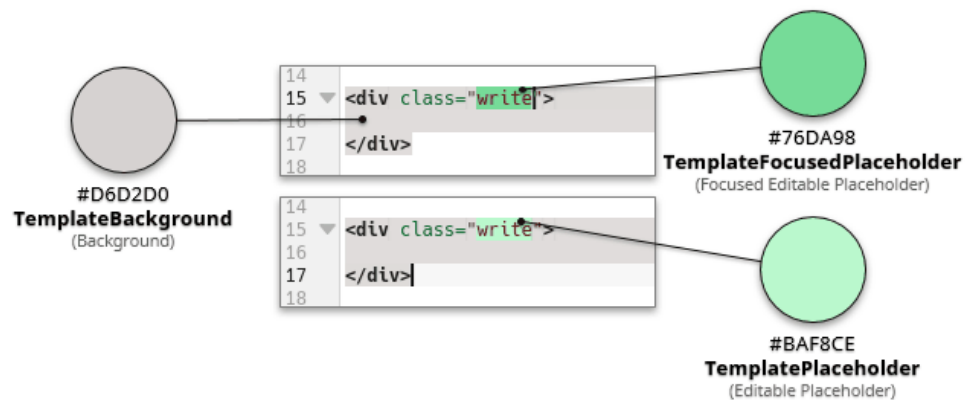
Deze kleur wordt door de fragmenten-plug-in van Kate gebruikt om de achtergrond van een fragment te markeren. Voor meer informatie, zie [de documentatie van fragmenten in Kate](#).

TemplatePlaceholder (Te bewerken plaatshouder)

Deze kleur wordt gebruikt door de Kate fragmenten-plug-in om een plaatshouder te markeren, waarop u kunt klikken om deze handmatig te bewerken. Voor meer informatie, zie [de documentatie van Kate-snippets](#).

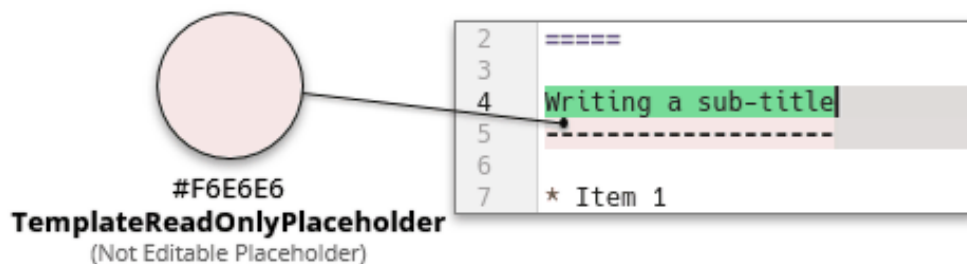
TemplateFocusedPlaceholder (Gefocuste bewerkbare plaatshouder)

Deze kleur wordt gebruikt door de Kate fragmenten-plug-in om een plaatshouder te markeren, die u op dit moment bewerkt. Voor meer informatie zie [de documentatie van Kate-snippets](#).



TemplateReadOnlyPlaceholder (Niet-bewerkbare plaatshouder)

Deze kleur wordt gebruikt door de Kate fragmenten-plug-in om een plaatshouder te markeren, die niet handmatig is te bewerken, zoals een die automatisch wordt gevuld. Voor meer informatie, zie [de documentatie van Kate-snippets](#).



6.3.4.2 Standaard tekststijlen

De standaard stijlen voor tekst zijn dezelfde als de stijlen voor geaccentueerde tekst, waardoor de editor tekst op een hele consistente manier kan presenteren. Zo wordt bijvoorbeeld voor commentaartekst dezelfde stijl gebruikt in bijna alle opmaakprofielen die kunnen worden geaccentueerd door KSyntaxHighlighting.

OPMERKING

Deze tekststijlen kunnen gerefereerd worden uit de **standaard stijlen** gebruikt in de XML definitiebestanden voor [syntaxisaccentuering](#), bijvoorbeeld, het attribuut 'Normaal' is gelijk aan 'dsNormal' in de XML bestanden en 'DataType' is gelijk aan 'dsDataType'. Zie Section 6.2.3.3 in de documentatie van syntaxisaccentuering.

TIP

Let op om leesbare kleuren met goed contrast te kiezen speciaal in combinatie met de **Kleuren van bewerker**. Zie Section 6.3.6.1.

In het [JSON-bestand](#) heeft de respectievelijke sleutel **text-styles** als waarde een *object* waar elke sleutel overeenkomt met de naam van een *standaard tekststijl*, die gelijk zijn aan die gebruikt in de definities voor syntaxisaccentuering. Hier zijn, **alle beschikbare tekststijlsleutels verplicht**, deze staan onderstaand in een lijst.

```
"text-styles": {
  "Normal" : {
    "text-color" : "#1f1c1b",
    "selected-text-color" : "#ffffff",
    "bold" : false,
    "italic" : false,
    "underline" : false,
    "strike-through" : false
  },
  "Keyword" : {
    "text-color" : "#1f1c1b",
    "selected-text-color" : "#ffffff",
    "bold" : true
  },
  "Function" : {
    "text-color" : "#644a9b",
    "selected-text-color" : "#452886"
  },
  De andere tekststijlsleutels...
}
```

Elke sleutel van de *standaard tekststijl* heeft een JSON-object als zijn waarde, waar waarden zoals *color*, *bold*, *italic*, etc. zijn gespecificeerd. Deze sleutels zijn als volgt:

text-color: het is een *tekenreeks* met de tekstkleur in hexadecimale kleurcodering. Deze sleutel/waarde is vereist.

selected-text-color: de tekstkleur wanneer het is geselecteerd is in het algemeen hetzelfde als 'text-color'. Wanneer de tekst is geselecteerd wordt de achtergrond gedefinieerd door de waarde van [TextSelection](#) in de [Kleuren van bewerker](#), u moet dus verzekeren dat de tekst goed contrast heeft en leesbaar is met deze achtergrond. De waarde is een *tekenreeks* met een hexadecimale kleurcode. Deze sleutel/waarde is vereist.

bold: het is een *boolean* die bepaalt of de tekst vet is. Deze sleutel is optioneel, de standaard waarde is **false**.

italic: het is een *boolean* die bepaalt of de tekst cursief is. Deze sleutel is optioneel, de standaard waarde is **false**.

underline: het is een *boolean* die bepaalt of de tekst onderstreept is. Deze sleutel is optioneel, de standaard waarde is **false**.

strike-through: het is een *boolean* die bepaalt of de tekst doorgestreept is. Deze sleutel is optioneel, de standaard waarde is **false**.

background-color: bepaalt de achtergrond van de tekst, bijvoorbeeld, gebruikt in opmerkelijk maken in commentaar. De waarde is een **tekenreeks** met een hexadecimale kleurcode. Deze key is optioneel, standaard is er geen achtergrond.

selected-background-color: bepaalt de achtergrond van de tekst wanneer deze is geselecteerd. De waarde is een **tekenreeks** met een hexadecimale kleurcode. Deze key is optioneel, standaard is er geen achtergrond.

In de [GUI om kleurenthema's van KTextEditor te beheren](#) kunnen deze attributen gewijzigd worden in het tabblad **Standaard tekststijlen**. De naam in de lijst met stijlen gebruikt de stijl geconfigureerd voor het item, waarmee u een onmiddellijk voorbeeld krijgt bij het configureren van een stijl. Elke stijl laat u gemeenschappelijke attributen selecteren evenals voor- en achtergrondkleuren. Om een achtergrondkleur niet in te stellen, klik rechts om het contextmenu te gebruiken.

De beschikbare tekststijlsleutels zijn als volgt; de sleutels gebruikt in het [JSON bestand](#) staan in de lijst *vet*, de namen gebruikt in de [GUI](#) worden getoond in ronde haakjes, als ze anders zijn.

Normale tekst & broncode

Normaal: standaard tekststijl voor normale tekst en broncode zonder speciale accentuering.

Sleutelwoord: tekststijl voor ingebouwde sleutelwoorden voor taal.

Functie: tekststijl voor functie-definities en functie-aanroepen.

Variable: tekststijl voor variabelen, indien van toepassing. Variabelen in bijvoorbeeld PHP/Perl starten typisch met een \$, dus alle identifiers volgens het patroon **\$foo** worden als variabele geaccentueerd.

ControlFlow (Loop van controle): Tekststijl voor loop van sleutelwoorden, zoals *if, then, else, return, switch, break, yield, continue*, etc.

Operator: tekststijl voor operators, zoals **+, -, *, / , %**, etc.

Ingebouwd (Ingebouwd): tekststijl voor ingebouwde klassen, functies en objecten van talen.

Extensie: tekststijl voor bekende extensies, zoals Qt™-klassen en functies/macro's in C++ en Python of boost.

Preprocessor: tekststijl voor preprocessor statements of macro-definities.

Attribuut: tekststijl voor annotaties of attributen van functies of objecten, bijv. **@override** in Java of **__declspec(...)** en **__attribute__((...))** in C++.

Getallen, typen & constanten

DataType (Type gegeven): tekststijl voor ingebouwde gegevenstypen zoals *int, char, float, void, u64*, etc.

DecVal (Decimaal/Waarde): tekststijl voor decimale waarden.

BaseN (Base-N geheel getal): tekststijl voor getallen met basis anders dan 10.

Float (Floating Point): tekststijl voor getallen met drijvende komma.

Constante: tekststijl voor taalconstanten en gebruikergedefinieerde constanten, bijv. *True, False, None* in Python of *nullptr* in C/C++; of wiskundige constanten zoals *PI*.

Tekenreeksen & tekens

Char (Teken): tekststijl voor losse tekens zoals **'x'**.

SpecialChar (Speciaal teken): tekststijl voor tekens met escape in tekenreeksen, bijv. **'hello\n'** en andere tekens met speciale betekenis in tekenreeksen, zoals substituties of regex-operators.

String: tekststijl voor tekenreeksen zoals **'hallo wereld'**.

VerbatimString (Letterlijke tekenreeks): tekststijl voor letterlijke of raw-tekenreeksen zoals `'raw \backslashbackslash'` in Perl, CoffeeScript en shells, evenals in `r'\raw'` in Python of zoals HIER-documenten.

SpecialString (Speciale tekenreeks): tekststijl voor speciale tekenreeksen, zoals reguliere expressies in ECMAScript, de L^AT_EX wiskundige modus, SQL, etc.

Import (Geïmporteerde dingen, modulen, invoegingen): tekststijl voor invoegingen, geïmporteerde dingen, modules of L^AT_EX pakketten.

Commentaar & documentatie

Comment: tekststijl voor normaal commentaar.

Documentation: tekststijl voor commentaar dat gaat over API-documentatie, zoals `/** doxygen comments */` of `"""docstrings"""`.

Annotation: tekststijl voor annotaties in commentaar of documentatiecommando's, zoals `@param` in Doxygen of JavaDoc.

CommentVar (Commentaarvariabele): tekststijl die verwijst naar variabelenamen gebruikt in bovenstaande commando's in een commentaar, zoals `foobar` in `'@param foobar'`, in Doxygen of JavaDoc.

RegionMarker (Gebiedsmarkering): tekststijl voor gebiedsmarkering, typisch gedefinieerd door `/ BEGIN` en `/ END` in commentaar.

Information: tekststijl voor informatie, notities en tips, zoals het sleutelwoord `@note` in doxygen.

Waarschuwing: tekststijl voor waarschuwingen, zoals het sleutelwoord `@warning` in doxygen.

Alert: tekststijl voor speciale woorden in commentaar, zoals `TODO`, `FIXME`, `XXXX` en `WAARSCHUWING`.

Diversen

Fout: tekststijl die een fout en verkeerde syntaxis aangeeft met accentuering.

Overigen: tekststijl voor attributen die niet overeenkomen met een van de andere standaard stijlen.

6.3.4.3 Aangepaste tekststijlen voor accentuering

Hier kunt u tekststijlen instellen voor een specifieke definitie voor syntaxisaccentuering, overschrijvend de **standaard tekststijl** beschreven in [de vorige sectie](#).

In het [JSON themabestand](#) correspondeert dit met de sleutel **custom-styles**, waarvan de waarde een *object* is waar elke subschemasleutel overeenkomt met de **naam van een definitie voor syntaxisaccentuering**. Zijn waarde is een *object* waar elke sleutel refereert naar de **stijlattribuutnaam** gedefinieerd in [de elementen itemData](#) van het XML-bestand met syntaxisaccentuering en de respectievelijke waarde een sub-object is met de sleutels *text-color*, *selected-text-color*, *bold*, *italic*, *underline*, *strike-through*, *background-color* en *selected-background-color*, gedefinieerd in [de vorige sectie](#). Elk van deze waarden zijn optioneel, omdat als ze niet aanwezig zijn, de stijl ingesteld in **text-styles** wordt beschouwd.

In dit stukje code, bijvoorbeeld, heeft de definitie voor 'ISO C++' syntaxisaccentuering een speciale tekststijl voor de attributen 'Type Modifiers' en 'Standard Classes'. In het overeenkomstige XML-bestand 'isocpp.xml' gebruikt het gedefinieerde attribuut 'Standard Classes' de standaard stijl **BuiltIn** (of **dsBuiltIn**). In dit attribuut wordt alleen de waarde van **text-color** overschreven door de nieuwe kleur '#6431b3'.

```
"custom-styles": {
  "ISO C++": {
    "Standard Classes": {
      "text-color": "#6431b3"
    },
  },
}
```

```
"Type Modifiers": {
  "bold": true,
  "selected-text-color": "#009183",
  "text-color": "#00b5cf"
}
```

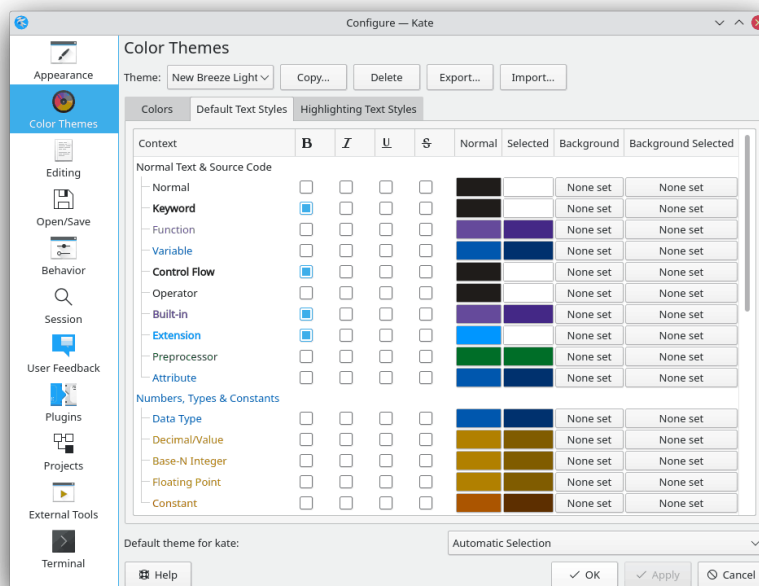
OPMERKING

- U zou deze tekststijlen moeten beschouwen als geassocieerd met de attribuutnamen gedefinieerd in de XML-bestanden voor syntaxisaccentuering. Als een XML bestand wordt bijgewerkt en sommige attributen worden hernoemd of verwijderd, zal de aangepaste stijl gedefinieerd in het thema niet langer van toepassing zijn.
- Definities voor syntaxisaccentuering bevatten vaak andere definities. Bijvoorbeeld, de module voor accentueren van 'QML' bevat deze voor 'JavaScript', omdat ze functionaliteit in accentueren delen.

In de GUI om thema's van *KTextEditor* te beheren kunnen deze attributen gewijzigd worden in het tabblad **Tekststijlen voor accentueren**. Standaard selecteert het module voor accentueren van het huidige document voor. U zult opmerken dat veel accentuering andere accentuering representeert door groepen in de lijst met stijlen. De meeste accentueringen importeren de accentuering 'Alert' en veel broncodeformaten importeren de accentuering 'Doxygen'.

6.3.5 De GUI voor kleurenthema's

De eenvoudigste manier om kleurenthema's aan te maken en te bewerken is via de GUI in de dialoog 'Configureren' geleverd door *KTextEditor*. Om er toegang toe te krijgen, selecteer **Instellingen** → **Toepassing configureren...** uit de menubalk in uw tekstbewerker. Dit toont het dialoogvenster **Configureren**, selecteer daar **Kleurenthema's** in het zijpaneel.



Het instellingendialoogvenster van Kate met het beheer van kleurenthema's.

In deze [dialoog](#) kunt u alle kleuren in elk thema dat u hebt configureren, evenals nieuwe thema's aanmaken/kopiëren, ze verwijderen, ze exporteren naar een bestand `.theme` in [JSON-formaat](#) of ze importeren uit externe bestanden `.theme`. Elk thema heeft instellingen voor tekstkleuren en stijlen.

De ingebouwde thema's kunnen niet standaard gewijzigd worden. Om dat te doen moet u ze kopiëren en een nieuwe naam geven.

Om een thema permanent in uw tekstbewerker te gebruiken moet u het in de keuzelijst met label **Standaard thema voor Toepassing** onderaan de dialoog selecteren en op **Toepassen** of **OK** drukken. Standaard is de optie **Automatische selectie** actief, die een meer toepasselijk kleurenthema volgens het *KDE Plasma kleurenschema* gebruikt in de tekstbewerkertoepassing; er wordt gewoonlijk gekozen tussen 'Breeze Light' en 'Breeze Dark' als het schema, respectievelijk licht of donker is.

TIP

U kunt het KDE globale kleurenschema in de module [Kleuren in Systeeminstellingen](#) aanpassen. U kunt het ook individueel wijzigen in sommige toepassingen zoals Kate of KDevelop, uit het menu **Instellingen** → **Kleurenschema**.

6.3.5.1 Een nieuw thema aanmaken

Om een nieuw thema aan te maken is het eerst noodzakelijk om een bestaande te kopiëren. Selecteer een bestaand thema die u als basis wilt gebruiken, zoals 'Breeze Light' of 'Breeze Dark' en klik op **Kopiëren**. Schrijf daarna een naam voor het nieuwe thema.

Als u een ingebouwde of alleen-lezen thema wilt wijzigen, moet u het eerst kopiëren onder een andere naam.

6.3.5.2 JSON themabestanden im- of exporteren

U kunt een geselecteerd thema (inclusief ingebouwden) exporteren naar een [JSON-bestand](#) met extensie `.theme`, met de knop **Exporteren**. Dit zal een dialoog openen om het bestand op te slaan. Om een kleurenthema uit een extern [JSON-bestand](#) toe te voegen, druk gewoon op de knop **Importeren** en selecteer het bestand `.theme` uit de dialoog.

TIP

- Zoals [boven genoemd](#) worden door gebruikers aangepaste themabestanden opgeslagen in de map `org.kde.syntax-highlighting/themes/`. Wanneer u hier een thema kopieert of maakt zal het automatisch daar verschijnen. Importeren of toevoegen van een thema is equivalent met kopiëren van een extern `.theme`-bestand in deze map. KSyntaxHighlighting pikt automatisch kleurenthemabestanden op uit deze map.
- Als u een door u gemaakt thema wilt publiceren is het essentieel om het [metagegevens](#)-object van het [JSON bestand](#) te controleren, met toevoeging van de respectievelijke licentie en controle op het revisienummer.

6.3.5.3 Kleurenthema's bewerken

6.3.5.3.1 Kleuren

Hier zijn de kleuren van het tekstbewerkingsgebied aangepast. Deze zijn gedetailleerd in [Section 6.3.4.1](#).

6.3.5.3.2 Standaard tekststijlen

De standaard stijlen voor tekst zijn dezelfde als de stijlen voor geaccentueerde tekst, waardoor de editor tekst op een hele consistente manier kan presenteren. Zo wordt bijvoorbeeld voor commentaartekst dezelfde stijl gebruikt in bijna alle opmaakprofielen die kunnen worden geaccentueerd door KSyntaxHighlighting.

De naam in de lijst met stijlen gebruikt de stijl die voor dat item is ingesteld, en u krijgt onmiddellijk een voorbeeld te zien wanneer u een stijl aanpast.

Bij elke stijl kunt u algemene kenmerken kiezen en voorgrond- en achtergrondkleuren. Om een achtergrondkleur uit te zetten klikt u met rechts zodat het contextmenu verschijnt.

De attributen in dit gebied zijn gedetailleerd in Section [6.3.4.2](#).

6.3.5.3.3 Stijlen voor accentuering van tekst

Hier kunt u de tekststijlen die gebruikt worden bij een specifieke accentueringsdefinitie bewerken. De editor zal de accentuering die gebruikt wordt door uw huidige document voorselecteren. Om aan een andere accentuering te werken kiest u er een in het combinatieveld **Accentueren** boven de lijst met stijlen.

De naam in de lijst met stijlen gebruikt de stijl die voor dat item is ingesteld, en u krijgt onmiddellijk een voorbeeld te zien wanneer u een stijl aanpast.

Bij elke stijl kunt u algemene kenmerken kiezen en voorgrond- en achtergrondkleuren. Om een achtergrondkleur uit te zetten klikt u met rechts zodat het contextmenu verschijnt. Bovendien kunt u zien of een stijl gelijk is aan de standaardstijl die voor het item gebruikt wordt en die al dan niet gebruiken.

U ziet dat vele accentueringen andere accentueringen bevatten die als groepen vertegenwoordigd zijn in de lijst met stijlen. Zo importeren bijvoorbeeld de meeste accentueringen de Alert-accentuering, en vele opmaken voor broncode importeren de Doxygen-accentuering.

6.3.6 Tips & trucs

6.3.6.1 Contrast van tekstkleuren

Een belangrijk aspect bij werken met kleurenthema's is een tekstcontrast te kiezen dat het gemakkelijk maakt om te lezen, speciaal in combinatie met de achtergrond.

De toepassing **Kontrast** is een controleur op kleurcontrast. Het vertelt dat de combinaties van tekstkleur en achtergrondkleur leesbaar en toegankelijk zijn, dit is dus een uitstekend hulpmiddel om u te helpen bij het maken van kleurenthema's.

U kunt **Kontrast** downloaden uit [de website voor KDE toepassingen](#) of uit [het Flatpak pakket op Flathub](#) (alleen in GNU/Linux).

De **Contrast** toepassing van GNOME is vergelijkbaar. U kunt [het Flatpak pakket op Flathub](#) downloaden (alleen in GNU/Linux).

Een python script in [Opslagruimte voor syntaxisaccentuering](#) laat u alle kleuren in een theme zien, evenals het contrast met de verschillende geconfigureerde achtergronden.

6.3.6.2 Suggesties over consistentie met syntaxisaccentuering

KSyntaxHighlighting omvat [meer dan 300 definities voor syntaxisaccentuering](#), het is daarom ideaal dat u nagaat dat uw nieuwe thema er goed uitziet in alle definities voor syntaxisaccentuering. De ingebouwde kleurenthema's hebben de volgende te volgen gelijkenissen die aanbevolen zijn (maar niet verplicht) om een juiste weergave van alle definities voor syntaxisaccentuering te verkrijgen:

- Vet gebruiken voor het 'Sleutelwoord' en 'loop van het programma' **tekststijlen**.
- Geen achtergrondkleur in elk **tekststijlen**, behalve 'Alert' en 'Gebiedsmarkeringen'.

De meeste syntaxisaccentueringen zijn bedoeld om er goed uit te zien met de standaard thema's 'Breeze Light' en 'Breeze Dark', een andere manier om consistentie te behouden is om soortgelijke kleuren in de **tekststijlen** te gebruiken, zoals *groen* voor 'Preprocessor' en 'Overigen', *blauw* voor 'Gegevenstype' en 'Attribuut' of *paars* voor 'Functie'.

Merk op dat deze aanbevelingen niet verplicht zijn bij aanmaken en publiceren van een thema.

6.4 Scripting met JavaScript

De editor-component van KatePart is eenvoudig uit te breiden met scripts. De scripttaal is ECMAScript (breed bekend als JavaScript). KatePart ondersteunt twee soorten scripts: inspringen en commandoregelscripts. Een functionaliteit gelijk aan opdrachtregelscripts wordt ook geleverd in de **snippets plug-in**.

6.4.1 Inspringscripts

Inspringscripts - ook genoemd indenteerders - laten de broncode inspringen bij het typen van tekst. Na het indrukken van de toets Enter zal het inspringniveau vaak toenemen.

De volgende secties beschrijven stap voor stap hoe het skelet voor een eenvoudige indenteerder te maken. Als eerste stap, maak een nieuw bestand *.js aan, bijv. met de naam javascript.js in de lokale thuismap \$XDG_DATA_HOME /katepart5/script/indentation. Waarin de omgevingsvariabele XDG_DATA_HOME typisch expandeert tot ofwel ~/.local of ~/.local/share.

Op Windows® zijn deze bestanden gelokaliseerd in %USERPROFILE%\AppData\Local\katepart5\script\indentation. %USERPROFILE% wordt gewoonlijk C:\\Users\\gebruiker.

6.4.1.1 De kop van het inspringscript

De kop van het bestand javascript.js is ingebed als JSON aan het begin van het document als volgt:

```
var kate-script = {
  * "name": JavaScript
  * "author": Voorbeeldnaam <voorbeeld.naam@een.adres.org>,
  * "license": "BSD License"
  * "revision": 1,
  * "kate-version": "5.1",
  * "required-syntax-style": javascript,
  * "indent-languages": ["javascript"],
  * "priority": 0,
  *}; // kate-script-header, moet aan het begin van het bestand staan zonder ←
      commentaar
```

Elk item wordt nu in detail uitgelegd:

- **name** [vereist]: Dit is de naam van de indenteerder die in het menu **Hulpmiddelen** → **Inspringen** verschijnt en in de configuratiedialoog.
- **author** [optioneel]: De naam van de auteur en contactinformatie.
- **license** [optioneel]: Korte vorm van de licentie, zoals BSD License of LGPLv3.

- `revision` [vereist]: De revisie van het script. Dit getal moet verhoogd worden wanneer het script wordt gewijzigd.
- `kate-version` [required]: Minimaal vereiste versie van KatePart.
- `required-syntax-style` [optioneel]: De vereiste stijl voor de syntaxis, die overeenkomt met de gespecificeerde `style` in bestanden voor accentuering van syntaxis. Dit is belangrijk voor indenteerders die vertrouwen op specifieke informatie voor accentuering in het document. Als een vereiste syntaxisstijl is gespecificeerd, zal de indenteerder alleen beschikbaar zijn wanneer het juiste programma voor accentuering actief is. Dit voorkomt 'ongedefinieerd gedrag' veroorzaakt door de indenteerder te gebruiken zonder het verwachte schema voor accentuering. De Ruby indenteerder, bijvoorbeeld, maakt hier gebruik van in de bestanden `ruby.js` en `ruby.xml`.
- `indent-languages` [optioneel]: JSON array van syntaxis stijlen die de indenteerder correct kan laten inspringen, bijv.: `[~c++, ~java]`.
- `priority` [optioneel]: Als verschillende indenteerders geschikt zijn voor een bepaald bestand om te accentueren, dan bepaalt de prioriteit welke indenteerder als standaard wordt gekozen.

6.4.1.2 De broncode van de indenteerder

Nadat de header is gespecificeerd, legt deze sectie uit hoe het script voor inspringen zelf werkt. Het basis sjabloon van de inhoud ziet er uit als volgt:

```
// vereiste js-bibliotheken van katepart, bijv. range.js als u Range ←
    gebruikt
require ("range.js");
triggerCharacters = "{}/:;";
function indent(line, indentWidth, ch)
{
    // aangeroepen voor elk teken nieuwe-regel (ch == '\n') en alle tekens ←
    gespecificeerd in
    // de globale variabele triggerCharacters. Bij het aanroepen van ←
    HulpmiddelenInspringen formatteren
    // is de variabele ch leeg, i.e. ch == ''.
    //
    // zie ook: Scripting API
    return -2;
}
```

De functie `indent()` heeft drie parameters:

- `line`: de regel die moet inspringen
- `indentWidth`: de inspringbreedte in aantal spaties
- `ch`: of een teken nieuwe-regel (`ch == '\n'`), het activeringsteken gespecificeerd in `triggerCharacters` of leeg als de gebruiker de actie **Hulpmiddelen** → **Uitlijnen** heeft gestart.

De terugkeerwaarde van de functie `indent()` specificeert hoe de regel zal worden ingesprongen. Als de terugkeerwaarde een eenvoudig geheel getal is, wordt deze geïnterpreteerd als volgt:

- terugkeerwaarde `-2`: niets doen
- terugkeerwaarde `-1`: behoudt het inspringen (zoekt naar voorgaande niet-blanko regel)
- terugkeerwaarde `0`: getallen `>= 0` specificeer de inspringdiepte in spaties

Alternatief, een reeks van twee elementen kan worden teruggegeven:

- geef [indent, align] terug;

In dit geval is het eerste element de inspringdiepte zoals hierboven met dezelfde betekenis van de speciale waarden. Het tweede element is een absolute waarde die een kolom aangeeft voor 'uitlijning'. Als deze waarde groter is dan de inspringwaarde, dan is het verschil een aantal spaties dat na het inspringen van de eerste parameter moet worden toegevoegd. Anders wordt het tweede getal genegeerd. Tabs en spaties voor inspringen gebruiken wordt vaak 'gemengde modus' genoemd.

beschouw het volgende voorbeeld: Neem aan dat tabs worden gebruikt voor inspringen en tabbreedte is ingesteld op 4. <tab> vertegenwoordigt hier een tab-teken en '.' een spatie:

```
1: <tab><tab>foobar("hello",
2: <tab><tab>....."world");
```

Bij inspringen van regel 2 zal de functie `indent()` [8, 15] als terugkeerwaarde geven. Als resultaat worden hier twee tabs ingevoegd om in te springen tot kolom 8 en worden 7 spaties toegevoegd om de tweede parameter uit te lijnen onder de eerste, zodat het uitgelijnd blijft als het bestand wordt bekeken met een andere tabbreedte.

Een standaard KDE installatie levert bij KatePart verschillende indenteerders. De overeenkomstige JavaScript broncode kan worden gevonden in `$XDG_DATA_DIRS /katepart5/script/indentation`.

Op Windows® zijn deze bestanden gelokaliseerd in `%USERPROFILE%\AppData\Local\katepart5\script\indentation`. `%USERPROFILE%` wordt gewoonlijk `C:\\Users\\gebruiker`.

Ontwikkeling van een indenteerder vereist herladen van de scripts om te zien of de wijzigingen zich netjes gedragen. In plaats van het opnieuw opstarten van de toepassing, schakel eenvoudig om naar de commandoregel en start het commando **reload-scripts**.

Als u bruikbare scripts ontwikkeld overweeg dan deze aan het KatePart-project bij te dragen door een mengverzoek naar [our GitLab project](#).

6.4.2 Commandoregelscripts

OPMERKING

Een functionaliteit gelijk aan opdrachtregelscripts wordt ook geleverd in de [snippets plug-in](#). Deze plug-in kan een gemakkelijker startpunt leveren, speciaal voor kleine eigen scripts.

Omdat het moeilijk is om aan de noden van iedereen te voldoen, ondersteunt KatePart kleine hulpmiddelen voor snelle tekstmanipulatie via de [ingebouwde commandoregel](#). Het commando **sort** is bijvoorbeeld geïmplementeerd als een script. Deze sectie legt uit hoe *.js bestanden aan te maken om KatePart uit te breiden met willekeurige hulpscripts.

Commandoregelscripts staan in dezelfde map als inspringscripts. Dus als eerste stap maakt u een nieuw *.js bestand aan genaamd `myutils.js` in de lokale map `$XDG_DATA_HOME /katepart5/script/commands`. De omgevingsvariabele `XDG_DATA_HOME` daarin expandeert typisch tot ofwel `~/.local` of `~/.local/share`.

Op Windows® zijn deze bestanden gelokaliseerd in `%USERPROFILE%\AppData\Local\katepart5\script\commands`. `%USERPROFILE%` wordt gewoonlijk `C:\\Users\\user`.

6.4.2.1 De kop van het commandoregelscript

De header van elke commandoregelscript is ingebed in JSON aan het begin van het script als volgt:

```
var katescript = {
  "author": "Voorbeeldnaam <voorbeeld.naam@een.adres.org>",
  "license": "LGPLv2+",
  "revision": 1,
  "kate-version": "5.1",
  "functions": ["sort", "moveLinesDown"],
  "actions": [
    {
      "function": "sort",
      "name": "Sort Selected Text",
      "category": "Editing",
      "interactive": "false"
    },
    {
      "function": "moveLinesDown",
      "name": "Move Lines Down",
      "category": "Editing",
      "shortcut": "Ctrl+Shift+Down",
      "interactive": "false"
    }
  ]
}; // kate-script-header, moet zich bevinden aan het begin van het bestand ↵
    zonder commentaar
```

Elk item wordt nu in detail uitgelegd:

- `author` [optioneel]: De naam van de auteur en contactinformatie.
- `license` [optioneel]: Korte vorm van de licentie, zoals BSD licentie of LGPLv2.
- `revision` [vereist]: De revisie van het script. Dit getal moet verhoogd worden wanneer het script wordt gewijzigd.
- `kate-version` [required]: Minimaal vereiste versie van KatePart.
- `functions` [vereist]: JSON-array met commando's in het script.
- `actions` [optioneel]: JSON Array van JSON objecten die de acties definiëren die verschijnen in het menu van de toepassing. Gedetailleerde informatie wordt in de sectie [Koppelen van sneltoetsen](#).

Omdat de waarde van `functions` een JSON array is, is een enkel script in staat een willekeurig aantal commandoregels te bevatten. Elke functie is beschikbaar via [ingebouwde commandoregel](#) van KatePart.

6.4.2.2 De broncode van het script

Alle in de kop gespecificeerde functies moeten worden geïmplementeerd in het script. Het script-bestand uit het bovenstaande voorbeeld moet de twee functies **sort** en **moveLinesDown** implementeren. Alle functies hebben de volgende syntaxis:

```
// vereiste js-bibliotheken van katepart, bijv. range.js als u Range ↵
    gebruikt
require ("range.js");

function <naam>(arg1, arg2, ...)
{
    // ... implementatie, zie ook: Scripting API
}
```

De argumenten in de commandoregel worden doorgegeven naar de functie als *arg1*, *arg2*, etc. Om documentatie voor elk commando te leveren, implementeert u eenvoudig de functie 'help' als volgt:

```
function help(cmd)
{
    if (cmd == "sort") {
        return i18n("Sort the selected text.");
    } else if (cmd == "...") {
        // ...
    }
}
```

uitvoeren van **help sort** in de commandoregel roept deze help-functie op met het argument *cmd* ingesteld op het gegeven commando, bijv. *cmd == "sort"*. KatePart laat dan de teruggeleverde tekst zien als documentatie voor de gebruiker. Zorg er voor dat [de tekenreeksen vertaald zijn](#).

Ontwikkeling van een commandoregelscript vereist herladen van de scripts om te zien of de wijzigingen zich netjes gedragen. In plaats van het opnieuw opstarten van de toepassing, schakel eenvoudig om naar de commandoregel en start het commando **reload-scripts**.

6.4.2.2.1 Koppelen van sneltoetsen

Om de scripts toegankelijk te maken in het menu van de toepassing en sneltoetsen toe te kunnen, moet het script een juiste script-header leveren. In het bovenstaande voorbeeld, verschijnen beide functies `sort` en `moveLinesDown` in het menu vanwege het volgende deel in de script-header:

```
var katescript = {
    ...
    "actions": [
        {
            "function": "sort",
            "name": "Sort Selected Text",
            "icon": "",
            "category": "Editing",
            "interactive": "false"
        },
        {
            "function": "moveLinesDown",
            "name": "Move Lines Down",
            "icon": "",
            "category": "Editing",
            "shortcut": "Ctrl+Shift+Down",
            "interactive": "false"
        }
    ]
};
```

De velden voor een actie zijn als volgt:

- **function** [vereist]: De functie die zou moeten verschijnen in het menu **Hulpmiddelen** → **Scripts**.
- **name** [vereist]: De tekst verschijnt het script-menu.
- **icon** [optioneel]: Het pictogram verschijnt naast de tekst in het menu. Alle pictogramnamen van KDE kunnen hier worden gebruikt.
- **category** [optioneel]: Als een categorie wordt gespecificeerd, verschijnt het script in een sub-menu.

- `shortcut` [optioneel]: De hier gegeven sneltoets is de standaard sneltoets. Voorbeeld: `Ctrl+Alt+t`. Zie de [Qt-documentatie](#) voor verdere details.
- `interactive` [optioneel]: Als het script invoer van de gebruiker in de commandoregel nodig heeft, zet dit dan op `true`.

Als u bruikbare scripts ontwikkeld overweeg dan deze aan het KatePart-project bij te dragen door een mengverzoek naar [our GitLab project](#).

6.4.3 API voor scripts maken

De API voor scripts hier aanwezig is beschikbaar voor alle scripts, bijv. scripts voor inspringen en opdrachtregels. De klassen `Cursor` en `Range` worden geleverd door bibliotheekbestanden in `$XDG_DATA_DIRS /katepart5/libraries`. Als u ze in uw script wilt gebruiken, wat nodig is om bepaalde Document of View-functies te gebruiken, voeg dan de benodigde bibliotheek in met:

```
// vereiste js-bibliotheken van katepart, bijv. range.js als u Range ←  
gebruikt  
require ("range.js");
```

Om de standaard API voor scripting uit te breiden met uw eigen functies en prototypen, maak gewoon een nieuw bestand aan in de lokale configuratiemap van KDE `$XDG_DATA_HOME /apps/katepart5/libraries` en voeg deze in in uw script met:

```
require ("mijnscripnaamhier.js");
```

Op Windows[®] zijn deze bestanden gelokaliseerd in `%USERPROFILE%\AppData\Local\katepart5\libraries`. `%USERPROFILE%` wordt gewoonlijk `C:\\Users\\user`.

Om bestaande prototypes zoals `Cursor` of `Reeks` uit te breiden, os de aanbevolen manier om *niet* de globale `*.js` bestanden te wijzigen. In plaats daarvan wijzigt u het prototype `Cursor` in JavaScript nadat `cursor.js` is ingevoegd in uw script via `require`.

6.4.3.1 Cursors en bereiken

Omdat KatePart is een tekstbewerker is, is alles voor de API voor scripting, waar mogelijk, gebaseerd op cursors en reeksen. Een cursor is een eenvoudig een paar (`regel`, `kolom`) die een tekstpositie in het document weergeeft. Een reeks omspant tekst van een begincursorpositie tot een eincursorpositie. De API wordt in de volgende secties in detail uitgelegd.

6.4.3.1.1 Het prototype van cursor

`Cursor()`;

Constructor. Levert een cursor terug op positie (0, 0).

Voorbeeld: `var cursor = new Cursor();`

`Cursor(int regel, int kolom);`

Constructor. Levert een cursor terug op positie (`regel`, `kolom`).

Voorbeeld: `var cursor = new Cursor(3, 42);`

`Cursor(Cursor andere);`

Constructor kopiëren. Geeft een kopie van de cursor *andere*.

Voorbeeld: `var copy = nieuwe cursor(andere);`

Cursor Cursor.clone();

Geeft een kloon van de cursor terug.

Voorbeeld: `var clone = cursor.clone();`

Cursor.setPosition(int regel, int kolom);

Stel de cursorpositie in op *regel* en *kolom*.

Sinds: KDE 4.11

bool Cursor.isValid();

Controleert of de cursor geldig is. De cursor is ongeldig indien lijn en/of kolom ingesteld zijn op -1.

Voorbeeld: `var valid = cursor.isValid();`

Cursor Cursor.invalid();

Geeft een nieuwe ongeldige cursor op locatie (-1, -1).

Voorbeeld: `var invalidCursor = cursor.invalid();`

int Cursor.compareTo(Cursor andere);

Vergelijkt deze cursor met de cursor *andere*. Geeft terug

- -1, indien deze cursor zich bevindt voor de cursor *andere*,
- 0, indien beide cursors gelijk zijn
- +1, als deze cursor gelokaliseerd is na de cursor *andere*.

bool Cursor.equals(Cursor andere);

Geeft `true` terug, als deze cursor en de cursor *andere* gelijk zijn, anders `false`.

String Cursor.toString();

Geeft de cursor terug als een tekenreeks met de vorm 'Cursor(*regel*, *kolom*)'.

6.4.3.1.2 Het prototype Range

Range();

Constructor. Aanroepen van `new Range()` geeft een bereik op (0, 0) - (0, 0) terug.

Range(Cursor begin, Cursor eind);

Constructor. Aanroepen van `new Range(begin, eind)` geeft de Reeks (*begin*, *eind*) terug.

Range(int beginRegel, int beginKolom, int eindRegel, int eindKolom);

Constructor. Aanroepen van `new Range(beginRegel, beginKolom, eindeRegel, eindKolom)` geeft de Reeks van (*beginRegel*, *beginKolom*) tot (*eindeRegel*, *eindKolom*).

Range(Range andere);

Constructor kopiëren. Geeft een kopie terug van Range *andere*.

Reeks Range.clone();

Geeft een kloon van de reeks terug.

Voorbeeld: `var clone = range.clone();`

bool Range.isEmpty();

Geeft `true` terug, als zowel begin als einde cursor gelijk zijn.

Voorbeeld: `var empty = range.isEmpty();`

Sinds: KDE 4.11

bool Range.isValid();

Geeft `true` terug, als zowel begin als einde cursor geldig zijn, anders `false`.

Voorbeeld: `var valid = range.isValid();`

Range Range.invalid();

Geeft de Reeks van `(-1, -1)` tot `(-1, -1)` terug.

bool Range.contains(Cursor cursor);

Geeft `true` terug, als de reeks de cursorpositie bevat, anders `false`.

bool Range.contains(Range andere);

Geeft `true` terug, als deze reeks de Reeks *andere* bevat, anders `false`.

bool Range.containsColumn(int kolom);

Geeft `true` terug, als *kolom* zich in het half open interval `[start.column, end.column)` bevindt, anders `false`.

bool Range.containsLine(int regel);

Geeft `true` terug, als *regel* zich in het half open interval `[start.line, end.line)` bevindt, anders `false`.

bool Range.overlaps(Range overig);

Geeft `true` terug, als deze reeks en de reeks *overig* een gebied delen, anders `false`.

bool Range.overlapsLine(int regel);

Geeft `true` terug, als *regel* zich in het interval `[start.line, end.line)` bevindt, anders `false`.

bool Range.overlapsColumn(int kolom);

Geeft `true` terug, als *kolom* zich in het interval `[start.column, end.column)` bevindt, anders `false`.

bool Range.onSingleLine();

Geeft `true` terug, als zowel begin als einde van de reeks op dezelfde regel, bijv. als `Range.start.line == Range.end.line`.

Sinds: KDE 4.9

bool Range.equals(Range overig);

Geeft `true` terug, als deze reeks de Reeks *andere* gelijk zijn, anders `false`.

String Range.toString();

Geeft de reeks terug als een tekenreeks van de vorm `'Range(Cursor(line, column), Cursor(line, column))'`.

6.4.3.2 Globale functies

Deze sectie geeft een lijst met alle globale functies.

6.4.3.2.1 Lezen & invoegen van bestanden

String read(String bestand);

Zal het gegeven *bestand* zoeken relatief ten opzichte van de map `katepart5/script/files` en de inhoud teruggeven als een tekenreeks.

void require(String bestand);

Zal het gegeven *bestand* zoeken relatief ten opzichte van de map `katepart5/script/libraries` en het evalueren. `require` is intern beschermd tegen meervoudig invoegen van hetzelfde *bestand*.

Sinds: KDE 4.10

6.4.3.2.2 Debugging

void debug(*String* tekst);

Stuurt *tekst* naar `stdout` in de console na het starten van de toepassing.

6.4.3.2.3 Vertaling

Om volledige lokalisatie te ondersteunen zijn er verschillende functies om tekenreeksen in scripts te vertalen, namelijk `i18n`, `i18nc`, `i18np` en `i18ncp`. Deze functies gedragen zich op precies dezelfde manier zoals in [Vertalingsfuncties van KDE](#).

De vertalingfuncties vertalen de omwikkelde tekenreeksen via het vertalingssysteem van KDE naar de in de toepassing gebruikte taal. Tekenreeksen in scripts die worden ontwikkeld in de officiële broncode van KatePart worden automatisch geëxtraheerd en vertaalbaar gemaakt. Met andere woorden, als ontwikkelaar van KatePart hoeft u zich niet bezig te houden met extraheren en vertalen. Opgemerkt moet echter worden dat de vertaling alleen werkt binnen de infrastructuur van `$kde`. Dat wil zeggen dat nieuwe tekenreeksen in scripts van derden ontwikkeld buiten KDE niet worden vertaald. Neem dus in overweging om uw scripts aan Kate bij te dragen zodat vertaling mogelijk wordt.

void i18n(*String* tekst, *arg1*, ...);

Vertaalt *text* in de taal die wordt gebruikt door de toepassing. De argumenten *arg1*, ..., zijn optioneel en worden gebruikt om plaatshouders `%1`, `%2`, etc. te vervangen.

void i18nc(*String* context, *String* tekst, *arg1*, ...);

Vertaalt *text* in de taal die wordt gebruikt door de toepassing. Bovendien is de tekenreeks *context* zichtbaar voor de vertalers, zodat zij een betere vertaling kunnen produceren. De argumenten *arg1*, ..., zijn optioneel en worden gebruikt om plaatshouders `%1`, `%2`, etc. te vervangen.

void i18np(*String* enkelvoud, *String* meervoud, *int* nummer, *arg1*, ...);

Vertaalt ofwel *enkelvoud* of *meervoud* in de taal die wordt gebruikt door de toepassing, afhankelijk van het gegeven *nummer*. De argumenten *arg1*, ..., zijn optioneel en worden gebruikt om plaatshouders `%1`, `%2`, etc. te vervangen.

void i18ncp(*String* context, *String* enkelvoud, *String* meervoud, *int* nummer, *arg1*, ...);

Vertaalt *enkelvoud* of *meervoud* in de taal die wordt gebruikt door de toepassing, afhankelijk van het gegeven *nummer*. Bovendien is de tekenreeks *context* zichtbaar voor de vertalers, zodat zij een betere vertaling kunnen produceren. De argumenten *arg1*, ..., zijn optioneel en worden gebruikt om plaatshouders `%1`, `%2`, etc. te vervangen.

6.4.3.3 De API voor weergave

Wanneer een script wordt uitgevoerd, dan is er een globale variabele 'view' die de huidige actieve editor-view representeert. Het volgende is een lijst met alle beschikbare weergavefuncties.

void view.copy()

Kopieer de selectie als er een is, anders de huidige regel als de optie [] **Kopieer/knip de huidige regel als er geen selectie is** is gezet.

Sinds: KDE Frameworks 5.79

void view.cut()

Knip de selectie als er een is, anders de huidige regel als de optie [] **Kopieer/knip de huidige regel als er geen selectie is** is gezet.

Sinds: KDE Frameworks 5.79

void view.paste()

De inhoud van het klembord plakken.

Sinds: KDE Frameworks 5.79

Cursor view.cursorPosition()

Geeft de huidige cursorpositie in het beeld terug.

**void view.setCursorPosition(int line, int column); void
view.setCursorPosition(Cursor cursor);**

Stelt de huidige cursorpositie in op ofwel (regel, kolom) of op de gegeven cursor.

Cursor view.virtualCursorPosition();

Geeft de virtuele cursorpositie terug waarin elke tab telt voor de overeenkomstige hoeveelheid spaties afhankelijk van de huidige tabbreedte.

**void view.setVirtualCursorPosition(int line, int column); void
view.setVirtualCursorPosition(Cursor cursor);**

Stelt de huidige virtuele cursorpositie in op (regel, kolom) of op de gegeven cursor.

String view.selectedText();

Geeft de geselecteerde tekst terug. Als er geen tekst is geselecteerd dan is de tekenreeks leeg.

bool view.hasSelection();

Geeft `true` terug, als het beeld geselecteerde tekst heeft, anders `false`.

Range view.selection();

Geeft de geselecteerde tekstreks terug. Als er geen tekst is geselecteerd dan is de tekenreeks ongeldig.

void view.setSelection(Range reeks);

Stelt de geselecteerde tekst in op de gegeven reeks.

void view.removeSelectedText();

Verwijder de geselecteerde tekst. Als het beeld geen geselecteerde tekst bevat, dan doet dit niets.

void view.selectAll();

Selecteert de gehele tekst in het document.

void view.clearSelection();

Wist de tekstselectie zonder de tekst te verwijderen.

void view.setBlockSelection(bool on);

Modus blokselectie aan- of uitzetten.

bool view.blockSelection();

Geeft `true` terug, als modus blokselectie aan is, anders `false`.

void view.align(Range reeks);

Regels opnieuw laten inspringen binnen *reeks* volgens de huidige instellingen voor inspringen.

```
void view.alignOn(Range reeks, String patroon = "");
```

Lijnt regels uit in de *reeks* op de kolom gegeven door de reguliere expressie *patroon*. Met een leeg *patroon* zal er standaard uitgelijnd worden op het eerste niet-blanke teken. Als het patroon een vangst heeft zal het inspringen op de gevangen overeenkomst.

Voorbeelden:

`view.alignOn(document.documentRange(), '-')`; zal spaties invoegen vóór de eerste - van elke regel om ze allemaal uit te lijnen op dezelfde kolom.

`view.alignOn(document.documentRange(), ':\s+(.)')`; zal spaties invoegen vóór het eerste niet-blanke teken dat verschijnt na een dubbele punt om ze allemaal op dezelfde kolom uit te lijnen.

```
object view.executeCommand(String commando, String argumenten, Range reeks);
```

Voert het [opdrachtregelcommando](#) *commando* uit met de optionele argumenten *argumenten* en de optionele *reeks*. Het terugontvangen *object* heeft een eigenschap boolean *object.ok* die aangeeft of uitvoeren van het *commando* met succes was. In geval van een fout, bevat de tekenreeks *object.status* een foutbericht.

Sinds: KDE Frameworks 5.50

```
Range view.searchText(Range reeks, String patroon, bool achterwaarts = false);
```

Naar het eerste voorkomen van *patroon* zoeken in *reeks* en de overeenkomende reeks teruggeven. Zoeken wordt achterwaarts uitgevoerd als de optionele booleaanse parameter *achterwaarts* is ingesteld op *true*.

De teruggegeven reeks is ongeldig (zie `Range.isValid()`) als *patroon* niet wordt gevonden in *reeks*.

Sinds: KDE Frameworks 5.97

6.4.3.4 De API van Document

Wanneer een script uitgevoerd wordt is er een globale variabele 'document' die het huidige actieve document representeert. Het volgende is een lijst met alle beschikbare documentfuncties.

```
String document.fileName();
```

Geeft de bestandsnaam van het document terug of een lege tekenreeks voor niet opgeslagen tekstbuffers.

```
String document.url();
```

Geeft de volledige URL van het document terug of een lege tekenreeks voor niet opgeslagen tekstbuffers.

```
String document.mimeType();
```

Geeft het MIME-type van het document terug of het MIME-type `application/octet-stream` als er geen toepasselijk MIME-type gevonden kan worden.

```
String document.encoding();
```

Geeft de nu gebruikte codering terug om het bestand op te slaan.

```
String document.highlightingMode();
```

Geeft de gebruikte globale accentueringmodus terug voor het gehele document.

```
String document.highlightingModeAt(Cursor pos);
```

Geeft de gebruikte accentueringmodus terug op de gegeven positie in het document.

```
Array document.embeddedHighlightingModes();
```

Geeft een reeks van accentueringmodi terug ingebed in dit document.

bool document.isModified();

Geeft `true` terug, als het document niet opgeslagen wijzigingen heeft(gemodificeerd), anders `false`.

String document.text();

Geef de gehele inhoud van het document terug in een enkele tekenreeks. Nieuwe regels zijn gemarkeerd met het teken nieuw-regel `'\n'`.

String document.text(int vanafRegel, int vanafKolom, int totRegel, int totKolom); String document.text(Cursor vanaf, Cursor tot); String document.text(Range bereik);

Geeft de tekst terug in het gegeven bereik. Het is aanbevolen de cursor te gebruiken en een versie gebaseerd op bereik voor betere leesbaarheid van de broncode.

String document.line(int regel);

Geeft de gegeven tekstregel terug als tekenreeks. De tekenreeks is leeg als de gevraagde regel buiten het bereik ligt.

String document.wordAt(int regel, int kolom); String document.wordAt(Cursor cursor);

Geeft het woord terug op de gegeven cursorpositie.

Range document.wordRangeAt(int regel, int kolom); Range document.wordRangeAt(Cursor cursor);

Geeft de reeks van het woord op de gegeven cursorpositie terug. De teruggeven reeks is ongeldig (zie `Range.isValid()`), als de tekstpositie zich na het eind van een regel bevindt. Als er geen woord is op de gegeven cursorpositie, zal een lege reeks worden teruggegeven.

Sinds: KDE 4.9

String document.charAt(int regel, int kolom); String document.charAt(Cursor cursor);

Geeft het teken terug op de gegeven cursorpositie.

String document.firstChar(int regel);

Geeft het eerste teken in de gegeven *regel* terug die geen witruimte is. Het eerste teken is in kolom 0. Als de regel leeg of alleen witruimte bevat, dan is de teruggegeven tekenreeks leeg.

String document.lastChar(int regel);

Geeft het laatste teken in de gegeven *regel* terug die geen witruimte is. Als de regel leeg of alleen witruimte bevat, dan is de teruggegeven tekenreeks leeg.

bool document.isSpace(int regel, int kolom); bool document.isSpace(Cursor cursor);

Geeft `true` terug, als het teken op de gegeven cursorpositie gelijk is aan een witruimte, anders `false`.

bool document.matchesAt(int regel, int kolom, String tekst); bool document.matchesAt(Cursor cursor, String tekst);

Geeft `true` terug, als de gegeven *tekst* overeenkomt met de overeenkomstige cursorpositie, anders `false`.

bool document.startsWith(int regel, String tekst, bool skipWhiteSpaces);

Geeft `true` terug als de regel begint met *tekst*, anders `false`. Het argument *skipWhiteSpaces* bepaalt of voorloopwitruimte genegeerd worden.

bool document.endsWith(int regel, String tekst, bool skipWhiteSpaces);

Geeft `true` terug als de regel eindigt met *tekst*, anders `false`. Het argument *skipWhiteSpaces* bepaalt of witruimte achteraan genegeerd wordt.

bool document.setText(*String* tekst);

Stelt de gehele documenttekst in.

bool document.clear();

Verwijdert de gehele tekst uit het document.

bool document.truncate(*int* regel, *int* kolom); bool document.truncate(*Cursor* cursor);

Kap de gegeven regel op de gegeven kolom of cursorpositie af. Geeft `true` terug bij succes of `false` als de gegeven regel geen deel is van het documentbereik.

bool document.insertText(*int* regel, *int* kolom, *String* tekst); bool document.insertText(*Cursor* cursor, *String* tekst);

Voeg de *tekst* op de cursorpositie in. Geeft `true` terug bij succes of `false`, als het document alleen-lezen is.

bool document.removeText(*int* vanafRegel, *int* vanafKolom, *int* totRegel, *int* totKolom); bool document.removeText(*Cursor* vanaf, *Cursor* tot); bool document.removeText(*Range* bereik);

Verwijdert de tekst in het gegeven bereik. Geeft `true` terug bij succes of `false`, als het document alleen-lezen is.

bool document.insertLine(*int* regel, *String* tekst);

Voegt tekst in de gegeven regel in. Geeft `true` terug bij succes of `false`, als het document alleen-lezen is of de regel niet in het documentbereik ligt.

bool document.removeLine(*int* regel);

Verwijdert de gegeven tekstregel. Geeft `true` terug bij succes of `false`, als het document alleen-lezen is of de regel niet in het documentbereik ligt.

bool document.wrapLine(*int* regel, *int* kolom); bool document.wrapLine(*Cursor* cursor);

Breekt de regel op de gegeven cursorpositie op. Geeft `true` terug bij succes, anders `false`, bijv. als regel < 0.

Sinds: KDE 4.9

void document.joinLines(*int* beginRegel, *int* eindRegel);

Voegt de regels vanaf *beginRegel* tot *eindRegel* samen. Twee elkaar opvolgende tekstregels worden altijd gescheiden door een enkele spatie.

int document.lines();

Geeft het aantal regels in het document terug.

bool document.isLineModified(*int* regel);

Geeft `true` terug, als *regel* nu niet opgeslagen gegevens bevat.

Sinds: KDE 5.0

bool document.isLineSaved(*int* regel);

Geeft `true` terug, als *regel* is gewijzigd, maar het document is opgeslagen. Dus bevat de regel nu geen niet opgeslagen gegevens.

Sinds: KDE 5.0

bool document.isLineTouched(*int* regel);

Geeft `true` terug, als *regel* nu niet opgeslagen gegevens bevat of eerder is gewijzigd.

Sinds: KDE 5.0

void document.findTouchedLine(int beginRegel, bool omlaag);
Zoek naar de volgende aangeraakte regel beginnend bij *regel*. Het zoeken wordt uitgevoerd ofwel omhoog of omlaag afhankelijk van de zoekrichting gespecificeerd in *omlaag*.
Sinds: KDE 5.0

int document.length();
Geeft het aantal tekens in het document terug.

int document.lineLength(int regel);
Geeft de lengte van de *regel* terug.

void document.editBegin();
Begint een bewerkingsgroep voor groepering van ongedaan maken/opnieuw doen. Controleer altijd of `editEnd()` is aangeroepen als u `editBegin()` aanroept. Aanroepen van `editBegin()` gebruikt intern een referentieteller, bijv. deze aanroep kan genest worden.

void document.editEnd();
Beëindigt een bewerkingsgroep. De laatst aanroep van `editEnd()` (bijv. diegene voor de eerste aanroep van `editBegin()`) beëindigt de bewerkingsstap.

int document.firstColumn(int regel);
Geeft de eerste niet-witruimte kolom in de gegeven *regel* terug. Als er alleen witruimte in de regel bevindt, dan is de terugkeerwaarde -1.

int document.lastColumn(int regel);
Geeft de laatste niet-witruimte kolom in de gegeven *regel* terug. Als er alleen witruimte in de regel bevindt, dan is de terugkeerwaarde -1.

int document.prevNonSpaceColumn(int regel, int kolom); int document.prevNonSpaceColumn(Cursor cursor);
Geeft de kolom terug met een niet-witruimte teken beginnend bij de gegeven cursorpositie en achterwaarts zoeken.

int document.nextNonSpaceColumn(int regel, int kolom); int document.nextNonSpaceColumn(Cursor cursor);
Geeft de kolom terug met een niet-witruimte teken beginnend bij de gegeven cursorpositie en voorwaarts zoeken.

int document.prevNonEmptyLine(int regel);
Geeft de volgende niet-lege regel terug met een niet-witruimte tekens en achterwaarts zoeken.

int document.nextNonEmptyLine(int regel);
Geeft de volgende niet-lege regel terug met een niet-witruimte teken en voorwaarts zoeken.

bool document.isInWord(String teken, int attribuut);
Geeft `true` terug, als het gegeven *teken* met het gegeven *attribuut* deel van een woord kan zijn, anders `false`.

bool document.canBreakAt(String teken, int attribuut);
Geeft `true` terug, als het gegeven *teken* met het gegeven *attribuut* geschikt is om een regel af te breken, anders `false`.

bool document.canComment(int beginAttribuut, int eindAttribuut);
Geeft `true` terug, als een bereik beginnend en eindigend met de gegeven attributen geschikt is om als commentaar te worden aangemerkt, anders `false`.

String document.commentMarker(int attribuut);
Geeft de markeerder voor commentaar terug voor commentaar op een enkele regel voor een gegeven *attribuut*.

String document.commentStart(int attribuut);

Geeft de markeerder voor commentaar terug voor begin van commentaar over meerdere regels voor een gegeven *attribuut*.

String document.commentEnd(int attribuut);

Geeft de markeerder voor commentaar terug voor het einde van commentaar over meerdere regels voor een gegeven *attribuut*.

Range document.documentRange();

Geeft een reeks terug die het gehele document omvat.

Cursor documentEnd();

Geeft een cursor terug gepositioneerd in de laatste kolom van de laatste regel in het document.

bool isValidTextPosition(int regel, int kolom); bool

isValidTextPosition(Cursor cursor);

Geeft *true* terug, als de gegeven cursorpositie is gepositioneerd op een geldige tekstpositie. Een tekstpositie is alleen geldig als het is gelokaliseerd aan het begin, in het midden, of aan het eind van een geldige regel. Verder is een tekstpositie ongeldig als het is gelokaliseerd in een Unicode-surrogaat.

Sinds: KDE 5.0

int document.attribute(int regel, int kolom); int document.attribute(Cursor cursor);

Geeft het attribuut terug op de gegeven cursorpositie.

bool document.isAttribute(int regel, int kolom, int attribuut); bool

document.isAttribute(Cursor cursor, int attribuut);

Geeft *true* terug, als het attribuut op de gegeven cursorpositie gelijk is aan *attribuut*, anders *false*.

String document.attributeName(int regel, int kolom); String

document.attributeName(Cursor cursor);

Geeft de attribuutnaam terug als leesbare tekst. Dit is gelijk aan de *itemData*-naam in de syntaxis-accentuering-bestanden.

bool document.isAttributeName(int regel, int kolom, String naam); bool

document.isAttributeName(Cursor cursor, String naam);

Geeft *true* terug, als de attribuutnaam op een bepaalde cursorpositie overeenkomt met de gegeven *naam*, anders *false*.

String document.variable(String sleutel);

Geeft de waarde terug van de gevraagde documentvariabele *sleutel*. Als de documentvariabele niet bestaat, wordt een lege tekenreeks teruggegeven.

void document.setVariable(String sleutel, String waarde);

Stelt de waarde in van de gevraagde documentvariabele *sleutel*.

Zie ook: [Documentvariabelen van Kate](#)

Sinds: KDE 4.8

int document.firstVirtualColumn(int regel);

Geeft de virtuele kolom terug van het eerste niet-witruimte-teken in de gegeven regel of -1, als de regel leeg is of alleen witruimte-tekens bevat.

int document.lastVirtualColumn(int regel);

Geeft de virtuele kolom terug van het laatste niet-witruimte-teken in de gegeven regel of -1, als de regel leeg is of alleen witruimte-tekens bevat.

```
int document.toVirtualColumn(int regel, int kolom);  
int document.toVirtualColumn(Cursor cursor); Cursor  
document.toVirtualCursor(Cursor cursor);
```

Converteert de gegeven 'echte' cursorpositie naar een virtuele cursorpositie, geeft ofwel een *int* terug of een *cursor-object*.

```
int document.fromVirtualColumn(int regel, int virtueleKolom);  
int document.fromVirtualColumn(Cursor virtueleCursor); Cursor  
document.fromVirtualCursor(Cursor virtueleCursor);
```

Converteert de gegeven virtuele cursorpositie naar een 'echte' cursorpositie, geeft ofwel een *int* terug of een *cursor-object*.

```
Cursor document.anchor(int regel, int kolom, Char teken); Cursor  
document.anchor(Cursor cursor, Char teken);
```

Zoekt achterwaarts naar het gegeven teken te beginnen vanaf de gegeven cursor. Als voorbeeld, als '(' als teken wordt meegegeven, dan geeft deze functie de positie terug van de openingings '('. Dit houdt rekening met aantallen, bijv. andere '(...)' worden genegeerd.

```
Cursor document.rfind(int regel, int kolom, String tekst, int attribuut =  
-1); Cursor document.rfind(Cursor cursor, String tekst, int attribuut =  
-1);
```

Zoek de gegeven tekst achterwaarts met het juiste *attribuut*. Het argument *attribuut* wordt genegeerd als het is ingesteld op -1. De teruggegeven cursor is ongeldig, als de tekst niet is gevonden.

```
int document.defStyleNum(int regel, int kolom); int  
document.defStyleNum(Cursor cursor);
```

Geeft de standaard stijl terug op de gegeven cursorpositie.

```
bool document.isCode(int regel, int kolom); bool document.isCode(Cursor  
cursor);
```

Geeft *true* terug, als het attribuut op de gegeven cursorpositie niet gelijk is aan alle volgende stijlen: *dsComment*, *dsString*, *dsRegionMarker*, *dsChar*, *dsOthers*.

```
bool document.isComment(int regel, int kolom); bool  
document.isComment(Cursor cursor);
```

Geeft *true* terug, als het attribuut van het teken op de cursorpositie gelijk is aan *dsComment*, anders *false*.

```
bool document.isString(int regel, int kolom); bool document.isString(Cursor  
cursor);
```

Geeft *true* terug, als het attribuut van het teken op de cursorpositie gelijk is aan *dsString*, anders *false*.

```
bool document.isRegionMarker(int regel, int kolom); bool  
document.isRegionMarker(Cursor cursor);
```

Geeft *true* terug, als het attribuut van het teken op de cursorpositie gelijk is aan *dsRegionMarker*, anders *false*.

```
bool document.isChar(int regel, int kolom); bool document.isChar(Cursor  
cursor);
```

Geeft *true* terug, als het attribuut van het teken op de cursorpositie gelijk is aan *dsChar*, anders *false*.

```
bool document.isOthers(int regel, int kolom); bool document.isOthers(Cursor  
cursor);
```

Geeft *true* terug, als het attribuut van het teken op de cursorpositie gelijk is aan *dsOthers*, anders *false*.

void document.indent(Range reeks, int aantal);

Laat alle regels in *reeks* inspringen door *aantal* tabs of *aantal* keren `tabSize` spaties in te voegen afhankelijk van de voorkeur van de gebruiker. De parameter *aantal* kan negatief zijn.

6.4.3.5 De API voor bewerken

Naast het document en weergave-API, is er een algemene bewerker-API die functies levert voor algemene functionaliteit voor bewerken met een script.

Tekenreeks editor.clipboardText();

Geeft de tekst terug die actief is in het globale klembord.

Sinds: KDE Frameworks 5.50

String editor.clipboardHistory();

De bewerker houdt een klembordgeschiedenis vast die tot 10 klemborditems bevat. Deze functie geeft alle items terug die actief zijn in de klembordgeschiedenis.

Sinds: KDE Frameworks 5.50

void editor.setClipboardText(String tekst);

Stel de inhoud van het klembord in op *tekst*. De *tekst* zal toegevoegd worden aan de geschiedenis van het klembord.

Sinds: KDE Frameworks 5.50

Hoofdstuk 7

KatePart configureren

Met het kiezen van **Instellingen** → **Toepassing instellen...** uit het menu verschijnt het dialoogvenster **Instellen**. Dit dialoogvenster kan gebruikt worden om een aantal verschillende instellingen te wijzigen. Welke instellingen gewijzigd kunnen worden hangt af van de categorie die u kiest uit een verticale lijst links van het dialoogvenster. Door middel van drie knoppen onderaan het venster kan de gebruiker het proces aansturen.

U kunt het **Help**systeem aanroepen, de huidige instellingen accepteren en het dialoogvenster sluiten met de knop **OK**, of het proces **Annuleren**. De categorieën **Uiterlijk**, **Lettertypen & Kleuren**, **Bewerking**, **Openen/opslaan** en **Extensies** worden hieronder besproken.

7.1 De instellingen voor de tekstbewerkercomponent

Deze groep bevat alle pagina's gerelateerd aan de tekstbewerkercomponent van Kate. De meeste instellingen hier zijn standards, zij kunnen overschreven worden door [een bestandstype te definiëren](#) door [Documentvariabelen](#) of door ze per document te wijzigen in een bewerkingssessie.

7.1.1 Uiterlijk

7.1.1.1 Algemeen

Lettertype voor editor

Hier kunt u het lettertype kiezen voor de tekst in de bewerker. U kunt kiezen uit elk beschikbare lettertype op uw systeem, en een standaardgrootte instellen. Onderaan het dialoogvenster wordt een voorbeeldtekst getoond, zodat u het effect van uw keuze kunt zien.

Voor meer informatie over het selecteren van een lettertype, zie de sectie [Lettertypen kiezen van de documentatie van KDE Fundamentals](#).

Witruimte-indicatoren tonen

Nooit

De tekstbewerker zal nooit punten tonen om het aanwezig zijn van witruimte aan te geven.

Aan het regeleinde

De tekstbewerker zal punten tonen om het aanwezig zijn van witruimte aan het eind van regels aan te geven.

Altijd

De tekstbewerker zal altijd punten tonen om het aanwezig zijn van extra witruimte aan te geven.

Grootte van witruimte-indicator

De schuifregelaar gebruiken om de grootte van de zichtbare indicatormarkering te wijzigen.

Tabindicatoren tonen

Indien geactiveerd zal de editor het symbool » tonen om de aanwezigheid van een tab in de tekst aan te geven.

Haken overeen laten komen

Bereik tussen geselecteerde haakjes accentueren

Wanneer dit ingeschakeld is, zal het bereik tussen de geselecteerde, bij elkaar horende haakjes geaccentueerd worden.

Voorbeeld van overeenkomend openhaakje tonen

Indien geactiveerd zal de editor een tekstballon van het bijbehorende opening vierkante haakje tonen.

Bijbehorend vierkante haakje laten knippen wanneer de cursor naar het andere haakje van het paar beweegt

Indien ingeschakeld, plaatsing op de haken ({, [,], }, (of)) zal snel de overeenkomstige haak laten knippen.

Inspronglijnen tonen

Wanneer dit aangevinkt is, toont de editor verticale lijnen om ingesprongen regels beter te herkennen.

Aantallen

Aantal woorden tonen

Toont het aantal woorden en tekens in het document en in de huidige selectie in de statusbalk. Deze optie is ook beschikbaar in het contextmenu van de statusbalk.

Aantal regels tonen

Toont het aantal regels in het document en in de statusbalk. Deze optie is ook beschikbaar in het contextmenu van de statusbalk.

Eerste regel invouwen

Indien ingeschakeld zal de eerste regel invouwen zijn, indien mogelijk. Dit is nuttig, als het bestand begint met een commentaarregel, zoals een copyright

Dynamische regelafbreking

Wanneer deze optie aangevinkt is, zullen de tekstregels afgebroken worden aan de rechter rand van het editorvenster.

Dynamisch regel afbreken op statische afbreekmarkering van woord

Indien geactiveerd zal de editor regels dynamisch afbreken op de [positie voor statische regelafbreking](#).

Niet op woordgrenzen letten voor dynamische regelafbreking

Indien geactiveerd, houdt de editor geen rekening met woordgrenzen bij afbreken van tekstregels.

Dynamische regelafbrekingindicatoren:

Kies wanneer de dynamische regelafbrekingindicatoren getoond moeten worden, ofwel **Uit**, **Volg regelnummers** of **Altijd aan**.

Afgebroken regels laten inspringen

Bovendien kunt u hier een maximale breedte van het venster, als percentage, instellen waarna dynamisch afgebroken regels niet langer verticaal worden uitgelijnd. Ingesteld op bijvoorbeeld 50 % zullen regels, waarvan het inspringniveau groter is dan 50 % van de vensterbreedte, na afbreking niet verticaal worden uitgelijnd.

Vermenigvuldiger van regelhoogte

Deze waarde zal vermenigvuldigd worden met de standaard regelhoogte van het lettertype. Een waarde van 1,0 betekent dat de standaard hoogte gebruikt zal worden.

7.1.1.2 Randen

Codeblok-invouwing

Pijlen tonen om codeblokken in te vouwen

Wanneer deze optie is aangevinkt, zullen in de huidige weergave markeringen voor code-invouwing getoond worden indien code-invouwing beschikbaar is.

Voorbeeld van de ingevouwen blokken tonen bij erboven zweven

Als deze optie is geselecteerd zal met de muis over het ingevouwen gebied gaan een voorbeeld van de ingevouwen tekst laten zien in een pop-up.

Zichtbaarheid van invouwpijlen

Schakel de invouwpijlen tussen **Bij er boven zweven tonen** en **Altijd tonen**.

Linkerkant

Markeringen tonen

Wanneer dit aangevinkt is, ziet u links een pictogramrand. In de pictogramrand ziet u bijvoorbeeld bladwijzerpictogrammen.

Regelnummering tonen

Wanneer dit aangevinkt is, ziet u links regelnummers.

Gewijzigde en niet opgeslagen regels accentueren

Als dit is geactiveerd zullen markeringen voor wijzigingen in regels zichtbaar zijn. Voor meer informatie, zie Section [3.9](#).

Schuifbalken

Markeringen tonen

Wanneer deze optie aangevinkt is, worden in de huidige weergave markeringen getoond op de verticale schuifbalk. Deze markeringen tonen bijvoorbeeld bladwijzers.

Voorbeeld tonen bij zweven boven schuifbalk

Als deze optie is geactiveerd en u zweeft met de muiswijzer boven de schuifbalk dan zal een klein tekstvoorbeeld met enkele regels van het huidige document rond de cursorpositie worden getoond. Dit stelt u in staat om snel te schakelen naar een ander gedeelte van het document.

Mini-weergave

Mini-weergave tonen

Als u deze optie selecteert zal elke nieuwe weergave een mini-weergave van het document op de verticale schuifbalk tonen.

Voor meer informatie over de schuifbalk met mini-map, zie [Section 3.10](#).

Breedte van mini-weergave

Pas de breedte van de schuifbalk van mini-weergave aan, gedefinieerd in pixels.

Zichtbaarheid van schuifbalken

Schakel de schuifbalk aan, uit of toon de schuifbalk alleen indien nodig. Druk met de linkermuisknop op de blauwe rechthoek om de reeks regelnummers van het getoonde document te tonen op het scherm. Houd de linkermuisknop ingedrukt buiten de blauwe rechthoek om automatisch door het document te rollen.

Bladwijzermenu sorteren

Op datum aangemaakt

Elke nieuwe bladwijzer zal onderaan worden toegevoegd, onafhankelijk van waar die geplaatst is in het document.

Op regelnummer

De bladwijzers zullen geordend worden volgens de regelnummers waarin ze staan.

7.1.2 Kleurenthema's

In deze sectie van het dialoogvenster kunt u alle kleuren in elk kleurenschema waarover u beschikt configureren. Tevens kunt u nieuwe thema's creëren, bestaande thema's wissen of gewoon **Systeemkleurenschema volgen**. Elk thema heeft instellingen voor kleuren en stijlen voor normale en geaccentueerde tekst.

KatePart zal het huidige actieve thema voor u voorselecteren. Als u aan een ander thema wilt werken moet u dat eerst kiezen uit de keuzelijst **Thema selecteren**. Met de knoppen **Kopiëren** en **Verwijderen** kunt u een nieuw thema (een bestaande kopiëren) creëren of een bestaande verwijderen.

Dit is in detail beschreven in [Section 6.3.5](#).

7.1.3 Bewerking

7.1.3.1 Algemeen

Regelafbreking

Regelafbreking is een functie die ervoor zorgt dat de editor automatisch een nieuwe tekstregel begint en de cursor naar het begin van die nieuwe regel verplaatst. KatePart zal automatisch een nieuwe tekstregel beginnen wanneer de huidige regel de positie bereikt die is opgegeven bij de optie [Regels afbreken op](#).

Regels op woord afbreken op een vaste kolom

Schakelt statische regelafbreking in of uit.

Verticale lijn tekenen op de kolom voor regelafbreking

Wanneer deze optie is aangevinkt, verschijnt er een verticale lijn in de regelafbreking-kolom zoals gedefinieerd op het tabblad **Bewerking** van de **Instellingen** → **Editor instellen**. Merk op dat de regelafbrekingmarkering alleen getoond wordt wanneer u een niet-proportioneel lettertype gebruikt.

Regels afbreken op:

Wanneer de optie **Regels afbreken op een vaste kolom** geselecteerd is, bepaalt de invoer hier de positie (in tekens) waarop de editor automatisch naar een nieuwe regel zal gaan.

Standaard invoermodus

De geselecteerde invoermethode zal geactiveerd worden wanneer er een nieuw venster wordt geopend. U kunt nog altijd schakelen tussen de een bepaald venster in het menu **Bewerken**.

Blokhaakjes

Als de optie **Automatisch haak sluiten wanneer openingshaak is getypt** is geselecteerd wanneer de gebruiker een linker haak ([, (of {) intypt, plaatst KatePart automatisch het rechter haakje (],) of }) rechts van de cursor.

Omsluitende tekens

Het is mogelijk de omsluitende tekens te selecteren met de bijbehorende afrollijst.

Wanneer tekst is geselecteerd, zorgt typen van één van deze tekens dat de geselecteerde tekst verdeelt wordt over meer regels.

Knippen en plakken

Geselecteerde tekst verplaatsen bij slepen

Deze optie schakelt slepen-en-loslaten in van de geselecteerde tekst in het venster van de bewerker.

Kopieer/knip de huidige regel als er geen geselecteerde tekst is

Als deze optie is ingeschakeld, en er geen tekst is geselecteerd, worden kopieer- en knipacties toegepast op de tekstregel waar de cursor zich bevindt.

De tekstcursor niet verplaatsen bij plakken met de muis

Als deze optie is ingeschakeld en u plakt enige tekst in het venster van de bewerker met klikken met de middelstemuisknop, zal KatePart de tekstcursor niet verplaatsen naar de positie van de klik.

7.1.3.2 Tekstnavigatie

Tekstcursorbeweging

Intelligente begin- en eindpositie

Wanneer dit geselecteerd is, zal bij het indrukken van de home-toets de cursor witruimte overslaan en naar het begin van de tekst in een regel gaan.

Page Up/Page Down beweegt de cursor

Deze optie wijzigt het gedrag van de cursor wanneer de gebruiker drukt op de toets **PgUp** of **PgDn**. Wanneer deze optie niet aangevinkt is, behoudt de cursor zijn relatieve positie binnen de zichtbare tekst in KatePart zodra nieuwe tekst zichtbaar wordt ten gevolge van deze handeling. Dus, wanneer de cursor zich in het midden van de zichtbare tekst bevindt wanneer er op de toets wordt gedrukt, zal die op die positie blijven (behalve wanneer het begin of einde bereikt wordt). Wanneer deze optie aangevinkt is, zorgt de eerste druk op de toets ervoor dat de cursor naar het begin of einde van de zichtbare tekst wordt verplaatst zodra een nieuwe pagina met tekst getoond wordt.

Cursorbeweging bij hoofdletters inschakelen

Deze optie wijzigt het gedrag van de cursor wanneer de gebruiker de sneltoets **Ctrl-Links** of **Ctrl-Rechts** indrukt. Indien niet geactiveerd zal de tekstcursor over de volledige woorden springen. Met deze optie geactiveerd zal de cursorsprong stoppen bij hoofdletters.

Cursor automatisch centreren:

Stelt het aantal regels in die zichtbaar blijven boven en onder de cursor indien mogelijk.

Tekstselectiemodus

Normaal

Selecties worden overschreven door getypte tekst en worden opgeheven bij verplaatsing van de cursor.

Blijvend

Selecties blijven bestaan, ook na verplaatsen van de cursor en typen.

Sta schuiven toe verder dan het einde van het document

Met deze optie kunt u verder dan het einde van het document schuiven. Dit kan gebruikt worden om de onderkant van het document verticaal te centreren of om het bovenaan de huidige weergave te plaatsen.

De toets **backspace** verwijdert het basis teken met zijn diakritische tekens

Indien geselecteerd worden samengestelde tekens verwijderd met hun diakritische tekens in plaats van alleen het basis teken te verwijderen. Dit is nuttig voor Indic taalgebieden.

Multicursor-modifier

Deze optie laat u de modifier instellen die gebruikt zal worden om meerdere cursors met een linkermuisknop-klik aan te maken. U moet de modifiers indrukken en op de linkermuisknop klikken om een cursor aan te maken op de gewenste locatie. Zie [Meerdere cursors aanmaken](#) om andere manieren te ontdekken bij het aanmaken van meerdere cursors.

7.1.3.3 Inspringmethode

Standaard inspringmodus:

Kies de automatische inspringmodus die u als standaard wilt gebruiken. Het is ten sterkste aan te raden hier **Geen** of **Normaal** te gebruiken en bestandstypeconfiguraties te gebruiken om andere inspringmodi in te stellen voor tekstformaten zoals C/C++ code of XML.

Inspringen gebruikt

Tabulators

Wanneer dit ingeschakeld is, zal de editor tab-tekens invoegen wanneer u de **Tab**-toets indrukt of gebruik [automatisch inspringen](#).

Spaties

Wanneer dit ingeschakeld is, zal de editor een bepaald aantal spaties invoegen, afhankelijk van de positie in de tekst en de instelling `tabbreedte`, wanneer u op de **Tab**-toets drukt of gebruik [automatisch inspringen](#).

Tabulators en spaties

Als dit is ingeschakeld zal de tekstbewerker spaties invoegen zoals hierboven beschreven bij inspringen of **Tab** indrukken aan het begin van een regel, maar voegt tabulators in bij het indrukken van de **Tab**-toets in het midden of einde van een regel.

Tabbreedte:

Dit stelt het aantal spaties in dat wordt getoond in plaats van een tab-teken.

Inspringbreedte:

De inspringbreedte is het aantal spaties dat gebruikt wordt om een regel in te springen. Wanneer is ingesteld dat tab-teken moet worden gebruikt, dan wordt er een tab-teken ingevoegd als de insprong deelbaar is door de tabbreedte.

Inspringeigenschappen

Extra spaties behouden

Wanneer deze optie uitgeschakeld is en het inspringniveau gewijzigd wordt, zal de regel uitgelijnd worden op een veelvoud van de breedte opgegeven bij **Inspringbreedte**.

Inspringen aanpassen van tekst geplakt vanaf het klembord

Wanneer deze optie geactiveerd is, springt tekst die van het klembord geplakt wordt in. Door te kiezen voor **Ongedaan maken** wordt het inspringen verwijderd.

Inspringacties

Backspace-toets in eerste witruimte maakt insprong ongedaan

Wanneer deze optie aangevinkt is, verlaagt de **Backspace**-toets het inspringniveau als de cursor in de witruimte aan het begin van de regel staat.

Tab-toets-actie (als er niets is geselecteerd)

Als u wilt dat **Tab** de huidige regel uitlijnt in het huidige codeblok zoals in Emacs, maak dan van **Tab** een sneltoets voor de actie **Inspringen formatteren**.

Altijd naar volgende tab-positie gaan

Wanneer deze optie geselecteerd is, voegt de **Tab**-toets altijd witruimte in zodat de volgende tab-positie wordt bereikt. Wanneer de optie **Spaties in plaats van tabs invoegen** op het tabblad **Algemeen** in de sectie **Bewerking** geactiveerd is, worden er spaties ingevoegd; anders wordt een enkele tabsprong ingevoegd.

Altijd inspringniveau vergroten

Wanneer deze optie geselecteerd is, wordt met de **Tab**-toets de huidige regel altijd ingesprongen met het aantal tekens opgegeven bij **Inspringbreedte**.

Inspringniveau vergroten wanneer in voorgaande witruimte

Wanneer deze optie geselecteerd is, zorgt de **Tab**-toets voor het inspringen van de huidige regel of de cursor gaat naar de volgende tab-positie. Als het invoegpunt zich op of voor het eerste niet-witruimteteken in de regel bevindt, of wanneer er een selectie is, wordt de huidige regel ingesprongen met het aantal tekens opgegeven bij **Inspringbreedte**. Bevindt het invoegpunt zich achter het eerste niet-witruimteteken in de regel en er is geen selectie, dan wordt er witruimte ingevoegd zodat de volgende tabpositie bereikt wordt: Wanneer de optie **Spaties in plaats van tabs invoegen** op het tabblad **Algemeen** in de sectie **Bewerking** geactiveerd is, worden er spaties ingevoegd; anders wordt een enkele tabsprong ingevoegd.

7.1.3.4 Automatische aanvulling

Algemeen

Automatisch aanvullen activeren

Indien geactiveerd komt er automatisch een vakje met aanvullingen te voorschijn bij het intypen die een lijst met teksten toont om de huidige tekst onder de cursor aan te vullen.

Automatische selecteren van item voor eerste aanvulling

Indien ingeschakeld is het eerste item met automatisch aanvulling altijd voorgeselecteerd zodat u het met **Enter** kunt invoegen. Als u dat gedrag niet wilt, bijv. als u **Enter** alleen om een nieuwe regel in te voegen, schakel dan dit item uit.

Minimale woordlengte aan te vullen

Tijdens het typen zoekt Woordaanvulling in het document naar woorden beginnend met de reeds getypte tekst. Deze optie stelt het minimum aantal tekens in, nodig om Woordaanvulling te activeren en een aanvullingslijst te tonen.

Achterste gedeelte bij aanvullen verwijderen

Verwijder het achterste deel van een vorig woord wanneer het aangevulde item is gekozen uit een lijst.

Automatische aanvulling van sleutelwoorden

Indien ingeschakeld gebruikt de ingebouwde automatische aanvulling de sleutelwoorden gedefinieerd door de accentuering van syntaxis.

7.1.3.5 Spellingcontrole

Deze instelopties worden beschreven in de documentatie voor de Systeeminstellingen-module [Spellingcontrole](#).

7.1.3.6 VI-invoermethode

Algemeen

VI-commando's in plaats van Kate-sneltoetsen gebruiken

Wanneer dit aangevinkt is, zullen VI-commando's de ingebouwde commando's van KatePart vervangen. Bijvoorbeeld: **Ctrl+R** voert het commando Opnieuw uit in plaats van de standaard actie (het tonen van het dialoogvenster Zoeken en vervangen).

Relatieve regelnummers tonen

als dit ingeschakeld is verwijst de huidige regel altijd naar lijn 0. Bovenliggende en onderliggende lijnen verhogen het regelnummer relatief.

Toetsmapping

Sneltoetsen instellen wordt gebruikt om de functie van toetsen te wijzigen. Hierdoor kunt u opdrachten aan andere toetsen toekennen of speciale toetsaanslagen maken voor het uitvoeren van een reeks opdrachten.

Voorbeeld:

F2 -> I-- Esc

Hiermee wordt **I--** voor een regel gezet wanneer **F2** wordt ingedrukt.

7.1.4 Openen/opslaan

7.1.4.1 Algemeen

Bestandsformaat

Codering

Dit definieert de standaard te gebruiken codering om bestanden te openen/op te slaan, indien niet gewijzigd in de dialoog openen/opslaan of door een commandoregel-optie te gebruiken.

Codering detecteren

Selecteer een item uit de keuzelijst: u kunt automatisch detecteren uitschakelen of **Universeel** gebruiken om het automatisch detecteren in te schakelen voor alle coderingen. Omdat dit mogelijk alleen utf-8/utf-16 detecteert, zal het kiezen van een regio waarbij aangepaste heuristieken worden gebruikt tot betere resultaten leiden. Als noch de hierboven standaard gekozen codering, noch de codering gespecificeerd op de commandoregel overeenkomen met de inhoud van het bestand, dan zal deze detectie worden uitgevoerd.

Terugvalcodering

Dit definieert de te proberen terugvalcodering bij het openen van bestanden als noch de boven als standaard gekozen codering, noch de codering gespecificeerd in de dialoog voor openen/opslaan, noch de codering gespecificeerd op de commandoregel overeenkomt met de inhoud van het bestand. Voordat dit wordt gebruikt, wordt een poging gedaan om te kijken naar een bytevolgordemarkering aan het begin van het bestand voor de te gebruiken codering: als deze is gevonden, wordt de juiste unicode-codering gekozen; anders wordt detectie van codering gedaan, als beide mislukken wordt terugvalcodering geprobeerd.

Regeleinde

Kies de gewenste regeleindemodus voor het actieve document. U kunt kiezen tussen UNIX®, DOS/Windows® of Macintosh.

Regeleinde automatisch detecteren

Vink dit aan als u wilt dat de editor automatisch het soort regeleinde detecteert. Het soort regeleinde dat het eerst gevonden wordt zal gebruikt worden voor het hele bestand.

Bytevolgordemarkering (BOM) inschakelen

De bytevolgordemarkering is een speciale reeks aan het begin van documenten gecodeerd in unicode. Het helpt gebruikers tekstdocumenten te openen met de juiste unicode-codering. Zie voor meer informatie: [Byte Order Mark](#).

Limiet van de regellengte

Helaas ervaart, vanwege zwakke zaken in Qt™, KatePart slechte prestaties bij het werken met extreem lange regels. Daarom zal KatePart regels automatisch opsplitsen wanneer ze langer zijn dan het aantal hier gespecificeerde tekens. Om dit uit te schakelen stelt u dit on op 0.

Automatisch opschonen bij opslaan

Spaties aan het einde van de regel tijdens het bewerken verwijderen

De bewerker zal automatisch extra spaties aan het einde van regels tekst verwijderen bij het opslaan van het bestand. U kunt **Nooit** kiezen om deze functie uit te schakelen, **Op gewijzigde regels** om dat alleen te doen op regels die u hebt gewijzigd sinds u het document eerder hebt opgeslagen of **In gehele document** om ze onvoorwaardelijk uit het gehele document te verwijderen.

Nieuwe-regel aan het eind van het bestand bij opslaan

De editor zal automatisch een nieuwe-regel aan het eind van het bestand voegen, als deze niet aanwezig is wanneer het bestand wordt opgeslagen.

Automatisch opslaan inschakelen (alleen lokale bestanden)

Activeer dit als u wilt dat de bewerker documenten automatisch opslaat terwijl u er aan werkt.

Document automatisch opslaan wanneer de bewerker geen focus meer heeft

De bewerker zal documenten automatisch opslaan wanneer u omschakelt naar iets buiten de bewerker, bijv., het terminalpaneel in Kate.

Tijdsinterval tussen automatisch opslaan

U kunt hier het interval, in seconden, voor automatisch opslaan bepalen. Als het interval 0 is, zal het document niet na intervallen automatisch worden opgeslagen.

7.1.4.2 Geavanceerd

Een backup-bestand schrijven bij opslaan voor

Het maken van een reservekopie bij opslaan houdt in dat KatePart het bestand op schijf (de eerder opgeslagen versie van het bestand) kopieert naar <voorvoegsel><bestandsnaam><achtervoegsel> voordat de nieuwe wijzigingen worden opgeslagen. Een reserve bestand kan helpen bij het herstellen van werk als er iets verkeerd gaat bij opslaan of als u later de vorige versie van het bestand wilt herstellen. Het standaard achtervoegsel is ~ en standaard is het voorvoegsel leeg.

Lokale bestanden

Vink dit aan als u reservekopieën wilt hebben van lokale bestanden wanneer die worden opgeslagen.

Externe bestanden

Vink dit aan als u reservekopieën wilt hebben van externe bestanden wanneer die worden opgeslagen.

Voorvoegsel voor backup-bestanden

Voer het voorvoegsel in dat u wilt gebruiken voor reservekopieën.

Achtervoegsel voor backup-bestanden

Voer het achtervoegsel in dat aan de naam van de reservekopie moet worden toegevoegd.

Modus van swap-bestand

KatePart kan (het meeste van) niet-opgeslagen werk, in het geval van een crash of stroomuitval herstellen. Een swapbestand (<bestandsnaam>.kate-swp) wordt aangemaakt wanneer een document wordt bewerkt. Als de gebruiker de wijzigingen niet opslaat en KatePart crasht, dan blijft het swapbestand op de schijf. Bij het openen van een bestand controleert KatePart of er een swapbestand voor het document bestaat en als het bestaat, dan vraagt het aan de gebruiker of hij de verloren gegevens wil herstellen of niet. De gebruiker heeft ook de mogelijkheid om de verschillen tussen het originele bestand en het herstelde te bekijken. Het swapbestand wordt verwijderd na elk opslaan en bij normaal beëindigen.

KatePart synchroniseert de swapbestanden op de schijf elke 15 seconden, maar alleen als ze zijn gewijzigd sinds de laatste synchronisatie. De gebruiker kan het synchroniseren van de swapbestanden uitschakelen, als hij dat wil, door het vakje **Uitschakelen** te selecteren, maar dat kan tot meer verlies van gegevens leiden.

Wanneer het swap-bestand is ingeschakeld is het mogelijk om te schakelen tussen twee modi, namelijk **Ingeschakeld, opslaan in standaard map** en **Ingeschakeld, opslaan in aangepaste map**.

Swap-bestanden opslaan in

Standaard worden de swap-bestanden opgeslagen in dezelfde map als het bestand. Wanneer **Ingeschakeld, opslaan in aangepaste map** is gekozen voor modus voor swap-bestanden, worden swap-bestanden aangemaakt in de gespecificeerde map. Dit is nuttig voor netwerkbestandssystemen om onnodig netwerkverkeer te voorkomen.

Swap-bestanden opslaan elke

KatePart synchroniseert de swap-bestanden op de schijf elke 15 seconden, maar alleen als ze zijn gewijzigd sinds de laatste synchronisatie. U kunt het interval voor synchronisatie maar eigen behoefte aanpassen.

7.1.4.3 Modi & bestandstypen

Op dit tabblad kunt u de standaardconfiguratie voor documenten van specifieke MIME-typen vervangen. Wanneer de editor een bestand laadt, wordt gekeken of het overeenkomt met de

bestandsmaskers of MIME-typen voor één van de gedefinieerde bestandstypen. Indien dat het geval is worden de gedefinieerde variabelen toegepast. Als er meerdere bestandstypen overeenkomen, zal die met de hoogste prioriteit gebruikt worden.

Bestandstype:

Het bestandstype met de hoogste prioriteit is het type dat in de eerste keuzelijst getoond wordt. Als er meerdere bestandstypen werden gevonden komen die ook in de lijst te staan.

Nieuw

Dit wordt gebruikt voor het aanmaken van een nieuw bestandstype. Nadat u op deze knop gedrukt hebt, zijn de velden eronder leeg en kunt u de gewenste eigenschappen voor het nieuwe bestandstype invullen.

Verwijderen

Om een bestaand bestandstype te verwijderen kiest u het uit de keuzelijst en drukt u op de knop Verwijderen.

Eigenschappen van huidige bestandstype

Het bestandstype met de hoogste prioriteit is het type dat in de eerste keuzelijst getoond wordt. Als er meerdere bestandstypen werden gevonden komen die ook in de lijst te staan.

Naam:

De naam van het bestandstype wordt de tekst van het overeenkomstige menu-item. Deze naam wordt getoond in het menu **Hulpmiddelen** → **Bestandstypen**.

Sectie:

De sectienaam wordt gebruikt om de bestandstypen in menus te organiseren. Deze wordt ook gebruikt in het menu **Hulpmiddelen** → **Bestandstypen**.

Variabelen:

Met deze tekenreeks kunt u de KatePart-instellingen configureren voor de bestanden die door dit MIME-type gekozen zijn, gebruik makend van KatePart-variabelen. U kunt bijna elke configuratie-optie instellen, zoals accentuering, inspringmethode, etc. Druk op **Bewerken** om een lijst te zien van alle beschikbare variabelen en hun beschrijving. Selecteer het keuzevakje links om een bepaalde variabele in te schakelen en stel dan de waarde van de variabele rechts in. Sommige variabelen bieden een afrolvak om mogelijke waarden te selecteren terwijl anderen vereisen dat u handmatig een waarde invoert. Voor volledige informatie over deze variabelen, zie [Configureren met documentvariabelen](#).

Accentuering:

Als u een nieuw bestandstype aanmaakt, kunt u in deze keuzelijst een bestandstype selecteren voor de accentuering.

Inspringmodus:

In de keuzelijst kunt u de inspringmodus voor nieuwe documenten selecteren.

Bestandsextensies:

Het jokertekensmasker stelt u in staat bestanden op bestandsnaam te kiezen. Een masker bestaat vaak uit een asterisk en de extensie van het bestand, bijvoorbeeld *.txt; *.text. De tekenreeks bestaat uit een lijst van maskers, gescheiden door een puntkomma.

MIME-bestandstypen:

Toont een assistent waarmee u gemakkelijk MIME-typen kunt selecteren.

Prioriteit:

Stelt een prioriteit in voor dit bestandstype. Als hetzelfde bestand door meer dan één bestandstype gekozen wordt, zal het type met de hoogste prioriteit gebruikt worden.

7.2 Configureren met documentvariabelen

KatePart-variabelen is de implementatie van documentvariabelen van KatePart, vergelijkbaar met modusregels van Emacs en vi. In katepart hebben de regels het volgende formaat **kate: VARIABELENAAM WAARDE; [VARIABELENAAM WAARDE; ...]** de regels kunnen zich natuurlijk in commentaar bevinden, als het bestand in een formaat is met commentaar. Variabelennamen zijn enkele woorden (geen witruimte) en alles tot aan de volgende puntkomma is de waarde. De puntkomma is vereist.

Hier is een voorbeeld van een variabelenregel, die instellingen voor inspringen forceert voor een bestand met C++, Java™ of JavaScript:

```
// kate: replace-tabs on; indent-width 4; indent-mode cstyle;
```

OPMERKING

Alleen in de eerste en laatste 10 regels worden gezocht naar variabelenregels.

Bovendien, kunnen documentvariabelen in een bestand geplaatst worden genaamd `.kateconfig` in elke map en de geconfigureerde instellingen zullen worden toegepast alsof de modusregels ingevoerd zijn op elk bestand in de map en zijn submappen. Documentvariabelen in `.kateconfig` gebruiken dezelfde syntax als die in modusregels, maar met [uitgebreide opties](#).

Er zijn variabelen om bijna alle configuraties in KatePart te ondersteunen, en extra plugins kunnen variabelen gebruiken, in welk geval het gedocumenteerd zou moeten zijn in de documentatie van de plugin.

KatePart heeft ondersteuning voor het lezen van configuraties uit `.editorconfig` bestanden, wanneer de bibliotheek [editorconfig](#) is geïnstalleerd. KatePart zoekt automatisch naar een `.editorconfig` wanneer u een bestand opent. Het geeft echter prioriteit aan bestanden `.kateconfig`.

7.2.1 Hoe KatePart variabelen gebruikt

Bij het lezen van de instellingen kijkt katepart op de volgende plaatsen (in die volgorde):

- De globale instellingen.
- Optionele sessiegegevens.
- De instellingen voor "Bestandstype".
- Documentvariabelen in `.kateconfig`.
- Documentvariabelen in het document zelf.
- Instellingen die zijn gemaakt tijdens het bewerken uit het menu of vanaf de commandoregel.

Zoals u kunt zien hebben documentvariabelen die gewijzigd zijn tijdens uitvoering voorrang. Wanneer een document wordt opgeslagen zullen de documentvariabelen worden herlezen en zullen de wijzigingen overschrijven die zijn gemaakt door menu-items of door de commandoregel te gebruiken.

Elke variabele, die niet hieronder wordt genoemd, wordt opgeslagen in het document en kan door andere objecten, zoals plugins, worden opgevraagd, die ze voor hun eigen doeleinden kunnen gebruiken. De variabele 'indent-mode' gebruikt bijvoorbeeld documentvariabelen voor zijn configuratie.

De hier getoonde variabelen documenteren KatePart versie 5.38. Meer variabelen kunnen in de toekomst worden toegevoegd. Er zijn 3 mogelijke typen waarden voor variabelen, met de volgende geldige expressies:

- **BOOL** - on | off | true | false | 1 | 0
- **INTEGER** - elk geheel getal
- **STRING** - iets anders

7.2.2 Beschikbare variabelen

auto-brackets [BOOL]

Automatische vierkante haakjes invoegen inschakelen.

auto-center-lines [INT]

Stelt het aantal autocentreer-regels in.

background-color [STRING]

Stelt de kleur in van de achtergrond van het document. De waarde moet iets zijn dat geëvalueerd kan worden tot een geldige kleur, bijvoorbeeld **#ff0000**.

backspace-indent [BOOL]

Indentering in- of uitschakelen wanneer **Backspace** wordt ingedrukt.

block-selection [BOOL]

Zet [selectie van blokken](#) aan of uit.

bom | **byte-order-mark** | **byte-order-marker** [BOOL]

Schakelt bytevolgordemarkering (BOM) in/uit bij het opslaan van bestanden in het formaat unicode (utf8, utf16, utf32).

Sinds: Kate 3.4 (KDE 4.4)

bracket-highlight-color [STRING]

Stel de kleur in voor accentuering van haakjes. De waarde moet iets zijn dat geëvalueerd kan worden tot een geldige kleur, bijvoorbeeld **#ff0000**”

current-line-color [STRING]

Stelt de kleur in van de huidige regel. De waarde moet iets zijn dat geëvalueerd kan worden tot een geldige kleur, bijvoorbeeld **#ff0000**”.

default-dictionary [STRING]

Stelt het standaard woordenboek in dat wordt gebruikt voor spellingcontrole.

Sinds: Kate 3.4 (KDE 4.4)

dynamic-word-wrap [BOOL]

Schakelt [dynamische regelafbreking](#) in of uit.

eol | **end-of-line** [STRING]

Stelt de modus van einde-regel in. Geldige instellingen zijn **unix**, **mac** en **dos**.

folding-markers [BOOL]

Zet het tonen van [invouwmarkering](#) aan of uit.

folding-preview [BOOL]

Voorbeeld van invouwen in de rand van bewerker.

font-size [INT]

Stelt de grootte van het lettertype van het document in in punten.

font [STRING]

Stelt het lettertype van het document in. De waarde moet de naam van een geldig lettertype zijn, bijvoorbeeld **courier**.

hl | syntax [STRING]

Stelt de accentuering van de syntax in. Geldige tekenreeksen zijn alle namen beschikbaar in de menu's. Bijvoorbeeld voor C++, schrijf eenvoudig **C++**.

icon-bar-color [STRING]

Stelt de kleur in de pictogrambalk. De waarde moet iets zijn dat geëvalueerd kan worden tot een geldige kleur, bijvoorbeeld **#ff0000**.

icon-border [BOOL]

Zet het tonen van de pictogramrand aan of uit.

indent-mode [STRING]

Stelt de auto-inspringmodus in. De opties **none**, **normal**, **cstyle**, **haskell**, **lilypond**, **lisp**, **python**, **ruby** en **xml** worden herkend. Zie de sectie Section 3.8 voor details.

indent-pasted-text [BOOL]

Inspringen in-/uitschakelen van tekst geplakt vanaf het klembord

Sinds: Kate 3.11 (KDE 4.11)

indent-width [INT]

Stelt de inspringbreedte in.

keep-extra-spaces [BOOL]

Stelt in of extra spaties behouden blijven wanneer de inspringbreedte wordt berekend.

line-numbers [BOOL]

Zet het tonen van regelnummers aan of uit.

newline-at-eof [BOOL]

Voeg een lege regel toe aan het eind van het bestand (EOF) wanneer document wordt opgeslagen.

Sinds: Kate 3.9 (KDE 4.9)

overwrite-mode [BOOL]

Overschrijfmodus aan/uitzetten.

persistent-selection [BOOL]

Zet [blijvende selecties](#) aan of uit.

replace-tabs-save [BOOL]

Zet conversie van tab naar spatie bij opslaan aan of uit.

replace-tabs [BOOL]

Zet dynamische conversie van tab naar spatie aan of uit.

remove-trailing-spaces [TEKENREEKS]

Spaties aan het einde van de regel bij het opslaan verwijderen. Geldige opties zijn:

- **none**, **-** of **0**: nooit spaties aan het einde verwijderen.
- **modified**, **mod**, **+** of **1**: spaties aan het eind alleen verwijderen in gewijzigde regels. De gewijzigde regels worden gemarkeerd door het regelwijzigingssysteem.
- **all**, ***** or **2**: spaties aan het eind in het gehele document verwijderen.

scrollbar-minimap [BOOL]

Schuifbalk met mini-weergave tonen

scrollbar-preview [BOOL]

Voorbeeld op schuifbalk tonen

scheme [STRING]

Stelt het kleurenschema in. De tekenreeks moet de naam van een kleurenschema zijn dat in uw configuratie bestaat om enig effect te hebben.

selection-color [STRING]

Stelt de kleur van een selectie in. De waarde moet iets zijn dat geëvalueerd kan worden tot een geldige kleur, bijvoorbeeld **#ff0000**.

show-tabs [BOOL]

Zet het visuele tab-teken aan of uit.

smart-home [BOOL]

Zet [intelligent navigeren naar de basismap](#) aan of uit.

tab-indents [BOOL]

Zet het inspringen met de **Tab**-toets aan of uit.

tab-width [INT]

Stelt de weergavebreedte van het tab-teken in.

undo-steps [INT]

Stelt het aantal te herinneren stappen in.

Opmerking: Verouderd sinds Kate 3 in KDE4. Deze variabele wordt genegeerd. Het maximum aantal stappen voor terugdraaien is ongelimiteerd.

word-wrap-column [INT]

Stelt breedte in van [statische regelafbreking](#).

word-wrap-marker-color [TEKENREEKS]

Stelt de kleur in van de marker voor regelafbreking. De waarde moet iets zijn dat geëvalueerd kan worden tot een geldige kleur, bijvoorbeeld **#ff0000**.

word-wrap [BOOL]

Zet statische regelafbreking aan of uit.

7.2.3 Uitgebreide opties in **.kateconfig** bestanden

KatePart zoekt altijd naar een **.kateconfig** bestand voor locale bestanden (geen bestanden op afstand). Bovendien is het mogelijk opties in te stellen gebaseerd op jokers (bestandsextensies) als volgt:

```
kate: tab-width 4; indent-width 4; replace-tabs on;
kate-wildcard(*.xml): indent-width 2;
kate-wildcard(Makefile): replace-tabs off;
```

In dit voorbeeld gebruiken alle bestanden een tab-breedte van 4 spaties, een indentatiebreedte van 4 spaties en tabs worden vervangen door spaties. Voor alle ***.xml** bestanden, is de indentatiebreedte ingesteld op 2 spaties. En Makefiles gebruiken tabs, bijv. tabs worden niet vervangen door spaties.

Jokers worden gescheiden door puntkomma's, bijv. u kunt ook meerdere bestandsextensies specificeren als volgt:

Het handboek van KatePart

```
kate-wildcard(*.json;*.xml): indent-width 2;
```

U kunt verder ook het MIME-type te gebruiken om overeen te laten komen, bijv. om alle C++ bronbestanden te indenteren met 4 spaties, u kunt schrijven:

```
kate-mimetype(text/x-c++src): indent-width 4;
```

OPMERKING

Naast de ondersteuning in `.kateconfig` bestandstypen, worden jokers en MIME-type afhankelijke documentvariabelen ook ondersteund in de bestandstype zelf als commentaar.

Hoofdstuk 8

Dankbetuigingen en licentie

KatePart en KWrite Copyright 2001-2022 door het team van Kate.

Gebaseerd op het originele KWrite, waarvoor geldt: Copyright 2000 door Jochen Wilhelmy digisnap@cs.tu-berlin.de

Bijdragen:

- Christoph Cullmann cullmann@kde.org
- Michael Bartl michael.bartl1@chello.at
- Phlip phlip_cpp@my-deja.com
- Anders Lund anders@alweb.dk
- Matt Newell newellm@proaxis.com
- Joseph Wenninger kde@jowenn.at
- Jochen Wilhely digisnap@cs.tu-berlin.de
- Michael Koch koch@kde.org
- Christian Gebauer gebauer@kde.org
- Simon Hausmann hausmann@kde.org
- Glen Parker glenebob@nwlk.com
- Scott Manson sdmanson@altel.net
- John Firebaugh jfirebaugh@kde.org
- Nibaldo González nibgonz@gmail.com

De documentatie van KatePart is gebaseerd op de originele documentatie van KWrite, gewijzigd om relevant te zijn voor alle KatePart gebruikers.

De originele documentatie van KWrite is geschreven door Thad McGinnis ctmcginnis@compuserve.com, met veel wijzigingen van Christian Tibirna tibirna@kde.org. Geconverteerd naar docbook/proofreading door Lauri Watts lauri@kde.org en bijgewerkt door Anne-Marie Mahfouf annma@kde.org en Anders Lund anders@alweb.dk

De huidige documentatie van KatePart wordt onderhouden door T.C. Hollingsworth thollingsworth@gmail.com. Om wijzigingen dien een bugrapport in op het [KDE Bugvolgsysteem](#) of lever een patch op ons [GitLab-project](#).

Het handboek van KatePart

Op- of aanmerkingen over de vertalingen van de toepassing en haar documentatie kunt u melden op <http://www.kde.nl/bugs>.

Dit document is vertaald in het Nederlands door Niels Reedijk nielx@kde.nl.

Dit document is vertaald in het Nederlands door Tom Albers tomalbers@kde.nl.

Dit document is vertaald in het Nederlands door Hannie Lafeber-Dumoleyn hannie@kde.nl.

Deze documentatie valt onder de bepalingen van de [GNU vrije-documentatie-licentie](#).

Deze toepassing valt onder de bepalingen van de [GNU General Public License](#).

Hoofdstuk 9

De VI-invoermethode

Erlend Hamberg
Vertaler/Nalezer: Freek de Kruijf

9.1 VI-invoermethode

Het doel van de VI-modus is niet om een volledige vervanging voor Vim te zijn en alle functies van Vim te ondersteunen. Zijn doel is om het op 'Vim manier' van tekstbewerken te doen - en de geleerde Vim-gewoonten - beschikbaar te maken voor programma's die de tekstbewerker KatePart als hun interne editor gebruiken.

Het doel van de VI-modus is te integreren netjes met het programma en wijkt af van het gedrag van Vim als dat zinvol is. Bijvoorbeeld :w zal een dialoog voor opslaan openen in de VI-modus van KatePart.

Om de VI-invoermodus voor alle nieuwe weergaven in te schakelen, gaat u naar **Instellingen** → **KatePart instellen...+Bewerken** → **VI-invoermodus**. Op dit tabblad kunt u opties instellen voor de VI-invoermodus en de toetsenmapping definiëren en bewerken in deze modus. VI-invoermodus kan ook omgeschakeld worden met de instelling **VI-invoermodus** in het menu **Bewerken**. (De standaard sneltoets is **Meta+Ctrl+V** - waar **Meta** gewoonlijk de toets **Windows** is).

OPMERKING

Veel toetsenbordcommando's in vi-modus zijn, anders dan de meeste sneltoetsen van KDE, gevoelig voor hoofd- en kleine letters. Dit betekent dan **y** en **Y** verschillende commando's zijn. Om het **y** (yank) commando in te voeren, moet u controleren dat **Caps Lock** is uitgeschakeld en **Y** indrukken. Om het commando **Y** (yank tot het eind van de regel), voert u **Shift+Y** uit.

Dit is niet van toepassing op commando's die de toets **Ctrl** gebruiken, die mogen worden ingevoerd onafhankelijk van de modus van **Caps Lock** en zonder **Shift** in te drukken. Sommige commando's vereisen het gebruik van een **Ctrl**-toetscombinatie gevolgd door een andere toets die wel gevoelig is voor hoofd- en kleine letters. Om '**Ctrl+W, h**' (omschakelen naar het rechter gespleten beeld) in te voeren, controleert u of **Caps Lock** is uitgeschakeld, drukt u **Ctrl+W** in, laat u los en drukt dan **H** in.

9.1.1 Incompatibiliteiten met Vim

Er zijn slechts een paar functies van de VI-modus van KatePart die incompatibel zijn met Vim (dingen die ontbreken niet meegenomen). Zij worden hieronder getoond samen met de respectievelijke redenen.

- KatePart: **U** en **Ctrl+R** is opnieuw doen.
Vim: **Ctrl+R** is normaal opnieuw doen, **U** wordt gebruikt om alle laatste wijzigingen op één regel ongedaan te maken.
De reden om **U** te laten reageren als opnieuw in de vi-modus van KatePart is dat de sneltoets **Ctrl+R** standaard in KatePart is ingenomen door de functie (zoeken en vervangen). De vi-modus zal standaard de sneltoetsen van KatePart niet overschrijven (dit kan ingesteld worden in **Instellingen** → **KatePart instellen...+Bewerken** → **Vi-invoermodus**), daarom moet een opnieuw-actie ook beschikbaar zijn als een 'reguliere' toets. Het gedrag van het **U** commando in Vim, dit terzijde, gaat niet goed samen met het interne opnieuw-systeem van KatePart, het is dus niet triviaal om het te ondersteunen.
- KatePart: **print** toont de dialoog **Afdrukken**.
Vim: **print** drukt de regels in de gegeven reeks af zoals zijn voorouder **ed**.
Commando's zoals **:print** zijn niet alleen beschikbaar in vi-modus maar ook voor gebruikers van het 'reguliere' KatePart - daarom opent het commando **:print** de printdialoog - volgens het principe van het minst verrassend, in plaats van het gedrag van Vim nadoen.
- KatePart: **Y** yanks tot het einde van de regel.
Vim: **Y** yanks de gehele regel, net als **yy**.
Het gedrag van vi op het commando **Y** is in de praktijk een bug. De beide commando's wijzigen en verwijderen, **cc** / **dd** doen hun actie op de huidige regel en **C/D** werken van de cursor tot het regeleinde. Echter, zowel **yy** en **Y** 'yanks' (kopiëren de huidige regel. In de vi-modus van KatePart kopieert **Y** tot het einde van de regel. Dit wordt als 'logischer' beschreven in de [documentatie](#) van Vim.
- KatePart: **O** en **o** opent [*aantal*] nieuwe regels en plaatst u in invoegmodus.
Vim: **O** en **o** opent een nieuwe regel en voegt een [*aantal*] keren tekst in bij het verlaten van de invoegmodus.
Dit is hoofdzakelijk gedaan als gevolg van het zien van velen die in verwarring zijn door dit gedrag van een vim IRC-kanaal (#vim op Libera Chat).

9.1.2 Modi omschakelen

- *Normale modus* stelt u in staat om commando's in te voeren om te navigeren of een document te bewerken en is de standaard modus. U kunt er naar terug keren vanuit elke andere modus door **Esc** in te drukken.
- *Visuele modus* stelt u in staat tekst in een document te accentueren. De meeste commando's in Normale modus zijn ook geldig in deze modus. U kunt ze invoeren door **v** in te drukken om tekens te selecteren of **V** om regels te selecteren.
- *Invoegmodus* stelt u in staat om het document direct te bewerken. U kunt in deze modus komen door **i** in te drukken of één van de verschillende andere onderstaande commando's.
- *Commandomodus* roept de commandoregel van KatePart op, waarmee u vele commando's kunt uitvoeren die beschikbaar zijn in Vi-implementaties evenals sommige specifiek voor KatePart. Voor meer informatie over deze commando's, zie Section 5.2. Om ze te gebruiken, druk op **:**, voer het commando in en druk op **Enter**.

9.1.3 Integratie met wat Kate kan

- Visuele modus wordt automatisch ingegaan wanneer er tekst wordt geselecteerd met de muis. Deze modus wordt ook ingegaan wanneer functies van Kate worden gebruikt die tekst selecteren, zoals 'Alles selecteren' (ofwel uit het menu of via **Ctrl+A**).
- Vi markeringen en [Bladwijzers van Kate](#) zijn geïntegreerd. Wanneer een markering aangemaakt is in Vi-modus, dan wordt er een overeenkomstige bladwijzer in Kate aangemaakt en verschijnt het in het menu **Bladwijzers**. Wanneer er een bladwijzer in Kate wordt aangemaakt, dan zal er een overeenkomstige Vi-markering op de 0-kolom worden aangemaakt.

9.1.4 Ondersteunde commando's in normale/visuele modus

a	Ga naar invoermodus; na de cursor invoegen
A	Ga naar invoermodus; na de regel invoegen
i	Ga naar invoermodus; voor de cursor invoegen
Ins	Ga naar invoermodus; voor de cursor invoegen
I	Ga naar invoermodus; voor de eerste niet-blanco teken op de regel invoegen
gi	Ga naar invoermodus; voor de plek invoegen waar de laatste invoermodus werd verlaten
v	Ga naar zichtmodus; selecteer tekens
V	Ga naar zichtmodus; selecteer regels
Ctrl+v	Ga naar zichtmodus; selecteer blokken
gb	Ga naar zichtmodus; selecteer opnieuw de laatste selectie
o	Open een nieuwe regel onder de huidige regel
O	Open een nieuwe regel boven de huidige regel
J	Regels samenvoegen
c	Wijziging: gevolgd door een beweging om te verwijderen en in invoegmodus te gaan
C	Wijzig tot het eind van de regel: verwijder tot het eind van de regel en ga naar invoermodus
cc	Wijzig de regel: verwijder de regel en ga naar invoermodus
s	Teken vervangen
S	Regel vervangen
dd	Regel verwijderen
d	Gevolgd door een beweging om te verwijderen
D	Tot het einde van de regel verwijderen
x	Verwijder teken rechts van de cursor
Del	Verwijder teken rechts van de cursor
X	Verwijder teken links van de cursor
gu	Gevolgd door een beweging om een kleine letter te maken
guu	Maak de huidige regel geheel kleine letters
gU	Gevolgd door een beweging om hoofdletter te maken
gUU	Maak de huidige regel geheel hoofdletters
y	Gevolgd door een beweging voor 'yank' (kopieëren)
yy	Yank (kopieer) regel
Y	Yank (kopieer) regel
p	Na cursor plakken
P	Voor cursor plakken
lp	Na de cursor ingesprongen plakken

[p	Voor de cursor ingesprongen plakken
r	Gevolg door een teken dat het teken na de cursor vervangt
R	In vervang-modus gaan
:	In commando-modus gaan
/	Zoeken
u	Ongedaan maken
Ctrl+R	Opnieuw
U	Opnieuw
m	Stel een markering in (kan later worden gebruikt door verplaatsingen)
n	Volgende zoeken
N	Vorige zoeken
>>	Regel in laten springen
<<	Inspringen in regel verwijderen
>	Regels in laten springen
<	Inspringen in regels verwijderen
Ctrl+F	Pagina omlaag
Ctrl+B	Pagina omhoog
ga	Toon de ASCII-waarde van het teken
.	Laatste wijziging herhalen
==	commandoRegelUitlijnen
=	commandoRegelsUitlijnen
~	Keer de hoofd-/kleine letter om
Ctrl+S	Splits het beeld horizontaal
Ctrl+V	Beeld verticaal splitsen
Ctrl+W, w	Ga rond naar het volgende gespleten venster
Ctrl+W, h CtrlW Left	Ga naar het linker gespleten venster
Ctrl+W, l CtrlW Right	Ga naar het rechter gespleten venster
Ctrl+W, k CtrlW Up	Ga naar het bovenste gespleten venster
Ctrl+W, j CtrlW Down	Ga naar het onderste gespleten venster

9.1.5 Ondersteunde verplaatsingen

Deze kunnen gebruikt worden om in een document rond te gaan in normale of visuele modus of in samenhang met een van de boven genoemde commando's. Ze kunnen vooraf gaan met een aantal, die aangeven hoe vaak de van toepassing zijnde verplaatsingen gemaakt moeten worden.

h	Links
Left	Links
Backspace	Links
j	Omlaag
Down	Omlaag
k	Up
Up	Up

Het handboek van KatePart

l	Rechts
Right	Rechts
Space	Rechts
\$	Einde van regel
End	Einde van regel
0	Eerste teken van de regel (kolom 0)
Home	Eerste teken van de regel
^	Eerste niet-witruimte-teken van de regel
f	Gevolgd door het teken om naar rechts van de cursor te gaan
F	Gevolgd door het teken om naar links van de cursor te gaan
t	Gevolgd door het teken om naar rechts van de cursor te gaan en de cursor op het teken ervoor te plaatsen
T	Gevolgd door het teken om naar links van de cursor te gaan en de cursor op het teken ervoor te plaatsen
gg	Eerste regel
G	Laatste regel
w	Volgende woord
W	Volgende woord gescheiden door witruimte
b	Vorige woord
B	Vorige woord gescheiden door witruimte
e	Einde van woord
E	Einde van woord gescheiden door witruimte
ge	Einde van vorige woord
gE	Einde van vorige woord gescheiden door witruimte
 	Gevolgd door een kolomnummer om naar die kolom te gaan
%	Gevolgd door een item om naar dat item te gaan
`	Markeren
`	Eerste niet-witruimte-teken van de regel waarop een markering zit
[[	Vorig recht openingshaakje
]]	Volgend recht openingshaakje
[]	Vorig recht sluihaakje
] [	Volgend recht sluihaakje
Ctrl+I	Naar de volgende locatie springen
Ctrl+O	Naar de vorige locatie springen
H	Ga naar de eerste regel op het scherm
M	Ga naar de middelste regel op het scherm
L	Ga naar de laatste regel op het scherm
%percentage	Ga naar het gespecificeerde percentage in het document
gk	Ga één zichtbare regel omhoog (bij gebruik van dynamische regelafbreking)
gj	Ga één zichtbare regel omlaag (bij gebruik van dynamische regelafbreking)
Ctrl+Left	Eén woord naar links

Ctrl+Right	Eén woord naar rechts
-------------------	-----------------------

9.1.6 Ondersteunde tekstobjecten

Deze kunnen gebruikt worden om een bepaald gedeelte van een document te selecteren.

iw	Inwendig woord: woord inclusief witruimte
aw	Een woord: woord exclusief witruimte
i~	Vorig aanhalingsteken (~) tot volgend aanhalingsteken, inclusief de aanhalingstekens
a~	Vorig aanhalingsteken (~) tot volgend aanhalingsteken, exclusief de aanhalingstekens
i'	Vorig accent (') tot volgend accent, inclusief de accenten
a'	Vorig accent (') tot volgend accent, exclusief de accenten
i(	Vorig rond openingshaakje [(] tot volgend rond sluihaakje [)], inclusief de haakjes
a(	Vorig rond openingshaakje [(] tot volgend rond sluihaakje [)], exclusief de haakjes
i[	Vorig recht openingshaakje [[] tot volgend recht sluihaakje []], inclusief de haakjes
a[	Vorig recht openingshaakje [[] tot volgend recht sluihaakje []], exclusief de haakjes
i{	Vorige startacolade { { } tot volgende sluitacolade { } }, inclusief de accolades
a{	Vorige startacolade { { } tot volgende sluitacolade { } }, exclusief de accolades
i<	Vorig kleiner-dan-teken (<) tot volgend groter-dan-teken (>), inclusief deze tekens
a<	Vorig kleiner-dan-teken (<) tot volgend groter-dan-teken (>), exclusief de haakjes
i`	Vorig accent-achterover (`) tot volgend accent-achterover, inclusief deze tekens
a`	Vorig accent-achterover (`) tot volgend accent-achterover, exclusief deze tekens

9.1.7 Ondersteunde commando's in invoegmodus

Ctrl+D	Inspringen verwijderen
Ctrl+T	Inspringen
Ctrl+E	Van onderaf invoegen
Ctrl+Y	Woord verwijderen

Ctrl+W	Woord verwijderen
Ctrl+U	Regel verwijderen
Ctrl+J	Nieuwe regel
Ctrl+H	Teken achterwaarts verwijderen
Ctrl+Home	Ga naar het eerste teken in het document
Ctrl+R n	De inhoud van register n invoegen
Ctrl+O , <i>commando</i>	Normale modus voor één commando ingaan
Ctrl+A	Nu geselecteerd getal met één verhogen
Ctrl+X	Nu geselecteerd getal met één verlagen

9.1.8 Het kommatekstobject

Dit object ontbreekt in Vim. Het kommatekstobject maakt het gemakkelijk om parameterlijsten in C-achtige talen en andere kommagescheidenlijsten. Het is in de basis het gebied tussen twee komma's of tussen een komma en een haakje. In de in de illustratie getoonde regel lichten de drie bereiken, die deze tekstobjecten kunnen overspannen, op.

```
int f(int arg1, double arg2, char arg3);
```

*Reeksen van kommatekstobjecten. Als de cursor boven bijv. `arg2` is en u drukt op **ci**, ('inwendige komma wijzigen') dan zou `double arg2` worden verwijderd en komt de cursor tussen de twee komma's in invoegmodus te staan. Een erg gemakkelijke manier om functieparameters te wijzigen.*

9.1.9 Ontbrekende functies

Zoals eerder aangegeven, is het doel van de VI-modus van KatePart niet om 100% van de functies van Vim te ondersteunen.

Bijlage A

Reguliere expressies

Deze Appendix bevat een beknopte maar hopelijk voldoende en dekkende introductie tot de wereld van *reguliere expressies*. Het documenteert reguliere expressies in de vorm beschikbaar in KatePart, wat niet compatibel is met de reguliere expressies van perl, noch met die van bijvoorbeeld **grep**.

A.1 Inleiding

Reguliere expressies bieden ons een manier om mogelijke inhoud van een tekenreeks te beschrijven zodat deze verstaan wordt door een klein stukje software, waarmee het kan nagaan of een tekst overeenkomt en ook in het geval van geavanceerde toepassingen met de middelen om stukjes overeenkomende tekst op te slaan.

Een voorbeeld: Zeg dat u naar een tekst in paragrafen wilt zoeken die begint met ofwel de namen 'Hendrik' of 'Piet' gevolgd door een werkwoordsvorm 'zegt'.

Met een normale zoekopdracht, zou u beginnen met te zoeken naar de eerste naam, 'Hendrik' misschien gevolgd door 'ze' zoals dit: **Hendrik ze** en, terwijl u zoekt naar overeenkomsten, moet u diegene negeren die niet het begin van een paragraaf zijn, evenals die waarin het woord beginnend met de letters 'ze' niet ofwel 'zegt', 'zei' of zo iets is. En dan natuurlijk dat alles herhalen met de volgende naam...

Met reguliere expressies kan die taak worden gedaan met een enkele zoekopdracht en met een grotere graad van precisie.

Om dit te bereiken definieert Reguliere Expressies regels voor het uitdrukken van details op een generaliserende manier van een overeenkomst met een tekenreeks. Ons voorbeeld, die we letterlijk als volgt uitdrukken: 'Een regel beginnend met ofwel 'Hendrik' of 'Piet' (mogelijk gevolgd door tot 4 spaties of een tab) gevolgd door een witruimte gevolgd door 'ze' en dan ofwel 'gt' of 'i' kan worden uitgedrukt met de volgende reguliere expressie:

```
^[ \t]{0,4}(Hendrik|Piet) ze(gt|i)
```

Het bovenstaande voorbeeld demonstreert alle vier belangrijkste concepten van moderne reguliere expressies namelijk:

- Patronen
- Toekenningen
- Aantallen
- Achterwaartse verwijzingen

Het dakje (^) waarmee de uitdrukking begint is een toekenning, die alleen waar is als de volgende overeenkomende tekenreeks zich aan het begin van de regel bevindt.

De tekenreeksen `[ \t]` en `(Hendrik|Piet) ze(gt|i)` zijn patronen. De eerste is een *tekenklasse* die ofwel overeenkomt met een spatie of een teken (horizontale) tab; het andere patroon bevat eerst een subpatroon overeenkomend met ofwel *Hendrik of Piet*, daarna iets wat exact overeenkomt met de tekenreeks *ze* en tenslotte een subpatroon overeenkomend met ofwel *gt of e*.

De tekenreeks `{0,4}` is een aantal dat aangeeft 'ergens van 0 t/m 4 van de vorige'.

Omdat software voor reguliere expressies, die het concept van *achterwaartse verwijzingen* ondersteunt, het gehele deel van de tekenreeks, dat overeenkomt, evenals subpatronen tussen haakjes, opslaat, geeft dus ook toegang tot die verwijzingen, zodat we onze handen kunnen leggen op ofwel de gehele overeenkomst (bij het zoeken in een tekstdocument in een tekstbewerker met een reguliere expressie, die vaak gemarkeerd is als geselecteerd) of ofwel de gevonden naam of het laatste deel van de zin.

Samenvattend zal de uitdrukking overeenkomen met wat we willen waarmee deze moet overeenkomen en alleen daarmee.

De volgende secties zullen in detail beschrijven hoe patronen, tekenklassen, toekenningen, aantallen en achterwaartse verwijzingen, te construeren en te gebruiken en de laatste sectie zal een paar bruikbare voorbeelden geven.

A.2 Patronen

Patronen bestaan uit letterlijke tekenreeksen en tekenklassen. Patronen mogen subpatronen bevatten, die patronen zijn tussen haakjes.

A.2.1 Escape-tekens

In patronen evenals in tekenklassen hebben sommige tekens een speciale betekenis. Om zulke tekens letterlijk overeen te laten komen met deze tekens, moeten ze zo gemarkeerd worden of *escaped* om de software voor reguliere expressies te laten weten dat het zulke tekens letterlijk moet interpreteren.

Dit wordt gedaan door het teken vooraf te laten gaan door een backslash (\).

De software voor reguliere expressies zal stiltejes een teken met een escape-teken ervoor negeren die geen speciale betekenis heeft in de context, dus, bijvoorbeeld, een 'j' met een escape ervoor (`\j`) is veilig. Als u er niet zeker van bent of een teken een speciale betekenis heeft, dan kunt u er veilig een escape-teken voor zetten.

Met het escape-teken werken omvat natuurlijk ook het teken backslash zelf, om letterlijk dat teken weer te geven, moet u schrijven `\\`.

A.2.2 Tekensklassen en afkortingen

Een *tekenklasse* is een uitdrukking die overeenkomt met een gedefinieerde set tekens. In Reguliere Expressies, worden tekenklassen gedefinieerd door de toegestane tekens voor de klasse tussen rechte haken te zetten, `[]`, of door een van de hieronder beschreven afgekorte klassen te gebruiken.

Eenvoudige tekenklassen bevatten letterlijk een of meer tekens, bijvoorbeeld `[abc]` (komt overeen met een van de letters 'a', 'b' of 'c') of `[0123456789]` (komt overeen met een cijfer).

Omdat letters en cijfers een logische volgorde hebben, kunt u deze afbreken door er reeksen van te specificeren: `[a-c]` is gelijk aan `[abc]` en `[0-9]` is gelijk aan `[0123456789]`. Combineren van deze constructies, bijvoorbeeld `[a-fy not 1-38]` is geheel toegestaan (de laatste zou natuurlijk overeenkomen met ofwel 'a', 'b', 'c', 'd', 'e', 'f', 'y', 'n', 'o', 't', '1', '2', '3' of '8').

Omdat hoofdletters andere tekens zijn dan hun kleine letter equivalenten, is het nodig om overeenkomsten zonder onderscheid van hoofd- en kleine letters te maken tussen 'a' of 'b', moet u het schrijven als **[aAbB]**.

Het is natuurlijk mogelijk om een 'negatieve' klasse overeenkomst te maken zoals 'alles behalve'. Er moet dan een dakje (^) aan het begin van de klasse staan:

[^abc] komt overeen met elk teken *behalve* 'a', 'b' of 'c'.

Naast letterlijke tekens, zijn er ook enige afkortingen gedefinieerd, om het leven een beetje gemakkelijker te maken:

\a

Dit komt overeen met het ASCII bel-teken (BEL, 0x07).

\f

Dit komt overeen met het ASCII formfeed-teken (FF, 0x0C).

\n

Dit komt overeen met het ASCII linefeed-teken (LF, 0x0A, Unix nieuwe-regel).

\r

Dit komt overeen met het ASCII carriage-return-teken (CR, 0x0D).

\t

Dit komt overeen met het ASCII horizontale tab-teken (HT, 0x09).

\v

Dit komt overeen met het ASCII verticale tab-teken (VT, 0x0B).

\xhhhh

Dit komt overeen met het Unicode-teken overeenkomend met het hexadecimale getal hhhh (tussen 0x0000 en 0xFFFF). \0ooo (bijv., \zero ooo) komt overeen met het ASCII/Latin-1 teken overeenkomend met het octale getal ooo (tussen 0 en 0377).

. (punt)

Dit komt overeen met elk teken (inclusief nieuwe-regel).

\d

Dit komt overeen met een cijfer. Gelijk aan [0-9]

\D

Dit komt overeen met een niet-cijfer. Gelijk aan [^0-9] of [^\d]

\s

Dit komt overeen met een witruimte-teken. Praktisch gelijk aan [\t\n\r]

\S

Dit komt overeen met een niet-witruimte-teken. Praktisch gelijk aan [^ \t\r\n] en gelijk aan [^\s]

\w

Komt overeen met elk 'woordteken' - in dit geval elke letter, elk cijfer of underscore. Gelijk aan [a-zA-Z0-9_]

\W

Dit komt overeen met een niet-woord-teken - alles behalve letters, cijfers of underscore. Gelijk aan [^a-zA-Z0-9_] of [^\w]

De *POSIX-notatie van klassen*, **[<class name>:]** wordt ook ondersteund. Bijvoorbeeld, **[:digit:]** is equivalent met **\d** en **[:space:]** met **\s**. Zie de volledige lijst met POSIX-tekenklassen [hier](#).

De afgekorte klassen kunnen in een zelf gemaakte klasse worden gezet, bijvoorbeeld om overeen te komen met een woord-teken, een spatie of een punt, u kunt dan schrijven **[\w \.]**

A.2.2.1 Tekens met een speciale betekenis in tekenklassen

De volgende tekens hebben een speciale betekenis binnen de constructie '[' van de tekenklasse en er moet een escape-teken voor om letterlijk ingevoegd te worden in een klasse:

]

Beëindigt de tekenklasse. Moet een escape-teken voor tenzij het het allereerste teken in de klasse is (mag na een dakje zonder escape-teken).

^ (dakje)

Geeft een negatieve klasse aan als het het eerste teken is. Moet een escape-teken voor om letterlijk overeen te komen als het het eerste teken in de klasse is.

– (streepje)

Geeft een logische reeks aan. Moet altijd een escape-teken voor binnen een tekenklasse.

\ (backslash)

Het escape-teken. Moet altijd een escape-teken voor.

A.2.3 Alternatieven: overeenkomen met 'één van'

Als u met een set van alternatieve patronen overeen wilt komen, dan kunt u die scheiden door | (teken verticale streep).

Om bijvoorbeeld ofwel 'Jan' of 'Harry' te zoeken, zou u een expressie **Jan|Harry** moeten gebruiken.

A.2.4 Subpatronen

Subpatronen zijn patronen omgeven door haakjes, ze kunnen op verschillende manieren in de wereld van de reguliere expressies worden gebruikt.

A.2.4.1 Alternatieven specificeren

U kunt een subpatroon gebruiken om een set alternatieven in een groter patroon te groeperen. De alternatieven zijn gescheiden door het teken '|' (verticale streep).

Om de woorden 'int', 'float' of 'double' te laten overeenkomen, kunt u het patroon **int | float | double** gebruiken. Als u het alleen wilt vinden wanneer het gevolgd wordt door enige witruimte en dan enige letters, stop de alternatieven dan in een subpatroon: **(int | float | double) \s+ \w+**.

A.2.4.2 Overeenkomende tekst vangen (achterwaartse verwijzingen)

Als u een achterwaartse referentie wilt gebruiken, gebruik dan een subpatroon **(PATTERN)** om het gewenste deel van het patroon te onthouden. Om te verhinderen het subpatroon te onthouden, gebruik een non-capturing groep **(?: PATTERN)**.

U wilt bijvoorbeeld twee exemplaren van hetzelfde woord gescheiden door een komma en mogelijk enige witruimte, dan kunt u schrijven **(\w+), \s* \1**. Het subpatroon **\w+** zou een groep tekens in een woord vinden en de gehele expressie zou overeenkomen als die gevolgd worden door een komma, 0 of meer witruimtes en dan een gelijke groep woordtekens. (De tekenreeks **\1** refereert naar *het eerste subpatroon ingesloten in haakjes*).

OPMERKING

Om meerduidigheid met het gebruik van `\1` met enige cijfers erachter (bijv. `\12` kan het 12de subpatroon zijn of gewoon het eerste subpatroon met 2) we gebruiken `\{12}` als syntax voor subpatronen met meerdere cijfers.

Voorbeelden:

- `\{12}1` is 'gebruik subpatroon 12'
- `\123` is 'gebruik gevangen 1 daarna 23 als de normale tekst'

A.2.4.3 Vooruitblik-toekenningen

Een vooruitblik-toekenning is een subpatroon die begint met ofwel `?=` of `?!`.

Om bijvoorbeeld een overeenkomst te vinden met de letterlijke tekenreeks 'Bill' maar alleen als deze niet wordt gevolgd door ' Gates', dan kunt u deze expressie gebruiken: **`Bill(?! Gates)`**. (Dit vindt 'Bill Clinton' evenals 'Billy the kid', maar negeert stillematen de andere overeenkomsten.)

Subpatronen voor toekenningen worden niet gevangen.

Zie ook [Toekenningen](#).

A.2.4.4 Achteruitblik-toekenningen

Een achteruitblik-toekenning is een subpatroon dat begint met ofwel `?<=` of `?<!`.

Achteruitblik heeft hetzelfde effect als de vooruitblik, maar werkt achterwaarts. Om de letterlijke tekenreeks 'fruit' overeen te laten komen maar alleen indien niet vooraf gegaan door 'grape', dan kunt u deze expressie gebruiken: **`(?<!grape) fruit`**.

Subpatronen voor toekenningen worden niet gevangen.

Zie ook [Toekenningen](#)

A.2.5 Tekens met een speciale betekenis in patronen

De volgende tekens hebben een speciale betekenis binnen een patroon en er moet een escape-teken voor om letterlijk met ze overeen te komen:

`\` (backslash)

Het escape-teken.

`^` (dakje)

Kent het begin van de tekenreeks toe.

`$`

Kent het einde van de tekenreeks toe.

`()` (linker en rechter haakje)

Geeft subpatronen aan.

`{ }` (linker en rechter accolade)

Geeft numerieke aantallen aan.

`[ ]` (linker en rechter rechte haakje)

Geeft tekenklassen aan.

| (verticale streep)

logische OR. Scheidt alternatieven.

+ (plusteken)

Aantal, 1 of meer.

*** (sterretje)**

Aantal, 0 of meer.

? (vraagteken)

Een optioneel teken. Kan geïnterpreteerd worden als een aantal, 0 of 1.

A.3 Aantallen

Aantallen biedt een reguliere expressie de mogelijkheid om een gespecificeerd aantal of reeks aantallen van ofwel een teken, tekenklasse of subpatroon.

Aantallen worden omgeven door accolades (`{` en `}`) en hebben de algemene vorm `{[minimum-aantal][, [maximum-aantal]]}`

Het gebruik wordt het beste uitgelegd door een voorbeeld:

`{1}`

Exact 1 keer

`{0,1}`

Nul of 1 keer

`{,1}`

Hetzelfde, met minder werk;)

`{5,10}`

Minstens 5 maar maximaal 10 keer.

`{5,}`

Minstens 5 keer, geen maximum.

Verder zijn er enkele afkortingen:

*** (sterretje)**

gelijk aan `{0,}`, elk aantal keren.

+ (plusteken)

gelijk aan `{1,}`, minstens 1 keer.

? (vraagteken)

gelijk aan `{0,1}`, nul of 1 keer.

A.3.1 Happig

Bij het gebruik van aantallen zonder maximum, is de standaard bij reguliere expressies om overeen te komen met zo veel als mogelijk van de gezochte tekenreeks, algemeen bekend staand als *happig* gedrag.

Moderne software voor reguliere expressies levert de middelen om 'happigheid uit te zetten', hoewel in een grafische omgeving het aan het interface is om toegang tot deze functie te leveren. Een zoekdialoog bijvoorbeeld, die een reguliere expressie biedt bij het zoeken, kan een keuzevakje hebben met het label 'Minimale overeenkomst' evenals het zou moeten aangeven of happigheid het standaard gedrag.

A.3.2 In voorbeelden van context

Hier zijn een paar voorbeelden van het gebruik van aantallen:

`^\d{4,5}\s`

Komt overeen met de cijfers in '1234 ga' en '12345 nu', maar niet in '567 elf' of in '223459 ergens'.

`\s+`

Komt overeen met één of meer witruimte tekens.

`(bla){1,}`

Komt overeen met alles van 'blablabla' en de 'bla' in 'blackbird' of 'tabla'.

`/?>`

Komt overeen met '>' in '<closeditem/>' evenals '>' in '<openitem>'.

A.4 Toekenningen

Toekenningen bieden de mogelijkheid om een reguliere expressie alleen te laten overeenkomen onder bepaalde gecontroleerde condities.

Een toekenning heeft geen teken nodig om mee overeen te komen, in plaats daarvan onderzoekt het de omgeving voor een mogelijke overeenkomst alvorens deze toe te kennen. Bijvoorbeeld de *woordgrens* toekenning probeert geen niet-woord-teken ten opzichte van een woordteken op zijn positie te vinden, in plaats daarvan moet er geen woordteken zijn. Dit betekent dat de toekenning overeen kan komen met iets waar geen teken is, bijv. aan het eind van een te zoeken tekenreeks.

Sommige toekenningen hebben in werkelijkheid een patroon waarmee overeen gekomen moet worden, maar het deel van de overeenkomst met de tekenreeks behoort niet bij een deel van het resultaat van overeenkomst met de volledige expressie.

Reguliere expressies zoals hier gedocumenteerd ondersteunt de volgende toekenningen:

`^` (dakje: begin van de tekenreeks)

Komt overeen met het begin van de tekenreeks.

De expressie **`^Peter`** komt overeen met 'Peter' in de tekenreeks 'Peter, hey!' maar niet in 'Hey, Peter!'

`$` (eind van de tekenreeks)

Komt overeen met het eind van de tekenreeks.

De expressie **`you\?$`** komt overeen met de laatste you in de tekenreeks 'You didn't do that, did you?' maar nergens in 'You didn't do that, right?'

`\b` (woordgrens)

Komt overeen als er een woordteken is aan één kant en niet aan de andere kant.

Dit is nuttig om wordeinden te vinden, bijvoorbeeld beide einden om een heel woord te vinden. De expressie **`\bin\b`** komt overeen met het aparte 'in' in de tekenreeks 'He came in through the window', maar niet met het 'in' in 'window'.

`\B` (niet-woordgrens)

Komt overeen waar '`\b`' dat niet doet.

Dit betekent dat het zal overeenkomen bijvoorbeeld in woorden: De expressie **`\Bin\B`** komt overeen in 'window' maar niet in 'integer' of 'I'm in love'.

(?=PATROON) (Positief vooruitkijken)

Een vooruitkijktokenning kijkt naar het deel van de tekenreeks die volgt op een mogelijke overeenkomst. Het positieve vooruitkijken voorkomt dat de tekenreeks overeenkomt als de tekst die volgt op de mogelijke overeenkomst niet overeenkomt met het *PATROON* van de tokenning, maar voorkomt dat de tekst daarmee overeenkomt en zal niet meegenomen worden in het resultaat.

De expressie **handy (=?\w)** komt overeen met 'handy' in 'handyman' maar niet in 'That came in handy!'

(?!PATROON) (Negatieve vooruitblik)

Het negatieve vooruitkijken voorkomt dat de tekenreeks overeenkomt als de tekst die volgt op de mogelijke overeenkomst overeenkomt met het *PATROON* van de tokenning.

De expressie **const \w+\b(?!s*&)** zal overeenkomen met 'const char' in de tekenreeks 'const char* foo' terwijl het niet kan overeenkomen met 'const QString' in 'const QString& bar' omdat de '&' overeenkomt met het patroon in de negatieve vooruitkijktokenning.

(?<=PATROON) (Positief achteruit kijken)

Een achteruitkijktokenning heeft hetzelfde effect als de vooruitkijk, maar werkt achterwaarts. Een achteruitkijktokenning kijkt naar het deel van de vorige tekenreeks met een mogelijke overeenkomst. Het positieve achteruitkijken zal alleen overeenkomen met een tekenreeks als het voorafgegaan wordt door het *PATROON* van de tokenning, maar voorkomt dat de tekst daarmee overeenkomt en zal niet meegenomen worden in het resultaat.

De expressie **(?<=cup) cake** zal overeenkomen met 'cake' als het volgt op 'cup' (in 'cup-cake' maar niet in 'cheesecake' of in alleen 'cake').

(?<!PATROON) (Negatieve achteruitblik)

Het negatieve achteruitkijken voorkomt dat de tekenreeks overeenkomt als het voorgaande deel van de zoektekst overeenkomt met het *PATROON*.

De expressie **(?<![\w\.\])[0-9]+** zal overeenkomen met '123' in de tekenreeksen '=123' en '-123' terwijl het niet kan overeenkomen met '123' in '.123' of 'word123'.

(PATROON) (Vanggroep)

Het subpatroon in de haakjes wordt gevangen en onthouden, zodat het gebruikt kan worden in achterwaartse referentie. Bijvoorbeeld, de expressie **("+)[^"]*\1** komt overeen met `""tekst""` en `"tekst"`.

Zie de sectie [Overeenkomende tekst vangen \(achterwaartse referenties\)](#) voor meer informatie.

(?:PATROON) (Non-capturing groep)

Het subpatroon in de haakjes wordt niet gevangen en niet onthouden. Het heeft de voorkeur om altijd non-capturing groepen te gebruiken als de vangst niet zal worden gebruikt.

Bijlage B

Index

C

commentaar toevoegen, [34](#)
commentaar verwijderen, [34](#)

V

vervangen, sed-stijl
zoeken, sed-stijl, [38](#)