

Handbok KatePart

Thad McGinnis
Anne-Marie Mahfouf
Anders Lund
T.C. Hollingsworth
Christoph Cullmann
Lauri Watts
Översättare: Stefan Asserhäll



Handbok KatePart

Innehåll

1	Inledning	8
2	En del grundläggande information	9
2.1	Drag och släpp	9
2.2	Genvägar	9
3	Arbeta med KatePart-editorn	12
3.1	Översikt	12
3.2	Flytta sig i texten	13
3.3	Arbeta med markeringar	13
3.3.1	Använda blockmarkering	14
3.3.2	Använda skriv över-markering	14
3.3.3	Att använda långlivade markeringar	14
3.4	Kopiera och klistra in text	14
3.5	Söka och ersätta text	15
3.5.1	Sök- och ersättningsrader	15
3.5.2	Söka efter text	16
3.5.3	Ersätta text	16
3.6	Att använda bokmärken	17
3.7	Automatisk radbrytning av text	17
3.8	Att använda automatisk indentering	18
3.9	Indikering av radändringar	18
3.10	Rullningslistens miniavbildning	19
3.11	Flera markörer	19
3.11.1	Skapa flera markörer	20
3.11.2	Arbeta med flera markörer	20
4	Menyalternativen	21
4.1	Menyn Arkiv	21
4.2	Menyn Redigera	22
4.3	Markeringsmenyn	24
4.4	Menyn Visa	26
4.5	Menyn Gå	27
4.6	Menyn Verktyg	28
4.7	Menyerna Inställningar och Hjälp	31

5	Avancerade redigeringsverktyg	32
5.1	Kommentera/avkommentera	32
5.2	Editorkomponentens kommandorad	32
5.2.1	Vanliga kommandon på kommandoraden	33
5.2.1.1	Kommandon för att anpassa editorn	33
5.2.1.2	Kommandon för redigering	35
5.2.1.3	Kommandon för navigering	39
5.2.1.4	Kommandon för grundläggande redigeringsfunktioner (dessa be- ror på programmet som editorkomponenten används i)	40
5.3	Använda kodvikning	41
6	Utöka KatePart	42
6.1	Inledning	42
6.2	Arbeta med syntaxfärgläggning	42
6.2.1	Översikt	42
6.2.2	KateParts syntaxfärgläggningssystem	43
6.2.2.1	Hur det fungerar	43
6.2.2.2	Regler	44
6.2.2.3	Sammanhangsstilar och nyckelord	44
6.2.2.4	Standardstilar	44
6.2.3	XML formatet för syntaxfärgläggningsdefinitioner	45
6.2.3.1	Översikt	45
6.2.3.2	Sektionerna i detalj	47
6.2.3.3	Tillgängliga standardstilar	49
6.2.4	Regler för syntaxdetektering	50
6.2.4.1	Reglerna i detalj	53
6.2.4.2	Tips och trick	56
6.3	Arbeta med färgteman	57
6.3.1	Översikt	57
6.3.2	Färgteman i KSyntaxHighlighting	58
6.3.3	JSON-format för färgteman	59
6.3.3.1	Översikt	59
6.3.3.2	JSON-strukturen	60
6.3.3.3	Huvudsektionerna i JSON-färgtemafilern	60
6.3.3.4	Metadata	62
6.3.4	Färger i detalj	62
6.3.4.1	Editorns färger	63
6.3.4.2	Förvalda textstilar	69
6.3.4.3	Egna färgläggningstextstilar	72
6.3.5	Det grafiska användargränssnittet för färgteman	73

6.3.5.1	Skapa ett nytt tema	74
6.3.5.2	Importera eller exportera JSON-temafile	74
6.3.5.3	Redigera färgteman	74
6.3.5.3.1	Färger	74
6.3.5.3.2	Förvalda textstilar	74
6.3.5.3.3	Färgläggningstextstilar	74
6.3.6	Tips och trick	75
6.3.6.1	Textfärgernas kontrast	75
6.3.6.2	Förslag för konsekvent syntaxfärgläggning	75
6.4	Skapa skript med JavaScript	75
6.4.1	Indenteringsskript	75
6.4.1.1	Indenteringsskriptets huvud	76
6.4.1.2	Indenteringens källkod	76
6.4.2	Kommandoradskript	78
6.4.2.1	Kommandoradskriptets huvud	78
6.4.2.2	Skriptets källkod	79
6.4.2.2.1	Tilldela genvägar	79
6.4.3	Programmeringsgränssnittet för skripthantering	80
6.4.3.1	Markörer och intervall	80
6.4.3.1.1	Prototypen för Cursor	80
6.4.3.1.2	Prototyp för Range	81
6.4.3.2	Globala funktioner	82
6.4.3.2.1	Att läsa och inkludera filer	83
6.4.3.2.2	Felsökning	83
6.4.3.2.3	Översättning	83
6.4.3.3	Programmeringsgränssnittet för vyn	84
6.4.3.4	Programmeringsgränssnittet för dokumentet	85
6.4.3.5	Programmeringsgränssnittet för editorn	91
7	Att anpassa KatePart	92
7.1	Inställning av editorkomponenten	92
7.1.1	Utseende	92
7.1.1.1	Allmänt	92
7.1.1.2	Kanter	94
7.1.2	Färgteman	95
7.1.3	Redigering	95
7.1.3.1	Allmänt	95
7.1.3.2	Textnavigering	96
7.1.3.3	Indentering	97
7.1.3.4	Automatisk komplettering	98

Handbok KatePart

7.1.3.5	Stavningskontroll	98
7.1.3.6	VI-inmatningsläge	99
7.1.4	Öppna/spara	99
7.1.4.1	Allmänt	99
7.1.4.2	Avancerat	100
7.1.4.3	Lägen och filtyper	101
7.2	Anpassa med dokumentvariabler	102
7.2.1	Hur KatePart använder variabler	103
7.2.2	Tillgängliga variabler	103
7.2.3	Utökade alternativ i .kateconfig-filer	106
8	Tack till och licens	107
9	Vi-inmatningsläget	108
9.1	Vi-inmatningsläge	108
9.1.1	Funktioner inkompatibla med Vim	108
9.1.2	Byta lägen	109
9.1.3	Integrering med funktioner i Kate	109
9.1.4	Kommandon som stöds i normalt och visningsläge	110
9.1.5	Förflyttningar som stöds	111
9.1.6	Textobjekt som stöds	112
9.1.7	Kommandon i infogningsläge som stöds	113
9.1.8	Komma-textobjektet	114
9.1.9	Saknade funktioner	114
A	Reguljära uttryck	115
A.1	Inledning	115
A.2	Mönster	116
A.2.1	Undanta tecken	116
A.2.2	Teckenklasser och förkortningar	116
A.2.2.1	Tecken med särskild betydelse inne i teckenklasser	118
A.2.3	Alternativ: matchar 'en av'	118
A.2.4	Delmönster	118
A.2.4.1	Ange alternativ	118
A.2.4.2	Spara text som matchar (bakåtreferenser)	118
A.2.4.3	Påståenden för sökning framåt	119
A.2.4.4	Påståenden för sökning bakåt	119
A.2.5	Tecken med särskild betydelse inne i mönster	119
A.3	Kvalificering	120
A.3.1	Gluskt beteende	120
A.3.2	Exempel i sammanhang	121
A.4	Påståenden	121
B	Sakregister	123

Sammanfattning

KatePart är en editorkomponent med fullständig funktionalitet av KDE.

Kapitel 1

Inledning

KatePart är en texteditorkomponent med fullständig funktionalitet som används av många QtTM- och KDE-program. KatePart är mer än en texteditor. Den är avsedd som en editor för programmerare, och kan delvis anses som ett alternativ till mer kraftfulla editorer. En av KateParts huvudfunktioner är syntaxfärgläggningen, som är anpassad för många olika programspråk som C/C++, JavaTM, Python, Perl, Bash, Modula 2, HTML och Ada.

KWrite är ett enkelt texteditorprogram baserat på KatePart. Det har ett enkelt dokumentgränssnitt (SDI) som gör det möjligt att redigera en fil åt gången per fönster. Eftersom KWrite är en mycket enkel implementation av KatePart behöver det inte någon egen dokumentation. Om du vet hur KWrite används, kan du använda KatePart var som helst.

Kapitel 2

En del grundläggande information

KWrite och många andra tillämpningar av KatePart är mycket lätta att använda. Alla som har använt en texteditor bör inte ha några problem.

2.1 Drag och släpp

KatePart använder KDE:s drag och släpp protokoll. Filer kan dras och släppas på KatePart från skrivbordet, filhanteraren Dolphin, eller någon FTP-plats som öppnats i ett av Dolphins fönster.

2.2 Genvägar

Många av snabbtangenterna kan anpassas via menyn [Inställningar](#). Normalt hanterar KatePart följande snabbtangenter:

Insert	Växla mellan lägena infoga och ersätt. I läget infoga lägger editorn till alla tecken som skrivs in till texten medan all data knuffas framåt till höger om textmarkören. Läget ersätt gör att varje tecken som skrivs in ersätter tecknet omedelbart till höger om textmarkören.
Vänsterpil	Flytta markören ett tecken åt vänster.
Högerpil	Flytta markören ett tecken åt höger.
Uppåtpil	Flytta markören en rad uppåt.
Neråtpil	Flytta markören en rad neråt.
Ctrl+E	Gå till föregående redigeringsplats i dokumentet.
Ctrl+Skift+E	Gå till nästa redigeringsplats i dokumentet.
Alt+Skift+Uppåtpil	Flytta markören till föregående motsvarande indentering.
Alt+Skift+neråtpil	Flytta markören till föregående motsvarande indentering.
Ctrl+6	Gå till motsvarande parentes.
Page Up	Flytta markören en sida uppåt.

Page Down	Flytta markören en sida neråt.
Home	Flytta markören till början av raden.
End	Flytta markören till slutet av raden.
Ctrl+Home	Gå till dokumentets början.
Ctrl+End	Gå till dokumentets slut.
Ctrl+Uppåtpil	Rulla uppåt en rad.
Ctrl+Neråtpil	Rulla nedåt en rad.
Ctrl>Högerpil	Gå ett ord åt höger.
Ctrl>Vänsterpil	Gå ett ord åt vänster.
Ctrl+Skift+Uppåtpil	Flytta upp rader.
Ctrl+Skift+Neråtpil	Flytta ner rader.
Ctrl+.	Duplicera markerade rader neråt.
Ctrl+B	Lägg till ett bokmärke.
Alt+Page Up	Föregående bokmärke.
Alt+Page Down	Nästa bokmärke.
Delete	Ta bort tecknet till höger om markören (eller all markerad text).
Backsteg	Ta bort tecknet till vänster om markören.
Ctrl+Delete	Ta bort höger ord.
Ctrl+Backsteg	Ta bort vänster ord.
Ctrl+K	Ta bort rad.
Skift+retur	Infoga nyrad inklusive inledande tecken på nuvarande rad som inte är bokstäver eller siffror. Det är t.ex. användbart för att skriva kommentarer i koden: I slutet av raden '// någon text' tryck på snabbtangenter så börjar den nästa raden redan med '// '. Du behöver alltså inte skriva in kommentartecknen i början av varje ny rad med kommentarer.
Ctrl+Skift+Enter	Skapa en ny rad under den aktuella raden.
Ctrl+Alt+Enter	Skapa en ny rad ovanför den aktuella raden.
Skift>Vänsterpil	Markera text ett tecken åt vänster.
Skift>Högerpil	Markera text ett tecken åt höger.
Ctrl+F1	Sök.
F3	Sök igen.
Skift+F3	Sök föregående.
Ctrl+H	Sök markering.
Ctrl+Skift+H	Sök markering bakåt.
Ctrl+Skift>Högerpil	Markera ord åt höger.
Ctrl+Skift>Vänsterpil	Markera ord åt vänster.
Skift+Home	Markera till radens början.
Skift+End	Markera till radens slut.
Skift+Uppåtpil	Markera till föregående rad.
Skift+Neråtpil	Markerad till nästa rad.
Ctrl+Skift+6	Markera till matchande parentes.
Ctrl+Skift+Page Up	Markera till överst i fönstret.
Ctrl+Skift+Page Down	Markera till nederst i fönstret.
Skift+Page Up	Markera uppåt en sida.
Skift+Page Down	Markera nedåt en sida.
Ctrl+Skift+Home	Markera till dokumentets början.
Ctrl+Skift+End	Markera till dokumentets slut.
Ctrl+Home	Markera alla.

Ctrl+Skift+A	Avmarkera.
Ctrl+Skift+B	Blockmarkeringsläge.
Ctrl+C / Ctrl+Insert	Kopiera den markerade texten till klippbordet.
Ctrl+D	Kommentera.
Ctrl+Skift+D	Avkommentera.
Ctrl+G	Gå till rad...
Ctrl+I	Indentera markering.
Ctrl+Skift+I	Avindentera markering.
Ctrl+J	Kombinera rader.
Ctrl+P	Skriv ut.
Ctrl+R	Ersätt.
Ctrl+S	Utför kommandot Spara .
Ctrl+Skift+S	Spara som.
Ctrl+U	Stora bokstäver.
Ctrl+Skift+U	Små bokstäver.
Ctrl+Alt+U	Inledande stor bokstav.
Ctrl+V / Skift+Insert	Klistrar in klippbordets text på raden som redigeras.
Ctrl+X / Skift+Insert	Klipp ut den markerade texten, och kopiera den till klippbordet.
Ctrl+Z	Ångra.
Ctrl+Skift+Z	Gör om.
Ctrl+-	Minska teckensnitt.
Ctrl++Ctrl+=	Öka teckensnitt.
Ctrl+Skift+-	Fäll ihop toppnivånoder.
Ctrl+Skift++	Expandera toppnivånoder.
Ctrl+Mellanslag	Utför kodkomplettering.
F5	Uppdatera.
F6	Visa/Dölj ikonkant.
F7	Byt till kommandorad.
F9	Visa/Dölj vikmarkörer.
F10	Dynamisk radbrytning.
F11	Visa/Dölj radnummer.
Ctrl+T	Byt två intilliggande tecken.
Ctrl+Skift+O	Automatisk stavningskontroll.
Ctrl+Skift+V	Byt till nästa inmatningsläge.
Ctrl+8	Återanvänd ord ovanför.
Ctrl+9	Återanvänd ord nedanför.
Ctrl+Alt+#	Expandera förkortning.
Ctrl+Alt+Uppåtpil	Lägg till en markör ovanför aktuell markör.
Ctrl+Alt+Neråtpil	Lägg till en markör under aktuell markör.
Skift+Alt+I	Skapa en markör i slutet av varje rad av markeringen.
Alt+J	Slå upp nästa förekomst av ordet under markören och markera det.
Ctrl+Alt+Skift+J	Slå upp alla förekomster av ordet under markören och markera dem.

Kapitel 3

Arbeta med KatePart-editorn

Anders Lund
Dominik Haumann
Översättare: Johan Thelmén

3.1 Översikt

KatePart-editorn är redigeringsområdet i KateParts fönster. Editorn delas mellan Kate och KWrite, och kan också användas i Konqueror för att visa textfiler från den lokala datorn, eller från nätverket.

Editorn består av följande komponenter:

Redigeringsområdet

Det här är platsen där texten i ditt dokument finns.

Rullningslisterna

Rullningslisterna anger positionen av den synliga delen av dokumentets text, och kan användas för att flytta omkring i dokumentet. Att dra i rullningslisterna gör inte att infogningsmarkören flyttar sig.

Rullningslisterna visas eller göms efter behov.

Ikonkanten

Ikonkanten är en smal ruta till vänster om editorn, som visar en liten ikon intill markerade rader.

Du kan lägga till eller ta bort ett [bokmärke](#) för en synlig rad genom att klicka med vänster musknapp i ikonkanten intill raden.

Visning av ikonkanten kan ändras med menyalternativet **Visa** → **Visa ikonkant**.

Radnummerrutan

Rutan med radnummer visar radnummer för alla synliga rader i dokumentet.

Visning av radnummer kan ändras med menyalternativet **Visa** → **Visa radnummer**.

Kodvikningsrutan

Kodvikningsrutan låter dig fälla ihop eller expandera block av rader som kan fällas ihop. Beräkningen av områden som kan fällas ihop görs enligt reglerna i definitionen av syntaxfärgläggning för dokumentet.

OCKSÅ I DET HÄR KAPITLET:

- [Flytta sig i texten](#)
- [Arbeta med markeringar](#)
- [Kopiera och klistra in text](#)
- [Söka och ersätta text](#)
- [Använda bokmärken](#)
- [Automatiskt radbryta text](#)
- [Använda automatisk indentering](#)

3.2 Flytta sig i texten

Att flytta omkring i texten i KatePart sker på liknande sätt som i de flesta grafiska texteditorer. Du flyttar markören med piltangenterna och tangenterna **Page Up**, **Page Down**, **Home** och **End** tillsammans med väljartangenterna **Ctrl** och **Skift**. Tangenten **Skift** används alltid för att skapa en markering, medan tangenten **Ctrl** har olika effekt för olika tangenter:

- För tangenterna **Upp** och **Ner** betyder den rulla istället för att flytta markören.
- För tangenterna **Vänster** och **Höger** betyder den hoppa över ord istället för tecken.
- För tangenterna **Page Up** och **Page Down** betyder den att gå till vyns synliga kant istället för att bläddra.
- För tangenterna **Home** och **End** betyder den att gå till dokumentets början eller slut istället för radens början eller slut.

KatePart tillhandahåller också ett sätt att snabbt gå till motsvarande klammer eller parentes: Placera markören innanför ett parentes- eller klammertecken och tryck på **Ctrl-6** för att gå till motsvarande parentes eller klammer.

Dessutom kan du använda [bokmärken](#) för att snabbt gå till platser som du själv definierar.

3.3 Arbeta med markeringar

Det finns två grundläggande sätt att markera text i KatePart: med musen, och med tangentbordet.

För att markera med musen, håll nere vänsterknappen medan du drar markören från stället där markeringen ska börja, till den önskade slutpunkten. Texten markeras medan du drar.

Dubbelklicka på ett ord för att markera ordet.

Trippelklicka på en rad markerar hela raden.

Om **Skift** hålls nere vid ett klick, så markeras texten.

- Om ingenting redan är markerat, från textmarkörens position till musens position.
- Om det finns en markering, från och inklusive den markeringen till musens position.

NOT

När text markeras genom att dra med musen, kopieras den markerade texten till klippbordet, och kan klistras in genom att klicka med musens mittenknapp i editorn, eller i vilket annat program som helst där du vill klistra in texten.

För att markera med tangentbordet, håll nere **Skift**-tangenten medan du använder förflyttningstangenterna (piltangenterna, **Page Up**, **Page Down**, **Home** och **End**, möjligen tillsammans med **Ctrl** för att utöka förflyttningen av textmarkören).

Se också avsnittet [Flytta sig i texten](#) i det här kapitlet.

För att Kopiera den nuvarande markeringen, använd menyalternativet **Redigera** → **Kopiera** eller snabbtangenten (normalt **Ctrl+C**).

För att avmarkera den nuvarande markeringen använd menyalternativet **Redigera** → **Avmarkera**, eller snabbtangenten (normalt **Ctrl+Skift+A**), eller klicka med vänster musknapp i editorn.

3.3.1 Använda blockmarkering

När blockmarkering är aktiverat, kan du göra 'vertikala markeringar' i texten, vilket betyder att markera vissa kolumner i flera rader. Det här är praktiskt för att till exempel arbeta med tabulatoråtskilda rader.

Blockmarkering kan väljas genom att använda menyalternativet **Redigera** → **Blockmarkeringsläge**. Den normala snabbtangenten är **Ctrl+Skift+B**.

3.3.2 Använda skriv över-markering

Om skriv över-markering är aktiverad, så ersätts markeringen genom att skriva in eller klistra in text. Om det inte är aktiverat, så läggs ny text till på markörens plats.

Skriv över-markering är normalt aktiverad.

För att ändra inställning av alternativet, använd sidan [Markör och markering](#) i [inställningsdialogrutan](#).

3.3.3 Att använda långlivade markeringar

När långlivade markeringar är aktiverade, så blir inte markerad text avmarkerad när tecken skrivs in eller markören flyttas. Det här betyder att du kan flytta markören från den markerade texten, och skriva in ny text.

Långlivade markeringar är normalt inte aktiverade.

Långlivade markeringar kan aktiveras på sidan [Markör och markering](#) i [inställningsdialogrutan](#).

VARNING

Om både långlivade markeringar och skriv över-markering är aktiverade, så ersätts markeringen och avmarkeras om ny text skrivs in eller klistras in, då textmarkören är inne i markeringen.

3.4 Kopiera och klistra in text

För att kopiera text, markera den och använd menyalternativet **Redigera** → **Kopiera**. Dessutom, vid markering av text med musen kopieras den markerade texten till X-markeringen.

För att klistra in text som för närvarande finns på klippbordet, använd menyalternativet **Redigera** → **Klistra in**.

Dessutom, så kan text som markeras med musen klistras in genom att klicka med musens mittknapp på önskad plats.

TIPS

Om du använder KDE-skrivbordet, kan du hämta text som tidigare kopierats från vilket program som helst med Klipper-ikonen i systembrickan.

3.5 Söka och ersätta text

3.5.1 Sök- och ersättningsrader

KatePart har en inkrementell sökrad och en avancerad sök- och ersättningsrad, som erbjuder möjlighet att skriva in en ersättningssträng tillsammans med några extra alternativ.

Raderna erbjuder följande gemensamma alternativ:

Sök

Här skriver du in söksträngen. Tolkningen av strängen beror på några av alternativen som beskrivs nedan.

Matcha skiftläge

Om aktiverad, begränsas sökningen till värden som motsvarar skiftläget (stora eller små bokstäver) för varje tecken i sökbegreppet.

Den avancerade sök- och ersättningsraden erbjuder några ytterligare alternativ:

Vanlig text

Matcha bokstavligen en förekomst av söksträngen.

Hela ord

Om markerad, så fås bara en träff om det finns en ordgräns på båda sidor om söksträngen, i betydelsen inte en bokstav eller siffra: antingen ett annat synligt tecken, eller ett radslut.

Escape-sekvenser

Om markerad, aktiveras menyalternativet **Lägg till** längst ner i den sammanhangsberoende menyn i textrutorna, och låter dig lägga till escape-sekvenser till sökbegreppet från en fördefinierad lista.

Reguljärt uttryck

Om markerad, tolkas söksträngen som ett reguljärt uttryck. Menyalternativet **Lägg till** längst ner i den sammanhangsberoende menyn i textrutorna aktiveras och låter dig lägga till reguljära uttryck till sökbegreppet från en fördefinierad lista.

Se [Reguljära uttryck](#) för mer om detta.

Sök bara i markeringen

Om markerad, utförs sökning och ersättning bara inom den markerade texten.

Sök alla

Genom att klicka på den här knappen markeras alla träffar i dokumentet, och antal träffar visas i en liten meddelanderuta.

3.5.2 Söka efter text

Starta den inkrementella sökraden med **Ctrl+F** eller med menyalternativet **Redigera → Sök...**, för att söka efter text.

Det här öppnar den inkrementella sökraden längst ner i editorfönstret. Till vänster på raden finns en knapp med en ikon för att stänga raden, följt av en liten textruta för att skriva in sökbegrepp."

När du börjar skriva in tecknen i ditt sökbegrepp, börjar sökningen omedelbart. Om det finns en träff i texten markeras den och bakgrundsfärgen i inmatningsfältet ändras till ljusgrönt. Om sökbegreppet inte motsvarar någon sträng i texten, indikeras det av en ljusröd bakgrundsfärg i inmatningsfältet.

Använd knappen  eller  för att gå till nästa eller föregående träff i dokumentet.

Träffar markeras även när sökraden har stängts. För att ta bort markeringarna, tryck på tangenten **Esc**.

Du kan välja om sökningen ska vara skiftlägeskänslig. Genom att välja  begränsas träffar till värden som motsvarar skiftläget (stora eller små bokstäver) för varje tecken i sökbegreppet.

Klicka på knappen  till höger på den inkrementella sökraden för att byta till den avancerade sök- och ersättningsraden.

För att upprepa den senaste sökningen, om någon, utan att visa den inkrementella sökraden, använd **Redigera → Sök igen (F3)** eller **Redigera → Sök föregående (Skift+F3)**.

3.5.3 Ersätta text

Starta den avancerade sök- och ersättningsraden med kommandot **Redigera → Ersätt**, eller genvägen **Ctrl+R**, för att ersätta text.

På radens övre vänstra sida finns en knapp med en ikon för att stänga raden, följt av en liten kombinationsruta för att skriva in sökbegrepp. Rutan kommer ihåg nyligen använda mönster.

Du kan styra sökläget genom att välja **Enkel text**, **Hela ord**, **Escape-sekvenser** eller **Reguljärt uttryck** i kombinationsrutan.

Om **Escape-sekvenser** eller **Reguljärt uttryck** är markerat, aktiveras menyalternativet **Lägg till...** längst ner i den sammanhangsberoende menyn i textrutorna och låter dig lägga till reguljära uttryckstecken i sökbegreppet eller ersätta mönster från fördefinierade listor.

Använd knappen  eller  för att gå till nästa eller föregående träff i dokumentet.

Skriv in texten att ersätta med i textrutorna med beteckningen **Ersätt** och klicka på knappen **Ersätt** för att bara ersätta markerad text eller **Ersätt alla** för att ersätta söktexten i hela dokumentet.

Du kan ändra sök- och ersättningsbeteendet genom att välja de olika alternativen längst ner på

raden. Genom att välja  begränsas träffar till poster som matchar skiftläget (stora eller små bokstäver) för varje tecken i sökbegreppet.  söker och ersätter bara inom nuvarande markering. Knappen **Sök alla** markerar alla träffar i dokumentet och visar antal hittade träffar i ett litet meddelandefönster.

Klicka på knappen  till höger på den avancerade sök- och ersättningsraden för att byta till den inkrementella sökraden.

TIPS

Om du använder ett reguljärt uttryck för att hitta texten som ska ersättas, kan du använda bakåttrefenser för att återanvända text som sparats i parentetiserade delmönster i uttrycket. Se [Reguljära uttryck](#) för mer information om dem.

TIPS

Du kan utföra **find** (sök), **replace** (ersätt) och **ifind** (inkrementell sökning) från [kommandoraden](#).

3.6 Att använda bokmärken

Bokmärkesfunktionen låter dig markera vissa rader, för att lätt kunna hitta dem igen.

Du kan lägga till och ta bort ett bokmärke för en rad på två olika sätt:

- Flytta infogningsmarkören till raden, och välj kommandot **Bokmärken** → **Nytt bokmärke (Ctrl+B)**.
- Klicka på ikonkanten intill raden.

Bokmärken är tillgängliga i menyn **Bokmärken**. Varje individuellt bokmärke är tillgängligt som ett menyalternativ, namngivet med radnumret för raden med bokmärket, och de första tecknen i texten på raden. För att flytta infogningsmarkören till början av en rad med ett bokmärke, öppna menyn och välj bokmärket.

För att snabbt gå mellan bokmärken eller till nästa/föregående bokmärke, använd kommandona **Bokmärken** → **Nästa (Alt+Page Down)** eller **Bokmärken** → **Föregående (Alt+Page Up)**.

3.7 Automatisk radbrytning av text

Den här funktionen låter dig formatera texten på ett mycket enkelt sätt: texten radbryts, så att inga rader överskrider ett maximalt antal tecken per rad, om det inte finns en längre sträng av andra tecken än blanktecken.

För att aktivera/inaktivera det, markera/avmarkera kryssrutan **Statisk radbrytning** på [redigeringssidan](#) i [inställningsdialogrutan](#).

För att ställa in maximal radlängd (maximalt antal tecken per rad), använd alternativet **Bryt rader vid** på [redigeringssidan](#) i [inställningsdialogrutan](#).

Om aktiverad, så har det följande effekter:

- Medan du skriver, lägger editorn automatiskt till en hård nyrad efter det sista blanktecknet på en position innan den maximala radlängden nås.
- När dokumentet laddas, radbryter editorn texten på ett liknande sätt, så att inga rader är längre än den maximala radlängden, om de innehåller några blanktecken som tillåter detta.

NOT

Det finns för närvarande inget sätt att ställa in radbrytning för typer av dokument, eller aktivera och inaktivera funktionen på dokumentnivå. Det här kommer att rättas i en framtida version av KatePart.

3.8 Att använda automatisk indentering

KateParts editorkomponent stöder en mängd olika automatiska indenteringslägen, skapade för olika textformat. Du kan välja bland tillgängliga lägen med menyn **Verktyg** → **Indentering**. Modulerna för automatisk indentering tillhandahåller också funktionen **Verktyg** → **Formatera indentering** som beräknar indenteringen för markeringen eller nuvarande rad igen. Alltså kan du indentera om hela dokumentet genom att markera all text och utföra åtgärden.

Alla indenteringslägen använder inställningar som hör ihop med indentering i det aktiva dokumentet.

TIPS

Du kan ställa in alla möjliga inställningsvariabler, inklusive de som hör ihop med indentering, genom att använda [dokumentvariabler](#) och [filtyper](#).

TILLGÄNGLIGA LÄGEN FÖR AUTOMATISK INDENTERING

Inget

Väljes det här läget stängs automatisk indentering av helt och hållet.

Normal

Den här indenteringen behåller helt enkelt en indentering som liknar den föregående raden med något innehåll som inte är blanktecken. Du kan kombinera den med alternativen indentera och avindentera för att indentera efter din egen smak.

C-stil

Indentering för C och liknande språk, såsom C++, C#, Java™, JavaScript och så vidare. Indenteringen fungerar inte med skriptspråk som Perl eller PHP.

Haskell

Indentering för det funktionella programspråket Haskell.

Lilypond

Indentering för notskriftsspråket Lilypond för musik.

Lisp

Indentering särskilt för skriptspråket Lisp och dialekter av Lisp.

Python

Indentering särskilt för skriptspråket Python.

XML-stil

Indentering särskilt för XML-liknande språk.

3.9 Indikering av radändringar

KatePart indikering av radändringar låter dig enkelt se vad du nyligen har ändrat i en fil. Sparade ändringar anges av en grön rad till vänster om ett dokument, medan osparade ändringar anges av en orange rad.

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

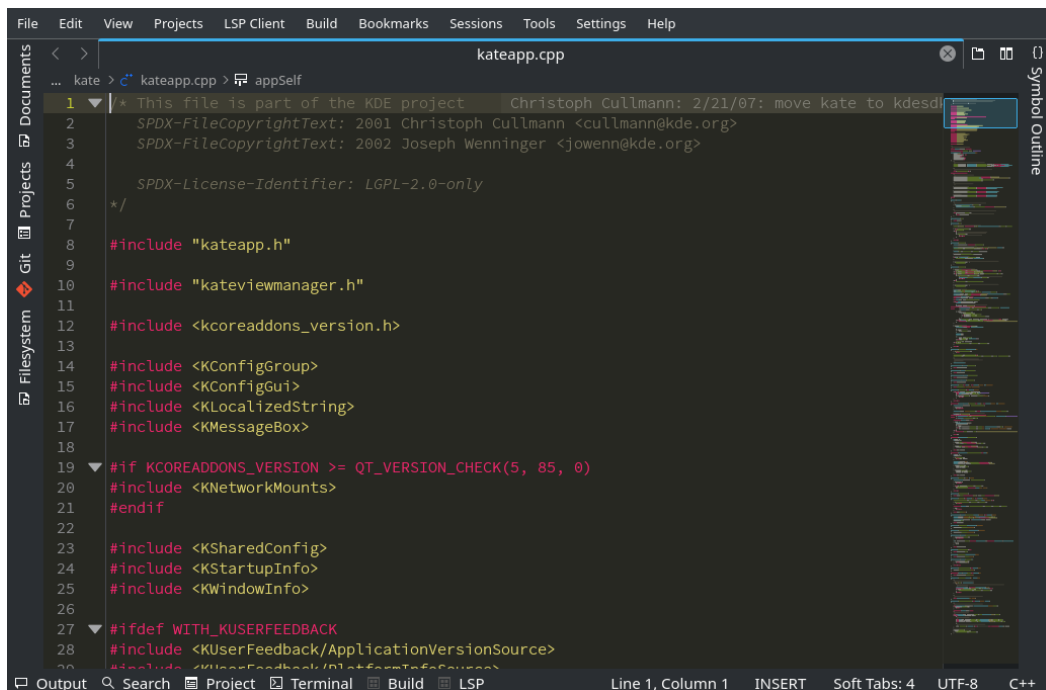
TextBlock *newBlock = new TextBlock (this, 0);
newBlock->appendLine (TextLine (new TextLineData()));
```

Indikering av radändringar under användning.

Färgerna som används kan ändras i [inställningsrutan Teckensnitt och färger](#), eller inaktivera funktionen helt under fliken [Kanter](#) i [inställningsrutan Utseende](#).

3.10 Rullningslistens miniavbildning

Rullningslistens miniavbildning i KatePart visar en förhandsgranskning av dokument på rullningslistens plats. Den del av dokumentet som för närvarande är synlig markeras.



Rullningslistens miniavbildning visar en förhandsgranskning av Kates källkod.

Miniavbildningen kan tillfälligt aktiveras eller inaktiveras genom att använda **Visa → Visa rullningslistens miniavbildning** eller permanent i [avdelningen Utseende](#) i [KateParts inställningar](#).

3.11 Flera markörer

Stöd för flera markörer introducerades med version 5.93 av Kate-delprogrammet.

3.11.1 Skapa flera markörer

- För att skapa dem med musen, använd **Alt** + vänster musknapp. Väljaren går att ställa in, se [Anpassa väljare för flera markörer](#).
- För att skapa dem med tangentbordet, tryck på **Ctrl+Alt+Upp** för att skapa en markör ovanför den primära markören och **Ctrl+Alt+Ner** för att skapa en markör nedanför den. Genvägarna är också inställningsbara.
- För att skapa markörer från en markering, markera först text och tryck sedan på **Skift+Alt+I**. Det skapar en markör sist på varje rad i markeringen.
- Använd **Alt+J** för att söka efter nästa förekomst av ordet under markören och markera det samt skapa en markör. Om du vill hoppa över det aktuella ordet under markören, tryck på **Alt+K** så markeras det aktuella ordet som överhoppat. När du trycker på **Alt+J** igen avmarkeras det aktuella ordet och sökningen efter nästa ord utförs.
- Använd **Ctrl+Alt+Skift+J** för att söka efter alla förekomster av ordet under markören, och markera dem med en markör i slutet av varje markering. Du kan använda **Alt+J** för att gå igenom de markerade orden och använda **Alt+K** för att avmarkera vilket ord som helst, som nämnt i föregående stycke.

3.11.2 Arbeta med flera markörer

När du väl har skapat några markörer, kan du utföra de flesta redigeringsåtgärder med dem som man skulle med en enda markör. Genom att exempelvis skriva en bokstav, skrivs den vid varje markör. På liknande sätt kan du utföra textomvandlingar, t.ex. göra om alla positionerna eller markeringarna till stora bokstäver.

Ibland vill man ta bort markörer. Du kan använda **Alt** + vänster musknapp på markören som du vill ta bort. Om du bara vill ta bort markörer på tomma rader, finns det en färdig åtgärd för det som gör det åt dig. Öppna kommandoraden genom att använda **Ctrl+Alt+I** och leta reda på **Ta bort markörer från tomma rader** för att utföra åtgärden,

Kapitel 4

Menyalternativen

4.1 Menyn Arkiv

Arkiv → Ny (Ctrl+N)

Det här startar ett nytt dokument i ett nytt och oberoende editorfönster.

Arkiv → Nytt fönster

Skapa ett annat fönster som innehåller det aktuella dokumentet. Alla ändringar av dokumentet i ett fönster reflekteras i det andra fönstret och tvärtom.

Arkiv → Öppna... (Ctrl+O)

Visar den vanliga dialogrutan **Öppna fil** i KDE. Använd filvyn för att välja filen du vill öppna, och klicka på **Öppna** för att öppna den.

Arkiv → Öppna senaste

Det här är en genväg för att öppna nyligen sparade dokument. Genom att klicka på det här alternativet öppnas en lista vid sidan om menyn med flera av de senast sparade filerna. Genom att klicka på en specifik fil öppnas den i KatePart, om filen fortfarande finns på samma plats.

Arkiv → Spara (Ctrl+S)

Det här sparar det nuvarande dokumentet. Om dokumentet redan har sparats skriver det här över den tidigare sparade filen utan att fråga efter användarens samtycke. Om det här är första gången ett nytt dokument sparas, visas dialogrutan Spara som (beskrivs nedan).

Arkiv → Spara som... (Ctrl+Skift+S)

Det här låter ett dokument sparas med ett nytt filnamn. Det här görs med fildialogrutan som beskrivs ovan under avsnittet **Öppna** i den här hjälpfilen.

Arkiv → Spara som med kodning

Spara ett dokumentet med ett nytt filnamn i en annan kodning.

Arkiv → Spara kopia som

Spara en kopia av dokumentet med ett nytt filnamn och fortsatt redigera originaldokumentet.

Arkiv → Återställ (F5)

Laddar den aktiva filen igen. Kommandot är användbart om ett annat program eller en annan process har ändrat filen medan den varit öppen i KatePart.

Arkiv → Skriv ut... (Ctrl+P)

Öppnar en enkel utskriftsdialog, som låter användaren ange vad, vart och hur utskrift ska ske.

Arkiv → Exportera som HTML

Spara dokumentet som för närvarande är öppet som en HTML-fil, som kommer att formateras med användning av de nuvarande inställningarna av syntaxfärgläggning och färgschema.

Arkiv → Stäng (Ctrl+W)

Stäng aktuell fil med det här kommandot. Om du har gjort osparade ändringar, kommer du att få en fråga om du vill spara ändringarna innan KatePart stänger filen.

Arkiv → Avsluta (Ctrl+Q)

Det här stänger editorfönstret. Om du har mer än en instans av KatePart startad, genom menyalternativen **Ny** eller **Nytt fönster**, så stängs inte de instanserna.

4.2 Menyn Redigera

Redigera → Ångra (Ctrl+Z)

Ångra det senaste redigeringskommandot (inmatning, kopiering, utklippning etc.)

NOT

Detta kan ångra flera redigeringskommandon av samma sort, som att skriva in tecken.

Redigera → Gör om (Ctrl+Skift+Z)

Det här återställer den senaste ändringen (om det finns någon) som gjorts med Ångra.

Redigera → Klipp ut (Ctrl+X)

Det här kommandot tar bort den nuvarande markerade texten och placerar den på klippbordet. Klippbordet fungerar osynligt och erbjuder ett sätt att överföra data mellan program.

Redigera → Kopiera (Ctrl+C)

Det här kopierar den nuvarande markerade texten till klippbordet så att den kan klistras in någon annanstans. Klippbordet fungerar osynligt och erbjuder ett sätt att överföra data mellan program.

Redigera → Kopiera som HTML

Kopiera markeringen som HTML, formaterad med användning av de nuvarande inställningarna av syntaxfärgläggning och färgschema.

Redigera → Klistra in (Ctrl+V)

Det här infogar innehållet på klippbordet vid markörens position. Klippbordet fungerar osynligt och erbjuder ett sätt att överföra data mellan program.

NOT

Om skriv över markering är aktiverad, så skriver den inklistrade texten över markerad text, om det finns någon.

Redigera → Klistra in markering (Ctrl+Skift+Insert)

Klistrar in [musmarkeringens](#) innehåll som tidigare valdes. Markera text med muspekaren för att klistra in den i filen som för närvarande är öppen med menyalternativet.

Redigera → Byt mot klippbordets innehåll

Byter den markerade texten med innehållet på [klippbordet](#).

Redigera → Klistra in klippbordshistorik

Öppnar en dialogruta för att välja och klistra in en post från klippbordshistoriken.

Redigera → Inmatningslägen

Byt mellan normalt och Vi-liknande redigeringsläge. Läget stöder de oftast använda kommandona och förflyttningarna från vim:s normala och visuella läge, och har en valfri Vi-statusrad. Statusraden visar kommandon medan de matas in, utmatning från kommandon och nuvarande läge. Lägets beteende kan anpassas under fliken [Vi-inmatningsläge](#) på sidan **Redigering** i KateParts inställningsdialogruta.

Redigera → Ersättningsläge (Ins)

Växlar mellan infoga och skriv över. När läget är **INFOGA**, infogar du tecken där markören befinner sig. När läget är **ERSÄTT** ersätter inskrivna tecken de nuvarande tecknen om markören är placerad före några tecken. Statusraden visar aktuell status av skriv över, antingen **INFOGA** eller **ERSÄTT**.

Redigera → Skrivskyddat läge

Ställer in aktuellt dokument till skrivskyddat läge. Det förhindrar tillägg av all text och alla ändringar av dokumentets formatering.

Redigera → Sök... (Ctrl+F)

Det här öppnar den inkrementella sökraden längst ner i editorfönstret. Till vänster på raden finns en knapp med en ikon för att stänga den, följt av en liten textruta för att skriva in sökbegrepp.

När du börjar skriva in tecknen i ditt sökbegrepp, börjar sökningen omedelbart. Om det finns en träff i texten markeras den och bakgrundsfärgen i inmatningsfältet ändras till ljusgrönt. Om sökbegreppet inte motsvarar någon sträng i texten, indikeras det av en ljusröd bakgrundsfärg i inmatningsfältet.

Använd knapparna  eller  för att gå till nästa eller föregående träff i dokumentet.

Träffar markeras även när sökraden har stängts. För att ta bort markeringarna, tryck på tangenten **Esc**.

Du kan välja om sökningen ska vara skiftlägeskänslig. Genom att välja  begränsas träffar till värden som motsvarar skiftläget (stora eller små bokstäver) för varje tecken i sökbegreppet.

Klicka på knappen  med den gröna pilikonen till höger på den inkrementella sökraden för att byta till den avancerade sök- och ersättningsraden.

Redigera → Sök varianter → Sök nästa (F3)

Det här upprepar den senaste sökningen, om det finns någon, utan att anropa den inkrementella sökraden, och söker framåt igenom dokumentet med början på markörens position.

Redigera → Sök varianter → Sök föregående (Skift+F3)

Det här upprepar den senaste sökningen, om det finns någon, utan att anropa den inkrementella sökraden, och söker bakåt istället för framåt igenom dokumentet.

Redigera → Sök varianter → Sök markering (Ctrl+H)

Slår upp nästa förekomst av markerad text.

Redigera → Sök varianter → Sök markering bakåt (Ctrl+Skift+H)

Slår upp föregående förekomst av markerad text.

Redigera → Ersätt... (Ctrl+R)

Det här kommandot öppnar den avancerade sök- och ersättningsraden. Längst upp till vänster på raden finns en knapp med en ikon för att stänga raden, följt av en liten textruta för att skriva in sökbegreppet.

Sökläget kan styras genom att välja **Enkel text**, **Hela ord**, **Escape-sekvenser** eller **Reguljärt uttryck** i kombinationsrutan.

Om **Escape-sekvenser** eller **Reguljärt uttryck** väljes, aktiveras knappen **Lägg till...** längst ner i textrutornas sammanhangsberoende meny, som låter dig lägga till escape-sekvenser eller reguljära uttryck i sök- eller ersättningsbegrepp från fördefinierade listor.

Använd knapparna  eller  för att gå till nästa eller föregående träff i dokumentet.

Skriv in texten att ersätta med i textrutan med beteckningen **Ersätt** och klicka på knappen **Ersätt** för att bara ersätta markerad text eller **Ersätt alla** för att ersätta söktexten i hela dokumentet.

Du kan ändra sök- och ersättningsbeteendet genom att välja olika alternativ längst ner på raden. Genom att välja  begränsas träffar till poster som matchar skiftläget (stora eller små bokstäver) för varje tecken i sökbegreppet.  söker och ersätter bara inom nuvarande markering. Knappen **Sök alla** markerar alla träffar i dokumentet och visar antal hittade träffar i ett litet meddelandefönster.

Klicka på knappen  med den gröna pilikonen till höger på den avancerade sök- och ersättningsraden för att byta till den inkrementella sökraden.

4.3 Markeringsmenyn

Markering → Markera alla (Ctrl+A)

Det här väljer hela dokumentet. Det här kan vara mycket användbart för att kopiera hela filen till ett annat program.

Markering → Avmarkera (Ctrl+Skift+A)

Avmarkerar den markerade texten i editorn, om det finns någon.

Markering → Blockmarkeringsläge (Ctrl+Skift+B)

Växlar markeringsläge. När markeringsläget är **BLOCK**, innehåller statusraden strängen **[BLOCK]** och du kan göra vertikala markeringar, t.ex. markera kolumn 5 till 10 på rad 9 till 15.

Markering → Kommentera (Ctrl+D)

Det här lägger till ett blanktecken i början på raden där textmarkören befinner sig eller till början av alla markerade rader.

Markering → Avkommentera (Ctrl+Skift+D)

Det här tar bort ett blanktecken (om det finns något) från början på raden där textmarkören befinner sig eller från början av alla markerade rader.

Markering → Kombinera rader (Ctrl+J)

Kombinerar markerade rader, eller den nuvarande raden och raden under den med ett blanktecken som avskiljare. Inledande eller efterföljande blanktecken tas bort för kombinerade rader i ändarna som påverkas.

Markering → Inledande stor bokstav (Ctrl+Alt+U)

Ändra den inledande bokstaven för orden i den markerade texten eller i aktuellt ord till stor bokstav.

Markering → Stora bokstäver (Ctrl+U)

Ändra den markerade texten eller bokstaven efter markören till stora bokstäver.

Markering → Små bokstäver (Ctrl+Skift+U)

Ändra den markerade texten eller bokstaven efter markören till små bokstäver.

Markering → Rensa indentering

Det här rensar indenteringen för nuvarande markering eller för raden där markören för närvarande befinner sig. Att rensa indenteringen försäkrar dig om att den markerade texten följer indenteringsläget du valt.

Markering → Formatera indentering

Gör att nuvarande rad eller markerade rader justeras om med indenteringsläget och indenteringsinställningarna i dokumentet.

Markering → Justera till...

Kommandot justerar rader i det markerade blocket eller hela dokumentet på kolumnen angiven av ett reguljärt uttryck som du blir tillfrågad om.

Om du anger ett tomt mönster, justeras den på radens första tecken som inte är ett blanktecken.

Om mönstret har en platsmarkör indenterar det där markören matchar.

Exempel:

Med `'-'` infogas mellanslag innan första - på varje rad för att justera alla på samma kolumn.

Med `:\s+(.)` infogas det mellanslag före det första tecknet som inte är ett blanktecken efter ett kolon för att justera alla till samma kolumn.

Markering → Bryt rader mellan ord i dokument

Tillämpa statisk radbrytning i hela dokumentet. Det betyder att en ny textrad automatiskt börjar när den aktuella raden överskrider längden som anges av alternativet **Bryt rader vid** under fliken Redigering i menyn **Inställningar → Anpassa editor....**

Markering → Lägg till insättningstecken över markör (Ctrl+Alt+Upp)

Lägger till ytterligare ett insättningstecken ovanför den aktuella markören. Insättningstecknet placeras på raden direkt ovanför och i samma kolumn som den aktuella markören.

Markering → Lägg till insättningstecken under markör (Ctrl+Alt+Ner)

Lägger till ytterligare ett insättningstecken nedanför den aktuella markören. Insättningstecknet placeras på raden direkt ovanför och i samma kolumn som den aktuella markören.

Markering → Lägg till markörer vid radslut (Alt+Skift+I)

Lägger till en markör på varje rad som för närvarande är markerad.

Markering → Sök och markera nästa förekomst (Alt+J)

Söker efter nästa förekomst av ordet för närvarande under markören, markerar det och lägger till en markör.

Markering → Sök och markera alla förekomster (Ctrl+Alt+Skift+J)

Söker efter alla förekomster av ordet för närvarande under markören, markerar dem och lägger till en markör för vart och ett av dem.

4.4 Menyn Visa

Visa → Öka teckenstorlek (Ctrl++)

Det här ökar teckenstorleken som används vid visning.

Visa → Minska teckenstorlek (Ctrl+-)

Det här minskar teckenstorleken som används vid visning.

Visa → Radbryt → Dynamisk radbrytning (F10)

Ändrar dynamisk radbrytning i den nuvarande vyn. Dynamisk ordbrytning gör att all text i en vy blir synlig utan att behöva rulla horisontellt genom att återge en verklig rad på fler synliga rader när det behövs.

Visa → Radbryt → Dynamiska radbrytningsmarkörer

Välj när och hur dynamiska radbrytningsmarkörer ska visas. Det är bara tillgängligt om alternativet **Dynamisk radbrytning** är markerat.

Visa → Radbryt → Visa statisk radbrytningsmarkör

Om det här alternativet är markerat, ritas en vertikal linje i radbrytningskolumnen som definieras i **Inställningar** → **Anpassa editor...** under fliken Redigering. Observera att radbrytningsmarkören bara ritas om du använder ett teckensnitt med fast breddsteg.

Visa → Kanter → Visa ikonkant (F6)

Det här är ett alternativ som växlar. Om det aktiveras, visas ikonkanten vid vänsterkanten i den aktiva editorn, och vice versa. Ikonkanten anger position för markerade rader i editorn.

Visa → Kanter → Visa radnummer (F11)

Det här är ett alternativ som växlar. Om det aktiveras, visas en ruta med radnummer längs vänsterkanten på den aktiva editorn, och vice versa.

Visa → Kanter → Visa rullningslistens miniavbildning

Om alternativet är markerat visar den aktuella vyn markeringar på den vertikala rullningslistan. Markeringarna är ekvivalenta med markeringarna på [ikonkanten](#).

Visa → Kanter → Visa rullningslistens miniavbildning

Ersätter rullningslistan med en visualisering av det aktuella dokumentet. För mer information om rullningslistens miniavbildning, se Avsnitt [3.10](#).

Visa → Kodvikning

Dessa alternativ rör [kodvikning](#):

Visa → Kanter → Kodvikning → Visa vikmarkörer

Ändrar visning av vikmarkörer till vänster om vyn.

Visa → Kodvikning → Fäll ihop aktuell nod

Fäll ihop området som innehåller markören.

Visa → Kodvikning → Expandera aktuell nod

Expandera området som innehåller markören.

Visa → Kodvikning → Fäll ihop toppnivånoder (Ctrl+Skift+-)

Fäll ihop alla toppnivåområden i dokumentet. Klicka på triangeln som pekar åt höger för att expandera alla toppnivåområden.

Visa → Kodvikning → Expandera toppnivånoder (Ctrl+Skift++)

Expandera alla toppnivåområden i dokumentet.

Visa ej utskrivningsbara mellanslag

Visa eller dölj omgivande ruta omkring ej utskrivningsbara mellanslag.

4.5 Menyn Gå

Gå → Gå till rad... (Ctrl+G)

Det här öppnar gå till raden längst ner i fönstret, som används för att få markören att gå till en viss rad (som anges med dess nummer) i dokumentet. Radnumret kan skrivas in direkt i textrutan eller anges grafiskt genom att klicka på uppåt- eller nedåtpilarna vid sidan av textrutan. Den lilla uppåtpilen ökar radnumret och nedåtpilen minskar det. Stäng raden med ett klick på knappen med ikonen till vänster på raden.

Gå → Gå till föregående redigeringsrad (Ctrl+E)

Åtgärden går till föregående redigeringsrad med [inställningen för flera markörer](#).

Gå → Gå till nästa redigeringsrad (Ctrl+Skift+E)

Åtgärden går till nästa redigeringsrad med [inställningen för flera markörer](#).

Gå → Gå till föregående ändrade rad

Rader som ändrades efter filen öppnades kallas ändrade rader. Det här alternativet går till föregående ändrade rad.

Gå → Gå till nästa ändrade rad

Rader som ändrades efter filen öppnades kallas ändrade rader. Det här alternativet går till nästa ändrade rad.

Gå → Gå till matchande parentes (Ctrl+6)

Flytta markören till samhörande inledande eller avslutande parentes.

Gå → Markera till matchande parentes (Ctrl+Skift+6)

Markerar texten mellan samhörande inledande och avslutande parenteser.

Gå → Bokmärken (Ctrl+Skift+6)

Under alternativen som beskrivs här, kommer det att finnas ett alternativ för varje bokmärke i det aktiva dokumentet. Texten är några få ord i början på den markerade raden. Välj ett alternativ för att flytta markören till början på den raden. Editorn panorerar om det behövs, för att göra raden synlig.

Gå → Bokmärken → Nytt bokmärke (Ctrl+B)

Lägger till eller tar bort ett bokmärke på den nuvarande raden i det aktiva dokumentet (om det redan finns, tas det bort, annars läggs det till).

Gå → Bokmärken → Rensa alla bokmärken

Det här kommandot tar bort alla bokmärken från dokumentet samt listan på bokmärken som läggs till längst ner i det här menyalternativet.

Gå → Bokmärken → Föregående (Alt+Page Up)

Det här flyttar markören till början av den första raden ovan med ett bokmärke. Menyalternativets text innehåller radnummer och den inledande texten på raden. Alternativet är bara tillgängligt när det finns ett bokmärke på en rad ovanför markören.

Gå → Bokmärken → Nästa (Alt+Page Down)

Det här flyttar markören till början av nästa rad med ett bokmärke. Menyalternativets text innehåller radnummer och den inledande texten på raden. Alternativet är bara tillgängligt när det finns ett bokmärke på en rad nedanför markören.

4.6 Menyn Verktg

Verktg → Lge

Vlj filtypschema som du f6redrar f6r det aktiva dokumentet. Det skriver 6ver det globala filtypslaget som stllts in i **Instllningar** → **Anpassa editor...** under fliken Filtyper, men bara f6r det aktuella dokumentet.

Verktg → Frglggning

Vlj frglggningsschema som du f6redrar f6r det aktiva dokumentet. Det skriver 6ver det globala frglggningsslaget som stllts in med **Instllningar** → **Anpassa editor...**, men bara f6r det aktuella dokumentet.

Verktg → Indentering

Vlj indenteringsstil som du vill ha f6r det aktiva dokumentet. Det skriver 6ver det globala indenteringslaget som stllts in i **Instllningar** → **Anpassa editor...**, men bara f6r det aktuella dokumentet.

Verktg → Kodning

Du kan skriva 6ver den f6rvalda kodningen instlld i **Instllningar** → **Anpassa editor...** p sidan **6ppna/spara** f6r att ange en annan kodning f6r det aktuella dokumentet. Kodningen du stller in h6r 6r bara giltig f6r det aktuella dokumentet.

Verktg → Lgg till byte-ordningsmarkering (BOM)

Genom att markera 6tg6rden kan du explicit l6gga till en byte-ordningsmarkering f6r dokument kodade med Unicode. Byte-ordningsmarkeringen (BOM) 6r ett Unicode-tecken som anv6nds f6r att ange byte-ordningen i en textfil eller str6m. F6r mer information se [Byte Order Mark](#).

Verktg → Radslut

Vlj radslutslaget som du f6redrar f6r det aktiva dokumentet. Det skriver 6ver det globala radslutslaget som stllts in med **Instllningar** → **Anpassa editor...**, men bara f6r det aktuella dokumentet.

Verktg → Skript

Den h6r undermenyn inneh6ller en lista med alla skriptstyrda 6tg6rder. Listan kan enkelt 6ndras genom att [skriva egna skript](#). P6 s6 s6tt kan KatePart ut6kas med anv6ndardefinierade verktg.

Verktg → Skript → Navigering

Verktg → Skript → Navigering → Flytta mark6ren till f6reg6ende motsvarande indentering (Alt+Skift+Upp)

Flyttar mark6ren till den f6rsta raden ovanf6r den nuvarande som 6r indenterad till samma niv6 som den nuvarande raden.

Verktg → Skript → Navigering → Flytta mark6ren till n6sta motsvarande indentering (Alt+Skift+Ner)

Flyttar mark6ren till den f6rsta raden nedanf6r den nuvarande som 6r indenterad till samma niv6 som den nuvarande raden.

Verktg → Skript → Redigering

Verktg → Skript → Redigering → Sortera markerad text

Sorterar den markerade texten eller hela dokumentet i stigande ordning.

Verktg → Skript → Redigering → Flytta ner rader (Ctrl+Skift+Ner)

Flytta markerade rader ner6t.

Verktg → Skript → Redigering → Flytta upp rader (Ctrl+Skift+Upp)

Flytta markerade rader upp6t.

Verktyg → Skript → Redigering → Duplicera markerade rader neråt (Ctrl+Skift+Ner)
Duplicerar markerade rader neråt.

Verktyg → Skript → Redigering → Duplicera markerade rader uppåt (Ctrl+Skift+Upp)
Duplicerar markerade rader uppåt.

Verktyg → Skript → Redigering → Koda markerad text till webbadress
Kodar den markerade texten så att den kan användas som en del av en frågesträng i en webbadress, och ersätter markeringen med den kodade texten.

Verktyg → Skript → Redigering → Avkoda markerad text från webbadress
Om en del av en frågesträng i en webbadress är markerad, avkodar detta den och ersätter markeringen med den ursprungliga obehandlade texten.

Verktyg → Skript → Emmet

Verktyg → Skript → Emmet → Utöka förkortning
Konverterar den markerade texten till ett par inledande och avslutande HTML- eller XML-taggar. Om exempelvis **div** är markerad, ersätter alternativet det med `<div></div>`.

Verktyg → Skript → Emmet → Radbryt enligt tagg
Omger den markerade texten med taggen som anges på [kommandoraden](#).

Verktyg → Skript → Emmet → Flytta markören till motsvarande tagg
Om markören är inne i en inledande HTML- eller XML-tagg, flyttar alternativet den till den avslutande taggen. Om markören är inne i den avslutande taggen, flyttas den istället till den inledande taggen.

Verktyg → Skript → Emmet → Markera HTML/XML-tagginnehåll inåt
När markören är inne i ett par av HTML- eller XML-taggar, ändrar alternativet markeringen till innehållet i dessa HTML- eller XML-taggar, utan att markera själva taggarna.

Verktyg → Skript → Emmet → Markera HTML/XML-tagginnehåll utåt
När markören är inne i ett par av HTML- eller XML-taggar, ändrar alternativet markeringen till innehållet i dessa HTML- eller XML-taggar, inklusive själva taggarna.

Verktyg → Skript → Emmet → Växla kommentar
Om den markerade delen inte är en kommentar, omger alternativet delen med HTML- eller XML-kommentarer (t.ex. `<!-- markerad text -->`). Om den markerade delen är en kommentar, tas kommentartaggarna istället bort.

Verktyg → Skript → Emmet → Ta bort tagg vid markören
Om markören för närvarande är inne i HTML- eller XML-tagg, tar alternativet bort hela taggen.

Verktyg → Skript → Emmet → Minska tal med 1
Alternativet subtraherar ett från texten som för närvarande är markerad om den är ett tal. Om exempelvis **5** är markerat, blir det 4.

Verktyg → Skript → Emmet → Minska tal med 10
Alternativet subtraherar 10 från texten som för närvarande är markerad om den är ett tal. Om exempelvis **15** är markerat, blir det 5.

Verktyg → Skript → Emmet → Minska tal med 0,1
Alternativet subtraherar 0,1 från texten som för närvarande är markerad om den är ett tal. Om exempelvis **4, 5** är markerat, blir det 4, 4.

Verktyg → Skript → Emmet → Öka tal med 1
Alternativet adderar ett till texten som för närvarande är markerad om den är ett tal. Om exempelvis **5** är markerat, blir det 6.

Verktyg → Skript → Emmet → Öka tal med 10
Alternativet adderar 10 till texten som för närvarande är markerad om den är ett tal. Om exempelvis **5** är markerat, blir det 15.

Verktyg → Skript → Emmet → Öka tal med 0,1
Alternativet adderar 0,1 till texten som för närvarande är markerad om den är ett tal. Om exempelvis **4, 5** är markerat, blir det 4, 6.

Verktyg → Byt till kommandorad (F7)

Visar KatePart-kommandorad längst ner i fönstret. Skriv **help** på kommandoraden för att få hjälp och **help list** för att få en lista med kommandon. Se [Editorkomponentens kommandorad](#) för mer information om kommandoraden.

Verktyg → Utför kodkomplettering (Ctrl+Mellanslag)

Aktivera kodkomplettering för hand, oftast genom att använda snabbtangenter som är kopplad till åtgärden.

Verktyg → Ordkomplettering

Återanvänd ord nedanför (Ctrl+9) och **Återanvänd ord ovanför (Ctrl+8)** kompletterar texten som för närvarande är inskriven genom att söka bakåt eller framåt efter liknande ord från den nuvarande markörpositionen. **Skalkomplettering** visar en kompletteringsruta med poster som matchar.

Verktyg → Stavning → Automatisk stavningskontroll (Ctrl+Skift+O)

När **Automatisk stavningskontroll** är aktiverad, stryks felstavad text under i dokumentet medan du skriver.

Verktyg → Stavning → Stavning...

Det här aktiverar stavningskontrollprogrammet, ett program som är konstruerat för att hjälpa användaren hitta och korrigera eventuella stavfel. Genom att klicka på det här alternativet startas kontrollen och visar rättstavningsdialogrutan som kan användas för att styra förloppet. Det finns fyra inställningar i en kolumn mitt i dialogrutan med tillhörande namn till vänster. Med början längst upp är de:

Okänt ord:

Här visar stavningskontrollen ordet som för närvarande hanteras. Det här sker när kontrollen stöter på ett ord som inte finns i dess ordbok, en fil som innehåller en lista på ord med riktig stavning, som den jämför varje ord i editorn med.

Ersätt med:

Om kontrollen har några liknande ord i sin ordbok, kommer det första att visas här. Användaren kan acceptera förslaget, skriva in sin egen rättning, eller välja ett annat förslag från nästa ruta.

Språk:

Om du har installerat flera ordlistor, kan du välja vilken ordlista/språk som ska användas.

På höger sida i dialogrutan finns sex knappar som låter användaren styra stavningskontrollprocessen. De är:

Lägg till i ordlista

Genom att klicka på den här knappen läggs **Okänt ord** till i stavningskontrollens ordlista. Det betyder att stavningskontrollen alltid kommer att anse att ordet är rättstavat i framtiden.

Föreslå

Här kan stavningskontrollen lista ett antal möjliga ersättningar av ordet som behandlas. Genom att klicka på något av förslagen gör att ordet läggs till i rutan ovanför, **Ersätt med**.

Ersätt

Den här knappen låter stavningskontrollen ersätta ordet som behandlas i dokumentet med ordet i rutan **Ersätt med**.

Ersätt alla

Den här knappen gör att stavningskontrollen inte bara ersätter nuvarande **Okänt ord**: men automatiskt gör samma ersättning för alla andra förekomster av **Okänt ord** i dokumentet.

Ignorera

Genom att aktivera den här knappen kommer kontrollen fortsätta utan att göra några ändringar.

Ignorera alla

Den här knappen talat om för kontrollen att inte göra något med det nuvarande **okända ordet** och att hoppa över alla andra förekomster av samma ord.

NOT

Det här gäller bara för den nuvarande körningen av stavningskontrollen. Om kontrollen körs igen kommer den att stanna på samma ord.

Ytterligare tre knappar är placerade horisontellt längs dialogrutan för stavningskontroll. De är:

Hjälp

Det här startar KDE:s hjälpsystem med hjälpsidan för den här dialogrutan.

Slutförd

Den här knappen avslutar stavningskontrollförloppet, och återgår till dokumentet.

Avbryt

Den här knappen avbryter stavningskontrollen, alla ändringar återställs och du återgår till dokumentet.

Verktyg → Stavning → Stavning (från markör)...

Det här aktiverar programmet för stavningskontroll, men det startar där markören finns istället för från början i dokumentet.

Verktyg → Stavning → Stavningskontroll av markering...

Stavningskontrollerar den aktuella markeringen.

Verktyg → Stavning → Byt ordlista

Visar en kombinationsruta med alla tillgängliga ordlistor för stavningskontroll längst ner i editorns fönster. Det gör det möjligt att enkelt byta ordlista för stavningskontroll, t.ex. för automatisk stavningskontroll av text på andra språk.

4.7 Menyerna Inställningar och Hjälp

Inställningar → Editorns färgtema

Den här menyn visar tillgängliga färgscheman. Du kan ändra schema för den nuvarande vyn här. För att ändra förvalt schema måste du använda sidan [Teckensnitt och färger](#) i inställningsdialogrutan.

KatePart har de vanliga menyalternativen i KDE, **Inställningar** och **Hjälp**. För mer information läs avsnitten om [Menyn Inställningar](#) och [Menyn Hjälp](#) i KDE:s grunder.

Kapitel 5

Avancerade redigeringsverktyg

Anders Lund
Dominik Haumann
Översättare: Johan Thelmén

5.1 Kommentera/avkommentera

Kommandona **Kommentera** och **Avkommentera**, som är tillgängliga i menyn **Verktyg**, låter dig lägga till eller ta bort kommentarmarkeringar i den markerade texten, eller den nuvarande raden, om ingen text är markerad, om kommentarer stöds av textformatet som du redigerar.

Reglerna för hur kommentarer ser ut definieras av syntaxdefinitionerna, så om syntaxfärgläggning inte används, så är det inte möjligt att kommentera eller avkommentera.

Vissa format definierar kommentarmarkeringar för enstaka rader, vissa för flera rader och vissa både ock. Om kommentarmarkeringar för flera rader inte är tillgängliga, så är det inte möjligt att kommentera en markering där den sista raden inte ingår helt och hållet.

Om det finns en kommentarmarkering för enstaka rader, så föredras kommentarer för enstaka rader när det är möjligt, eftersom detta hjälper till att undvika problem med nästlade kommentarer.

När kommentarmarkeringar tas bort, så ska inte någon okommenterad text markeras. När kommentarmarkeringar för flera rader tas bort från en markering, så ignoreras eventuella blanktecken utanför kommentarmarkeringen.

För att lägga till kommentarmarkeringar, använd menyalternativet **Verktyg** → **Kommentera** eller motsvarande snabbtangens, som normalt är **Ctrl+D**.

För att ta bort kommentarmarkeringar, använd menyalternativet **Verktyg** → **Avkommentera** eller motsvarande snabbtangens, som normalt är **Ctrl+Skift+D**.

5.2 Editorkomponentens kommandorad

KateParts editorkomponent har en intern kommandorad, som låter dig utföra vissa åtgärder från ett minimalt grafiskt gränssnitt. Kommandoraden är ett textinmatningsfält längst ner i redigeringsområdet. För att visa den, välj **Visa** → **Byt till kommandorad** eller använd snabbtangensen (normalt **F7**). Editorn tillhandahåller en uppsättning kommandon som är dokumenterade nedan, och ytterligare kommandon kan tillhandahållas av insticksprogram.

Skriv in kommandot och tryck på returtangenten för att utföra det. Kommandoraden anger om kommandot lyckades och visar kanske ett meddelande. Om du kom till kommandoraden genom

att trycka på **F7** döljs den automatiskt efter några sekunder. För att rensa meddelandet och skriva in ett nytt kommando, tryck på **F7** igen.

Kommandoraden har ett inbyggt hjälpsystem, skriv in kommandot **help** för att komma igång. För att se en lista med alla tillgängliga kommandon, skriv in **help list**. För att visa hjälp om ett visst kommando, skriv **help kommando**.

Kommandoraden har en inbyggd historik, så att du kan återanvända kommandon som redan skrivits in. För att navigera i historiken, använd tangenterna **Uppåtpil** och **Neråtpil**. När historikkommandon visas är kommandots argument markerade, vilket låter dig enkelt skriva över argumenten.

5.2.1 Vanliga kommandon på kommandoraden

ARGUMENTTYPER

BOOLESK

Denna används med kommandon som sätter på eller stänger av någonting. Möjliga värden är **on** (på), **off** (av), **true** (sant), **false** (falskt), **1** eller **0**.

HELTAL

Ett heltal.

STRÄNG

En sträng, omgiven av enkla citationstecken (') eller dubbla citationstecken (") när den innehåller mellanslag.

5.2.1.1 Kommandon för att anpassa editorn

Dessa kommandon tillhandahålls av editorkomponenten och låter dig bara anpassa det aktiva dokumentet och vyn. Det är praktiskt om du vill använda en inställning som skiljer sig från den förvalda inställningen, till exempel för indentering.

set-tab-width HELTAL bredd

Ställer in tabulatorbredden till talet **bredd**.

set-indent-width HELTAL bredd

Ställer in indenteringsbredden till talet **bredd**. Används bara om du indenterar med mellanslag.

set-word-wrap-column HELTAL bredd

Ställer in radbredden för hård radbrytning till **bredd**. Detta används om du använder automatisk radbrytning för din text.

set-icon-border BOOLESK aktivera

Ställer in synligheten hos ikonkanten.

set-folding-markers BOOLESK aktivera

Ställer in synligheten hos vikmarkörrutan.

set-line-numbers BOOLESK aktivera

Ställer in synligheten hos radnummerrutan.

set-replace-tabs BOOLESK aktivera

Om aktiverad, ersätts tabulatortecken med mellanslag medan du skriver.

set-remove-trailing-space BOOLESK aktivera

Om aktiverad, tas avslutande blanktecken bort så fort markören lämnar en rad.

set-show-tabs BOOLESK aktivera

Om aktiverad, synliggörs tabulatortecken och avslutande blanktecken med en liten punkt.

set-show-indent BOOLESK aktivera

Om aktiverad, synliggörs indentering av en vertikal punktlinje.

set-indent-spaces BOOLESK aktivera

Om aktiverad, indenterar editorn med `indenteringsbredd` mellanslag för varje indenteringsnivå, istället för ett tabulatortecken.

set-mixed-indent BOOLESK aktivera

Om aktiverad, använder KatePart en blandning av tabulatortecken och mellanslag för indentering. Varje indenteringsnivå har bredden `indenteringsbredd`, och fler indenteringsnivåer optimeras så att så många tabulatortecken som möjligt används.

När det utförs aktiverar kommandot dessutom indentering med mellanslag, och om indenteringsbredden inte anges ställs den in till halva tabulatorbredden för dokumentet.

set-word-wrap BOOLESK aktivera

Aktiverar dynamisk radbrytning enligt **aktivera**.

set-replace-tabs-save BOOLESK aktivera

Om aktiverad, ersätts tabulatortecken med blanktecken så fort dokumentet sparas.

set-remove-trailing-space-save BOOLESK aktivera

Om aktiverad, tas avslutande mellanslag bort från alla rader så fort dokumentet sparas.

set-indent-mode STRÄNG namn

Ställer in automatiskt indenteringsläge till **namn**. Om **namn** inte känns igen, ställs läget in till 'none'. Giltiga lägen är 'none', 'normal', 'cstyle', 'haskell', 'lilypond', 'lisp', 'python', 'ruby' och 'xml'.

set-auto-indent BOOLESK skript

Aktivera eller inaktivera automatisk indentering.

set-highlight STRÄNG färgläggning

Ställer in dokumentets syntaxfärgläggningssystem. Argumentet måste vara ett giltigt färgläggningsnamn, som visas i menyn **Verktyg** → **Färgläggning**. Kommandot tillhandahåller en lista för automatisk komplettering av dess argument.

reload-scripts

Läs in alla **JavaScript-skript** som används av Kate, inklusive indenterare och kommandoradskript.

set-mode STRÄNG läge

Välj filtypschema för aktuellt dokument.

nn[oremap] STRÄNG original STRÄNG avbildad

Avbilda tangentsekvensen **original** till **avbildad**.

5.2.1.2 Kommandon för redigering

Dessa kommandon ändrar aktuellt dokument.

indent

Indenterar de markerade raderna eller den nuvarande raden.

unindent

Avindenterar markerade rader eller nuvarande rad.

cleanindent

Rensar indenteringen för markerade rader eller nuvarande rad enligt dokumentets indenteringsinställningar.

comment

Infogar kommentarmarkeringar för att göra markeringen, markerade rader eller nuvarande rad en kommentar enligt textformatet som definieras av dokumentets syntaxfärgläggningsdefinition.

uncomment

Tar bort kommentarmarkeringar från markeringen, markerade rader eller nuvarande rad enligt textformatet som definieras av dokumentets syntaxfärgläggningsdefinition.

kill-line

Tar bort den nuvarande raden.

replace STRÄNG mönster STRÄNG ersättning

Ersätter text som matchar **mönster** med **ersättning**. Om du vill att **mönster** ska innehålla blanktecken, måste både **mönster** och **ersättning** placeras inom enkla eller dubbla citationstecken. Om argumenten inte är citerade, används det första ordet som **mönster** och resten som **ersättning**. Om **ersättning** är tom, tas varje förekomst av **mönster** bort.

Du kan anpassa kommandot genom att lägga till ett kolon följt av en eller flera bokstäver som var och en representerar en inställning, genom att ange **replace:alternativ mönster ersättning**. Tillgängliga alternativ är:

b

Sök bakåt.

c

Sök från markörens position.

e

Sök bara i markeringen.

r

Utför en sökning med ett reguljärt uttryck. Om det anges, kan du använda **\N**, där N är ett tal, för att representera hittad text i ersättningssträngen.

s

Utför skiftlägeskänslig sökning.

p

Fråga efter tillåtelse att ersätta nästa förekomst.

w

Matcha bara hela ord.

date STRÄNG format

Infogar en datum- och tidssträng som definieras av angivet **format**, eller formatet 'yyyy-MM-dd hh:mm:ss' om inget anges. Följande översättningar görs när **format** tolkas:

d	Dagen som ett tal utan inledande nolla (1-31).
dd	Dagen som ett tal med inledande nolla (01-31).
ddd	Dagens förkortade lokala namn (t.ex. 'mån', 'sön').
dddd	Dagens långa lokala namn (t.ex. 'måndag', 'söndag').
M	Månaden som ett tal utan inledande nolla (1-12).
MM	Månaden som ett tal med inledande nolla (01-12).
MMMM	Månadens långa lokala namn (t.ex. 'januari', 'december').
MMM	Månadens förkortade lokala namn (t.ex. 'jan', 'dec').
YY	Året som ett tal med två siffror (00-99).
YYYY	Året som ett tal med fyra siffror (1752-8000)
h	Timman utan inledande nolla (0-23 eller 1-12 om FM/EM visas).
hh	Timman med inledande nolla (00-23 eller 01-12 om FM/EM visas).
m	Minuten utan inledande nolla (0-59).
mm	Minuten med inledande nolla (00-59).
s	Sekunden utan inledande nolla (0-59).
ss	Sekunden med inledande nolla (00-59).
z	Millisekunden utan inledande nolla (0-999).
zzz	Millisekunden med inledande nolla (000-999).
AP	Använd FM/EM visning. AP ersätts antingen av "FM" eller "EM".
ap	Använd fm/em visning. ap ersätts antingen av "fm" eller "em".

char STRÄNG identifierare

Det här kommandot låter dig infoga enskilda tecken med deras numeriska kod, på decimal, oktal eller hexadecimal form. För att använda det, starta dialogrutan för redigeringskommandot och skriv **char: [nummer]** i inmatningsrutan, och tryck sedan på **Ok**.

Example 5.1 Exempel på användning av char

Inmatning: **char: 234**

Utmatning: ê

Inmatning: **char: 0x1234**

Utmatning: jué

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

Det här kommandot söker och ersätter på ett sätt som liknar sed, för den nuvarande raden, eller för hela filen (%s///).

I kortet, så söks texten igenom efter en text som matchar *sökmönstret*, det reguljära uttrycket mellan det första och andra snedstreckat, och vid en träff så ersätts texten med uttrycket mellan det andra och sista snedstreckat. Parenteser i sökmönstret skapar *bakåtreferenser*, det vill säga kommandot kommer ihåg vilken del av strängen som matchade inom parenteserna. Strängarna kan återanvändas i ersättningsmönstret, med referenserna \1, för första paret parenteser, \2 för det andra, och så vidare.

För att söka efter tecknen (eller), måste du *undanta* dem med ett bakstreckstecken \ (\) .

Om du lägger till **i** sist i uttrycket, blir matchningen skiftlägesokänslig. Om du lägger till **g** sist, ersätts alla förekomster av mönstret, annars ersätts bara den första förekomsten.

Example 5.2 Ersätta text på den nuvarande raden

Din vänliga kompilator stannade just, och talade om för dig att klassen `minKlass` som nämns på rad 3902 i din källkodsfil inte är definierad.

”Tusan också!” tänker du. Det ska förstås vara `MinKlass`. Du går till rad 3902, och istället för att försöka hitta ordet i texten, så startar du dialogrutan för redigeringskommandot, och skriver in **s/minklass/MinKlass/i**, trycker på knappen **Ok**, sparar filen och kompilerar – utan problem med felet borta.

Example 5.3 Ersätta text i hela filen

Antag att du har en fil där du nämner namnet ‘Anna Johansson’ flera gånger, och någon kommer in och talar om för dig att hon precis har gift sig och bytt namn till ‘Anna Svensson’. Du vill, förstås, byta ut alla förekomster av ‘Anna Johansson’ mot ‘Anna Svensson’.

Starta kommandoraden, och skriv in kommandot: **%s/Anna Johansson/Anna Svensson/** , tryck på returtangenten och du är klar.

Example 5.4 Ett mer avancerat exempel

Det här exemplet använder *bakåtreferenser* samt en *ordklass* (om du inte vet vad det är, se tillhörande dokumentation som nämns nedan).

Antag att du har följande rad:

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

Nu inser du att det inte är bra kod, och bestämmer dig för att du vill använda nyckelordet `const` för alla ‘adress’- parametrar, de som markeras med operatoren `&` framför argumentets namn. Du skulle också vilja förenkla blanktecknen, så att det bara finns ett blanktecken mellan varje ord.

Starta dialogrutan för redigeringskommandot, skriv in: **s/\s+(\w+)\s+(\&)/ const \1 \2/ g** och tryck sedan på knappen **Ok**. Tecknet **g** i slutet på raden gör att det reguljära uttrycket görs om för varje träff, så att *bakåtreferenserna* sparas.

Utmatning: `void MinKlass::DoStringOps( const String &x, const String &y String *p, const int &a, const int &b )`

Åtgärden utförd! Vad hände nu egentligen? Ja, vi letade efter blanktecken (\s+), följt av en eller flera bokstäver (\w+), följt av ytterligare blanktecken (\s+), följt av ett och-tecken, och sparade bokstäverna samt och-tecknet för att återanvända i ersättningsoperationen. Därefter ersatta vi den matchande delen av vår rad, med ett blanktecken följt av ‘const’, följt av ett blanktecken, följt av våra sparade bokstäver (\1), följt av vårt sparade och-tecken (\2)

I vissa fall var bokstäverna ‘String’, i andra ‘int’, så att använda teckenklassen `\w` och tecknet + visade sig vara en värdefull tillgång.

sort

Sorterar den markerade texten eller hela dokumentet.

natsort

Sorterar de markerade raderna eller hela dokumentet i naturlig ordning.

Example 5.5 sort mot natsort

sort (**a10**, **a1**, **a2**) ger resultatet **a1**, **a10**, **a2**

natsort (**a10**, **a1**, **a2**) ger resultatet **a1**, **a2**, **a10**

moveLinesDown

Flytta markerade rader neråt.

moveLinesUp

Flytta markerade rader uppåt.

uniq

Ta bort duplicerade rader från den markerade texten eller hela dokumentet.

rtrim

Ta bort avslutande blanktecken från den markerade texten eller hela dokumentet.

ltrim

Ta bort inledande blanktecken från den markerade texten eller hela dokumentet.

join [STRÄNG separator]

Sammanfoga markerade linjer eller hela dokumentet. Har en valfri parameter som definierar ett separatorecken, till exempel: **join** ' , '.

rmblank

Ta bort alla blanktecken från den markerade texten eller hela dokumentet.

alignon

Kommandot justerar rader i det markerade blocket eller hela dokumentet på kolumnen angiven av ett reguljärt uttryck som anges som ett argument.

Om du anger ett tomt mönster, justeras den på radens första tecken som inte är ett blanktecken.

Om mönstret har en platsmarkör indenterar det där markören matchar.

Exempel:

alignon - infogas mellanslag innan första '-' på varje rad för att justera alla på samma kolumn.

alignon :\\s+(.) infogar mellanslag före det första tecknet som inte är ett blanktecken efter ett kolon för att justera alla till samma kolumn.

unwrap

Ta bort radbrytning för markerad text eller hela dokumentet.

each STRÄNG skript

Givet en JavaScript-funktion som argument, anropa den för listan med markerade rader och ersätt dem med returvärdet från anropet.

Example 5.6 Sammanfoga markerade linjer

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

Eller, kortare:

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

filter STRÄNG skript

Givet en JavaScript-funktion som argument, anropa den för listan med markerade rader och ta bort de där anropet returnerar falskt.

Example 5.7 Ta bort tomma linjer

```
filter 'function(1){return 1.length > 0;}'
```

Eller, kortare:

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

map STRÄNG skript

Givet en JavaScript-funktion som argument, anropa den för listan med markerade rader och ersätt raden med värdet från anropet.

Example 5.8 Ta bort tomma linjer

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

Eller, kortare:

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

duplicateLinesUp

Duplicera markerade rader ovanför den nuvarande markeringen.

duplicateLinesDown

Duplicera markerade rader nedanför den nuvarande markeringen.

5.2.1.3 Kommandon för navigering

goto HELLTAL rad

Det här kommandot går till den angivna raden.

grep STRÄNG mönster

Sök i dokumentet efter det reguljära uttrycket **mönster**. För mer information, se [appendix A](#).

find STRÄNG mönster

Kommandot går till första förekomsten av **mönster** enligt inställningen. Följande förekomster kan hittas genom att använda **Redigera** → **Sök igen** (normal snabbtangent är **F3**).

Kommandot kan anpassas genom att lägga till ett kolon följt av ett eller flera alternativ. Formatet är **find:alternativ mönster**. Följande alternativ stöds:

- b** Sök bakåt.
- c** Sök från markörens position.
- e** Sök bara i markeringen.
- r** Utför en sökning med ett reguljärt uttryck. Om det anges, kan du använda **\N**, där **N** är ett tal, för att representera hittad text i ersättningssträngen.
- s** Utför skiftlägeskänslig sökning.
- w** Matcha bara hela ord.

ifind STRÄNG mönster

Kommandot tillhandahåller 'sökning medan du skriver'. Du kan ställa in beteendet eller sökningen genom att lägga till ett kolon följt av ett eller flera alternativ, på detta sätt: **ifind:alternativ mönster**. Tillåtna alternativ är:

- b** Sök bakåt.
- r** Gör en sökning med ett reguljärt uttryck.
- s** Utför skiftlägeskänslig sökning.
- c** Sök från markörens position.

5.2.1.4 Kommandon för grundläggande redigeringsfunktioner (dessa beror på programmet som editorkomponenten används i)

- w** Spara aktuellt dokument.
- wa** Spara alla dokument som för närvarande är öppna.
- q** Stäng aktuellt dokument.
- qa** Stäng alla öppna dokument.
- wq** Spara och stäng aktuellt dokument.
- wqa** Spara och stäng alla dokument som för närvarande är öppna.
- x** Spara och stäng aktuellt dokument, men bara om det har ändrats.
- x** Spara och stäng alla dokument som för närvarande är öppna, men bara om de har ändrats.
- bp** Gå till föregående dokument i dokumentlistan.
- bn** Gå till nästa dokument i dokumentlistan.
- new** Öppna ett nytt dokument i en horisontellt delad vy.
- vnew** Öppna ett nytt dokument i en vertikalt delad vy.
- e** Läs in aktuellt dokument om det har ändrats på disk.
- enew** Redigera ett nytt dokument.
- print** Öppna utskriftsdialogrutan för att skriva ut det aktuella dokumentet.

5.3 Använda kodvikning

Kodvikning låter dig dölja delar av ett dokument i editorn, vilket gör det lättare att få en överblick av stora dokument. I KatePart görs beräkningen av områden som kan fällas ihop genom att använda reglerna som anges i definitionerna av syntaxfärgläggning, och därför är det bara tillgängligt i vissa format, typiskt i programkällkod, XML-markering och liknande. De flesta definitioner av syntaxfärgläggning som stöder kodvikning låter dig också manuellt definiera områden som kan fällas ihop, typiskt genom att använda nyckelorden **BEGIN** och **END**.

För att använda kodvkningsfunktionen, aktivera vikmarkörer genom att använda menyalternativet **Visa** → **Visa vikmarkörer** om de inte redan är synliga. Vikmarkörrutan till vänster på skärmen visar en grafisk vy av områden som kan fällas ihop, med triangelsymboler för att ange möjliga åtgärder för ett givet område: en triangel med spetsen neråt betyder att området är expanderat, och att klicka på den fäller ihop området så att en triangel som pekar åt höger visas istället.

Tre kommandon tillhandahålls för att hantera tillståndet hos områden som kan fällas ihop, se [menydokumentationen](#).

De ihopfällda raderna blir ihågkomna när en fil stängs, så när filen öppnas igen är de ihopfällda delarna fortfarande ihopfällda. Det gäller också vid uppdatering.

Om du inte vill använda kodvkningsfunktionen kan du inaktivera alternativet **Visa vikmarkörer (om tillgängliga)** på sidan [Utseende](#) i editorns inställningar.

Kapitel 6

Utöka KatePart

T.C. Hollingsworth

Översättare: Johan Thelmén

6.1 Inledning

Som alla avancerade editorer erbjuder KatePart en mängd sätt att utöka dess funktionalitet. Du kan [skriva enkla skript för att lägga till funktionalitet med JavaScript](#). Till sist, när du väl har utökat KatePart, är du välkommen att [gå med](#) och dela med dig av dina förbättringar med hela världen.

6.2 Arbeta med syntaxfärgläggning

6.2.1 Översikt

Syntaxfärgläggning är det som gör att editorn automatiskt visar text med olika stilar/färger, beroende på strängens funktion i relation till filens syfte. Till exempel i programkällkod, så kan kontrollsatser visas i fetstil, medan datatyper och kommentarer får annorlunda färg än resten av texten. Det här ökar textens läsbarhet väsentligt, och hjälper på så sätt författaren att vara effektivare och mer produktiv.

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

En C++ funktion i Perl, återgiven med syntaxfärgläggning.

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

Samma C++ funktion, utan syntaxfärgläggning.

Av de två exemplen, vilket är lättast att läsa?

KatePart levereras med ett flexibelt, anpassningsbart och avancerat system för att göra syntaxfärgläggning, och standarddistributionen tillhandahåller definitioner för ett brett område av programspråk, formaterings- och skriptspråk samt andra textfilformat. Dessutom kan du tillhandahålla dina egna definitioner med enkla XML-filer.

KatePart detekterar automatiskt de riktiga syntaxreglerna när du öppnar en fil, baserat på filens MIME-typ, som avgörs av dess filändelse, eller om den inte har någon, dess innehåll. Skulle du råka ut för ett dåligt val, kan du ställa in syntaxen som används för hand med menyn **Verktyg** → **Färgläggning**.

Stilarna och färgerna som används av varje syntaxfärgläggningsläge kan anpassas under fliken [Färgläggningstextstilar](#) i [inställningsdialogrutan](#), medan MIME-typerna och filändelserna det ska användas för hanteras under fliken [Lägen och filtyper](#).

NOT

Syntaxfärgläggning finns för att förbättra läsbarheten för riktig text, men du kan inte lita på att den validerar din text. Att markera text för syntax kan vara svårt, beroende på formatet som du använder, och i vissa fall är upphovsmännen till syntaxreglerna stolta om 98 procent av texten visas korrekt, även om du behöver en ovanlig stil för att se de felaktiga 2 procenten.

6.2.2 KateParts syntaxfärgläggningssystem

Det här avsnittet beskriver KateParts syntaxfärgläggningsmekanism i mer detalj. Det är avsett för dig, om du vill veta mer om den, eller om du vill skapa och ändra syntaxdefinitioner.

6.2.2.1 Hur det fungerar

Så fort du öppnar en fil, är en av de första sakerna som KatePart-editorn gör att avgöra vilken syntaxdefinition som ska användas för filen. När filens text läses in, och medan du skriver i den, så analyserar syntaxfärgläggningssystemet texten enligt reglerna som definieras i syntaxdefinitionen och markerar var i den som olika sammanhang och stilar börjar och slutar.

När du skriver in text i dokumentet, så analyseras och markeras den nya texten i farten, så att om du tar bort ett tecken som markeras som början eller slutet på ett sammanhang, så ändras stilen på den omgivande texten i enlighet med detta.

Syntaxdefinitioner som används av KateParts syntaxfärgläggningssystem är XML-filer, som innehåller

- Regler för att detektera funktionen hos text, organiserade i sammanhangsblock
- Listor med nyckelord
- Definitioner av stilobjekt

När texten analyseras utvärderas detekteringsreglerna i den ordning som de definierades, och om början på den nuvarande strängen matchar en regel, så används motsvarande sammanhang. Startpunkten i texten flyttas till den sista punkten där regeln matchade, och en ny genomgång av reglerna sker, med början i sammanhanget som anges av den matchande regeln.

6.2.2.2 Regler

Detekteringsreglerna är centrala för färgläggningsdetekteringssystemet. En regel är en sträng, ett tecken eller ett reguljärt uttryck som texten som ska analyseras matchas mot. Den innehåller information om vilken stil som ska användas för den delen av texten som matchar. Den kan byta arbetssammanhanget för systemet, antingen till ett sammanhang som anges explicit, eller till det föregående sammanhanget som användes av texten.

Reglerna organiseras i sammanhangsgrupper. En sammanhangsgrupp används för de huvudsakliga koncepten i formatets text, till exempel textsträngar inom citationstecken eller kommentarblock i programkällkod. Det här försäkrar att färgläggningsystemet inte behöver gå igenom alla regler när det inte är nödvändigt, och att vissa teckensekvenser i texten kan hanteras annorlunda beroende på det nuvarande sammanhanget.

Sammanhang kan skapas dynamiskt för att till exempel tillåta användning av instansspecifik data i regler.

6.2.2.3 Sammanhangsstilar och nyckelord

I vissa programspråk, hanteras heltal annorlunda än flyttal av kompilatorn (programmet som översätter källkoden till körbart binärformat), och det kan finnas tecken som har en särskild mening i en sträng med citationstecken. I sådana fall är det vettigt att visa dem på ett annat sätt än omgivningen så att de är lätta att identifiera när texten läses. Så även om de inte representerar speciella sammanhang, så kan de betraktas som sådana av syntaxfärgläggningsystemet, så att de kan markeras för att visas på ett annorlunda sätt.

En syntaxdefinition kan innehålla så många stilar som krävs för att täcka koncepten i det format den används för.

I många format finns det en lista på ord som representerar ett speciellt koncept. Till exempel i programspråk, så är kontrollsatser ett koncept, namn på datatyper ett annat, och inbyggda funktioner i språket ett tredje. KateParts syntaxfärgläggningsystem kan använda sådana listor för att detektera och markera ord i texten för att visa koncepten för textformaten.

6.2.2.4 Standardstilar

Om du öppnar en C++ källkodsfil, en Java™ källkodsfil och ett HTML-dokument i KatePart, så ser du att även om formaten är olika, och olika ord därför väljes för särskild behandling, så är färgerna som används samma. Det här beror på att KatePart har en fördefinierad lista med standardstilar, som används av varje individuell syntaxdefinition.

Det här gör det lätt att känna igen liknande koncept i olika textformat. Kommentarer finns till exempel i nästa alla program-, skript- eller formateringsspråk, och när de visas med samma stil i alla språk, behöver du inte stanna och tänka efter för att identifiera dem i texten.

TIPS

Alla stilar i en syntaxdefinition använder en av standardstilarna. Några få syntaxdefinitioner använder fler stilar än det finns som standard, så om du ofta använder ett format, kan det vara värt att titta i inställningsdialogrutan för att se om några koncept använder samma stil. Det finns till exempel bara en standardstil för strängar, men eftersom programspråket Perl hanterar två sorters strängar, kan du utöka färgläggningen genom att ställa in dessa så att de skiljer sig något. Alla [tillgängliga standardstilar](#) förklaras senare.

6.2.3 XML formatet för syntaxfärgläggningsdefinitioner

6.2.3.1 Översikt

KatePart använder ramverket för syntaxfärgläggning från KDE Ramverk. XML-standardfilen för färgläggning som levereras med KatePart kompileras normalt i syntaxfärgläggningsbiblioteket.

Det här avsnittet är en översikt av färgläggningsdefinitionernas XML-format. Det beskriver huvudkomponenterna och deras betydelse och användning, och går djupare in i detaljerna för de-tekteringsreglerna.

Den formella definitionen, som också kallas XSD finns i [arkivet för syntaxfärgläggning](#) i filen `language.xsd`.

Egna `.xml` färgläggningsdefinitionsfiler finns i `org.kde.syntax-highlighting/syntax/` i användarkatalogen som hittas med `qtpaths --paths GenericDataLocation` som oftast är `$HOME/.local/share/` och `/usr/share/`.

För Flatpak- och Snap-paket, fungerar inte ovanstående katalog eftersom dataplatsen är olika för varje program. I ett Flatpak-program, är platsen för anpassade XML-filer oftast `$HOME/.var/app/p/flatpak-paketnamn/data/org.kde.syntax-highlighting/syntax/` och i ett Snap-program är platsen `$HOME/snap/snap-paketnamn/current/.local/share/org.kde.syntax-highlighting/syntax/`.

På Windows[®] är filerna placerade i `%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\syntax`. `%USERPROFILE%` som oftast expanderas till `C:\Users\user`.

Sammanfattningsvis, för de flesta konfigurationer är katalogen för anpassade XML-filer följande:

För lokal användare	<code>\$HOME/.local/share/org.kde.syntax-highlighting/syntax/</code>
För alla användare	<code>/usr/share/org.kde.syntax-highlighting/syntax/</code>
För Flatpak-paket	<code>\$HOME/.var/app/flatpak-paketnamn/data/org.kde.syntax-highlighting/syntax/</code>
För Snap-paket	<code>\$HOME/snap/snap-paketnamn/current/.local/share/org.kde.syntax-highlighting/syntax/</code>
På Windows [®]	<code>%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\syntax</code>

Om flera filer finns för samma språk, läses filen med det största värdet på attributet **version** i elementet **language** in.

HUVUDKOMponenter i KATEPARTS FÄRGLÄGgnINGSDEFINITIONER

En syntaxfärgläggningsfil innehåller ett huvud som anger XML-versionen:

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

Definitionsfilens rot är elementet **language**. Följande egenskaper är tillgängliga:

Egenskaper som krävs:

name anger språkets namn. Det visas senare i menyer och dialogrutor.

section anger kategorin.

extensions definierar filändelser, såsom `"*.cpp;*.h"`.

version anger aktuell version av definitionsfilen i form av ett heltal. När en definitionsfil för färgläggning ändras, försäkra dig om att talet ökas.

kateversion anger senaste version av KatePart som stöds.

Valfria egenskaper:

mimetype associerar filers MIME-typ.

casesensitive definierar om nyckelord är skiftlägeskänsliga eller inte.

priority krävs om en annan syntaxfärgläggningsfil använder samma filändelse. Den högsta prioriteten vinner.

author innehåller författarens namn och e-postadress.

license innehåller licensen, oftast MIT-licensen för nya syntaxfärgläggningsfiler.

stil innehåller språket som tillhandahålls och används av indenterare för egenskapen `required-syntax-style`.

indenter definierar vilken indenterare som normalt används. Tillgängliga indenterare är: *ada, normal, cstyle, cmake, haskell, latex, lilypond, lisp, lua, pascal, python, replicode, ruby* och *xml*.

hidden definierar om namnet ska visas i KateParts menyer.

Nästa rad kan alltså se ut så här:

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

Därefter kommer elementet **highlighting** som innehåller det valfria elementet **list** och det nödvändiga elementen **contexts** och **itemDatas**.

list element innehåller en lista med nyckelord. I det här fallet är nyckelorden *class* och *const*. Du kan lägga till så många listor som du behöver.

Sedan KDE Ramverk 5.53 kan en lista inkludera nyckelord från en annan lista, språk eller fil, genom att använda elementet **include** element.## är att separera listnamnet och språkdefinitionens namn, på samma sätt som regeln **IncludeRules**. Det är användbart för att undvika att duplicera nyckelordslistor, om du behöver inkludera nyckelorden från ett annat språk eller fil. Exempelvis innehåller listan *annat-namn* nyckelordet *str* och alla nyckelorden från listan *types*, vilken hör till språket *ISO C++*.

Elementet **contexts** innehåller alla sammanhang. Det första sammanhanget är förvalt som start för färgläggningen. Det finns två regler i sammanhanget *Normal Text* som matchar listan av nyckelord med namnet *något-namn* och en regel som detekterar ett citationstecken och byter sammanhang till *string*. För att lära dig mer om regler, läs nästa kapitel.

Den tredje delen är elementet **itemDatas**. Det innehåller alla färger och teckenstilar som behövs av sammanhangen och reglerna. I det här exemplet används **itemData** *Normal Text*, *String* och *Keyword*.

```
<highlighting>
  <list name="något-namn">
    <item> class </item>
    <item> const </item>
  </list>
  <list name="annat-namn">
    <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="något-namn ↔
      " />
```

```

        <DetectChar attribute="String" context="String=" annat-namn" />
        <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>

```

Den sista delen av färgläggningsdefinitionen är den valfria sektionen **general**. Den kan innehålla information om nyckelord, kodvikning, kommentarer, indentering, tomma rader och stavningskontroll.

Sektionen **comment** definierar den sträng som enradskommentarer inleds med. Du kan också definiera en flerradskommentar med *multiLine* och den ytterligare egenskapen *end*. Det är användbart när användaren trycker på snabbtangenten som motsvarar *kommentera/avkommentera*.

Sektionen **keywords** definierar om listor med nyckelord är skiftlägeskänsliga eller inte. Andra egenskaper förklaras senare.

Övriga sektioner, **folding**, **emptyLines** och **spellchecking**, behövs oftast inte och förklaras senare.

```

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

```

6.2.3.2 Sektionerna i detalj

Den här delen beskriver alla tillgängliga egenskaper för sammanhang, itemData, nyckelord, kommentarer, kodvikning och indentering.

Elementet **context** hör till gruppen **contexts**. Själva sammanhanget definierar sammanhangsspecifika regler, som vad som ska ske om färgläggningssystemet når slutet på en rad. Tillgängliga egenskaper är:

name anger sammanhangets namn. Regler använder namnet för att ange sammanhanget att byta till om regeln matchar.

lineEndContext definierar sammanhanget som färgläggningssystemet byter till om det når ett radslut. Det kan antingen vara namnet på ett annat sammanhang, **#stay** för att inte byta sammanhang (dvs. göra ingenting) eller **#pop** som gör att det lämnar sammanhanget. Det är till exempel möjligt att använda **#pop#pop#pop** för att återgå tre steg, eller till och med **#pop#pop!Annat_Sammanhang** för att återgå två steg och byta till sammanhanget som heter **Annat_Sammanhang**. Det är också möjligt att byta till ett sammanhang som hör till en annan språkdefinition, på samma sätt som för reglerna **IncludeRules**, t.ex. **Något_Sammanhang##JavaScript**. Observera att det inte går att använda det här sammanhangsbytet tillsammans med **#pop**, exempelvis är **#pop!Något_Sammanhang##JavaScript** inte giltigt. Sammanhangsbyten beskrivs också i Avsnitt 6.2.4.

lineEmptyContext definierar sammanhanget om en tom rad påträffas. Terminologin för sammanhangsbyte är samma som tidigare beskrivits i *lineEndContext*. Förval: **#stay**.

fallthroughContext anger nästa sammanhang att byta till om ingen regel matchar. Terminologin för sammanhangsbyte är samma som tidigare beskrivits i *lineEndContext*. Förval: **#stay**.

fallthrough definierar om färgläggningssystemet byter till sammanhanget som anges i **fallthroughContext** om ingen regel matchar. Observera att sedan KDE Frameworks 5.62 avråds användning av egenskapen, använd istället **fallthroughContext**, eftersom om egenskapen **fallthroughContext** finns förstås implicit att värdet på **fallthrough** är *true*. Förval: *false*.

noIndentationBasedFolding inaktiverar indenteringsbaserad vikning i sammanhanget. Om indenteringsbaserad vikning inte är aktiverad är egenskapen meningslös. Det definieras av elementet *folding* i gruppen *general*. Förval: *false*.

Elementet **itemData** är i gruppen **itemDatas**. Det definierar teckenstilen och färgerna. Det är alltså möjligt att definiera egna stilar och färger. Dock rekommenderar vi att du håller dig till standardstilarna om möjligt, så att användaren alltid ser samma färger använda i olika språk. Ibland finns det dock inget annat sätt, och det är nödvändigt att ändra färg- och teckenegenskaper. Egenskapens namn och **defStyleNum** krävs, övriga är valfria. Tillgängliga egenskaper är:

name anger namnet på **itemData**. Sammanhang och regler använder namnet i egenskapen *attribute* för att referera till **itemData**.

defStyleNum definierar vilken standardstil som ska användas. Tillgängliga standardstilar förklaras i detalj senare.

color definierar en färg. Giltiga format är **#rrggbb** eller **#rgb**.

selColor definierar markeringens färg.

italic om *true*, är texten kursiv.

bold om *true*, är texten i fetstil.

underline om *true*, är texten understruken.

strikeout om *true*, är texten överstruken.

spellChecking om *true*, stavningskontrolleras texten.

Elementet **keywords** i gruppen **general** definierar nyckelordens egenskaper. Tillgängliga egenskaper är:

casesensitive kan vara *true* eller *false*. Om det är *true*, matchas alla nyckelord skiftlägeskänsligt.

weakDelimiter är en lista med tecken som inte fungerar som ordavgränsare. Punkt **' . '** är till exempel en ordavgränsare. Med antagandet att ett nyckelord i en **list** innehåller en punkt, matchar det bara om du anger att punkten är en svag avgränsare.

additionalDelimiter definierar ytterligare avgränsare.

wordWrapDelimiter definierar tecken som en radbrytning kan ske efter.

Standardavgränsare och radbrytningsavgränsare är tecknen `. ( ) : ! + , - < = > % & * / ; ? [ ] ^ { | } ~ \`, mellanslag (`' '`) och tabulator (`'\t'`).

Elementet **comment** i gruppen **comments** definierar kommentaregenskaper som används för Verktyg → Kommentera, Verktyg → Avkommentera och Verktyg → Växla kommentar. Tillgängliga egenskaper är:

name är antingen *singleLine* eller *multiLine*. Om du väljer *multiLine* krävs egenskaperna *end* och *region*. Om du väljer *singleLine* kan du lägga till den valfria egenskapen *position*.

start definierar strängen som används för att inleda en kommentar. I C++ skulle det vara `"/**"`. Egenskapen krävs för typerna *multiLine* och *singleLine*.

end definierar strängen som används för att avsluta en kommentar. I C++ skulle det vara `**/"`. Egenskapen är bara tillgänglig för och krävs för kommentarer av typen *multiLine*.

region ska vara namnet på den vikbara flerraderskommentaren. Med antagandet att du har *beginRegion*=`"Comment "` ... *endRegion*=`"Comment "` i dina regler, ska du använda *region*=`"Comment "`. På så sätt fungerar avkommentering även om du inte markerar all text i en flerraderskommentar. Markören måste bara vara inne i flerraderskommentaren. Egenskapen är bara tillgänglig för typen *multiLine*.

position definierar var enraderskommentaren infogas. Normalt placeras enraderskommentaren i början av raden på kolumn 0, men om *position*=`"afterwhitespace"` används infogas kommentaren efter inledande blanktecken, precis före det första tecknet som inte är ett blanktecken. Det är användbart för att placera kommentarer korrekt i språk där indentering är viktig, såsom Python eller YAML. Egenskapen är valfri och det enda möjliga värdet är *afterwhitespace*. Den är bara tillgänglig för typen *singleLine*.

Elementet **folding** i gruppen **general** definierar kodvikningsegenskaper. Tillgängliga egenskaper är:

indentationsensitive om *true* läggs kodvkningsmarkörerna till baserat på indentering, som i skriptspråket Python. Oftast behöver du inte ange det, eftersom det har det förvalda värdet *false*.

Elementet **emptyLine** i gruppen **emptyLines** definierar vilka rader som ska behandlas som tomma rader. Det gör det möjligt att ändra beteendet hos egenskapen *lineEmptyContext* i elementets **context**. Tillgängliga egenskaper är:

regexpr definierar ett reguljärt uttryck som behandlas som en tom rad. Normalt innehåller tomma rader inte några tecken, därför lägger det här till ytterligare tomma rader, om du exempelvis vill att rader med mellanslag också ska anses vara tomma rader. Dock behöver man inte ange den här egenskapen i de flesta syntaxdefinitioner.

Elementet **encoding** i gruppens **spellchecking** definierar en teckenkodning för stavningskontroll. Tillgängliga egenskaper:

char är ett kodat tecken.

string är en teckenföljd som kodas som tecknet *char* i stavningskontrollen. Exempelvis representerar strängen `\{A}` tecknet **Ä** i språket LaTeX.

6.2.3.3 Tillgängliga standardstilar

Standardstilar har [redan förklarats](#). En kort sammanfattning: Standardstilar är fördefinierade stilar för teckensnitt och färger.

Allmänna standardstilar:

dsNormal, när ingen särskild färgläggning krävs.

dsKeyword, nyckelord för inbyggda språk.

dsFunction, funktionsanrop och definitioner.

dsVariable, om tillämpligt: variabelnamn (t.ex. \$enVariabel i PHP eller Perl).

dsControlFlow, nyckelord för kontrollflöde såsom if, else, switch, break, return, yield, ...

dsOperator, operatorer såsom + - * / :: < >

dsBuiltin, inbyggda funktioner, klasser och objekt.

dsExtension, vanliga ändelser såsom Qt™-klasser och funktioner eller makron i C++ och Python.

dsPreprocessor, preprocessorsatser eller makrodefinitioner.

dsAttribute, anmärkningar som @override och __declspec(...).

Strängrelaterade standardstilar:

dsChar, enstaka tecken såsom 'x'.

dsSpecialChar, tecken med speciell betydelse i strängar som undantag, ersättningar eller operatorer i reguljära uttryck.

dsString, strängar såsom "hej allihop".

dsVerbatimString, ordagranna eller obehandlade strängar som 'raw \backslash' i Perl, CoffeeScript och skal, samt r'\raw' i Python.

dsSpecialString, SQL-satser, reguljära uttryck, HERE-dokument, L^AT_EX-matematikläge, ...

dsImport, import, inkludering och require för moduler.

Nummerrelaterade standardstilar:

dsDataType, inbyggda datatyper såsom int, void, u64.

dsDecVal, decimala värden.

dsBaseN, värden med en bas skild från 10.

dsFloat, flyttalsvärden.

dsConstant, inbyggda och användardefinierade konstanter såsom PI.

Kommentar- och dokumentationsrelaterade standardstilar:

dsComment, kommentarer.

dsDocumentation, /** dokumentationskommentarer */ eller """dokumentsträngar""".

dsAnnotation, dokumentationskommandon som @param, @brief.

dsCommentVar, variabelnamnen som används i ovanstående kommandon, som "exempel" i @param exempel.

dsRegionMarker, områdesmarkeringar som //BEGIN, //END i kommentarer.

Andra standardstilar:

dsInformation, anmärkningar och tips som @note i doxygen.

dsWarning, varningar som @warning i doxygen.

dsAlert, specialord som TODO, FIXME, XXXX.

dsError, färgläggning av fel och felaktig syntax.

dsOthers, när ingenting annat passar in.

6.2.4 Regler för syntaxdetektering

Den här sektionen beskriver reglerna för syntaxdetektering.

Varje regel kan matcha noll eller flera tecken i början av strängen som de ska testas med. Om regeln matchar, så tilldelas de matchande tecknen stilen eller *egenskapen* som definieras av regeln. En regel kan också begära att det nuvarande sammanhanget byts.

En regel ser ut så här:

```
<Regelnamn attribute="(identifierare)" context="(identifierare)" [ ←
    regelspecifika egenskaper] />
```

Ordet *attribute* identifierar stilen som ska användas för tecken som matchar med namn eller index, och *context* identifierar sammanhanget som ska användas i fortsättningen.

Sammanhanget kan identifieras av:

- En *identifierare*, som är namnet på det andra sammanhanget.
- En *ordning* som talar om att färgläggningen ska stanna kvar i det nuvarande sammanhanget (**#stay**), eller gå tillbaka till det föregående sammanhanget som använts i strängen (**#pop**).
För att gå tillbaka flera steg, kan nyckelordet **#pop** upprepas: **#pop#pop#pop**
- En *ordning* följt av ett utropstecken (!) och en *identifierare*, som gör att färgläggningen först följer ordningen och därefter byter till det andra sammanhanget, t.ex. **#pop#pop!Annat_sammanhang**.
- En *identifierare*, som är ett sammanhangsnamn, följt av två nummertecken (##) och en annan *identifierare*, som är namnet på en språkdefinition. Namngivningen liknar den som används i regeln **IncludeRules** och låter dig byta till ett sammanhang som hör till en annat syntaxfärgläggningsdefinition, t.ex. **Något_Sammanhang##JavaScript**. Observera att det inte går att använda det här sammanhangsbytet tillsammans med **#pop**, exempelvis är **#pop!Något_Sammanhang##JavaScript** inte giltigt.

Regelspecifika egenskaper varierar och beskrivs i följande avsnitt.

GEMENSAMMA EGENSKAPER

- *attribute*: En egenskap avbildas på en definierad *itemData*.
- *context*: Anger sammanhanget som färgläggningssystemet byter till om regeln matchar.
- *beginRegion*: Inled ett kodvikningsblock. Förval: unset.
- *endRegion*: Avsluta ett kodvikningsblock. Förval: unset.
- *lookAhead*: Om *true* behandlar inte färgläggningssystemet matchningens längd. Förval: *false*.
- *firstNonSpace*: Matcha bara om strängen är den första förutom blanktecken på raden. Förval: *false*.
- *column*: Matcha bara om kolumnen matchar. Förval: unset.

DYNAMISKA REGLER

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

Hur det fungerar:

I länken för [reguljärt uttryck](#) i **RegExpr** reglerna, lagras all text inom vanliga kurvp parenteser i (**MÖNSTER**) och blir ihågkomna. Sådana platsmarkörer kan användas i sammanhanget som man går till i reglerna med egenskapen **dynamic true**, med *%N* (i *String*) eller *N* (i *char*).

Det är viktigt att nämna att en text lagrad i en **RegExpr** regel bara lagras för sammanhanget som man byter till, angivet i dess egenskap **context**.

TIPS

- Om platsmarkörerna inte kommer att används, varken av dynamiska regler eller i samma reguljära uttryck, ska **icke-lagrade grupper** användas: **(?:MÖNSTER)**
Grupperna *framåtreferens* eller *bakåtreferens* såsom **(?=MÖNSTER)**, **(?!MÖNSTER)** eller **(?<=MÖNSTER)** lagras inte. Se [Reguljära uttryck](#) för mer information.
- Lagringsgrupperna kan användas inom samma reguljära uttryck, använder `\N` istället för `%N`. För mer information, se [Spara text som matchar \(bakåtreferenser\)](#) under [Reguljära uttryck](#).

Exempel 1:

I det här enkla exemplet lagras texten som matchas av det reguljära uttrycket `=*` och infogas med `%1` i den dynamiska regeln. Det tillåter att kommentaren slutar med samma antal `=` som i början. Det matchar text som: `[ [ comment ] ]`, `[=[ comment ]=]` eller `[=====[ comment ]=====]`.

Dessutom är lagrad text bara tillgänglig i sammanhanget *Multi-line Comment* man byter till.

```
<context name="Normal" attribute="Normal Text" lineEndContext="#stay">
  <Regex 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>
```

Exempel 2:

I den dynamiska regeln motsvarar `%1` platsmarkörerna `#+`, och `%2` till `"+>`. Det matchar text som: `#beteckning~~~~inne i sammanhanget~~~~#`.

Platsmarkörerna är inte tillgängliga i andra sammanhang som *OtherContext*, *FindEscapes* eller *SomeContext*.

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

Exempel 3:

Det matchar text som: **Class::function<T>(...)**.

```
<context name="Normal" attribute="Normal Text" lineEndContext="#stay">
  <Regex context="FunctionName" lookAhead="true"
    String="\b([a-zA-Z_][\w-]*) (?:) ([a-zA-Z_][\w-]*) (?<[ \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>
```

LOKALA AVGRÄNSARE

- *weakDelimiter*: lista med tecken som inte fungerar som ordavgränsare.
- *additionalDelimiter* definierar ytterligare avgränsare.

6.2.4.1 Reglerna i detalj

DetectChar

Detektera ett enda specifikt tecken. Används ofta för att till exempel hitta slutet på strängar inom citationstecken.

```
<DetectChar char="(tecken)" (gemensamma egenskaper) (dynamisk) />
```

Egenskapen **char** definierar tecknet som ska matchas.

Detect2Chars

Detektera två angivna tecken i en definierad ordning.

```
<Detect2Chars char="(tecken)" char1="(tecken)" (gemensamma egenskaper) ↔
/>
```

Egenskapen **char** definierar det första tecknet som ska matcha, **char1** det andra.

AnyChar

Detektera ett tecken i en angiven teckenmängd.

```
<AnyChar String="(sträng)" (gemensamma egenskaper) />
```

Egenskapen **String** definierar teckenmängden.

StringDetect

Detektera en sträng exakt.

```
<StringDetect String="(sträng)" [insensitive="TRUE|FALSE;"] (gemensamma ↔
egenskaper) (dynamisk) />
```

Egenskapen **String** definierar strängen som ska matcha. Egenskapen **insensitive** (okänslig) är normalt **FALSE** (FALSK) och skickas med till strängjämförelsefunktionen. Om värdet är **TRUE** (SANT) så används en jämförelse som inte bryr sig om skiftläge.

WordDetect

Detektera en exakt sträng, men kräv dessutom ordgränser som en punkt ' .' eller ett blanktecken i början och slutet av ordet. Betrakta **\b<string>\b** som ett reguljärt uttryck, men snabbare än regeln **RegExpr**.

```
<WordDetect String="(sträng)" [insensitive="TRUE|FALSE;"] (gemensamma ↔
egenskaper) (lokala avgränsare)/>
```

Egenskapen **String** definierar strängen som ska matcha. Egenskapen **insensitive** (okänslig) är normalt **FALSE** (FALSK) och skickas med till strängjämförelsefunktionen. Om värdet är **TRUE** (SANT) så används en jämförelse som inte bryr sig om skiftläge.

Sedan: Kate 3.5 (KDE 4.5)

RegExp

Matchar med ett reguljärt uttryck.

```
<RegExp String="(sträng)" [insensitive="TRUE|FALSE;"] [minimal="TRUE|FALSE"] (gemensamma egenskaper) (dynamisk) />
```

Egenskapen **String** definierar det reguljära uttrycket.

Egenskapen **insensitive** (okänslig) är normalt **FALSE** (FALSK) och skickas med det reguljära uttrycket.

Egenskapen **minimal** är normalt **FALSE** (FALSK) och skickas med det reguljära uttrycket.

Eftersom reglerna alltid matchas mot början av den nuvarande strängen, så anger ett reguljärt uttryck som börjar med en hatt (^) att regeln bara ska matchas mot radens början.

Se [Reguljära uttryck](#) för mer information om dem.

keyword

Detektera ett nyckelord från en angiven lista.

```
<keyword String="(listnamn)" (gemensamma egenskaper) (lokala avgränsare) />
```

Egenskapen **String** identifierar en lista på nyckelord med namn. En lista med det namnet måste finnas.

Färgläggningssystemet behandlar nyckelordsregler på ett mycket optimerat sätt. Det gör det absolut nödvändigt att alla nyckelord som ska matchas måste omges med definierade avgränsare, antingen underförstått (standardavgränsare) eller explicit definierade inne i egenskapen *additionalDelimiter* i taggen *keywords*.

Om ett nyckelord som ska matchas måste innehålla ett skiljetecken, måste detta tecken läggas till i egenskapen *weakDelimiter* i taggen *keywords*. Detta tecken förlorar då sin avgränsningsegenskap i alla regler med *keyword*. Det är också möjligt att använda egenskapen *weakDelimiter* i *keyword* så att ändringen bara gäller den regeln.

Int

Detektera ett heltal (som det reguljära uttrycket: **\b[0-9]+**).

```
<Int (gemensamma egenskaper) (lokala avgränsare) />
```

Den här regeln har inga specifika egenskaper.

Float

Detektera ett flyttal (som det reguljära uttrycket: **(\b[0-9]+\.[0-9]*|\.[0-9]+)([eE][-+]?[0-9]+)?**).

```
<Float (gemensamma egenskaper) (lokala avgränsare) />
```

Den här regeln har inga specifika egenskaper.

HlCOct

Detektera en oktal talrepresentation (som det reguljära uttrycket: **\b0[0-7]+**).

```
<HlCOct (gemensamma egenskaper) (lokala avgränsare) />
```

Den här regeln har inga specifika egenskaper.

HlCHex

Detektera en hexadecimal talrepresentation (som det reguljära uttrycket: `\b0[xX][0-9a-fA-F]+`).

```
<HlCHex (gemensamma egenskaper) (lokala avgränsare) />
```

Den här regeln har inga specifika egenskaper.

HLCStringChar

Detektera ett undantaget tecken.

```
<HLCStringChar (gemensamma egenskaper) />
```

Den här regeln har inga specifika egenskaper.

Det matchar bokstavsrepresentationer av osynliga tecken som ofta används i programkod, till exempel `\n` (nyrad) eller `\t` (tabulator).

Följande tecken matchar om de följer ett bakstreck (`\`): `abefnrtv~' ?`. Dessutom matchar undantagna hexadecimala tal, som till exempel `\xff` och undantagna oktala tal, till exempel `\033`.

HLCChar

Detektera ett C-tecken.

```
<HLCChar (gemensamma egenskaper) />
```

Den här regeln har inga specifika egenskaper.

Det matchar C-tecken omgivna av apostrofer (till exempel `'c'`). Apostroferna kan innehålla ett enkelt tecken eller en teckenföljd. Se `HLCStringChar` för matchade teckenföljder.

RangeDetect

Detektera en sträng med definierade start- och sluttecken.

```
<RangeDetect char="(tecken)" char1="(tecken)" (gemensamma egenskaper) ←  
/>
```

char definierar tecknet som inleder intervallet, och **char1** tecknet som avslutar intervallet. Användbar för att till exempel detektera små strängar inom citationstecken och liknande, men observera att eftersom färgläggningen arbetar med en rad i taget, så hittar det inte strängar som fortsätter på nästa rad.

LineContinue

Matchar ett angivet tecken vid radens slut.

```
<LineContinue (gemensamma egenskaper) [char="\"] />
```

Egenskaper **char** definierar valfritt tecken som ska matchas, förval är bakstreck (`'\'`). Ny sedan KDE 4.13.

Regeln är användbar för att byta sammanhang vid radslut. Det behövs till exempel i C/C++ för att fortsätta makron eller strängar.

IncludeRules

Inkludera regler från ett annat sammanhang eller språk/fil.

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

Egenskaper **context** definierar vilket sammanhang som ska inkluderas.

Om den är en enkelt sträng innehåller den alla definierade regler i det nuvarande sammanhanget, till exempel:

```
<IncludeRules context="annat-sammanhang" />
```

Om strängen innehåller **##** letar färgläggningssystemet efter ett sammanhang från en annan språkdefinition med angivet namn, exempelvis skulle

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

inkludera sammanhanget *String* från färgläggningsdefinitionen av C++.

Om egenskapen **includeAttrib** är *true*, ändras målegenskapen till källans egenskap. Det krävs till exempel för att kommentarer ska fungera om text som matchas av det inkluderade sammanhanget har en annan färgläggning än värdsammanhanget.

DetectSpaces

Detektera blanktecken.

```
<DetectSpaces (gemensamma egenskaper) />
```

Den här regeln har inga specifika egenskaper.

Använd den här regeln om du vet att det kan finnas flera blanktecken framför, till exempel i början av indenterade rader. Regeln hoppar över alla blanktecken på en gång, istället för att prova flera regler och hoppa över en åt gången eftersom den inte matchar.

DetectIdentifier

Detektera strängar för identifierare (som det reguljära uttrycket: **[a-zA-Z_][a-zA-Z0-9_]***).

```
<DetectIdentifier (gemensamma egenskaper) />
```

Den här regeln har inga specifika egenskaper.

Använd den här regeln för att hoppa över en sträng med ordtecken på en gång, istället för att testa den med flera regler och hoppa ett steg i taget beroende på att ingenting matchar.

6.2.4.2 Tips och trick

- Om du bara matchar två tecken, använd **Detect2Chars** istället för **StringDetect**. Samma sak gäller för **DetectChar**.
- Reguljära uttryck är enkla att använda, men oftast finns det ett annat mycket snabbare sätt att uppnå samma resultat. Antag att du bara vill matcha tecknet '**#**' om det är det första tecknet på en rad. En lösning baserad på reguljära uttryck skulle se ut så här:

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

Du kan uppnå samma sak mycket snabbare med:

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

Om du vill matcha det reguljära uttrycket '**^#**' kan du fortfarande använda **DetectChar** med egenskapen **column="0"**. Egenskapen **column** räknar tecken, så en tabulator är bara ett tecken.

- Använd egenskapen **column="0"** i **Regex** reglerna om mönstret **^MÖNSTER** kommer att användas för att matcha text i början av en rad. Det förbättrar prestanda, eftersom det undviker sökning efter matchningar i resten av kolumnerna.

- Använd icke-lagrande grupper (**?:MÖNSTER**) istället för lagrande grupper (**MÖNSTER**), om platsmarkörerna inte kommer att användas i samma reguljära uttryck eller i dynamiska regler. Det undviker att platsmarkörer lagras i onödan.
- Du kan byta sammanhang utan att behandla tecken. Antag att du vill byta sammanhang när du stöter på strängen `*/`, men måste behandla denna sträng i nästa sammanhang. Regeln nedan matchar, och egenskapen **lookAhead** gör att färgläggningen behåller den matchade strängen för nästa sammanhang.

```
<Detect2Chars attribute="Comment" context="#pop" char="*" char1="/" ↵
lookAhead="true" />
```

- Använd **DetectSpaces** om du vet att många blanktecken förekommer.
- Använd **DetectIdentifier** istället för ett reguljärt uttryck `'[a-zA-Z_]\w*'`.
- Använd standardstilar överallt du kan. På så sätt finner användaren en bekant miljö.
- Titta i andra XML-filer för att se hur andra implementerade knepiga regler.
- Det går att validera alla XML-filer genom att använda kommandot **validatehl.sh language.xsd min-syntax.xml**. Filerna `validatehl.sh` och `language.xsd` är tillgängliga i [arkivet för syntax-färgläggning](#).
- Om du mycket ofta upprepar komplexa reguljära uttryck kan du använda **ENTITETER**. Till exempel:

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

Nu kan du använda *&myref*; istället för det reguljära uttrycket.

6.3 Arbeta med färgteman

6.3.1 Översikt

Färgteman definierar färgerna i [textredigeringsområdet](#) och [syntaxfärgläggningen](#). Ett färgtema omfattar följande:

- Textstilen, använd för syntaxfärgläggning via *förvalda stilegenskaper*. Exempelvis textfärgen och den markerade textfärgen.
- Textredigeringsområdets bakgrund, inklusive textmarkeringen och aktuell rad.
- Textområdets ikonkant: deras bakgrund, separationslinjen, radnumren, markörerna för radbrytning, de modifierade radmarkörerna och kodvikningen.
- Textdekorationer såsom sökmarkörerna, indenteringen och radmarkörer för tabulator och mellanslag, parentesmatchningen och stavningskontrollen.
- Bokmärken och textsnuttar

För att undvika förvirring, följande är omfattas inte:

- Teckensnittstypen och teckenstorleken.

- Textredigeringsprogrammets färger, såsom rullningslistens avbildning, menyerna, flikraden, fönsterfärgen, etc. I KDE-program, såsom Kate eller KDevelop definieras de här färgerna av **KDE Plasma globala färgschema**, som ställs in i modulen [Färger i systeminställningarna](#) eller från programmet själv i menyn **Inställningar** → **Färgschema**.

```

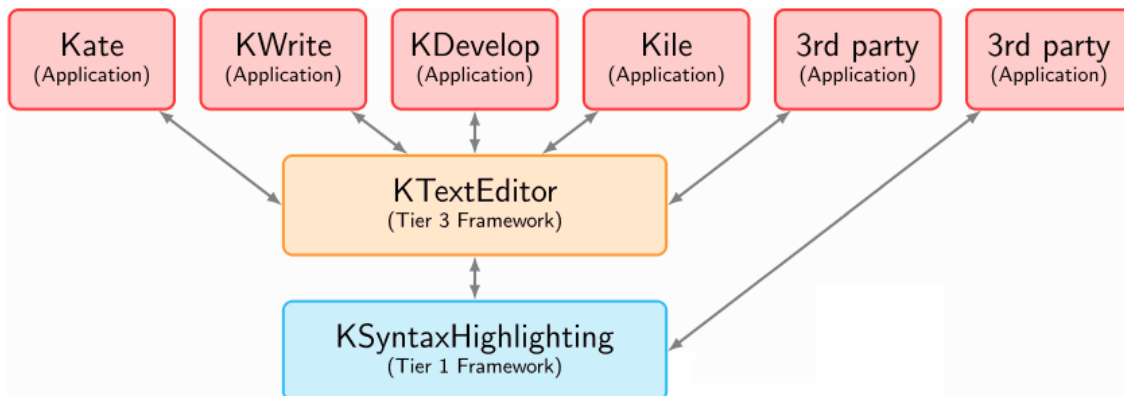
1  /**
2   * SPDX-FileCopyrightText: 2020 Christoph Cullmann <cullmann@kde.org>
3   * SPDX-License-Identifier: MIT
4   */
5
6  // BEGIN
7  #include <string>
8  #include <QString>
9  // END
10
11 /**
12  * Todo: improve documentation
13  * @param magicArgument some magic argument
14  * @return magic return 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 Light’ och ‘Breeze mörk’ färgteman med ‘C++’ syntaxfärgläggning.

6.3.2 Färgteman i KSyntaxHighlighting

Ramverket [KSyntaxHighlighting](#), som är gränssnittet för [syntaxfärgläggning](#), är biblioteket som **tillhandahåller och hanterar färgteman**. Det är en del av KDE Ramverk och används i KDE:s texteditorer såsom [Kate](#), [KWrite](#), [Kile](#) och [KDevelop](#). Beroendet ser ut på följande sätt:

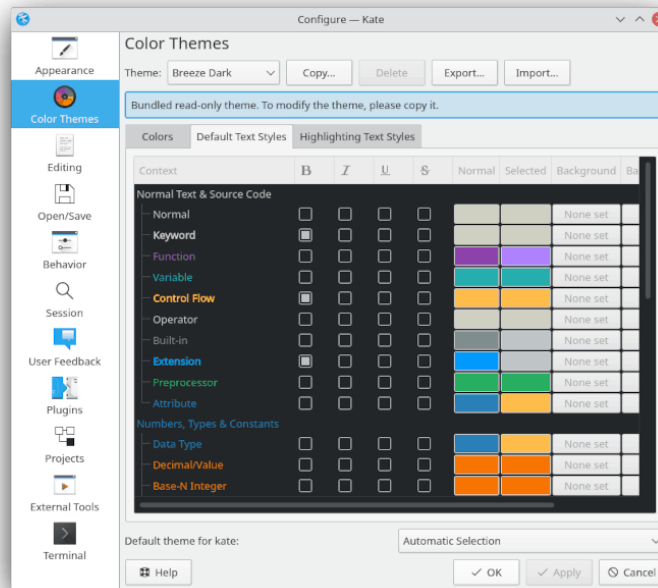


Beroende av texteditorer i KDE Ramverk biblioteken.

[KSyntaxHighlighting](#) inkluderar ett antal inbyggda teman som visas på sidan [Färgteman på Kate editorns webbplats](#).

Ramverket [KTextEditor](#), som är textredigeringsgränssnittet, tillhandahåller ett användargränssnitt för att skapa och redigera färgteman, inklusive ett verktyg för att exportera och importera

teman. Det är det enklaste sättet att skapa och redigera dem, och du kan komma åt det från dialogrutan 'Anpassa' i texteditorn. Mer detaljerad information finns i Avsnitt 6.3.5.



Grafiskt användargränssnitt för att hantera färgteman i Kates inställningar.

Det är viktigt att nämna att i KDE texteditorer som Kate eller KDevelop, används KSyntaxHighlighting färgteman sedan KDE Ramverk 5.75, utgivet 10:e oktober, 2020. Tidigare, användes Kates färgscheman (KConfig baserad schemainställning) och avråds nu från. Dock är det möjligt att konvertera de gamla Kate schema till KSyntaxHighlighting färgteman. Arkivet KSyntaxHighlighting inkluderar skriptet `utils/kateschema_to_theme_converter.py` och verktyget `utils/schema-converter/` i det syftet.

6.3.3 JSON-format för färgteman

6.3.3.1 Översikt

Färgteman lagras i filer med JSON-format, med filändelsen `.theme`.

I källkoden för KSyntaxHighlighting, finns JSON-filerna för inbyggda teman i katalogen `data/themes/`. Observera att i texteditorer kompileras inbyggda teman i biblioteket KSyntaxHighlighting, därför är sättet att komma åt dem via källkoden eller genom att exportera dem från det grafiska användargränssnittet för att hantera teman i KTextEditor.

Det är också möjligt att enkelt lägga till ytterligare eller egna teman, som läses in från filsystemet. Användaranpassade temafilerna finns i katalogen `org.kde.syntax-highlighting/themes/` i användarkatalogen, som hittas med kommandot `qtpaths --paths GenericDataLocation` och som oftast är `$HOME/.local/share/` och `/usr/share/`.

För Flatpak- och Snap-paket, fungerar inte ovanstående katalog eftersom dataplatsen är olika för varje program. I ett Flatpak-program, är platsen för anpassade temafilerna oftast `$HOME/.var/app/p/flatpak-paketnamn/data/org.kde.syntax-highlighting/themes/` och i ett Snap-program är platsen `$HOME/snap/snap-paketnamn/current/.local/share/org.kde.syntax-highlighting/themes/`.

På Windows® är filerna placerade i `%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\themes`. `%USERPROFILE%` expanderas oftast till `C:\Users\användarnamn`.

Sammanfattningsvis, för de flesta konfigurationer är den egna temakatalogen följande:

För lokal användare	\$HOME /.local/share/org.kde.syntax-highlighting/themes/
För alla användare	/usr/share/org.kde.syntax-highlighting/themes/
För Flatpak-paket	\$HOME /.var/app/ flatpak-paketnamn /data/org.kde.syntax-highlighting/themes/
För Snap-paket	\$HOME /snap/ snap-paketnamn /current/.local/share/org.kde.syntax-highlighting/themes/
På Windows®	%USERPROFILE%\AppData\Local\org.kde.syntax-highlighting\themes

Om flera temafilerna finns med samma namn, läses den med största **revision** in.

6.3.3.2 JSON-strukturen

Strukturen för en JSON-fil förklaras på [deras webbplats](#). I huvudsak består en JSON-formaterad fil av:

- Samlingar av nyckel-värde par, åtskilda av kommatecken och grupperade inom { }, som vi kallar 'objekt'.
- Ordade listor av värden, åtskilda av kommatecken och grupperade inom [], vilket vi kallar 'fält'.

Terminologin 'nyckel', 'värde', 'objekt' och 'fält' används i det här avsnittet. Om det är första gången du arbetar med JSON-filer, är det enkelt nog att förstå dem genom att titta på exemplen nedan.

6.3.3.3 Huvudsektionerna i JSON-färgtemafilerna

Färgtemats rotobjekt i JSON-filer innehåller följande schemanycklar:

- **metadata:** Det är nödvändigt. Värdet är ett objekt med temats metadata, såsom ett namn, revision och licens.
Det beskrivs detaljerat i Avsnitt [6.3.3.4](#).
- **editor-colors:** Det är nödvändigt. Värdet är ett objekt med färgerna i textredigeringsområdet, såsom bakgrunden, ikonkanten och textdekorationen.
Det beskrivs detaljerat i Avsnitt [6.3.4.1](#).
- **text-styles:** De är nödvändiga. Värdet är ett objekt med egenskapen *förvald textstil* för syntaxfärgläggningen. Varje egenskap definierar exempelvis sin *textfärg*, sin *markerad textfärg*, eller om den är *fetstil* eller *kursiv*. Textstilar kan refereras från [egenskaperna i syntaxdefinitionens XML-filer](#).
Det beskrivs detaljerat i Avsnitt [6.3.4.2](#).
- **custom-styles:** De är valfria. Definierar textstilar för egenskaperna av specifika syntaxfärgläggningsdefinitioner. Exempelvis, i en färgläggningsdefinition såsom **Python** eller **Markdown** kan man ange en annorlunda textstil som överskrider förval definierade av **text-styles**.
Det beskrivs detaljerat i Avsnitt [6.3.4.3](#).

JSON-språket stöder inte kommentarer. Dock kan du använda den valfria nyckeln `_comments` i rotobjektet för att skriva kommentarer. Om du exempelvis anpassar ett befintligt tema kan du lägga till webbadressen för originalarkivet. Det mest praktiska sättet är att använda ett fält av strängar.

Nedan finns en exempelfil för temat 'Breeze Light'. Du kanske märker att för att undvika att exemplet blir för stort, innehåller inte objekten `editor-colors` och `text-styles` alla nödvändiga nycklar. Du hittar hela arkivet av [temat Breeze Light i arkivet för KSyntaxHighlighting](#).

```
{
  "_comments": [
    "Det här är en kommentar.",
    "Om temat är en anpassning av ett annat, ange länken till ↵
      originalarkivet."
  ],
  "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",
    "Andra färgnycklar för editorn ..."
  },
  "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"
    },
    "Nycklar för andra textstilar..."
  }
}
```

```

    },
    "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 Metadata

JSON-objektet för nyckeln **metadata** innehåller relevant information om temat. Objektet har följande nycklar:

- **name:** Det är en *sträng* som anger språkets namn. Det visas senare i menyer och dialogrutor. Det är nödvändigt.
- **revision:** Det är ett *heltal* som anger aktuell version av temafilen. När du uppdaterar en färgtemafil, säkerställ att talet ökas. Det är nödvändigt.
- **license:** Det är en *sträng* som definierar temats licens, som använder identifieraren **SPDX-License-Identifier** från standarden [SPDX license communication format](#). Det är valfritt. Du kan se hela listan med SPDX-licensidentifierare [här](#).
- **copyright:** Det är ett *fält* av *strängar* som anger temats upphovsman, och använder identifieraren **SPDX-FileCopyrightText** från standarden [SPDX license communication format](#). Det är valfritt.

```

"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 Färger i detalj

Det här avsnittet beskriver alla tillgängliga färgegenskaper och tillgängliga färginställningar i detalj.

6.3.4.1 Editorns färger

Motsvarar färgerna i [textredigeringsområdet](#).

I [JSON-temafil](#), respektive nyckel **editor-colors** har ett värde, ett *object*, där varje nyckel-referens refererar till en egenskapsfärg i texteditorn. Här är **alla tillgängliga nycklar nödvändiga**, och deras värden är **strängar** med hexadecimala färgkoder, som '#00B5CF'.

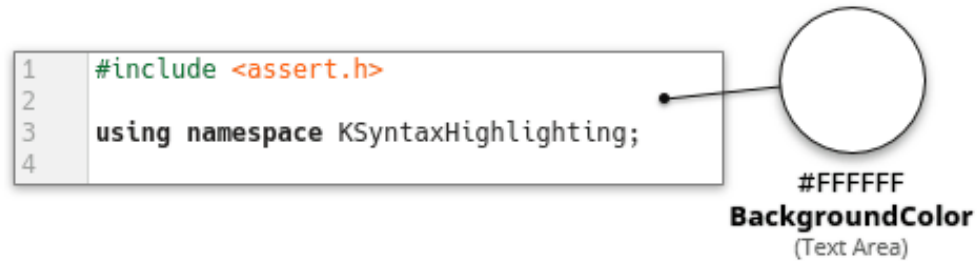
I [det grafiska användargränssnitt](#) för att hantera teman i [KTextEditor](#), kan egenskaperna ändras under fliken **Färger**.

Följande nycklar är tillgängliga: nycklarna använda i [JSON-filen](#) listas med *fetstil*, och namnen använda i [det grafiska användargränssnittet](#) visas inom parentes.

Editorns bakgrundsfärger

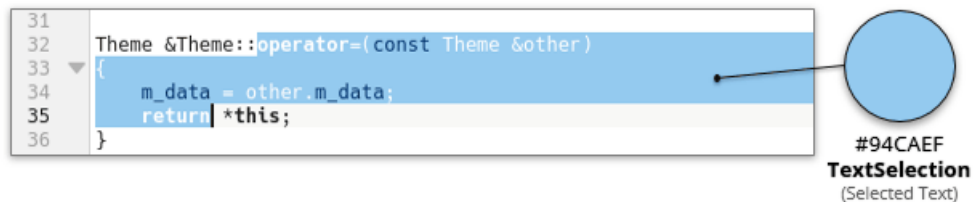
BackgroundColor (Textområde)

Det här är förvald bakgrund för redigeringsområdet. Det kommer att vara den dominerande färgen i redigeringsområdet.



TextSelection (Markerad text)

Det är bakgrunden för markerad text.



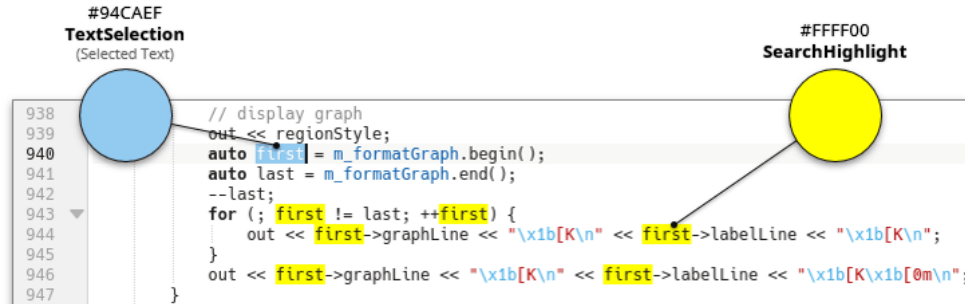
CurrentLine (Aktuell rad)

Ställ in färgen för aktuell rad. Att ställa in den något annorlunda jämfört med den normala textbakgrunden hjälper till att hålla fokus på den aktuella raden.



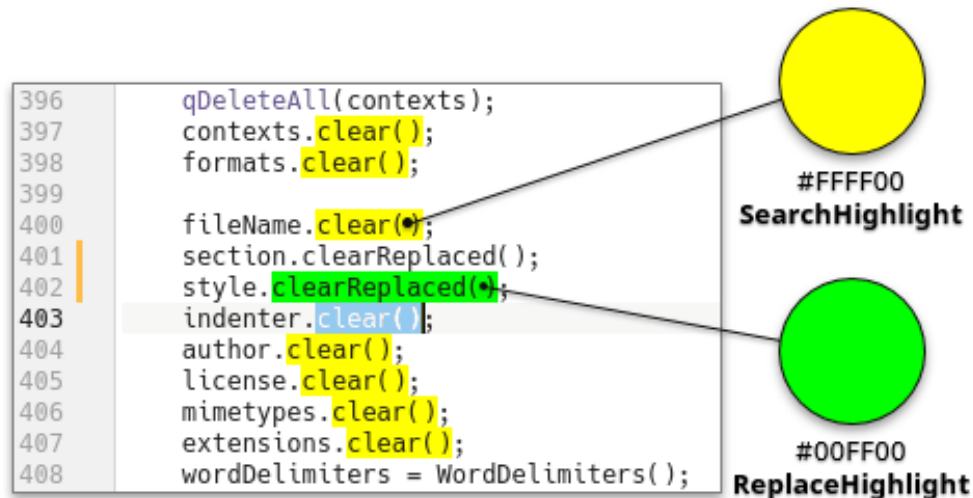
SearchHighlight (Sökfärgläggning)

Ställ in färgen på texten som matchar den senaste sökningen.



ReplaceHighlight (Ersättningsfärgläggning)

Ställ in färgen på texten som matchar den senaste ersättningsåtgärden.



Ikonkant

IconBorder (Bakgrundsområde)

Den här färgen används för markeringar, radnummer och vikmarkörkanter till vänster i redigeringsvyn när de visas.

LineNumbers (Radnummer)

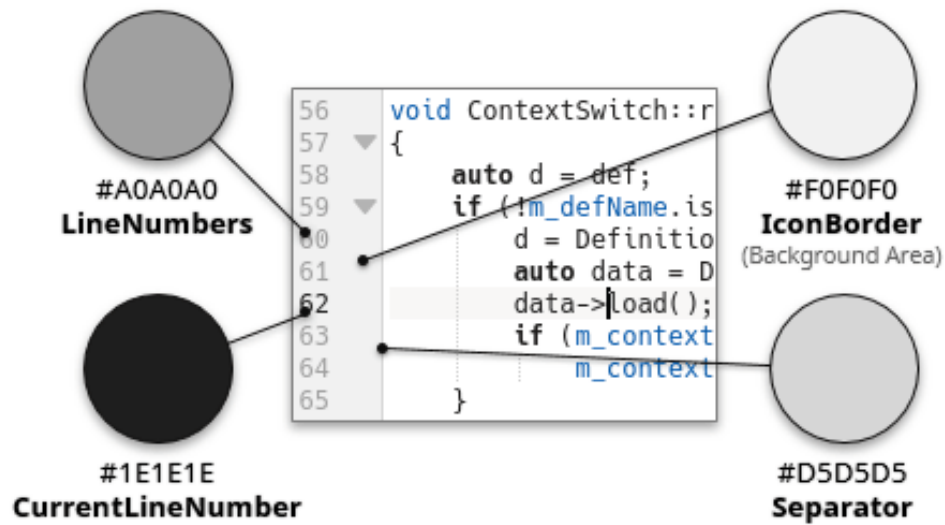
Den här färgen används för att visa radnummer till vänster i vyn när de visas.

CurrentLineNumber (Aktuellt radnummer)

Den här färgen används för att återge den aktuella radens radnummer, på vänster sida om vyn när det visas. Att ställa in det lite annorlunda från 'LineNumbers' hjälper till att fokusera på den aktuella raden.

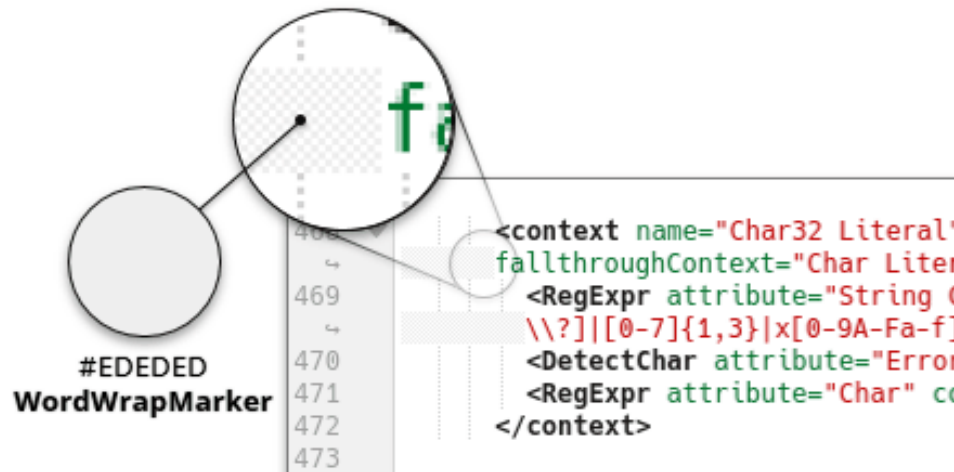
Separator (Separator)

Den här färgen används för att återge den vertikala linjen som separerar ikonkanten från textområdets bakgrund.



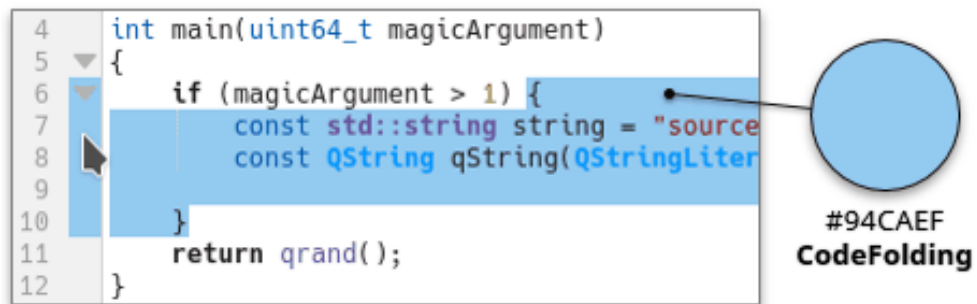
WordWrapMarker (Radbrytningsmarkör)

Den här färgen används för att rita ett mönster till vänster om rader med dynamisk radbrytning när de är vertikalt justerade, samt för statiska radbrytningsmarkörer.



CodeFolding (Kodvikning)

Den här färgen används för att markera delen av koden som skulle kodvikas när du klickar på kodvikningspilen till vänster om ett dokument. För mer information, se [dokumentationen av kodvikning](#).



ModifiedLines (Ändrade rader)

Den här färgen används för att färglägga rader som har ändrats men inte ännu sparats till vänster om ett dokument. För mer information se Avsnitt 3.9.

SavedLines (Sparade rader)

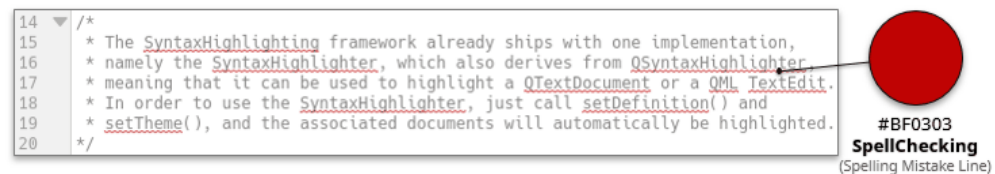
Den här färgen används för att färglägga rader som har ändrats under den här sessionen och sparats till vänster om ett dokument. För mer information se Avsnitt 3.9.



Textdekorationer

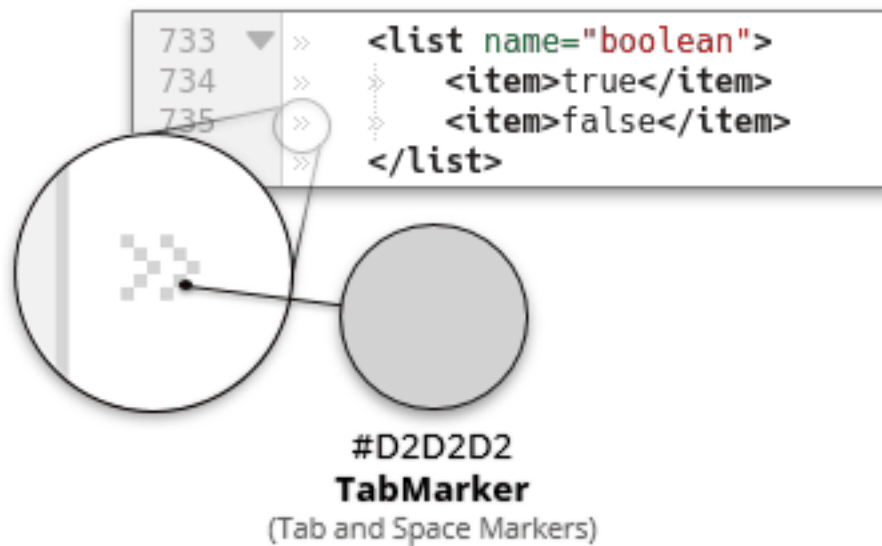
SpellChecking (Rad med stavfel)

Den här färgen används för att indikera stavfel.



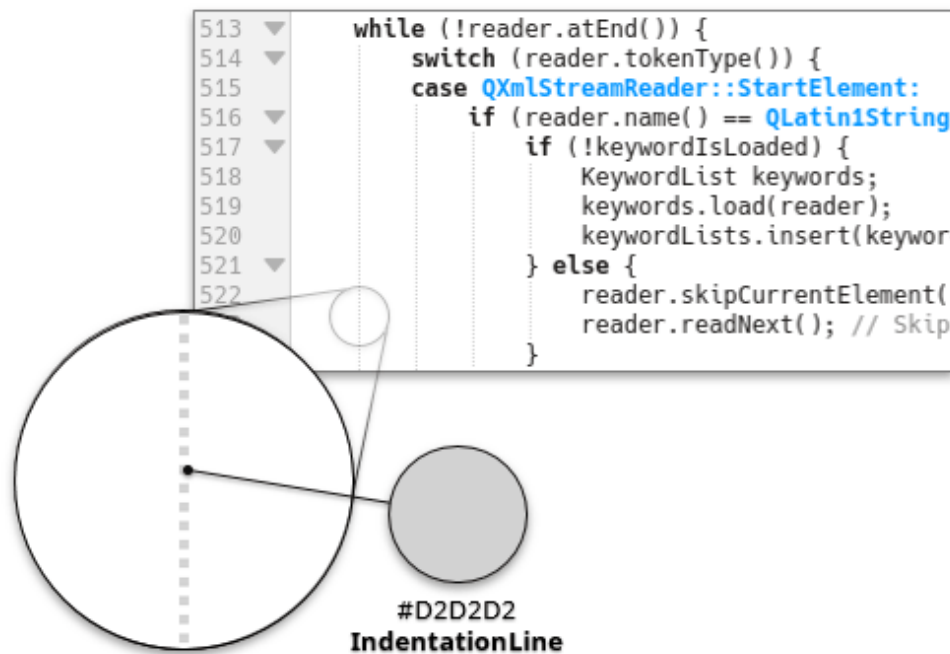
TabMarker (Tabulator- och mellanslagsmarkörer)

Den här färgen används för att visa indikering av blanktecken, när de är aktiverade.



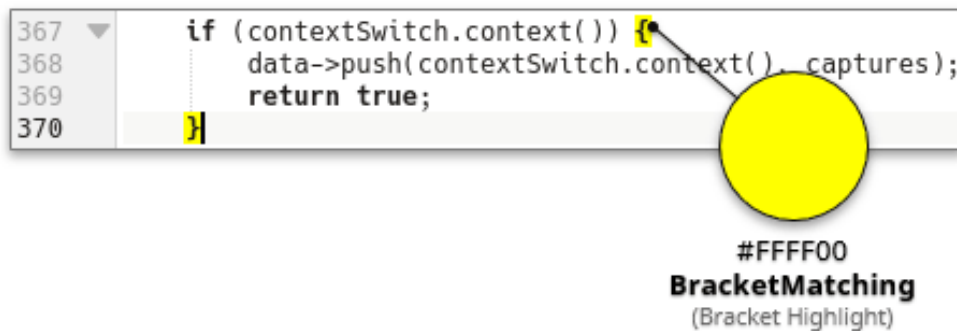
IndentationLine (Indenteringslinje)

Den här färgen används för att rita en linje till vänster om indenterade block, om funktionen är aktiverad.



BracketMatching (Färglägg parenteser)

Den här färgen används för att markera bakgrunden för matchande parenteser.



Markeringsfärger

MarkBookmark (Bokmärke)

Den här färgen används för att ange bokmärken. Observera att färgen har ogenomskinlighet 22 % (33 % för den aktuella raden). För mer information, se [Avsnitt 3.6](#).



MarkBreakpointActive (Aktiv brytpunkt)

Den här färgen används av GDB-insticksprogrammet för att ange en aktiv brytpunkt. Observera att färgen är ogenomskinlig mot bakgrunden. För mer information, se [dokumentationen av GDB-insticksprogrammet](#).

MarkBreakpointReached (Nådd brytpunkt)

Den här färgen används av GDB-insticksprogrammet för att ange en brytpunkt du har nått under avlusning. Observera att färgen är ogenomskinlig mot bakgrunden. För mer information, se [dokumentationen av GDB-insticksprogrammet](#).

MarkBreakpointDisabled (Inaktiv brytpunkt)

Den här färgen används av GDB-insticksprogrammet för att ange en inaktiv brytpunkt. Observera att färgen är ogenomskinlig mot bakgrunden. För mer information, se [dokumentationen av GDB-insticksprogrammet](#).

MarkExecution (Körning)

Den här färgen används av GDB-insticksprogrammet för raden som för närvarande körs. Observera att färgen är ogenomskinlig mot bakgrunden. För mer information, se [dokumentationen av GDB-insticksprogrammet](#).

MarkWarning (Varning)

Den här färgen används av bygginsticksprogrammet för att ange en rad som har orsakat en kompilatorvarning. Observera att färgen är ogenomskinlig mot bakgrunden. För mer information, se [dokumentationen av bygginsticksprogrammet](#).

MarkError (Fel)

Den här färgen används av bygginsticksprogrammet för att ange en rad som har orsakat ett kompilatorfel. Observera att färgen är ogenomskinlig mot bakgrunden. För mer information, se [dokumentationen av bygginsticksprogrammet](#).

Textmallar och snuttar

TemplateBackground (Bakgrund)

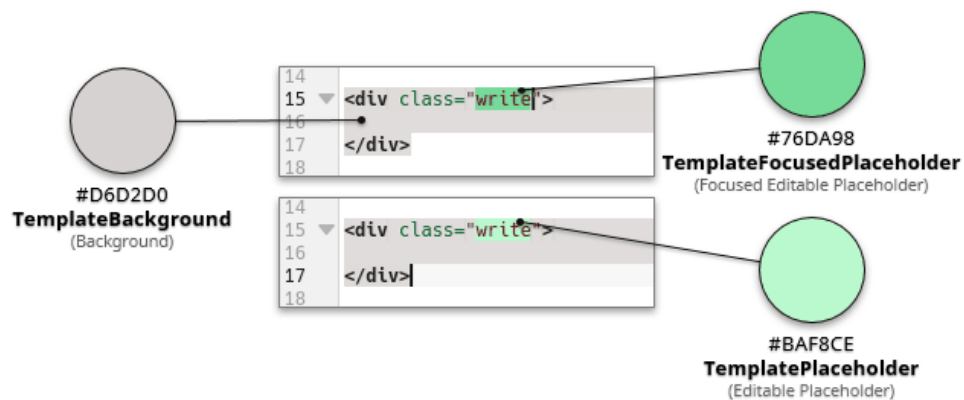
Den här färgen används av Kates insticksprogram för textsnuttar för att markera bakgrunden för en textsnutt. För mer information, se [dokumentationen av Kates textsnuttar](#).

TemplatePlaceholder (Redigerbar platsmarkör)

Den här färgen används av Kates insticksprogram för textsnuttar för att markera en platsmarkör som du kan klicka för att redigera manuellt. För mer information, se [dokumentationen av Kates insticksprogram för textsnuttar](#).

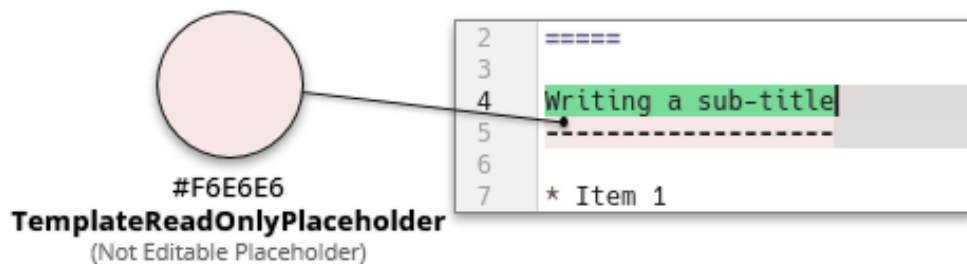
TemplateFocusedPlaceholder (Fokuserad redigerbar platsmarkör)

Den här färgen används av Kates insticksprogram för textsnuttar för att markera en platsmarkör som du för närvarande redigerar. För mer information, se [dokumentationen av Kates insticksprogram för textsnuttar](#).



TemplateReadOnlyPlaceholder (Icke-redigerbar platsmarkör)

Färgen används av Kates insticksprogram för textsnuttar för att markera en platsmarkör som inte kan redigeras manuellt, såsom en som fylls i automatiskt. För mer information, se [dokumentationen av Kates insticksprogram för textsnuttar](#).



6.3.4.2 Förvalda textstilar

De förvalda textstilarna ärvs av färgläggningstextstilarna, vilket gör det möjligt för editorn att visa text på ett mycket konsekvent sätt, till exempel använder kommentartext samma stil i nästan alla textformat som KSyntaxHighlighting kan färglägga.

NOT

Textstilarna kan refereras till från **default styles** använda i XML-definitionerna för [syntaxfärgläggning](#), exempelvis egenskapen 'Normal' är ekvivalent med 'dsNormal' i XML-filer, och 'DataType' är ekvivalent med 'dsDataType'. Se Avsnitt 6.2.3.3 i dokumentationen av syntaxfärgläggning.

TIPS

Säkerställ att läsbara färger med god kontrast väljes, särskilt i kombination med **editorns färg**. Se Avsnitt [6.3.6.1](#).

I [JSON-filen](#) har respektive **text-styles** nyckel värdet *object* där varje nyckel motsvarar namnet på en *förvald textstil*, som är ekvivalent med de som används i syntaxfärgläggningsdefinitioner. Här är **alla tillgängliga nycklarna för textstilar nödvändiga**, och de listas nedan.

```
"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"
  },
  Nycklar för andra textstilar...
}
```

Varje nyckel för en *förvald textstil* har ett JSON-objekt som sitt värde, där värden såsom *color*, *bold*, *italic*, etc. anges. Nycklarna är följande:

text-color: Det är en *sträng* med textfärg i hexadecimal färgkod. Det här nyckel/värdet är nödvändigt.

selected-text-color: Textfärg när den är markerad är i allmänhet samma värde som 'text-color'. När texten är markerad, definieras bakgrunden av värdet [TextSelection](#) i [editorns färger](#), så du måste säkerställa att texten har god kontrast och är läsbar med den bakgrunden. Värdet är en *sträng* med en hexadecimal färgkod. Det här nyckel/värdet är nödvändigt.

bold: Det är en *boolean* som bestämmer om texten är i fetstil. Nyckeln är valfri, och förvalt värde är **false**.

italic: Det är en *boolean* som bestämmer om texten är kursiv. Nyckeln är valfri, och förvalt värde är **false**.

underline: Det är en *boolean* som bestämmer om texten är understruken. Nyckeln är valfri, och förvalt värde är **false**.

strike-through: Det är en *boolean* som bestämmer om texten är överstruken. Nyckeln är valfri, och förvalt värde är **false**.

background-color: Bestämmer textens bakgrund, exempelvis används för varningar i kommentarer. Värdet är en *sträng* med en hexadecimal färgkod. Nyckeln är valfri, normalt finns ingen bakgrund.

selected-background-color: Bestämmer textens bakgrund när den är markerad. Värdet är en *sträng* med en hexadecimal färgkod. Nyckeln är valfri, normalt finns ingen bakgrund.

De här egenskaperna kan ändras i [det grafiska användargränssnittet för att hantera färgteman i KTextEditor](#), under fliken **Förvalda textstilar**. Namnet i listan över stilar använder stilen inställd för det objektet, vilket ger dig en omedelbar förhandsgranskning när en stil anpassas. Varje stil låter dig välja vanliga egenskaper samt förgrunds- och bakgrundsfärger. För att välja bort en bakgrundsfärg, högerklicka för att använda den sammanhangsberoende menyn.

Följande nycklar för textstil är tillgängliga: nycklarna använda i [JSON-filen](#) listas med *fetstil*, och namnen använda i [det grafiska användargränssnittet](#) visas inom parentes om de är annorlunda.

Normal text och källkod

Normal: Förvald textstil för normal text och källkod utan speciell färgläggning.

Keyword: Textstil för inbyggda nyckelord i språket.

Function: Textstil för funktionsdefinitioner och funktionsanrop.

Variable: Textstil för variabler, om tillämplig. Exempelvis variabler i PHP/Perl typiskt börjar med \$, så alla identifierare som följer mönstret **\$exempel** färgläggs som variabler.

ControlFlow (Kontrollflöde): Textstil för kontrollflödesnyckelord, såsom *if, then, else, return, switch, break, yield, continue*, etc.

Operator: Textstil för operatörer, såsom *+, -, *, / , %*, etc.

BuiltIn (Inbyggda): Textstil för inbyggda språkklasser, funktioner och objekt.

Extension: Textstil för välkända ändelser såsom QtTM-klasser och funktioner eller makron i C++ och Python eller boost.

Preprocessor: Textstil för preprocessorsatser eller makrodefinitioner.

Attribute: Textstilar för anmärkningar eller attribut för funktioner eller objekt, t.ex. `@override` i Java, eller `__declspec(...)` och `__attribute__((...))` i C++.

Tal, typer och konstanter

DataType (Datatyp): Textstilar för inbyggda datatyper såsom *int, char, float, void, u64*, etc.

DecVal (Decimal/Värde): Textstil för decimala värden.

BaseN (Bas-N heltal): Textstil för nummer med en bas som skiljer sig från 10.

Float (Flyttal): Textstil för flyttal.

Constant: Textstil för språkkonstanter och användardefinierade konstanter, t.ex. *True, False, None* i Python eller *nullptr* i C/C++, eller matematiska konstanter som *PI*.

Strängar och tecken

Char (Tecken): Textstil för enstaka tecken såsom *'x'*.

SpecialChar (Specialtecken): Textstil för undantagna tecken i strängar, t.ex. *'hej\n'*, och andra tecken med specialmening i strängar, såsom ersättningar eller operatörer i reguljära uttryck.

String: Textstil som strängar såsom *'hej allesamman'*.

VerbatimString (Ordagrann sträng): Textstil för ordagranna eller obehandlade strängar som *'obehandlad \bakstreck'* i Perl, CoffeeScript och skal, samt *r'\obehandlad'* i Python, eller sådant som HERE-dokument.

SpecialString (Specialsträng): Textstil för specialsträng, såsom reguljära uttryck i ECMAScript, L^AT_EX matematikläge, SQL, etc.

Import (Importer, Moduler, Inkluderade filer): Textstil för inkluderade filer, importer, moduler eller L^AT_EX-paket.

Kommentarer och dokumentation

Comment: Textstil för normala kommentarer.

Dokumentation: Textstil för kommentarer som anger dokumentation av programmeringsgränssnitt, såsom */** doxygen-kommentarer */* eller *"""docstrings"""*.

Annotation: Textstilar för anmärkningar i kommentarer eller dokumentationskommandon, såsom `@param` i Doxygen eller JavaDoc.

CommentVar (Kommentarvariabel): Textstil som refererar till variabelnamn använda i kommandona ovan i en kommentar, såsom `exempel` i `'@param exempel'`, i Doxygen eller JavaDoc.

RegionMarker (Områdesmarkering): Textstil för områdesmarkeringar, ofta definierade av `/ /BEGIN` och `/ /END` i kommentarer.

Information: Textstil för information, anteckningar och tips, såsom nyckelordet `@note` i Doxygen.

Warning: Textstil för varningar, såsom nyckelordet `@warning` i Doxygen.

Alert: Textstil för speciella ord i kommentarer, såsom `TODO`, `FIXME`, `XXXX` och `WARNING`.

Diverse

Error: Textstil som anger felmarkering och felaktig syntax.

Others: Textstil för egenskaper som inte motsvarar någon av de andra förvalda stilarna.

6.3.4.3 Egna färgläggningstextstilar

Här kan du etablera textstilar för en specifik syntaxfärgläggningsdefinition, och överskrida **förvald textstil** beskriven i [föregående avsnitt](#).

i [JSON-temafilen](#) motsvarar det nyckeln `custom-styles` vars värde är ett *objekt* där varje delschema nyckel motsvarar **namnet på en syntaxfärgläggningsdefinition**. Dess värde är ett *objekt* där en nyckel refererar till **stilens egenskapsnamn** definierat i [elementen itemData](#) i XML-filens syntaxfärgläggning, och respektive värde är ett delobjekt med nycklarna `text-color`, `selected-text-color`, `bold`, `italic`, `underline`, `strike-through`, `background-color` och `selected-background-color`, definierade i [föregående avsnitt](#). Vart och ett av värdena är valfritt, eftersom om de inte anges, beaktas stilen angiven i **text-styles**.

Exempelvis, i det här kodavsnittet, har 'ISO C++' syntaxfärgläggningsdefinition en särskild textstil för egenskaperna 'Type Modifiers' och 'Standard Classes'. I motsvarande XML-fil 'isocpp.xml' använder den definierade egenskapen 'Standard Classes' förvald stil **BuiltIn** (eller `dsBuiltIn`). I denna egenskap överskrids bara värdet **text-color** av den nya färgen '#6431b3'.

```
"custom-styles": {
  "ISO C++": {
    "Standard Classes": {
      "text-color": "#6431b3"
    },
    "Type Modifiers": {
      "bold": true,
      "selected-text-color": "#009183",
      "text-color": "#00b5cf"
    }
  }
}
```

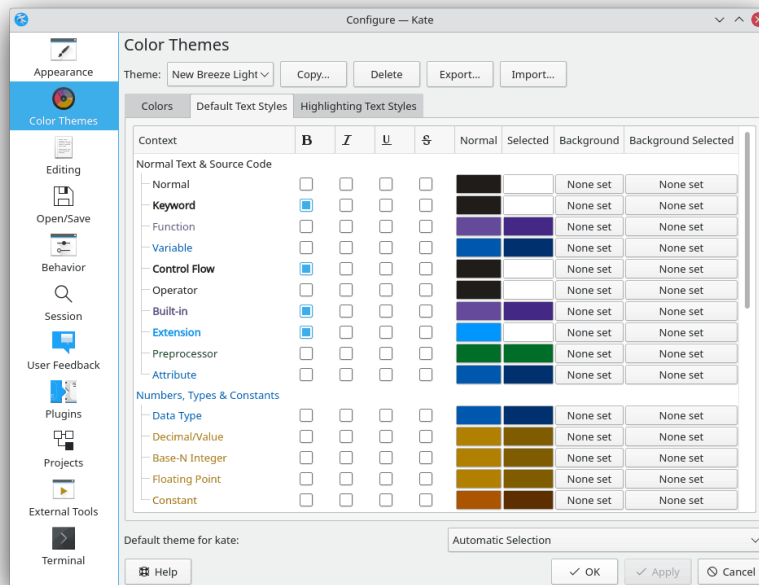
NOT

- Du bör ta hänsyn till att de här textstilarna hör ihop med egenskapsnamnen definierade i XML-filerna för syntaxfärgläggning. Om en XML-fil uppdateras och namnet ändras på vissa egenskaper eller de tas bort, gäller inte längre den egna stilen definierad i temat.
- Syntaxfärgläggningsdefinitioner inkluderar ofta andra definitioner. Exempelvis inkluderar 'QML'-färgläggningen 'JavaScript'-färgläggning, eftersom de delar funktionaliteten i färgläggning.

I det grafiska användargränssnitt för att hantera teman i **KTextEditor**, kan egenskaperna ändras under fliken **Färgläggningstextstilar**. Normalt väljer editorn färgläggningen av det aktuella dokumentet i förväg. Du kommer att märka att många färgläggningar innehåller andra färgläggningar som representeras av grupper i stillistan. Exempelvis importerar de flesta färgläggningar 'Alert', och många källkodsformat importerar 'Doxygen'.

6.3.5 Det grafiska användargränssnittet för färgteman

Det enklaste sättet att skapa och redigera färgteman är via det grafiska användargränssnittet i dialogrutan 'Anpassa' som tillhandahålls av **KTextEditor**. För att komma åt det, välj **Inställningar** → **Anpassa program...** i texteditorns menyrad. Det visar dialogrutan **Anpassa**. Välj där **Färgteman** i sidopanelen.



Kates inställningsdialogruta med färgtemahantering.

I den här dialogrutan kan du ställa in alla färgerna i de teman du har, samt skapa/kopiera nya teman, ta bort dem, exportera dem till **.theme**-filer med **JSON-format** eller importera dem från externa **.theme**-filer. Varje tema har inställningar för textfärger och stilar.

Inbyggda teman kan normalt inte ändras. För att göra det måste du kopiera dem och ge dem nya namn.

För att använda ett tema permanent i din texteditor måste du välja det i kombinationsrutan med beteckningen **Standardtema för program** längst ner i dialogrutan och klicka på **Verkställ** eller **Ok**. Normalt är alternativet **Automatiskt val** aktivt, vilket väljer ett lämpligt färgtema enligt det **KDE Plasma färgschema** som används i textredigeringsprogrammet. Det väljer oftast mellan 'Breeze Light' och 'Breeze mörk' beroende på om schemat är ljus eller mörkt.

TIPS

Du kan justera KDE:s globala färgschema i modulen **Färger i systeminställningarna**. Du kan också ändra det individuellt i vissa program, såsom Kate eller KDevelop, i menyn **Inställningar** → **Färgschema**.

6.3.5.1 Skapa ett nytt tema

För att skapa ett nytt tema är det först nödvändigt att kopiera ett befintligt. Välj ett befintligt tema som du vill använda som grund, såsom 'Breeze Light' eller 'Breeze mörk', och klicka på **Kopiera**. Skriv sedan in ett namn på det nya temat.

Om du vill ändra en inbyggt eller skrivskyddat tema, måste det först kopieras till ett annat namn.

6.3.5.2 Importera eller exportera JSON-temafil

Du kan exportera ett valt tema (inklusive inbyggda) till en **JSON-fil** med filändelsen **.theme**, med knappen **Exportera**. Det visar en dialogruta för att spara filen. För att lägga till ett färgtema från en extern **JSON-fil**, klicka helt enkelt på knappen **Importera** och välj **.theme**-filen i dialogrutan.

TIPS

- Som nämnts ovan, lagras användaranpassade temafil i katalogen `org.kde.syntax-highlighting/themes/`. När du kopierar eller skapar ett tema hamnar det automatiskt där. Dessutom, är import eller tillägg av ett tema ekvivalent med att kopiera en extern **.theme**-fil till katalogen. KSyntaxHighlighting hämtar automatiskt färgtemafil från den här katalogen.
- Om du vill publicera ett tema du skapar, är det ytterst viktigt att kontrollera **metadata**-objektet i **JSON-filen**, lägga till respektive licens och kontrollera revisionsnumret.

6.3.5.3 Redigera färgteman

6.3.5.3.1 Färger

Här justerar färgerna i textredigeringsområdet. De beskrivs detaljerat i Avsnitt 6.3.4.1.

6.3.5.3.2 Förvalda textstilar

De förvalda textstilarna ärvs av färgläggningstextstilarna, vilket gör det möjligt för editorn att visa text på ett mycket konsekvent sätt, till exempel använder kommentartext samma stil i nästan alla textformat som KSyntaxHighlighting kan färglägga.

Namnet i listan med stilar använder stilen som är inställd för objektet, vilket ger en omedelbar förhandsgranskning när en stil ställs in.

Varje stil låter dig välja gemensamma egenskaper samt förgrunds- och bakgrundsfärger. För att inte använda en bakgrundsfärg, högerklicka och använd den sammanhangsberoende menyn.

Det här områdets egenskaper beskrivs detaljerat i Avsnitt 6.3.4.2.

6.3.5.3.3 Färgläggningstextstilar

Här kan du redigera textstilarna som används av en viss färgläggningsdefinition. Editorn väljer färgläggningen som används av aktuellt dokument i förväg. För att arbeta med en annan färgläggning, välja en i kombinationsrutan ovanför listan med stilar.

Namnet i listan med stilar använder stilen som är inställd för objektet, vilket ger en omedelbar förhandsgranskning när en stil ställs in.

Varje stil låter dig välja gemensamma egenskaper samt förgrunds- och bakgrundsfärger. För att inte använda en bakgrundsfärg, högerklicka och använd den sammanhangsberoende menyn.

Dessutom kan du se om en stil är samma som den förvalda stilen för objektet, och ställa in den till det om den inte är det.

Du märker att många färgläggningar innehåller andra färgläggningar representerade av grupper i stillistan. De flesta färgläggningar importerar till exempel färgläggningen Alerts, och många källkodsformat importerar färgläggningen Doxygen. Redigering av färger i grupperna påverkar bara stilarna när de används i det redigerade färgläggningsformatet.

6.3.6 Tips och trick

6.3.6.1 Textfärgernas kontrast

En viktig aspekt vid arbete med färgteman är att välja en textkontrast som gör det lättare att läsa, särskilt i kombination med bakgrunden.

Programmet **Kontrast** är ett verktyg för kontroll av färgkontrast. Det talar om för dig att kombinationen av textfärg och bakgrundsfärg är läsbara och handikappanpassade, så det är ett utmärkt verktyg för att hjälpa dig skapa färgteman.

Du kan ladda ner **Kontrast** från [webbplatsen för KDE program](#) eller från [Flatpak-paketet på Flathub](#) (bara i GNU/Linux).

GNOME-programmet **Contrast** är liknande. Du kan ladda ner [Flatpak-paketet på Flathub](#) (bara på GNU/Linux).

6.3.6.2 Förslag för konsekvent syntaxfärgläggning

KSyntaxHighlighting inkluderar [mer än 300 syntaxfärgläggningsdefinitioner](#), och därför är det idealiskt att du säkerställer att ditt nya tema ser bra ut i alla syntaxfärgläggningsdefinitioner. De inbyggda färgteman har följande likheter som det rekommenderas (men inte krävs) att du följer för att åstadkomma korrekt visning av alla syntaxfärgläggningsdefinitioner:

- Använd fetstil för [textstilarna](#) 'Keyword' och 'ControlFlow'.
- Använd inte bakgrundsfärg i någon [textstil](#), utom 'Alert' och 'RegionMarker'.

De flesta av syntaxfärgläggningarna är avsedda att se bra ut med standardteman 'Breeze Light' och 'Breeze mörk', därför är ett annat sätt att behålla likformighet att använda liknande färger [textstilar](#), som *grön* för 'Preprocessor' och 'Others', *blå* för 'DataType' och 'Attribute', eller *violet* för 'Function'.

Observera att rekommendationerna inte är obligatoriska när ett tema skapas och publiceras.

6.4 Skapa skript med JavaScript

KateParts editorkomponent kan enkelt utökas genom att skriva skript. Skriptspråket är ECMAScript (känt överallt som JavaScript). KatePart stöder två sorters skript: indenterings- och kommandoradskript.

6.4.1 Indenteringsskript

Indenteringsskript, som också kallas indenterare, indenterar automatiskt källkoden medan texten skrivs in. Som ett exempel, efter att ha tryckt på returtangenten ökar ofta indenteringsnivån.

Följande avsnitt beskriver steg för steg hur skelettet för en enkel indenterare skapas. Som ett första steg, skapa en ny *.js-fil, som t.ex. heter javascript.js i den lokala hemkatalogen \$XDG_DATA_HOME /katepart5/script/indentation. I detta expanderas miljövariabeln XDG_DATA_DIRS typiskt antingen som ~/.local eller ~/.local/share.

På Windows® är filerna placerade i %USERPROFILE%\AppData\Local\katepart5\script\indentation. %USERPROFILE% expanderas oftast till C:\\Users\\användare.

6.4.1.1 Indenteringsskriptets huvud

Huvudet i filen javascript.js är inbäddat som JSON i början av dokumentet enligt följande:

```
var katescript = {
  "name": "JavaScript",
  "author": "Exempelnamn <exempel.namn@någon.adress.org>",
  "license": "BSD-licens",
  "revision": 1,
  "kate-version": "5.1",
  "required-syntax-style": "javascript",
  "indent-languages": ["javascript"],
  "priority": 0,
}; // kate-script-header, måste finnas i början av filen utan kommentarer
```

Varje post förklaras nu i detalj:

- name [krävs]: Detta är indenterarens namn som visas i menyn **Verktyg** → **Indentering** och i inställningsdialogrutan.
- author [valfri]: Upphovsmannens namn och kontaktinformation.
- license [valfri]: Kort form av licensen, såsom BSD-licensen eller LGPLv3.
- revision [krävs]: Skriptets utgåva. Numret ska ökas så fort skriptet ändras.
- kate-version [krävs]: Lägsta version av KatePart som krävs.
- required-syntax-style [valfri]: Syntaxstilen som krävs, som matchar angiven stil i syntaxfärgläggningsfiler. Det är viktigt för indenterare som förlitar sig på specifik syntaxfärgläggningsinformation i dokumentet. Om en krävd syntaxstil specificeras, är bara indenteraren tillgänglig när lämplig syntaxfärgläggning är aktiv. Det förhindrar 'odefinierat beteende' orsakat av att använda indenteraren utan den förväntade färgläggningsmetoden. Exempelvis använder sig Ruby-indenteraren sig av det i filen ruby.js och ruby.xml.
- indent-languages [valfri]: JSON-fält av syntaxstilar som indenteraren kan indentera korrekt, t.ex. ["c++", "java"].
- priority [valfri]: Om flera indenterare är lämpliga för en viss färglagd fil, bestämmer prioriteten vilken indenterare som väljes som standardindenterare.

6.4.1.2 Indenteringens källkod

Efter att ha specificerat huvudet, förklarar det här avsnittet hur själva indenteringsskriptet fungerar. Det grundläggande skelettet för implementeringen ser ut så här:

```
// nödvändiga katepart js bibliotek, t.ex. range.js om Range används
require ("range.js");

triggerCharacters = "{}/;";
function indent(line, indentWidth, ch)
```

```
{
    // anropas för varje nyrad (ch == '\n') och alla tecken angivna i
    // den globala variabeln triggerCharacters. Vid anrop av ↵
    VerktygFormatera indentering
    // är variabeln ch tom, dvs. ch == ''.
    //
    // se också: Programmeringsgränssnittet för skripthantering
    return -2;
}
```

Funktionen `indent()` har tre parametrar:

- `line`: raden som ska indenteras
- `indentWidth`: indenteringsbredden i antal mellanslag
- `ch`: antingen ett nyradstecken (`ch == '\n'`), avtryckartecknet angivet i `triggerCharacters` eller tom om användaren utförde åtgärden **Verktyg** → **Formatera indentering**.

Returvärde från funktionen `indent()` anger hur raden kommer att indenteras. Om returvärdet är ett enkelt heltal, tolkas det på följande sätt:

- returvärde `-2`: gör ingenting
- returvärde `-1`: behåll indentering (söker efter föregående icke tomma rad)
- returvärde `0`: tal ≥ 0 anger indenteringsdjupet i mellanslag

Som alternativ kan ett fält med två element returneras.

- `return [ indent, align ];`

I detta fall är det först elementet indenteringsdjupet ovan med samma betydelse för specialvärden. Det andra elementet är dock ett absolutvärde som representerar en kolumn för 'justering'. Om värdet är större än indenteringsvärdet, representerar skillnaden ett antal mellanslag att lägga till efter indentering av den första parametern. Annars ignoreras det andra numret. Att använda tabulator och mellanslag för indentering kallas ofta 'blandat läge'.

Betrakta följande exempel: Antag att tabulator används för indentering, och tabulatorbredden är inställd till 4. Här representerar `<tab>` en tabulator, och `'.'` ett mellanslag.

```
1: <tab><tab>anropa ("Hej",
2: <tab><tab>....."allesamman");
```

Vid indentering av rad 2, returnerar funktionen `indent()` [8, 15]. Som resultat infogas två tabulatorer för att indentera till kolumn 8, och 7 mellanslag läggs till för att justera den andra parametern under den första, så att den förblir justerad om filen visas med en annan tabulatorbredd.

En standardinstallation av KDE levererar KatePart med flera indenterare. Den motsvarande JavaScript-koden finns i `$XDG_DATA_DIRS /katepart5/script/indentation`.

På Windows® är filerna placerade i `%USERPROFILE%\AppData\Local\katepart5\script\indentation`. `%USERPROFILE%` expanderas oftast till `C:\Users\användare`.

Att utveckla en indenterare kräver att skript läses in igen för att se om ändringar betär sig lämpligt. Istället för att starta om programmet, byt helt enkelt till kommandoraden och utför kommandot **reload-scripts**.

Om du utvecklar användbara skript, fundera på att bidra med det till KatePart-projektet genom att [kontakta e-postlistan](#).

6.4.2 Kommandoradskript

Eftersom det är svårt att uppfylla allas behov, stöder KatePart små hjälpverktyg för snabb text-behandling via den [inbyggda kommandoraden](#). Exempelvis är kommandot **sort** implementerat som ett skript. Det här avsnittet förklarar hur man skapar *.js-filer för att utöka KatePart med godtyckliga hjälpskript.

Kommandoradskript är placerade i samma katalog som indenteringsskript. Som ett första steg, skapa alltså en ny *.js-fil som heter `mina_verktyg.js` i den lokala hemkatalogen `$XDG_DATA_HOME /katepart5/script/commands`. I detta expanderas miljövariabeln `XDG_DATA_DIRS` typiskt antingen som `~/.local` eller `~/.local/share`.

På Windows® är filerna placerade i `%USERPROFILE%\AppData\Local\katepart5\script\commands`. `%USERPROFILE%` expanderas oftast till `C:\\Users\\användare`.

6.4.2.1 Kommandoradskriptets huvud

Huvudet för varje kommandoradskript är inbäddat som JSON i början av skriptet enligt följande:

```
var katescript = {
  "author": "Exempelnamn <exempel.namn@någon.adress.org>",
  "license": "LGPLv2+",
  "revision": 1,
  "kate-version": "5.1",
  "functions": ["sort", "moveLinesDown"],
  "actions": [
    { "function": "sort",
      "name": "Sortera markerad text",
      "category": "Editing",
      "interactive": "false"
    },
    { "function": "moveLinesDown",
      "name": "Flytta rader neråt",
      "category": "Editing",
      "shortcut": "Ctrl+Shift+Down",
      "interactive": "false"
    }
  ]
}; // kate-script-header, måste finnas i början av filen utan kommentarer
```

Varje post förklaras nu i detalj:

- **author** [valfri]: Upphovsmannens namn och kontaktinformation.
- **license** [valfri]: Kort form av licensen, såsom BSD-licensen eller LGPLv2.
- **revision** [krävs]: Skriptets utgåva. Numret ska ökas så fort skriptet ändras.
- **kate-version** [krävs]: Lägsta version av KatePart som krävs.
- **functions** [krävs]: JSON-fält av kommandon i skriptet.
- **actions** [valfri]: Ett JSON-fält av JSON-objekt som definierar åtgärderna som visas i programmenyn. Detaljerad information tillhandahålls i avsnittet [Tilldela genvägar](#).

Eftersom värdet på **functions** är ett JSON-fält, kan ett enda skript innehålla godtyckligt antal kommandon för kommandoraden. Varje funktion är tillgänglig via KateParts [inbyggda kommandorad](#).

6.4.2.2 Skriptets källkod

Alla funktioner specificerade i huvudet måste implementeras i skriptet. Exempelvis måste skriptfilen i exemplet ovan implementera de två funktionerna **sort** och **moveLinesDown**. Alla funktioner har följande syntax:

```
// nödvändiga katepart.js bibliotek, t.ex. range.js om Range används
require ("range.js");

function <namn>(arg1, arg2, ...)
{
    // ... implementering, se också: Programmeringsgränssnittet för ↔
    skripthantering
}
```

Argument på kommandoraden skickas till funktionen som *arg1*, *arg2*, etc. För att ge dokumentation av varje kommando, implementera helt enkelt funktionen 'help' på följande sätt:

```
function help(cmd)
{
    if (cmd == "sort") {
        return "Sortera markerad text.";
    } else if (cmd == "...") {
        // ...
    }
}
```

Att utföra **help sort** på kommandoraden anropar hjälpfunktionen med argumentet *cmd* inställd till det givna kommandot, dvs. *cmd* == "sort". KatePart visar därefter den returnerade texten som dokumentation för användaren. Försäkra dig om att [översätta strängarna](#).

Att utveckla ett kommandoradskript kräver att skript läses in igen för att se om ändringar beterar sig lämpligt. Istället för att starta om programmet, byt helt enkelt till kommandoraden och utför kommandot **reload-scripts**.

6.4.2.2.1 Tilldela genvägar

För att göra det möjligt att komma åt skripten i programmenyn och tilldela genvägar, måste skriptet tillhandahålla ett lämpligt skripthuvud. I exemplet ovan, visas båda funktionerna **sort** och **moveLinesDown** i menyn på grund av följande del av skripthuvudet:

```
var katescript = {
    ...
    "actions": [
        {
            "function": "sort",
            "name": "Sortera markerad text",
            "icon": "",
            "category": "Editing",
            "interactive": "false"
        },
        {
            "function": "moveLinesDown",
            "name": "Flytta rader neråt",
            "icon": "",
            "category": "Editing",
            "shortcut": "Ctrl+Shift+Down",
            "interactive": "false"
        }
    ]
};
```

Fälten för en åtgärd är följande:

- `function` [krävs]: Funktionen som ska visas i menyn **Verktyg** → **Skript**.
- `name` [krävs]: Texten visas i skriptmenyn.
- `icon` [valfri]: Ikonen visas intill texten i menyn. Alla KDE:s ikonnamn kan användas här.
- `category` [valfri]: Om en kategori anges, visas skriptet i en undermeny.
- `shortcut` [valfri]: Genvägen som anges här är förvald genväg. Exempel: `Ctrl+Alt+t`. Se [Qt:s dokumentation](#) för ytterligare detaljinformation.
- `interactive` [valfri]: Om skriptet kräver användarinmatning via kommandoraden, ange det här som `true`.

Om du utvecklar användbara skript, fundera på att bidra med det till KatePart-projektet genom att [kontakta e-postlistan](#).

6.4.3 Programmeringsgränssnittet för skripthantering

Programmeringsgränssnittet för skripthantering som presenteras här är tillgängligt i alla skript, dvs. indenteringsskript och kommandon för kommandoraden. Klasserna `Cursor` och `Range` tillhandahålls av biblioteksfiler i `$XDG_DATA_DIRS /katepart5/libraries`. Om du vill använda dem i skriptet, vilket behöver använda vissa av funktionerna i `Document` eller `View`, inkludera det nödvändiga biblioteket med:

```
// nödvändiga katepart.js bibliotek, t.ex. range.js om Range används
require ("range.js");
```

För att utöka det vanliga skriptprogrammeringsgränssnittet med egna funktioner och prototyper, skapa helt enkelt en ny fil i KDE:s lokala inställningskatalog `$XDG_DATA_HOME /katepart5/libraries` och inkludera den i skriptet med:

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

På Windows® är filerna placerade i `%USERPROFILE%\AppData\Local\katepart5\libraries`. `%USERPROFILE%` expanderas oftast till `C:\Users\användare`.

För att utöka befintliga prototyper som `Cursor` eller `Range`, är det rekommenderade sättet *inte* att ändra den globala `*.js`-filen. Ändra istället prototypen för `Cursor` i JavaScript efter att `cursor.js` har inkluderats i skriptet via `require`.

6.4.3.1 Markörer och intervall

Eftersom KatePart är en texteditor är alla programmeringsgränssnitt baserade på markörer och intervall om det är möjligt. En markör (`Cursor`) är ett enkelt par (`rad`, `kolumn`), som representerar en textposition i dokumentet. Ett intervall (`Range`) omfattar text från en markörs startposition till en markörs slutposition. Programmeringsgränssnittet förklaras i detalj i följande avsnitt.

6.4.3.1.1 Prototypen för `Cursor`

`Cursor()`;

Konstruktör. Returnerar en markör på position (`0`, `0`).

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

Cursor(int rad, int kolumn);

Konstruktör. Returnerar en markör på position (rad, kolumn).

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

Cursor(Cursor annan);

Kopieringskonstruktör. Returnerar en kopia av markören *annan*.

Exempel: var copy = new Cursor(other);

Cursor Cursor.clone();

Returnerar en klon av markören.

Exempel: var clone = cursor.clone();

Cursor.setPosition(int rad, int kolumn);

Ställer in markörens position till *rad* och *kolumn*.

Sedan: KDE 4.11

bool Cursor.isValid();

Kontrollera om markören är giltig. Markören är ogiltig om rad och/eller kolumn har värdet -1.

Exempel: var valid = cursor.isValid();

Cursor Cursor.invalidate();

Returnerar en ny ogiltig markör placerad på (-1, -1).

Exempel: var invalidCursor = cursor.invalidate();

int Cursor.compareTo(Cursor annan);

Jämför den här markören med markören *annan*. Returnerar

- -1, om den här markören är placerad innan markören *annan*,
- 0, om båda markörerna är lika, och
- +1, om den här markören är placerad efter markören *annan*.

bool Cursor.equals(Cursor annan);

Returnerar sant om den här markören och markören *annan* är lika, annars falskt.

String Cursor.toString();

Returnerar markören som en sträng på formen 'Cursor(rad, kolumn)'.

6.4.3.1.2 Prototyp för Range

Range();

Konstruktör. Att anropa new Range() returnerar intervallet (0, 0) - (0, 0).

Range(Cursor start, Cursor slut);

Konstruktör. Att anropa new Range(start, slut) returnerar intervallet (*start*, *slut*).

Range(int startrad, int startkolumn, int slutrad, int slutkolumn);

Constructor. Calling new Range(startrad, startkolumn, slutrad, slutkolumn) returnerar intervallet från (*startrad*, *startkolumn*) to (*startrad*, *slutkolumn*).

Range(Range annat);

Kopieringskonstruktör. Returnerar en kopia av intervallet *annat*.

Range Range.clone();

Returnerar en klon av intervallet.

Exempel: var clone = range.clone();

bool Range.isEmpty();

Returnerar sant om både start- och slutmarkörerna är lika.

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

Sedan: KDE 4.11

bool Range.isValid();

Returnerar sant om både start- och slutmarkörerna är giltiga, annars falskt.

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

Range Range.invalid();

Returnerar intervallet från (-1, -1) till (-1, -1).

bool Range.contains(Cursor markör);

Returnerar `true` om intervallet innehåller markörpositionen, annars `false`.

bool Range.contains(Range annat);

Returnerar sant om det här intervallet innehåller intervallet *annat*, annars falskt.

bool Range.containsColumn(int kolumn);

Returnerar sant om *kolumn* är i det halvöppna intervallet [startkolumn, slutkolumn), annars falskt.

bool Range.containsLine(int rad);

Returnerar sant om *rad* är i det halvöppna intervallet [startrad, slutrad), annars falskt.

bool Range.overlaps(Range annat);

Returnerar sant om det här intervallet och intervallet *annat* delar ett gemensamt område, annars falskt.

bool Range.overlapsLine(int rad);

Returnerar sant om *rad* är i intervallet [startrad, slutrad], annars falskt.

bool Range.overlapsColumn(int kolumn);

Returnerar sant om *kolumn* är i intervallet [startkolumn, slutkolumn], annars falskt.

bool Range.onSingleLine();

Returnerar sant om intervallet startar och slutar på samma rad, dvs. om *Range.start.line == Range.end.line*.

Sedan: KDE 4.9

bool Range.equals(Range annan);

Returnerar sant om det här intervallet och intervallet *annat* är lika, annars falskt.

String Range.toString();

Returnerar intervallet som en sträng på formen 'Range(Cursor(rad, kolumn), Cursor(rad, kolumn))'.

6.4.3.2 Globala funktioner

Det här avsnittet listar alla globala funktioner.

6.4.3.2.1 Att läsa och inkludera filer

String read(String fil);

Söker efter angiven *fil* relativt i förhållande till katalogen `katepart5/script/files` och returnerar innehållet som en sträng.

void require(String fil);

Söker efter angiven *fil* relativt i förhållande till katalogen `katepart5/script/libraries` och utvärdera den. Internt skyddas `require` från att samma *fil* inkluderas flera gånger.

Sedan: KDE 4.10

6.4.3.2.2 Felsökning

void debug(String text);

Skriver ut *text* på standardutmatningen i terminalen som startade programmet.

6.4.3.2.3 Översättning

För att stödja fullständig landsanpassning, finns det flera funktioner för att översätta strängar i skript, närmare bestämt `i18n`, `i18nc`, `i18np` och `i18ncp`. Funktionerna beter sig exakt likadant som [KDE:s översättningsfunktioner](#).

Översättningsfunktionerna översätter strängarna de omger via KDE:s översättningssystem till det språk som används i programmet. Strängar i skript som utvecklas i KateParts officiella källkod extraheras automatiskt och är översättningsbara. Med andra ord, som utvecklare av KatePart behöver man inte bry dig om att extrahera och översätta meddelanden. Observera dock att översättningen bara fungerar inne i KDE:s infrastruktur, nya strängar i skript från tredjepart som utvecklas utanför KDE, översätts inte. Överväg därför gärna att bidra med dina skript till Kate så att en passande översättning blir möjlig.

void i18n(String text, arg1, ...);

Översätter *text* till språket som används av programmet. Parametrarna *arg1*, ..., är valfria och används för att ersätta platsmarkörerna `%1`, `%2`, etc.

void i18nc(String sammanhang, String text, arg1, ...);

Översätter *text* till språket som används av programmet. Dessutom är strängen *sammanhang* synlig för översättare, så att de kan tillhandahålla en bättre översättning. Parametrarna *arg1*, ..., är valfria och används för att ersätta platsmarkörerna `%1`, `%2`, etc.

void i18np(String singularis, String pluralis, int antal, arg1, ...);

Översätter antingen *singularis* eller *pluralis* till språket som används av programmet, beroende på angivet *antal*. Parametrarna *arg1*, ..., är valfria och används för att ersätta platsmarkörerna `%1`, `%2`, etc.

void i18ncp(String sammanhang, String singularis, String pluralis, int antal, arg1, ...);

Översätter antingen *singularis* eller *pluralis* till språket som används av programmet, beroende på angivet *antal*. Dessutom är strängen *sammanhang* synlig för översättare, så att de kan tillhandahålla en bättre översättning. Parametrarna *arg1*, ..., är valfria och används för att ersätta platsmarkörerna `%1`, `%2`, etc.

6.4.3.3 Programmeringsgränssnittet för vyn

När ett skript håller på att köras, finns det en global variabel 'view' som representerar den för närvarande aktiva editorvyn. Det följande är en lista med alla tillgängliga funktioner i View.

void view.copy()

Kopiera markeringen om det finns någon, annars aktuell rad om alternativet [] **Kopiera eller klipp ut nuvarande rad om markering saknas** är aktiverat.

Sedan: KDE Ramverk 5.79

void view.cut()

Klipp ut markeringen om det finns någon, annars aktuell rad om alternativet [] **Kopiera eller klipp ut nuvarande rad om markering saknas** är aktiverat.

Sedan: KDE Ramverk 5.79

void view.paste()

Klistra in klippbordets innehåll.

Sedan: KDE Ramverk 5.79

Cursor view.cursorPosition()

Returnerar den nuvarande markörpositionen i vyn.

void view.setCursorPosition(int rad, int kolumn); void

view.setCursorPosition(Cursor markör);

Ställ in den nuvarande markörpositionen till antingen (rad, kolumn) eller till den givna markören.

Cursor view.virtualCursorPosition();

Returnerar den virtuella markörpositionen, där varje tabulator räknas som motsvarande antal mellanslag, beroende på nuvarande tabulatorbredd.

void view.setVirtualCursorPosition(int rad, int kolumn); void

view.setVirtualCursorPosition(Cursor markör);

Ställ in den nuvarande virtuella markörpositionen till (rad, kolumn) eller till den givna markören.

String view.selectedText();

Returnerar den markerade texten. Om ingen text är markerad, är den returnerade strängen tom.

bool view.hasSelection();

Returnerar *true* om vyn har markerad text, annars *false*.

Range view.selection();

Returnerar det markerade textintervallet. Det returnerade intervallet är ogiltigt om det inte finns någon markerad text.

void view.setSelection(Range intervall);

Ställer in markerad text till angivet intervall.

void view.removeSelectedText();

Tar bort den markerade texten. Om vyn inte har någon markerad text, händer ingenting.

void view.selectAll();

Markerar hela texten i dokumentet.

void view.clearSelection();

Rensar textmarkeringen utan att ta bort texten.

void view.setBlockSelection(bool on);

Sätt på eller stäng av blockmarkering.

bool view.blockSelection();

Returnerar `true` om blockmarkering är på, annars `false`.

void view.align(Range intervall);

Indentera rader på ett riktigt sätt igen inom *intervall* enligt aktuella indenteringsinställningar.

void view.alignOn(Range intervall, String mönster = "");

Justerar rader i *intervall* i kolumnen som anges av det reguljära uttrycket *mönster*. Med ett tomt *mönster* justeras de normalt på den första tecknet som inte är ett blanktecken. Om mönstret har en platsmarkör indenterar det där markören matchar.

Exempel:

`view.alignOn(document.documentRange(), '-');` infogar mellanslag innan första - på varje rad för att justera alla på samma kolumn.

`view.alignOn(document.documentRange(), ':\s+(.)');` infogar mellanslag före det första tecknet som inte är ett blanktecken efter ett kolon för att justera alla till samma kolumn.

object view.executeCommand(String kommando, String argument, Range intervall);

Kör [kommandoradskommandot](#) *kommando* med de valfria argumenten *argument* och ett valfritt *intervall*. Returnerat *objekt* har den Booleska egenskapen *objekt.ok* som anger om körning av *kommando* lyckades. I händelse av fel, innehåller strängen *objekt.status* ett felmeddelande.

Sedan: KDE Ramverk 5.50

Range view.searchText(Range intervall, String mönster, bool bakåt = false);

Sök efter den första förekomsten av *mönster* i *intervall* och returnerar det matchade intervallet. Sökningen utförs bakåt om den valfria booleska parametern *bakåt* är inställd till `true`.

Det returnerade intervallet är ogiltigt (se `Range.isValid()`) om *mönster* inte hittas i *intervall*.

Sedan: KDE Ramverk 5.97

6.4.3.4 Programmeringsgränssnittet för dokumentet

När ett skript håller på att köras, finns det en global variabel 'document' som representerar det för närvarande aktiva dokumentet. Det följande är en lista med alla tillgängliga funktioner i Document.

String document.fileName();

Returnerar dokumentets filnamn eller en tom sträng för osparade textbuffrar.

String document.url();

Returnerar dokumentets fullständiga webbadress eller en tom sträng för osparade textbuffrar.

String document.mimeType();

Returnerar dokumentets Mime-typ eller MIME-typen `application/octet-stream` om ingen lämplig MIME-typ kunde hittas.

String document.encoding();

Returnerar den nuvarande kodningen använd för att spara filen.

String document.highlightingMode();

Returnerar det allmänna färgläggningsläget som används för hela dokumentet.

String document.highlightingModeAt(Cursor pos);

Returnerar färgläggningsläget som används på den givna positionen i dokumentet.

Array document.embeddedHighlightingModes();

Returnerar ett fält med färgläggningslägen inbäddade i det här dokumentet.

bool document.isModified();

Returnerar `true` om dokumentet har osparade ändringar (är modifierat), annars `false`.

String document.text();

Returnerar dokumentets hela innehåll i en enda textsträng. Nya rader är markerade med nyradstecknet `'\n'`.

String document.text(int från_rad, int från_kolumn, int till_rad, int till_kolumn); String document.text(Cursor från, Cursor till); String document.text(Range intervall);

Returnerar texten i det angivna intervallet. Det rekommenderas att använda markör- och intervallbaserade versioner för bättre läsbarhet hos källkoden.

String document.line(int rad);

Returnerar den givna textraden som en sträng. Strängen är tom om den begärda raden är utanför tillgängligt intervall.

String document.wordAt(int rad, int kolumn); String document.wordAt(Cursor markör);

Returnerar ordet på den angivna markörposition.

Range document.wordRangeAt(int rad, int kolumn); Range document.wordRangeAt(Cursor markör);

Returnerar intervallet för ordet på den givna markörpositionen. Det returnerade intervallet är ogiltigt (se `Range.isValid()`) om textpositionen är efter ett radslut. Om det inte finns något ord vid den givna markörpositionen returneras ett tomt intervall.

Sedan: KDE 4.9

String document.charAt(int rad, int kolumn); String document.charAt(Cursor markör);

Returnerar tecknet på den angivna markörpositionen.

String document.firstChar(int rad);

Returnerar det första tecknet på angiven `rad` som inte är ett blanktecken. Det första tecknet finns på kolumn 0. Om raden är tom, eller bara innehåller blanktecken, är den returnerade strängen tom.

String document.lastChar(int rad);

Returnerar det sista tecknet på angiven `rad` som inte är ett blanktecken. Om raden är tom, eller bara innehåller blanktecken, är den returnerade strängen tom.

bool document.isSpace(int rad, int kolumn); bool document.isSpace(Cursor markör);

Returnerar `sant` om tecknet på den givna markörpositionen är ett blanktecken, annars `falskt`.

bool document.matchesAt(int rad, int kolumn, String text); bool document.matchesAt(Cursor markör, String text);

Returnerar `sant` om given `text` matchar på den motsvarande markörpositionen, annars `falskt`.

bool document.startsWith(int rad, String text, bool hoppa_över_blanktecken);
 Returnerar sant om raden börjar med *text*, annars falskt. Argumentet *hoppa_över_blanktecken* bestämmer om inledande blanktecken ignoreras.

bool document.endsWith(int rad, String text, bool hoppa_över_blanktecken);
 Returnerar sant om raden slutar med *text*, annars falskt. Argumentet *hoppa_över_blanktecken* bestämmer om inledande blanktecken ignoreras.

bool document.setText(String text);
 Anger hela dokumentets text.

bool document.clear();
 Tar bort hela texten i dokumentet.

bool document.truncate(int rad, int kolumn); bool document.truncate(Cursor markör);
 Avkortar den givna raden på den givna kolumnen eller markörpositionen. Returnerar sant om det lyckas, eller falskt om den givna raden inte ingår i dokumentets intervall.

bool document.insertText(int rad, int kolumn, String text); bool document.insertText(Cursor markör, String text);
 Infogar *text* på den givna markörpositionen. Returnerar sant om det lyckas, eller falskt om dokumentet är skrivskyddat.

bool document.removeText(int från_rad, int från_kolumn, int till_rad, int till_kolumn); bool document.removeText(Cursor från, Cursor till); bool document.removeText(Range intervall);
 Tar bort text i det givna intervallet. Returnerar sant om det lyckas, eller falskt om dokumentet är skrivskyddat.

bool document.insertLine(int rad, String text);
 Infogar text på den givna raden. Returnerar sant om det lyckas, eller falskt om dokumentet är skrivskyddat eller raden inte ingår i dokumentets intervall.

bool document.removeLine(int rad);
 Tar bort den givna textrad. Returnerar sant om det lyckas, eller falskt om dokumentet är skrivskyddat eller raden inte ingår i dokumentets intervall.

bool document.wrapLine(int rad, int kolumn); bool document.wrapLine(Cursor markör);
 Radbryter raden vid den givna markörpositionen. Returnerar sant om det lyckas, eller falskt, t.ex. om $rad < 0$.
 Sedan: KDE 4.9

void document.joinLines(int startrad, int slutrad);
 Sammanfogar rader från *startrad* till *slutrad*. Två på varandra följande rader åtskiljs alltid med exakt ett mellanslag.

int document.lines();
 Returnerar antal rader i dokumentet.

bool document.isLineModified(int rad);
 Returnerar sant om *rad* för närvarande innehåller osparad data.
 Since: KDE 5.0

bool document.isLineSaved(int rad);
 Returnerar sant om *rad* har ändrats, men dokumentet har sparats. Sålunda innehåller raden för närvarande ingen osparad data.
 Since: KDE 5.0

bool document.isLineTouched(int rad);

Returnerar sant om *rad* för närvarande innehåller osparad data eller tidigare har ändrats.
Since: KDE 5.0

bool document.findTouchedLine(int startrad, bool ner);

Sök efter nästa ändrade rad med början på *rad*. Sökningen utförs antingen uppåt eller neråt beroende på sökdiriktningen angiven med *ner*.
Since: KDE 5.0

int document.length();

Returnerar antal tecken i dokumentet.

int document.lineLength(int rad);

Returnerar längden på *rad*.

void document.editBegin();

Startar en redigeringsgrupp för gruppering av ångra/gör om. Försäkra dig om att alltid anropa `editEnd()` lika många gånger som du anropar `editBegin()`. Att anropa `editBegin()` använder en referensräknare internt, dvs. anropet kan nästlas.

void document.editEnd();

Avslutar en redigeringsgrupp. Det sista anropet av `editEnd()` (dvs. det för det första anropet till `editBegin()`) avslutar redigeringssteget.

int document.firstColumn(int rad);

Returnerar det första tecknet som inte är ett blanktecken på given *rad*. Om det bara finns blanktecken på raden, är returvärdet -1.

int document.lastColumn(int rad);

Returnerar det sista tecknet som inte är ett blanktecken på given *rad*. Om det bara finns blanktecken på raden, är returvärdet -1.

int document.prevNonSpaceColumn(int rad, int kolumn); int

document.prevNonSpaceColumn(Cursor markör);

Returnerar kolumnen med ett tecken som inte är ett blanktecken med början på given markörposition genom att söka bakåt.

int document.nextNonSpaceColumn(int rad, int kolumn); int

document.nextNonSpaceColumn(Cursor markör);

Returnerar kolumnen med ett tecken som inte är ett blanktecken med början på given markörposition genom att söka framåt.

int document.prevNonEmptyLine(int rad);

Returnerar nästa rad som inte är tom och som innehåller tecken som inte är blanktecken, genom att söka bakåt.

int document.nextNonEmptyLine(int rad);

Returnerar nästa rad som inte är tom och som innehåller tecken som inte är blanktecken, genom att söka framåt.

bool document.isInWord(String tecken, int egenskap);

Returnerar sant, om givet *tecken* med given *egenskap* kan vara en del av ett ord, annars falskt.

bool document.canBreakAt(String tecken, int egenskap);

Returnerar sant, om givet *tecken* med given *egenskap* är lämpat att bryta en rad, annars falskt.

bool document.canComment(int startegenskap, int slutegenskap);
 Returnerar *sant* om ett intervall som börjar och slutar med den givna egenskapen är lämpat att kommenteras bort, annars *falskt*.

String document.commentMarker(int egenskap);
 Returnerar kommentarmarkören för enraderskommentarer för en given *egenskap*.

String document.commentStart(int egenskap);
 Returnerar kommentarmarkören för början av flerraderskommentarer för en given *egenskap*.

String document.commentEnd(int egenskap);
 Returnerar kommentarmarkören för slutet av flerraderskommentarer för en given *egenskap*.

Range document.documentRange();
 Returnerar ett intervall som omfattar hela dokumentet.

Cursor document.End();
 Returnerar en markör placerad på den sista radens sista kolumn i dokumentet.

bool isValidTextPosition(int rad, int kolumn); bool isValidTextPosition(Cursor markör);
 Returnerar *sant* om den angivna markörpositionen är placerad på en giltig textposition. En textposition är bara giltig om den finns i början av, mitten av eller slutet av en giltig rad. Dessutom är en textposition ogiltig om den är placerad i ett Unicode-surrogat.
 Since: KDE 5.0

int document.attribute(int rad, int kolumn); int document.attribute(Cursor markör);
 Returnerar egenskapen på den angivna markörpositionen.

bool document.isAttribute(int rad, int kolumn, int egenskap); bool document.isAttribute(Cursor markör, int egenskap);
 Returnerar *sant* om egenskapen på den givna markörpositionen är lika med *egenskap*, annars *falskt*.

String document.attributeName(int rad, int kolumn); String document.attributeName(Cursor markör);
 Returnerar egenskapsnamnet som läsbar text. Det är lika med namnet *itemData* i syntaxfärgläggningsfilerna.

bool document.isAttributeName(int rad, int kolumn, String namn); bool document.isAttributeName(Cursor markör, String namn);
 Returnerar *sant* om egenskapsnamnet på en viss markörposition motsvara givet *namn*, annars *falskt*.

String document.variable(String nyckel);
 Returnerar värdet på den begärda dokumentvariabeln *nyckel*. Om dokumentvariabeln inte finns, är returvärdet en tom sträng.

void document.setVariable(String nyckel, String värde);
 Ställ in värdet på den begärda dokumentvariabeln *nyckel*.
 Se också: [Kate-dokumentvariabler](#)
 Sedan: KDE 4.8

int document.firstVirtualColumn(int rad);
 Returnerar den virtuella kolumnen för det första tecknet som inte är ett blanktecken på den givna raden, eller -1 om raden är tom eller bara innehåller blanktecken.

int document.lastVirtualColumn(int rad);

Returnerar den virtuella kolumnen för det sista tecknet som inte är ett blanktecken på den givna raden, eller -1 om raden är tom eller bara innehåller blanktecken.

int document.toVirtualColumn(int rad, int kolumn); int

document.toVirtualColumn(Cursor markör); Cursor document.toVirtualCursor(Cursor markör);

Konverterar den givna 'verkliga' markörpositionen till en virtuell markörposition, och returnerar antingen ett heltal eller ett markörobjekt.

int document.fromVirtualColumn(int rad, int virtuell_kolumn);

int document.fromVirtualColumn(Cursor virtuell_markör); Cursor

document.fromVirtualCursor(Cursor virtuell_markör);

Konverterar den givna virtuella markörpositionen till en 'verklig' markörposition, och returnerar antingen ett heltal eller ett markörobjekt.

Cursor document.anchor(int rad, int kolumn, Char tecken); Cursor

document.anchor(Cursor markör, Char tecken);

Söker bakåt efter det givna tecknet med början på den givna markören. Om exempelvis '(' anges som tecken, returnerar funktionen positionen på ett inledande '('. Det sker med referensräkning, dvs. andra '(...)' ignoreras.

Cursor document.rfind(int rad, int kolumn, String text, int egenskap = -1);

Cursor document.rfind(Cursor markör, String text, int egenskap = -1);

Söker bakåt efter den givna texten med lämplig *egenskap*. Argumentet *egenskap* ignoreras om det anges som -1. Den returnerade markören är ogiltig om texten inte kunde hittas.

int document.defStyleNum(int rad, int kolumn); int document.defStyleNum(Cursor markör);

Returnerar standardstilen som används på den givna markörpositionen.

bool document.isCode(int rad, int kolumn); bool document.isCode(Cursor markör);

Returnerar sant om egenskapen på given markörposition inte är lika med någon av följande stilar: dsComment, dsString, dsRegionMarker, dsChar, dsOthers.

bool document.isComment(int rad, int kolumn); bool document.isComment(Cursor markör);

Returnerar sant, om egenskapen för tecknet på markörpositionen är dsComment, annars falskt.

bool document.isString(int rad, int kolumn); bool document.isString(Cursor markör);

Returnerar sant, om egenskapen för tecknet på markörpositionen är dsString, annars falskt.

bool document.isRegionMarker(int rad, int kolumn); bool

document.isRegionMarker(Cursor markör);

Returnerar sant, om egenskapen för tecknet på markörpositionen är dsRegionMarker, annars falskt.

bool document.isChar(int rad, int kolumn); bool document.isChar(Cursor markör);

Returnerar sant, om egenskapen för tecknet på markörpositionen är dsChar, annars falskt.

bool document.isOthers(int rad, int kolumn); bool document.isOthers(Cursor markör);

Returnerar sant, om egenskapen för tecknet på markörpositionen är dsOthers, annars falskt.

void document.indent (Range intervall, int ändring);

Indentera alla rader i *intervall* med *ändring* tabulatortecken eller *ändring* gånger tabulatorstorlek mellanslag beroende på användarens inställningar. Parametern *ändring* kan vara negativ.

6.4.3.5 Programmeringsgränssnittet för editorn

Förutom dokumentet och programmeringsgränssnittet för vyn, finns ett generellt programmeringsgränssnitt för editorn som tillhandahåller funktioner för allmän redigeringskriptfunktionalitet.

String editor.clipboardText ();

Returnerar texten som för närvarande finns i det globala klippbordet.

Sedan: KDE Ramverk 5.50

String editor.clipboardHistory ();

Editorn har en klippbordshistorik som innehåller upp till tio klippbordsposter. Funktionen returnerar alla poster som för närvarande finns i klippbordshistoriken.

Sedan: KDE Ramverk 5.50

void editor.setClipboardText (String text);

Ställ in klippbordets innehåll till *text*. *text* läggs till i klippbordshistoriken.

Sedan: KDE Ramverk 5.50

Kapitel 7

Att anpassa KatePart

Genom att välja **Inställningar** → **Anpassa program...** i menyn visar dialogrutan **Anpassa**. Dialogrutan kan användas för att ändra ett antal olika inställningar. Inställningarna som är tillgängliga för att ändras varierar beroende på vilken kategori som användaren väljer i den vertikala listan till vänster i dialogrutan. Förloppet kan styras med de tre knapparna längs rutans nederkant.

Du kan starta **Hjälp**-systemet, acceptera de nuvarande inställningarna och stänga dialogrutan med användning av knappen **Ok**, eller **Avbryta** förfarandet. Kategorierna **Utseende**, **Teckensnitt och färger**, **Redigering**, **Öppna/spara** och **Utökningar** beskrivs i detalj nedan.

7.1 Inställning av editorkomponenten

Den här gruppen innehåller alla sidor som hör till editorkomponenten i KatePart. De flesta inställningarna här är förvalda, och kan överskridas genom att [definiera en filtyp](#), genom [dokumentvariabler](#) eller genom att ändra dem per dokument under en redigeringsession.

7.1.1 Utseende

7.1.1.1 Allmänt

Editorns teckensnitt

Här kan du välja teckensnitt för editorns text. Du kan välja bland alla teckensnitt som är tillgängliga i systemet, och ange en normal storlek och kodning. Ett textexempel visas längst ner i dialogrutan, så att du kan se effekten av det du väljer.

För mer information om hur man väljer ett teckensnitt, se avsnittet [Välja teckensnitt i dokumentet KDE:s grunder](#).

Visa indikering av blanktecken

Aldrig

Editorn visar aldrig punkter för att indikera närvaron av blanktecken.

Vid radens slut

Editorn kommer att visa punkter som indikerar närvaron av extra blanktecken i slutet av rader.

Alltid

Editorn visar alltid punkter för att indikera närvaron av blanktecken.

Storlek på indikering av blanktecken

Använd skjutreglaget för att ändra storlek på den synliga markören.

Visa tabulatormarkörer

När markerat, visar editorn symbolen » för att indikera närvaron av en tabulator i texten.

Visa fokusram omkring editorn

När markerat, visar editorn en fokusram omkring den huvudsakliga textpanelen.

Parentesmatchning

Markera område mellan samhörande parenteser

Om det här är aktiverat, markeras området mellan de valda samhörande parenteserna.

Visa förhandsgranskning av motsvarande inledande parentes

När aktiverat, visar editorn ett verktygstips för motsvarande inledande parentes.

Blinka matchande parentes när markören flyttas till den andra parentesen i ett par

Om aktiverad och man går till en parentes ({, [,], }, (eller)) så blinkar den motsvarande parentesen snabbt.

Visa indenteringslinjer

Om det här är markerat, visar editorn vertikala linjer för att hjälpa till att identifiera indenterade rader.

Antal

Visa antal ord

Visar antal ord och tecken i dokumentet och aktuell markering i statusraden. Alternativet är också tillgängligt i statusradens sammanhangsberoende meny.

Visa antal rader

Visar totalt antal rader i dokumentet i statusraden. Alternativet är också tillgängligt i statusradens sammanhangsberoende meny.

Fäll ihop första raden

Om aktiverad fälls den första raden ihop om möjligt. Det är användbart om filen börjar med en kommentar, såsom en copyright.

Dynamisk radbrytning

Om det här alternativet markeras, bryts textrader vid fönstrets kant på skärmen.

Dynamisk radbrytning vid statisk radbrytningsmarkör

När markerat, bryter editorn rader dynamiskt vid den [statiska radbrytningspositionen](#).

Bortse från ordgränser vid dynamisk radbrytning

När markerat, tar inte editorn hänsyn till ordgränser när textrader brytes.

Dynamiska radbrytningsmarkörer

Välj när dynamiska radbrytningsmarkörer ska visas, antingen **Av**, **Följ radnummer** eller **Alltid på**.

Indentera radbrutna rader

Dessutom låter det dig att ställa in en maximal bredd av skärmen, som ett procenttal. Därefter justeras inte längre dynamiskt radbrutna rader vertikalt. Vid 50 % får exempelvis inte rader vars indenteringsdjup är mer än 50 % av skärmens bredd någon vertikal justering för efterföljande radbrutna rader.

Radhöjdsmultiplikator

Värdet multipliceras med teckensnittets förvalda radhöjd. Värdet 1,0 betyder att förvald höjd används.

7.1.1.2 Kanter

Vikning av kodblock

Visa pilar för att dra ihop kodblock

Om det här alternativet är markerat, visar den aktuella vyn markeringar för kodvikning, om kodvikning är tillgänglig.

Visa förhandsgranskning av vikta block när muspekaren hålls över

Om markerat, visas en förhandsgranskning av den hopvikta texten i en meddelanderuta när musen hålls över en hopvikt region.

Vikningspilarnas synlighet

Ändra vikningspilarna mellan **Visa när musen hålls över** och **Visa alltid**.

Vänsterkant

Visa markörer

Om det här är markerat, ser du en ikonkant till vänster. Ikonkanten visar till exempel bokmärkesmarkeringar.

Visa radnummer

Om det här är markerat, ser du radnummer till vänster.

Markera ändrade och osparade rader

Om det här är markerat, visas markörer av radändringar. För mer information, se Avsnitt 3.9.

Rullningslistor

Visa markörer

Om det här alternativet är markerat visar aktuell vy markeringar på den vertikala rullningslistan. Markeringarna visar till exempel bokmärken.

Visa förhandsgranskning när muspekaren hålls över rullningslistor

Om alternativet är markerat, och du håller muspekaren över rullningslistan, visas en liten textförhandsgranskning med flera rader i det nuvarande dokumentet omkring pekarens position. Det låter dig snabbt gå till en annan del av dokumentet.

Miniavbildning

Visa miniavbildning

Om det här alternativet är markerat visar varje ny vy en miniavbildning av dokumentet på den vertikala rullningslistan.

För mer information om rullningslistans miniavbildning, se Avsnitt 3.10.

Miniavbildningens bredd

Justerar bredden på rullningslistans miniavbildning, definierad i bildpunkter.

Rullningslistornas synlighet

Visa rullningslistan, dölj den eller visa bara rullningslistan när det behövs. Klicka med vänster musknapp på den blåa rektangeln för att visa radnummerintervallet för dokumentet som visas på skärmen. Håll vänster musknapp nertryckt utanför den blåa rektangeln för automatiskt förflyttning genom dokumentet.

Sortera menyn bokmärken

Enligt skapat datum

Varje nytt bokmärke läggs till längst ner, oberoende av var det är placerat i dokumentet.

Enligt radnummer

Bokmärken ordnas enligt radnumret där de är placerade.

7.1.2 Färgteman

Den här sidan i dialogrutan låter dig ställa in alla färger i de olika färgteman du har, och skapa nya teman, ta bort befintliga eller bara **Följ systemets färgschema**. Varje tema har inställningar av färger, normal och markerad textstil.

KatePart väljer temat som för närvarande är aktivt i förväg åt dig. Om du vill arbeta med ett annat tema, börja med att välja det med kombinationsrutan **Välj tema**. Du kan skapa ett nytt tema eller ta bort befintliga med knapparna **Ny** och **Ta bort**.

Det beskrivs detaljerat i Avsnitt [6.3.5](#).

7.1.3 Redigering

7.1.3.1 Allmänt

Radbrytning

Radbrytning är en funktion som orsakar att editorn automatiskt börjar en ny rad med text och flyttar (bryter) markören till början på den nya raden. KatePart startar automatiskt en ny rad med text när den nuvarande raden når längden som anges av alternativet [Bryt rader vid:](#).

Bryt rader vid en fast kolumn

Sätter på eller stänger av statisk radbrytning.

Rita vertikal linje vid radbrytningskolumnen

Om det här alternativet är markerat, ritas en vertikal linje i radbrytningskolumnen som definieras i **Inställningar** → **Anpassa editor...** under fliken Redigering. Observera att radbrytningsmarkören bara ritas om du använder ett teckensnitt med fast breddsteg.

Bryt rader vid:

Om alternativet [Bryt rader vid en fast kolumn](#) är valt avgör det här värdet längden (i tecken) där editorn automatiskt påbörjar en ny rad.

Standardläge för inmatning

Det valda inmatningsläget aktiveras när en ny vy öppnas. Du kan fortfarande sätta på eller stänga av Vi-inmatningsläge för en viss vy från menyn **Redigera**.

Parenteser

Om alternativet **Fyll automatiskt i parenteser när inledande parentes skrivs in** är aktiverat när användaren skriver en vänsterparentes ([, (, eller {) skriver KatePart automatiskt in högerparentesen (],), eller }) till höger om markören.

Omslutande tecken

Det är möjligt att markera omslutande tecken genom att använda motsvarande kombinationslista.

När text markeras, radbryts den markerade texten om ett av tecknen skrivs in.

Kopiera och klistra in

Flytta markerad text vid drag och släpp

Alternativet aktiverar dra och släpp av den markerade texten inne i editorfönstret.

Kopiera eller klipp ut nuvarande rad utförd utan markerad text

Om alternativet är aktiverat och textmarkeringen är tom, utförs åtgärderna kopiera och klipp ut för textraden på den verkliga markörpositionen.

Flytta inte textmarkören vid inklistring med musen

Om alternativet är aktiverat och du klistrar in någon text i editorfönstret genom att klicka med musens mittenknapp, flyttar inte KatePart textmarkören till den klickade positionen.

7.1.3.2 Textnavigering

Förflyttning av textmarkör

Smarta Home- och End-tangenter

Om markerad, hoppar markören över blanktecken och går till början av radens text när tangenten Home trycks ner.

Page Up och Page Down flyttar markören

Det här alternativet ändrar beteendet hos markören om användaren trycker på tangenterna **Page Up** eller **Page Down**. Om det här inte är valt, behåller textmarkören sin relativa position i den synliga texten i KatePart när ny text blir synlig som ett resultat av åtgärden. Så om markören befinner sig mitt i den synliga texten när åtgärden sker kommer den att förbli där (utom när man kommer till början eller slutet). Om det här alternativet är valt, gör den första tangentryckningen att markören antingen flyttas längst upp eller längst ner när en ny sida med text visas.

Aktivera markörförflyttning till stora bokstäver

Alternativet ändrar beteendet hos markören när användaren trycker på genvägen **Ctrl-Vänsterpil** eller **Ctrl-Högerpil**. Om avmarkerat går textmarkören förbi hela ord. Med alternativet markerat, stoppar markörförflyttningen vid stora bokstäver.

Centrera markör automatiskt:

Anger antalet rader som behålls synliga ovanför och under markören om möjligt.

Textmarkeringsläge

Normal

Markeringar skrivs över av inskriven text, och försvinner när markören flyttas.

Långlivat

Markeringar blir kvar också efter markören flyttas och text skrivs in.

Tillåt att rulla förbi dokumentets slut

Det här alternativet låter dig rulla förbi dokumentets slut. Det kan användas för att centrera dokumentets sista del vertikalt, eller placera det ovanpå den nuvarande vyn.

Backstegstangenten tar bort grundtecknet med diakritiska markeringar

När de är markerade, tas sammansatta tecken bort med sina diakritiska markeringar istället för att bara ta bort grundtecknet. Det är användbart för Indiska landsinställningar.

Väljare för flera markörer

Alternativet låter dig ange väljare som används för att skapa flera markörer med vänsterklick. Du måste hålla nere väljarna och klicka med vänster musknapp för att skapa en markör på önskad plats. Se [Skapa flera markörer](#) för att ta reda på andra sätt att skapa flera markörer.

7.1.3.3 Indentering

Standardläge för indentering:

Välj automatiskt indenteringsläge som du normalt vill använda. Du rekommenderas starkt att använda **Inget** eller **Normal** här, och använda inställning av filtyper för att ställa in andra indenteringslägen för textformat som C/C++ kod eller XML.

Indentera med

Tabulatorer

När det här är aktiverat, infogar editorn tabulatortecken när du trycker på tabulatortangenten eller använder [automatisk indentering](#).

Mellanslag

När det här är aktiverat, infogar editorn ett beräknat antal mellanslag enligt positionen i texten och inställningen `tab-width` när du trycker på tabulatortangenten eller använder [automatisk indentering](#).

Tabulatorer och mellanslag

När det här är aktiverat, infogar editorn mellanslag som beskrivs ovan vid indentering eller när tabulatortangenten används i början av en rad, men infogar tabulatorer när tabulatortangenten används i mitten eller slutet av en rad.

Tabulatorbredd:

Detta ställer in antal mellanslag som visas istället för ett tabulatortecken.

Indenteringsbredd:

Indenteringsbredden är antalet mellanslag som används för att indentera en rad. Om indentera med tabulator är inställt, infogas ett tabulatortecken om indenteringen är jämnt delbar med tabulatorbredden.

Indenteringsegenskaper

Behåll extra mellanslag

Om alternativet är inaktiverat, justeras en rad till en multipel av bredden som anges med **indenteringsbredd** när indenteringsnivån ändras.

Justera indentering av text som klistrats in från klippbordet

Om alternativet markeras, indenteras text som klistras in från klippbordet. Genom att använda alternativet **ångra** kan indenteringen tas bort.

Indenteringsåtgärder

Backsteg i inledande blanktecken avindenterar

Om alternativet är markerat, minskar tangenten **Backsteg** indenteringsnivån om markören är placerad i de inledande blanktecknen på en rad.

Åtgärd för tabulatortangent (om det inte finns någon markering)

Om du vill att Tabulator ska justera nuvarande rad i nuvarande kodblock som i Emacs, gör Tabulator till en genväg till åtgärden **Formatera indentering**.

Gå alltid vidare till nästa tabulatorposition

Om alternativet är markerat, infogar alltid tangenten Tabulator blanktecken så att nästa tabulatorposition nås. Om alternativet **Infoga mellanslag i stället för tabulatorer** under fliken **Allmänt** på sidan **Redigering** är aktiverat, infogas mellanslag. Annars infogas ett enda tabulatortecken.

Öka alltid indenteringsnivån

Om alternativet är markerat, indenterar alltid tangenten Tabulator raden med det antal teckenpositioner som anges av **Indenteringsbredd**.

Öka indenteringsnivån om i inledande blanktecken

Om alternativet är markerat, indenterar antingen tangenten Tabulator nuvarande rad eller går vidare till nästa tabulatorposition. Om markörens position är på eller före det första tecknet som inte är ett blanktecken på raden, eller om det finns en markering, indenteras nuvarande rad med det antal teckenpositioner som anges av **Indenteringsbredd**. Om markörens position är efter det första tecknet som inte är ett blanktecken på raden och det inte finns någon markering, infogas blanktecken så att nästa tabulatorposition nås: om alternativet **Infoga mellanslag i stället för tabulatorer** under **Redigering** är aktiverat, infogas mellanslag. Annars infogas ett enda tabulatortecken.

7.1.3.4 Automatisk komplettering

Allmänt

Aktivera automatisk komplettering

Om aktiverad, dyker en ordkompletteringsruta automatiskt upp vid inskrivning, som visar en lista med texter som kompletterar texten som för närvarande är under markören.

Välj första kompletteringspost automatiskt

Om aktiverad, är alltid det första objektet i automatisk komplettering förvalt så att du kan infoga det med returtangenten. Om du inte vill ha det beteendet, t.ex. om du vill trycka på returtangenten för att bara infoga en nyrad, inaktivera då det här alternativet.

Minimal ordlängd att komplettera

Medan du skriver in text, söker ordkomplettering efter ord i dokumentet som börjar med texten som redan skrivits in. Alternativet ställer in det minsta antal tecken som behövs för att aktivera ordkompletteringen och visa en kompletteringsruta.

Ta bort slutet vid komplettering

Ta bort slutet av ett föregående ord när kompletteringsobjektet väljes i en lista.

Nyckelordskomplettering

Om aktiverad, använder den inbyggda automatiska kompletteringen nyckelorden definierade av syntaxfärgläggningen.

7.1.3.5 Stavningskontroll

De här inställningsalternativen beskrivs i dokumentationen för modulen [Stavningskontroll](#) i systeminställningarna.

7.1.3.6 VI-inmatningsläge

Allmänt

Låt Vi-kommandon överskrida genvägar i Kate

Om markerad, överskrider Vi-kommandon KateParts inbyggda kommandon. Exempel: **Ctrl+R** gör om, och överskrider standardåtgärden (att visa dialogrutan för sökning och ersättning).

Visa relativa radnummer

Om det här är aktiverat, refererar aktuell rad alltid till rad 0. Rader ovanför och nedanför ökar radnumret relativt.

Tangentavbildning

Tangentavbildning används för att ändra betydelsen hos inskrivna tecken. Det låter dig flytta kommandon till andra tangenter, eller trycka på särskilda tangenter för att utföra en serie kommandon.

Exempel:

F2 -> **I--** Esc

Lägger till **!--** först på en rad när **F2** trycks ner.

7.1.4 Öppna/spara

7.1.4.1 Allmänt

Filformat

Kodning

Det här definierar standardkodningen som ska användas för att öppna och spara filer, om den inte ändras i dialogrutan för att öppna eller spara, eller genom att använda en kommandoradsväljare.

Detektering av kodning

Välj ett värde i kombinationsrutan, för att antingen inaktivera automatisk detektering eller använda **Allmän** för att aktivera automatisk detektering av alla kodningar. Men eftersom den troligen bara detekterar UTF-8 eller UTF-16, erhålls bättre resultat genom att välja ett område för att använda anpassade metoder. Om varken kodningen vald som standard ovan, eller kodningen angiven i dialogrutan för att öppna eller spara motsvarar filens innehåll, utförs den här detekteringen.

Reservkodning

Det här definierar reservkodningen att prova för att öppna filer, om kodningen definierad som standard ovan, eller den som anges via dialogrutan för att öppna eller spara, inte motsvarar filens innehåll. Innan den provas, görs ett försök att bestämma kodningen genom att läsa en möjlig byte-ordningsmarkering i början av filen. Om någon sådan hittas, väljes riktig Unicode-kodning, annars utförs kodningsdetektering. Om båda misslyckas provas reservkodningen.

Radens slut

Välj radslutsläget du föredrar för det aktiva dokumentet. Du kan välja bland UNIX®, DOS/Windows® eller MacIntosh.

Automatisk detektering av radslut

Markera det här om du vill att editorn automatiskt ska detektera radslutstyp. Den första radslutstypen som hittas används för hela filen.

Aktivera byte-ordningsmarkering (BOM)

Byte-ordningsmarkören är en speciell sekvens i början av dokument kodade med Unicode. Den hjälper editorer att öppna textdokument med riktig Unicode-kodning. För mer information, se [Byte Order Mark](#).

Radlängdsgräns

Tyvärr får KatePart dålig prestanda vid arbete med extremt långa rader, på grund av brister i Qt™. Av den orsaken bryter KatePart automatiskt rader när de är längre än antalet tecken som anges här. För att inaktivera det, ställ in den till 0.

Automatisk rensning vid spara

Ta bort mellanslag i slutet av en rad

Editorn tar automatiskt bort extra mellanslag i slutet på textrader när filen sparas. Du kan välja **Aldrig** för att inaktivera funktionen, **På ändrade rader** för att bara göra det på rader som du har ändrat sedan dokumentet senast sparades, eller **I hela dokumentet** för att ta bort dem ovillkorligt i hela dokumentet.

Lägg till radbrytning sist i filen när den sparas

Editorn lägger automatiskt till en nyrad i slutet av filen om det inte redan finns en när filen sparas.

Aktivera spara automatisk (bara för lokala filer)

Markera det om du vill att editorn ska spara dokument automatiskt medan du arbetar med dem.

Spara dokument automatiskt när fokus lämnar editorn

Editorn sparar automatiskt dokument när du byter till någonting utanför editorn, t.ex. terminalrutan i Kate.

Intervall för spara automatiskt

Du kan bestämma intervallet för spara automatiskt i sekunder här. Om intervallet är 0, sparas inte dokumentet efter ett intervall.

7.1.4.2 Avancerat

Skriv en säkerhetskopia vid spara för

Säkerhetskopiering när du sparar gör att KatePart kopierar filen på disk (den tidigare sparade filen) till <prefix><filnamn><suffix> innan de nya ändringarna sparas. En säkerhetskopia kan vara till hjälp för att återställa arbete om något går fel när du sparar, eller om du senare vill återställa den tidigare versionen av filen. Suffixet har normalt värdet ~ och prefixet är normalt tomt.

Lokala filer

Markera det här om du vill ha säkerhetskopior av lokala filer när du sparar.

Fjärrfiler

Markera det här om du vill ha säkerhetskopior av fjärrfiler när du sparar.

Prefix för säkerhetskopierade filer

Skriv in prefixet som ska läggas till i början av den säkerhetskopierade filens namn.

Suffix för säkerhetskopierade filer

Skriv in suffixet som ska läggas till sist i den säkerhetskopierade filens namn.

Växlingsfilsläge

KatePart kan återställa (det mesta) osparade arbetet i händelse av en krasch eller strömbrott. En växlingsfil (<filnamn>.kate-swp) skapas när ett dokument redigeras. Om användaren inte sparar ändringarna och KatePart kraschar, förblir växlingsfilen på disk. När en fil öppnas, kontrollerar KatePart om det finns en växlingsfil för dokumentet, och om det gör det, frågar användaren om förlorad data ska återställas eller inte. Användaren har också möjlighet att granska skillnaderna mellan originalfilen och den återställda filen. Växlingsfilen tas bort varje gång dokumentet sparas, och vid normal avslutning.

KatePart synkroniserar växlingsfilen på disk var 15:e sekund, men bara om det finns ändringar sedan den senaste synkroniseringen. Användaren kan inaktivera synkroniseringen av växlingsfiler om det önskas, genom att välja **Inaktivera**, men det kan leda till förlust av mer data.

När växlingsfilen är aktiverad är det möjligt att byta mellan två lägen, närmare bestämt **Aktiverad, lagra i förvald katalog** och **Aktiverad, lagra i egen katalog**.

Lagra växlingsfiler i

Normalt sparas växlingsfiler i samma katalog som filen. När **Aktiverad, lagra i egen katalog** väljes, skapas växlingsfiler i den angivna katalogen. Det är användbart för nätverksfilsystem för att undvika onödig nätverkstrafik.

Spara växlingsfiler var

KatePart synkroniserar växlingsfilen på disk var 15:e sekund, men bara om det finns ändringar sedan den senaste synkroniseringen. Du kan ändra synkroniseringsintervallet som du vill.

7.1.4.3 Lägen och filtyper

Den här sidan låter dig överskrida förvalda inställningar för dokument med angivna MIME-typer. När editorn laddar ett dokument, kontrollerar den om det matchar filmasken eller MIME-typerna för en av de definierade filtyperna, och om det är fallet tillämpas variablerna som är definierade. Om mer än en filtyp matchas, används den med högst prioritet.

Filtyp:

Filtypen med högst prioritet är den som visas först i kombinationsrutan. Om fler filtyper hittas, anges de också.

Ny

Det här används för att skapa en ny filtyp. Efter du klickat på knappen, blir fälten nedanför tomma och du kan fylla i de egenskaper du vill ha för den nya filtypen.

Ta bort

För att ta bort en befintlig filtyp, markera den i kombinationsrutan och tryck på knappen Ta bort.

Egenskaper för *aktuell filtyp*

Filtypen med högst prioritet är den som visas först i kombinationsrutan. Om fler filtyper hittas, anges de också.

Namn:

Filtypens namn är texten i motsvarande menyalternativ. Namnet visas i menyn **Verktyg** → **Filtyp**.

Sektion:

Sektionsnamnet används för att organisera filtyper i menyer. Det används också i menyn **Verktyg** → **Filtyp**.

Variabler:

Strängen låter dig ställa in KateParts inställningar för filer som väljes av den här Mime-typen genom att använda variabler i KatePart. Du kan ställa in nästan alla inställningsalternativ, såsom färgläggning, indenteringsläge, etc.

Klicka på **Redigera** för att se en lista med alla tillgängliga variabler och deras beskrivningar. Markera kryssrutan till vänster för att aktivera en viss variabel och ange därefter variabelns värde till höger. Vissa variabler tillhandahåller en kombinationsruta för att välja möjliga värden medan andra kräver att du matar in ett giltigt värde manuellt.

För fullständig information om dessa variabler, se [Anpassa med dokumentvariabler](#).

Färgläggning:

Om du skapar en ny filtyp, låter den här kombinationsrutan dig välja en filtyp för färgläggning.

Indenteringsläge:

Kombinationsrutan anger indenteringsläget för nya dokument.

Filändelser:

Jokerteckenmasken låter dig välja filer enligt filnamn. En typisk mask använder en asterisk och filändelsen, till exempel *.txt; *.text. Strängen är en lista med masker åtskilda av semikolon.

Mime-typer:

Visar en guide som hjälper dig att enkelt välja MIME-typer.

Prioritet:

Anger en prioritet för filtypen. Om fler filtyper väljer samma fil, används den med högst prioritet.

7.2 Anpassa med dokumentvariabler

Variabler i KatePart är KateParts implementering av dokumentvariabler, som liknar modelines i Emacs och Vi. I Kate-programdelen har raderna följande format: **kate: VARIABELNAMN VÄRDE; [VARIABELNAMN VÄRDE; . . .]**. Raderna kan förstås finnas i en kommentar, om filen har ett format med kommentarer. Variabelnamn är enstaka ord (inga blanktecken), och allt fram till nästa semikolon är värdet. Semikolonet är nödvändigt.

Här är ett exempel på en variabelrad, som styr indenteringsinställningar för en C++, Java™ eller JavaScript-fil:

```
// kate: replace-tabs on; indent-width 4; indent-mode cstyle;
```

NOT

Bara de första och sista 10 raderna söks igenom efter variabelrader.

Dessutom kan dokumentvariabler placeras i en fil som heter `.kateconfig` i vilken katalog som helst, och de angivna inställningarna verkställs som om modelines skrivits in i varje fil i katalogen och i dess underkataloger. Dokumentvariabler i `.kateconfig` använder samma syntax som modelines, men med [utökade alternativ](#).

Det finns variabler som stöder nästan alla inställningar i KatePart, och ytterligare insticksprogram kan använda variabler, som i så fall ska vara dokumenterade i insticksprogrammets dokumentation.

KatePart har stöd för att läsa inställningar från `.editorconfig`-filer, där biblioteket `editorconfig` är installerat. KatePart söker automatiskt efter `.editorconfig` så fort en fil öppnas. Den ger dock prioritet till `.kateconfig`-filer.

7.2.1 Hur KatePart använder variabler

När inställningarna läses, letar Kate-programdelen på följande ställen (i denna ordning):

- Den allmänna inställningen.
- Valfri sessionsdata.
- Inställningen under "filtyp".
- Dokumentvariabler i `.kateconfig`.
- Dokumentvariabler i själva dokumentet.
- Inställningar som görs under redigering från menyn eller kommandoraden.

Som du märker överskrids dokumentvariabler av ändringar som görs under körning. Så fort ett dokument sparas, läses dokumentvariablerna om, och skriver över ändringar som gjorts med menyalternativ eller från kommandoraden.

Alla variabler som inte anges nedan lagras i dokumentet och kan läsas av andra objekt, exempelvis insticksprogram, som kan använda dem för egna syften. Det variabelbaserade indenteringsläget använder till exempel dokumentvariabler för sin inställning.

Variablerna som anges här dokumenterar KatePart version 5.38. Fler variabler kan läggas till i framtiden. Det finns tre möjliga typer av värden för variabler, med följande giltiga uttryck:

- **BOOL** - `on` | `off` | `true` | `false` | `1` | `0`
- **HELTAL** - vilket heltal som helst
- **STRÄNG** - allt annat

7.2.2 Tillgängliga variabler

auto-brackets [BOOL]

Aktivera automatisk infogning av hakparenteser.

auto-center-lines [HELTAL]

Ange rader för automatisk centrering.

background-color [STRÄNG]

Ställ in dokumentets bakgrundsfärg. Värdet måste vara något som kan tolkas som en giltig färg, till exempel `#ff0000`.

backspace-indents [BOOL]

Aktivera eller inaktivera avindentering när backsteg trycks ner.

block-selection [BOOL]

Sätt på eller stäng av [blockmarkering](#).

bom | **byte-order-mark** | **byte-order-marker** [BOOL]

Aktivera eller inaktivera byte-ordningsmarkering (BOM) när filer sparas med Unicode-format (utf8, utf16, utf32).

Sedan: Kate 3.4 (KDE 4.4)

bracket-highlight-color [STRÄNG]

Ställer in färgen för markering av parenteser. Värdet måste vara något som kan tolkas som en giltig färg, till exempel `#ff0000`.

current-line-color [STRÄNG]

Ställer in färgen för nuvarande rad. Värdet måste vara något som kan tolkas som en giltig färg, till exempel **#ff0000**.

default-dictionary [STRÄNG]

Ställer in förvald ordlista använd vid stavningskontroll.

Sedan: Kate 3.4 (KDE 4.4)

dynamic-word-wrap [BOOL]

Sätter på eller stänger av [dynamisk radbrytning](#).

eol | end-of-line [STRÄNG]

Ställer in radslutsläget. Giltiga inställningar är **unix**, **mac** och **dos**.

folding-markers [BOOL]

Sätter på eller stänger av visning av [vikmarkörer](#).

folding-preview [BOOL]

Aktivera förhandsgranskning av vikning längs editorns kant.

font-size [HELTAL]

Ange dokumentets teckenstorlek i punkter.

font [STRÄNG]

Ställer in dokumentets teckensnitt. Värdet måste vara ett giltigt teckensnittsnamn, till exempel **courier**.

hl | syntax [STRÄNG]

Ställer in syntaxfärgläggning. Giltiga strängar är alla namn tillgängliga i menyerna. Skriv exempelvis helt enkelt **C++** för C++.

icon-bar-color [STRÄNG]

Ställer in ikonradens färg. Värdet måste vara något som kan tolkas som en giltig färg, till exempel **#ff0000**.

icon-border [BOOL]

Sätter på eller stänger av visning av ikonkanten.

indent-mode [STRÄNG]

Ställer in automatiskt indenteringsläge. Alternativen **none**, **normal**, **cstyle**, **haskell**, **lilypond**, **lisp**, **python**, **ruby** och **xml** känns igen. Se avsnittet Avsnitt 3.8 för detaljinformation.

indent-pasted-text [BOOL]

Aktivera eller inaktivera Justering av indentering för text som klistrats in från klippbordet.

Sedan: Kate 3.11 (KDE 4.11)

indent-width [HELTAL]

Ange indenteringsbredden.

keep-extra-spaces [BOOL]

Ställer in om extra mellanslag ska behållas när indenteringsbredden beräknas.

line-numbers [BOOL]

Sätt på eller stäng av visning av radnummer.

newline-at-eof [BOOL]

Lägg till en tom rad i slutet av filen (EOF) när dokumentet sparas.

Sedan: Kate 3.9 (KDE 4.9)

overwrite-mode [BOOL]

Sätter på eller stänger av överskrivningsläge.

persistent-selection [BOOL]

Sätt på eller stäng av [långlivade markeringar](#).

replace-tabs-save [BOOL]

Sätter på eller stänger av konvertering av tabulatortecken till mellanslag.

replace-tabs [BOOL]

Sätter på eller stänger av dynamisk konvertering av tabulatortecken till mellanslag.

remove-trailing-spaces [STRÄNG]

Tar bort avslutande mellanslag när dokumentet sparas. Giltig alternativ är:

- **none**, **-** eller **0**: ta aldrig bort avslutande mellanslag.
- **modified**, **mod**, **+** eller **1**: ta bara bort avslutande mellanslag på ändrade rader. De ändrade raderna markeras av radändringssystemet.
- **all**, ***** eller **2**: ta bort avslutande mellanslag i hela dokumentet.

scrollbar-minimap [BOOL]

Visa rullningslistens miniavbildning.

scrollbar-preview [BOOL]

Visa rullningslistens förhandsgranskning.

scheme [STRÄNG]

Ställer in färgschemat. Strängen måste vara namnet på ett färgschema som finns i dina inställningar för att ha någon effekt.

selection-color [STRÄNG]

Ställer in markeringsfärgen. Värdet måste vara något som kan tolkas som en giltig färg, till exempel **#ff0000**.

show-tabs [BOOL]

Sätter på eller stänger av synligt tabulatortecken.

smart-home [BOOL]

Sätter på eller stänger av [smart Home-navigering](#).

tab-indents [BOOL]

Sätter på eller stänger av indentering med tabulatortecken.

tab-width [HELTAL]

Ställer in tabulatorvisningsbredden.

undo-steps [HELTAL]

Ange antalet ångra-steg som ska kommas ihåg (= oändigt)

Observera: Användning avråds från sedan Kate 3 i KDE 4. Variabeln ignoreras. Det maximala antalet steg som kan ångras är obegränsat.

word-wrap-column [HELTAL]

Ställer in bredden för [hård radbrytning](#).

word-wrap-marker-color [STRÄNG]

Ställer in färgen som markerar radbrytningar. Värdet måste vara något som kan tolkas som en giltig färg, till exempel **#ff0000**.

word-wrap [BOOL]

Sätter på eller stänger av hård radbrytning.

7.2.3 Utökade alternativ i **.kateconfig**-filer

KatePart letar alltid efter en **.kateconfig**-fil för lokala filer (inte fjärrfiler). Dessutom är det möjligt att ange alternativ baserat på jokertecken (filändelser) enligt följande:

```
kate: tab-width 4; indent-width 4; replace-tabs on;
kate-wildcard(*.xml): indent-width 2;
kate-wildcard(Makefile): replace-tabs off;
```

I exemplet använder alla filer tabulatorbredd (tab-width) 4 mellanslag, indenteringsbredd (indent-width) 4 mellanslag och tabulatorer ersätts expanderade till mellanslag. För alla ***.xml**-filer är dock indenteringsbredden inställd till 2 mellanslag. Och Makefile-filer använder tabulatorer, dvs. tabulatorer ersätts inte med mellanslag.

Jokertecken skiljs åt av semikolon, dvs. det går också att ange flera filändelser enligt följande:

```
kate-wildcard(*.json;*.xml): indent-width 2;
```

Dessutom kan MIME-typer också användas för att matcha vissa filer, för att t.ex. indentera all C++ källkodsfiler med 4 mellanslag kan man skriva:

```
kate-mimetype(text/x-c++src): indent-width 4;
```

NOT

Förutom stödet i **.kateconfig**-filer, stöds också dokumentvariabler som beror på jokertecken och MIME-typer i själva filerna som kommentarer.

Kapitel 8

Tack till och licens

KatePart och KWrite Copyright 2001-2022 av Kate-gruppen.

Baserad på den ursprungliga KWrite, som var copyright 2000 av Jochen Wilhelmy digisnap@cs.tu-berlin.de

Bidragsgivare:

- Christoph Cullmann cullmann@kde.org
- Michael Bartl michael.bartl1@chello.at
- Philip phlip_cpp@my-deja.com
- Anders Lund anders@alweb.dk
- Matt Newell newellm@proaxis.com
- Joseph Wenninger kde@jowenn.at
- Jochen Wilhelmy digisnap@cs.tu-berlin.de
- Michael Koch koch@kde.org
- Christian Gebauer gebauer@kde.org
- Simon Hausmann hausmann@kde.org
- Glen Parker glenebob@nwlinc.com
- Scott Manson sdmanson@altel.net
- John Firebaugh jfirebaugh@kde.org
- Nibaldo González nibgonz@gmail.com

Dokumentationen om KatePart är baserad på den ursprungliga dokumentationen av KWrite, ändrad för att vara relevant för alla användare av KatePart.

Den ursprungliga dokumentationen av KWrite är skriven av Thad McGinnis ctmcginnis@compuserve.com, med många ändringar av Christian Tibirna tibirna@kde.org. Konvertering till docbook och korrekturläsning av Lauri Watts lauri@kde.org och uppdaterad av Anne-Marie Mahfouf annma@kde.org och Anders Lund anders@alweb.dk

Den nuvarande dokumentationen av KatePart underhålls av T.C. Hollingsworth thollingsworth@gmail.com. Skicka gärna kommentarer eller förslag till utvecklingslistan för KatePart på kwrite-devel@kde.org eller skicka in en felrapport med KDE:s felspårningssystem.

Översättning Stefan Asserhäll stefan.asserhall@bredband.net

Den här dokumentationen licensieras under villkoren i [GNU Free Documentation License](#).

Det här programmet licensieras under villkoren i [GNU General Public License](#).

Kapitel 9

Vi-inmatningsläget

Erlend Hamberg

Översättare: Stefan Asserhäll

9.1 Vi-inmatningsläge

Målet för VI-läget är inte att vara en fullständig ersättning av Vim och stödja alla funktioner i Vim. Dess mål är att göra 'Vim-sättet' att redigera (och vanorna inlärd med Vim) tillgängligt för program som använder texteditorn KatePart som intern editor.

VI-lägets mål är att integreras väl med programmet och skilja sig från Vims beteende när det är lämpligt. Exempelvis öppnas en dialogruta för att spara med **:w** i KateParts VI-läge.

För att aktivera VI-inmatningsläget för alla nya vyer, gå till **Inställningar** → **Anpassa KatePart...+Redigering** → **Vi-inmatningsläge**. Under denna flik kan du ställa in alternativ för VI-inmatningsläget samt definiera och redigera tangentbindningarna för läget. Vi-inmatningsläge kan också växlas med inställningen **Vi-inmatningsläge** i menyn **Redigera**. (Normal genväg är **Meta+Ctrl+V**, där **Meta** oftast är **Windows**-tangenten).

NOT

Många tangentbordskommandon i VI-läget är skiftlägeskänslig, i motsats till de flesta tangentbordsgenvägar i KDE. Det betyder att **y** och **Y** är olika kommandon. För att skriva in kommandot **y** (ryck ut), försäkra dig om att **Caps Lock** inte är aktivt och tryck på **Y**. För att skriva in kommandot **Y** (ryck ut till radens slut), tryck på **Skift+Y**.

Detta gäller inte för kommandon som använder **Ctrl**-tangenten, vilka kan skrivas in oberoende av värdet på **Caps Lock** och utan att hålla ner **Skift**. Dock kräver vissa kommandon användning av **Ctrl**-tangenten följt av en annan tangent som är skiftlägeskänslig. För att exempelvis skriva in '**Ctrl+Wh**' (byt till delad vy höger) försäkra dig om att **Caps Lock** inte är aktivt, tryck på **Ctrl+W**, släpp, och tryck därefter på **H**.

9.1.1 Funktioner inkompatibla med Vim

Det finns bara ett fåtal funktioner i KateParts VI-läge som är inkompatibla med Vim (om man inte räknar saknade funktioner). De listas nedan tillsammans med respektive orsak.

- KatePart: **U** och **Ctrl+R** är gör om.

Vim: **Ctrl+R** är vanlig gör om, **U** används för att ångra alla senaste ändringarna på en rad.

Orsaken att låta **U** fungera som ångra i KateParts VI-läge är att genvägen **Ctrl+R** normalt används av KateParts ersättningsfunktion (sök och ersätt). Normalt överskrider inte VI-läget KateParts genvägar (det kan ställas in i **Inställningar** → **Anpassa KatePart...+Redigering** → **Vi-inmatningsläge**), och därför måste åtgärden Gör om också vara tillgänglig som en 'vanlig' tangent. Dessutom motsvarar inte beteendet hos kommandot **U** i Vim KateParts interna ång-ringssystem väl, så det skulle hur som helst inte vara trivialt att stödja.

- KatePart: **print** visar dialogrutan **Skriv ut**.

Vim: **print** skriver ut raderna i det angivna intervallet, liksom i dess föregångare ed.

Kommandon som **:print** är inte bara tillgängliga i VI-läge, utan också för användare som använder KatePart på 'vanligt' sätt. Därför visar kommandot **:print** utskriftsdialogrutan, med principen att minimera överraskningar istället för att härma Vims beteende.

- KatePart: **Y** rycker ut till radens slut.

Vim: **Y** rycker ut hela raden, precis som **yy**.

Vims beteende för kommandot **Y** är i praktiken ett fel. Både för ändrings- och borttagningskommandon utför **cc/ dd** åtgärden för aktuell rad och **C/D** fungerar från markörens kolumn till radens slut. Dock rycker både **yy** och **Y** ut aktuell rad. I KateParts VI-läge rycker **Y** ut till radens slut. Det beskrivs som 'mer logiskt' i Vims [dokumentation](#).

- KatePart: **O** och **o** påbörjar [*antal*] nya rader och går till infogningsläge.

Vim: **O** och **o** påbörjar en ny rad och infogar text [*antal*] gånger när infogningsläget lämnas.

Det har i huvudsak gjorts som en följd av att ha stött på många som blivit förvirrade av beteendet på en Vim IRC-kanal (#vim på Libera Chat).

9.1.2 Byta lägen

- *Normalt läge* låter dig ange kommandon för att navigera eller redigera dokumentet, och är förvalt läge. Du kan återgå till det från vilket annat läge som helst genom att trycka på Esc.
- *Visningsläge* låter dig markera text i ett dokument. De flesta kommandon från normalläget är också giltiga i detta läget. Du kan gå till det genom att trycka på **v** för att markera tecken, eller **V** för att markera rader.
- *Infogningsläge* låter dig redigera dokumentet direkt. Du kan gå till det genom att trycka på **i** eller ett av flera andra kommandon som listas nedan.
- *Kommandoläget* aktiverar KateParts kommandorad, vilket möjliggör att köra många kommandon tillgängliga i Vi-implementeringar samt vissa specifika för KatePart. För mer information om dessa kommandon, se Avsnitt [5.2](#). För att använda det, tryck på **:**, skriv in kommandot och tryck på returtangenten.

9.1.3 Integrering med funktioner i Kate

- Visuellt läge aktiveras automatiskt när text markeras med musen. Det aktiveras också när funktioner i Kate som markerar text används, såsom Markera alla (antingen från menyn eller via **Ctrl+A**).
- Markörer i Vi och [bokmärken i Kate](#) är integrerade. När en markör skapas i VI-läge, skapas ett motsvarande bokmärke i Kate, och dyker upp i menyn **Bokmärken**. Omvänt, när ett bokmärke skapas i Kate, skapas en motsvarande markör i Vi på kolumn 0 också.

9.1.4 Kommandon som stöds i normalt och visningsläge

a	Gå till infogningsläge, lägg till efter markören
A	Gå till infogningsläge, lägg till efter raden
i	Gå till infogningsläge, lägg till före markören
Insert	Gå till infogningsläge, lägg till före markören
I	Gå till infogningsläge, infoga före första tecken på raden som inte är ett blanktecken
gi	Gå till infogningsläge, lägg till före stället där det senaste infogningsläget lämnades
v	Gå till visningsläge, markera tecken
V	Gå till visningsläge, markera rader
Ctrl+v	Gå till visningsläge, markera block
gb	Gå till visningsläge, markera den senaste markeringen igen
o	Påbörja en ny rad under den aktuella raden
O	Påbörja en ny rad ovanför den aktuella raden
J	Kombinera rader
c	Ändra: Följ med en förflyttning för att ta bort och gå till infogningsläge
C	Ändra till radens slut: Ta bort till radens slut och gå till infogningsläge
cc	Ändra rad: Ta bort till raden och gå till infogningsläge
s	Ersätt tecken
S	Ersätt rad
dd	Ta bort rad
d	Följ med en förflyttning för att ta bort
D	Ta bort till radens slut
x	Ta bort tecken till höger om markören
Delete	Ta bort tecken till höger om markören
X	Ta bort tecken till vänster om markören
gu	Följ med en förflyttning för att ändra till små bokstäver
guu	Gör den aktuella raden till små bokstäver
gU	Följ med en förflyttning för att ändra till stora bokstäver
gUU	Gör den aktuella raden till stora bokstäver
y	Följ med en förflyttning för att 'rycka ut' (kopiera)
yy	Ryck ut (kopiera) rad
Y	Ryck ut (kopiera) rad
p	Klistra in efter markören
P	Klistra in före markören
]p	Klistra in efter markören indenterat
[p	Klistra in före markören indenterat
r	Följ med ett tecken för att ersätta tecknet efter markören
R	Gå till ersättningsläge

:	Gå till kommandoläge
/	Sök
u	Ångra
Ctrl+R	Gör om
U	Gör om
m	Ange markering (kan användas av senare förflyttningar)
n	Sök igen
N	Sök föregående
>>	Indentera rad
<<	Avindentera rad
>	Indentera rader
<	Avindentera rader
Ctrl+F	Sida ner
Ctrl+B	Sida upp
ga	Skriv ut tecknets ASCII-värde
.	Upprepa senaste ändring
==	commandAlignLine
=	commandAlignLines
~	Ändra skiftläge för aktuellt tecken
Ctrl+S	Dela vyn horisontellt
Ctrl+V	Dela vyn vertikalt
Ctrl+W, w	Gå cykliskt till nästa delade fönster
Ctrl+W, h CtrlW Vänster	Gå till det delade fönstret till vänster
Ctrl+W, l CtrlW Höger	Gå till det delade fönstret till höger
Ctrl+W, k CtrlW Upp	Gå till det delade fönstret ovanför
Ctrl+W, j CtrlW Ner	Gå till det delade fönstret nedanför

9.1.5 Förflyttningar som stöds

Dessa kan användas för att flytta omkring i dokumentet i Normalt läge eller Visningsläge, eller i samband med något av ovanstående kommandon. De kan föregås av ett antal, vilket anger hur många av motsvarande förflyttningar som ska göras.

h	Vänster
Vänster	Vänster
Baksteg	Vänster
j	Ner
Ner	Ner
k	Upp
Upp	Upp
l	Höger
Höger	Höger
Mellanslag	Höger
\$	Radens slut

End	Radens slut
0	Radens första tecken (kolumn 0)
Home	Radens första tecken
^	Radens första tecken som inte är ett blanktecken
f	Följ av tecken för att flytta till höger om markören
F	Följ av tecken för att flytta till vänster om markören
t	Följ med tecken höger om markören att gå till, och placera markören på tecknet före det
T	Följ med tecken vänster om markören att gå till, och placera markören på tecknet före det
gg	Första raden
G	Sista raden
w	Nästa ord
W	Nästa ord åtskilt av blanktecken
b	Föregående ord
B	Föregående ord åtskilt av blanktecken
e	Ordets slut
E	Ordets slut åtskilt av blanktecken
ge	Föregående ords slut
gE	Föregående ords slut åtskilt av blanktecken
 	Följ med kolumnnummer för att gå till den kolumnen
%	Följ med ett objekt för att gå till det objektet
`	Markera
`	Första tecken på raden som inte är ett blanktecken där markeringen är
[[	Föregående inledande hakparentes
]]	Nästa inledande hakparentes
[]	Föregående avslutande hakparentes
] [	Nästa avslutande hakparentes
Ctrl+I	Gå till nästa position
Ctrl+O	Gå till föregående position
H	Gå till skärmens första rad
M	Gå till raden på skärmens mitt
L	Gå till skärmens sista rad
%procentvärde	Gå till angivet procentvärde av dokumentet
gk	Gå en rad uppåt visuellt (när dynamisk radbrytning används)
gj	Gå en rad neråt visuellt (när dynamisk radbrytning används)
Ctrl+Vänster	Gå ett ord åt vänster
Ctrl+Höger	Gå ett ord åt höger

9.1.6 Textobjekt som stöds

De kan användas för att markera vissa delar av ett dokument.

iw	Inre ord: Ord inklusive blanktecken
aw	Ett ord: Ord exklusive blanktecken
i"	Föregående dubbla citationstecken (") till nästa dubbla citationstecken, inklusive citationstecken
a"	Föregående dubbla citationstecken (") till nästa dubbla citationstecken, exklusive citationstecken
i'	Föregående enkla citationstecken (') till nästa enkla citationstecken, inklusive citationstecken
a'	Föregående enkla citationstecken (') till nästa enkla citationstecken, exklusive citationstecken
i (	Föregående inledande parentes [(] till nästa avslutande parenthesis [)], inklusive parenteserna
a (	Föregående inledande parentes [(] till nästa avslutande parentes [)], exklusive parenteserna
i [	Föregående inledande hakparentes ([] till nästa avslutande hakparentes (]), inklusive parenteserna
a [	Föregående inledande hakparentes ([] till nästa avslutande hakparentes (]), exklusive parenteserna
i {	Föregående inledande klammerparentes ({ } till nästa avslutande klammerparentes (}), inklusive parenteserna
a {	Föregående inledande klammerparentes ({ } till nästa avslutande klammerparentes (}), exklusive parenteserna
i <	Föregående inledande vinkelparentes (< >) till nästa avslutande vinkelparentes (>), inklusive parenteserna
a <	Föregående inledande vinkelparentes (< >) till nästa avslutande vinkelparentes (>), exklusive parenteserna
i `	Föregående grava accent (`) till nästa grava accent, inklusive accenterna
a `	Föregående grava accent (`) till nästa grava accent, exklusive accenterna

9.1.7 Kommandon i infogningsläge som stöds

Ctrl+D	Avindentera
Ctrl+T	Indentera
Ctrl+E	Infoga nerifrån
Ctrl+Y	Ta bort ord

Ctrl+W	Ta bort ord
Ctrl+U	Ta bort rad
Ctrl+J	Ny rad
Ctrl+H	Ta bort tecken bakåt
Ctrl+Home	Gå till första tecknet i dokumentet
Ctrl+R n	Infoga innehållet i register n
Ctrl+O , <i>kommando</i>	Gå till normalt läge för endast ett kommando
Ctrl+A	Inkrementera talet som för närvarande är markerat
Ctrl+X	Dekrementera talet som för närvarande är markerat

9.1.8 Komma-textobjektet

Objektet saknas i Vim. Komma-textobjektet gör det enkelt att ändra parameterlistor i C-liknande språk och andra listor åtskilda med kommatecken. Det är egentligen området mellan två kommatecken eller mellan ett kommatecken och en parentes. På raden som visas i illustrationen är de tre områden som textobjektet kan omfatta färglagda.

```
int f(int arg1, double arg2, char arg3);
```

*Intervall för komma-textobjekt. Om markören exempelvis är på `arg2`, skulle **ci**, ('ändra inre kommatecken') ta bort `double arg2` och placera markören mellan de två kommatecknen i infogningsläge. Ett mycket bekvämt sätt att ändra en funktions parametrar.*

9.1.9 Saknade funktioner

Som tidigare nämnts, är inte målet med KateParts VI-läge att stödja 100 % av Vims funktioner.

Bilaga A

Reguljära uttryck

Detta appendix innehåller en kortfattad, men förhoppningsvis tillräcklig och heltäckande introduktion till de *reguljära uttryckens* värld. Det dokumenterar reguljära uttryck på den form som är tillgänglig i KatePart, som inte är likadana som reguljära uttryck i Perl, och inte heller i till exempel **grep**.

A.1 Inledning

Reguljära uttryck ger oss en möjlighet att beskriva ett möjligt innehåll i en textsträng på ett sätt som enkelt kan förstås av programvara, så att den kan undersöka om texten matchar, och för mer avancerade tillämpningar, ge möjlighet att spara delar av texten som matchar.

Ett exempel: Säg att du vill leta i en text efter stycken som börjar med något av namnen 'Henrik' eller 'Pernilla', följt av någon form av verbet 'säga'.

Med en normal sökning, skulle du börja med att leta efter det första namnet, 'Henrik', kanske följt av 's' på det här sättet: **Henrik s**, och medan du letar efter träffar måste du bortse från alla de som inte inleder ett stycke, samt de där ordet som börjar med 's' inte var antingen 'säger', 'sade' etc. Och därefter måste du förstås repetera allt för nästa namn ...

Med reguljära uttryck, kan den här uppgiften utföras med en enda sökning, och med högre noggrannhet.

För att åstadkomma detta, definierar reguljära uttryck detaljerade regler för att beskriva en generell sträng som ska matcha. Vårt exempel, som vi kan beskriva så här med ord: 'En rad som börjar antingen med 'Henrik' eller 'Pernilla' (möjligen efter upp till fyra mellanslag eller tabulatorer), följt av ett mellanslag, följt av 's' och sedan antingen 'äger' eller 'ade', skulle kunna uttryckas med följande reguljära uttryck:

```
^[ \t]{0,4}(Henrik|Pernilla) s(äger|ade)
```

Ovanstående exempel demonstrerar alla fyra huvudkoncepten i moderna reguljära uttryck, nämligen:

- Mönster
- Påståenden
- Kvalificering
- Bakåtreferenser

Hatten (^) som inleder uttrycket är ett påstående, som bara är sant om den följande strängen som matchar är först på raden.

Strängarna [\t] och (Henrik|Pernilla) s(äger|ade) är mönster. Det första är en *teckenklass* som antingen matchar ett blanktecken eller ett (horisontellt) tabulatorstecken. Det andra mönstret innehåller först ett delmönster som antingen matchar Henrik *eller* Pernilla, därefter en del som exakt matchar strängen s och slutligen ett delmönster som antingen matchar äger *eller* ade.

Strängen {0,4} är en kvalificering som säger 'allt från noll upp till fyra av det föregående'.

Eftersom programvara för reguljära uttryck stöder konceptet med *bakåtreferenser*, så sparas hela den matchande delen av strängen, samt delmönster som finns mellan parenteser. Givet något sätt att komma åt de här referenserna, skulle vi kunna få tag på hela matchningen (när man söker i ett textdokument i en editor med ett reguljärt uttryck, så visas den ofta markerad), eller namnet som hittades, eller den sista delen av verbet.

I sin helhet, så matchar uttrycket vad vi ville att det skulle göra, och bara det.

Följande avsnitt beskriver i detalj hur man skapar och använder mönster, teckenklasser, påståenden, kvalificering och bakåtreferenser. Det sista avsnittet ger några användbara exempel.

A.2 Mönster

Mönster består av vanliga strängar och teckenklasser. Mönster kan innehålla delmönster, vilket är mönster som är inneslutna i parenteser.

A.2.1 Undanta tecken

I mönster och teckenklasser har vissa tecken en särskilt mening. För att uttryckligen matcha något av de här tecknen, måste de markeras eller *undantas* för att tala om för programvaran för reguljära uttryck att den ska tolka sådana tecken uttryckligen.

Det här görs genom att lägga till ett bakstreck (\) före tecknet.

Programvaran för reguljära uttryck ignorerar tecken som undantas även om de inte har någon särskild betydelse i sammanhanget, så att undanta till exempel ett 'j' (\j) är säkert. Om du är tveksam om ett tecken kan ha en särskild betydelse, kan du därför alltid undanta det på ett säkert sätt.

Undantag omfattar förstås också bakstrecket själv, för att uttryckligen matcha ett sådant, skulle du skriva \\.

A.2.2 Teckenklasser och förkortningar

En *teckenklass* är ett uttryck som matchar ett tecken i en mängd angivna tecken. I reguljära uttryck definieras teckenklasser genom att ange de giltiga tecknen i klassen inom hakparenteser, [], eller genom att använda en av de förkortade klasserna som beskrivs nedan.

Enkla teckenklasser innehåller bara uttryckligen ett eller flera tecken, till exempel [abc] (som matchar något av tecknen 'a', 'b' eller 'c') eller [0123456789] (som matchar alla siffror).

Eftersom bokstäver och siffror har en logisk ordning, kan du förkorta dem genom att ange intervall: [a-c] är samma sak som [abc] och [0-9] är samma sak som [0123456789]. Att kombineras dessa sammansättningar, till exempel med [a-fynot1-38] är fullständigt riktigt (detta skulle förstås matcha antingen 'a','b','c','d','e','f','y','n','o','t','1','2','3' eller '8').

Eftersom stora bokstäver är andra tecken än motsvarande små bokstäver, så måste du alltså skriva [aAbB], för att skapa en teckenklass som matchar både de små och stora bokstäverna 'a' eller 'b'.

Det är förstås möjligt att skapa en 'negativ' klass som matchar 'allting utom'. För att göra detta lägg till en hatt (^) först i klassen:

`[^abc]` matchar alla tecken *utom* 'a', 'b' eller 'c'.

Förutom uttryckliga tecken, definieras några förkortningar, som gör livet lite enklare:

`\a`

Det här matchar ASCII-tecknet BEL (0x07).

`\f`

Det här matchar ASCII-tecknet ny sida (FF, 0x0C).

`\n`

Det här matchar ASCII-tecknet nyrad (LF, 0x0A, Unix nyrad).

`\r`

Det här matchar ASCII returtecken (CR, 0x0D).

`\t`

Det här matchar ASCII-tecknet horisontell tabulator (HT, 0x09).

`\v`

Det här matchar ASCII-tecknet vertikal tabulator (VT, 0x0B).

`\xhhh`

Det här matchar Unicode-tecknet som motsvarar det hexadecimala värdet hhhh (mellan 0x0000 och 0xFFFF). `\0ooo` (dvs. `\noll ooo`) matchar ASCII/Latin-1 tecknet som motsvarar det oktala värdet ooo (mellan 0 och 0377).

. (punkt)

Det här matchar vilket tecken som helst (inklusive nyrad).

`\d`

Det här matchar en siffra. Samma som `[0-9]`.

`\D`

Det här matchar allt utom en siffra. Samma som `[^0-9]` eller `[^\d]`.

`\s`

Det här matchar ett blanktecken. Nästan samma sak som `[ \t\n\r]`.

`\S`

Det här matchar allt utom ett blanktecken. Nästan samma sak som `[^ \t\r\n]`, och samma som `[^\s]`.

`\w`

Matchar alla 'ord-tecken' - i det här fallet alla bokstäver, siffror eller understreck. Samma som `[a-zA-Z0-9_]`.

`\W`

Matchar alla tecken som inte ingår i ord - allt utom bokstäver, siffror eller understreck. Samma som `[^a-zA-Z0-9_]` eller `[^\w]`.

POSIX klassnotation, `[<class namn>:]` stöds också. Exempelvis är `[:digit:]` ekvivalent med `\d`, och `[:space:]` med `\s`. Se den fullständiga listan över POSIX teckenklasser [här](#).

De förkortade klasserna kan användas inne i en annan klass. För att till exempel matcha ett ord-tecken, ett mellanslag eller en punkt, skulle du kunna skriva `[\w \.]`.

A.2.2.1 Tecken med särskild betydelse inne i teckenklasser

Följande tecken har en särskild betydelse inne i konstruktionen '[' för teckenklasser, och måste undantas för att användas uttryckligen i en teckenklass:

]

Avslutar teckenklassen. Måste undantas om det inte är det allra första tecknet i klassen (kan följa en hatt som inte undantas).

^ (hatt)

Anger en negativ klass, om det är det första tecknet. Måste undantas för att uttryckligen matcha om det är det första tecknet i klassen.

– (minus)

Anger ett logiskt intervall. Måste alltid undantas inne i en teckenklass.

\ (bakstreck)

Undantagstecknet. Måste alltid undantas.

A.2.3 Alternativ: matchar 'en av'

Om du vill matcha en av ett antal olika mönster, kan du skilja dem åt med | (ett vertikalt streck). För att till exempel antingen hitta 'John' eller 'Harry' kan du använda uttrycket **John|Harry**.

A.2.4 Delmönster

Delmönster är mönster som innesluts i parenteser, och som har flera användningar i de reguljära uttryckens värld.

A.2.4.1 Ange alternativ

Du kan använda ett delmönster för att gruppera ett antal alternativ i ett större mönster. Alternativen delas av tecknet '|' (vertikalt streck).

För att till exempel matcha antingen orden 'int', 'float' eller 'double', skulle du kunna använda mönstret **int|float|double**. Om du bara vill hitta ordet om det följs av blanktecken och därefter några bokstäver, placera alternativen i ett delmönster: **(int|float|double)\s+\w+**.

A.2.4.2 Spara text som matchar (bakåtreferenser)

Om du vill använda en bakåtreferens, använd ett delmönster **(MÖNSTER)** för att komma ihåg önskad del av mönstret. För att förhindra att delmönstret lagras, använd en icke-lagrande grupp **(?:PATTERN)**.

Om du till exempel vill hitta två förekomster av samma ord åtskilda av ett kommatecken och möjligen några blanktecken, skulle du kunna skriva **(\w+),\s*\1**. Delmönstret **\w+** skulle hitta en grupp med ord-tecken, och hela uttrycket skulle matcha om de följdes av ett komma, noll eller flera blanktecken och sedan en likadan grupp med ord-tecken. (Strängen **\1** refererar till *det första delmönstret inneslutet i parenteser*).

NOT

För att undvika tvetydigheter vid användning av \1 med efterföljande siffror (t.ex. kan \12 vara det tolfte delmönstret eller bara det första delmönstret med 2) använder vi \{12} som syntax för delmönster med flera siffror.

Exempel:

- \{12}1 är 'använd delmönster 12'
- \123 är 'använd delmönster 1 därefter 23 som normal text'

A.2.4.3 Påståenden för sökning framåt

Ett påstående för sökning framåt är ett delmönster, som antingen börjar med ?= eller ?!.

För att till exempel uttryckligen matcha strängen 'Bill' men bara om den inte följs av ' Gates', skulle du kunna använda det här uttrycket: **Bill(?! Gates)**. (Det här skulle hitta 'Bill Clinton' samt 'Billy the kid', men tyst ignorera de andra träffarna).

Delmönster som används för påståenden sparas inte.

Se också [Påståenden](#).

A.2.4.4 Påståenden för sökning bakåt

Ett påstående för sökning bakåt är ett delmönster, som antingen börjar med ?<= eller ?<!.

Sökning bakåt har samma effekt som sökning framåt, men fungerar baklänges. För att exempelvis matcha den exakta strängen 'frukt' men bara om den inte föregås av 'grape', skulle du kunna använda det här uttrycket: **(?<!grape) frukt**.

Delmönster som används för påståenden sparas inte.

Se också [Påståenden](#).

A.2.5 Tecken med särskild betydelse inne i mönster

Följande tecken har särskild betydelse inne i mönster, och måste undantas om du uttryckligen vill matcha dem:

\ (bakstreck)

Undantagstecknet.

^ (hatt)

Anger början på strängen.

\$

Anger slutet på strängen.

() (vänster och höger parentes)

Anger delmönster.

{ } (vänster och höger klammer)

Anger numerisk kvalificering.

[] (vänster och höger hakparentes)

Anger teckenklasser.

| (vertikalt streck)

Logiskt ELLER. Skiljer alternativ.

+ (plustecken)

Kvalificering, en eller flera.

*** (asterisk)**

Kvalificering, noll eller flera.

? (frågetecken)

Ett extra tecken. Kan tolkas som en kvalificering, noll eller ett.

A.3 Kvalificering

En *kvalificering* låter ett reguljärt uttryck antingen matcha ett angivet antal eller intervall av ett tecken, en teckenklass eller ett delmönster.

En kvalificering innesluts av klamrar (`{` och `}`) och har den allmänna formen `{[minimalt antal] [, [maximalt antal]]}`.

Användningen förklaras bäst med exempel:

`{1}`

Exakt en förekomst

`{0, 1}`

Noll eller en förekomst

`{, 1}`

Samma sak, med mindre ansträngning ;)

`{5, 10}`

Minst 5 men maximalt 10 förekomster

`{5, }`

Minst 5 förekomster, inget maximum

Det finns dessutom några förkortningar:

*** (asterisk)**

liknar `{0, }`, hittar ett godtyckligt antal förekomster.

+ (plustecken)

liknar `{1, }`, åtminstone en förekomst.

? (frågetecken)

liknar `{0, 1}`, noll eller en förekomst.

A.3.1 Glupskt beteende

När kvalificeringar utan maximum används, så matchar reguljära uttryck normalt så mycket av söksträngen som möjligt, ett beteende som ofta kallas för *glupskt*.

Modern programvara för reguljära uttryck tillhandahåller en möjlighet att 'stänga av det glupska beteendet', även om det är det grafiska gränssnittet som måste ge dig tillgång till den här funktionen. En sökdialogruta som tillhandahåller sökning med reguljära uttryck skulle kunna ha en kryssruta som heter 'Minimal matchning', och borde också ange att glupskt beteende är det normala.

A.3.2 Exempel i sammanhang

Här är några exempel på att använda kvalificering:

`^\d{4,5}\s`

Matchar siffrorna i '1234 gå' och '12345 nu', men varken '567 elva' eller '223459 någonstans'.

`\s+`

Matchar en eller flera blanktecken.

`(bla){1,}`

Matchar samtliga 'blablabla' och 'bla' i 'blad' eller 'dubbla'.

`/?>`

Matchar '/>' i '<slut/>' samt '>' i '<start>'.

A.4 Påståenden

Påståenden gör att ett reguljärt uttryck bara matchar under vissa kontrollerade omständigheter.

Ett påstående behöver inte ett tecken för att matcha, utan undersöker istället omgivningen runt en möjlig matchning innan den erkänns. Påståendet *ordgräns* försöker till exempel inte hitta ett tecken som inte ingår i ett ord intill ett ord-tecken på platsen, utan kontrollerar istället att det inte finns ett ord-tecken. Det här betyder att påståendet matchar där det inte finns något tecken alls, dvs. vid slutet på söksträngen.

Vissa påståenden har verkligen ett mönster som ska matcha, men den delen av strängen som matchar kommer inte att ingå i resultatet för matchningen av hela uttrycket.

Reguljära uttryck som de dokumenteras här stöder följande påståenden:

^ (hatt: början på strängen)

Matchar början på söksträngen.

Uttrycket **^Peter** matchar vid 'Peter' i strängen 'Peter, hej då!' men inte 'Hej, Peter!'.

\$ (slutet på strängen)

Matchar slutet på söksträngen.

Uttrycket **du\?\$** matchar det sista du i strängen 'Du gjorde det väl inte, eller var det verkligen du?' men ingenstans i 'Du gjorde det väl inte, eller hur?'.

\b (ordgräns)

Matchar om det finns ett ordtecken på ena sidan och något annat än ett ordtecken på andra sidan.

Det här är användbart för att hitta slutet på ord, till exempel för både början och slutet för att hitta ett helt ord. Uttrycket **\bin\b** matchar ett ensamt 'in' i strängen 'Han kom in genom ingången', men inte det 'in' som finns i 'ingången'.

\B (icke ordgräns)

Matchar överallt där '\b' inte gör det.

Det här betyder att det matchar till exempel inne i ord: Uttrycket **\Bin\B** matchar in i 'minne' men inte i 'ingång' eller 'Han kom in genom fönstret'.

(?=MÖNSTER) (Positiv sökning framåt)

Ett påstående som söker framåt tittar på strängen som följer en möjlig matchning. Den positiva sökningen förhindrar att strängen matchar om inte texten som följer den möjliga matchningen matchar *MÖNSTER* i påståendet, men texten som matchas av det ingår inte i resultatet.

Uttrycket **vakt** **(?=\\w)** matchar för 'vakt' in 'vaktmästare' men inte i 'Han är en vakt!'

(?!MÖNSTER) (Negativ sökning framåt)

Den negativa sökningen framåt förhindrar en möjlig matchning från att bekräftas om den följande delen av söksträngen matchar sitt *MÖNSTER*.

Uttrycket **const** **\\w+\\b(?!\\s*&)** matchar vid 'const char' i strängen 'const char* x' medan den inte kan matcha 'const QString' i 'const QString& y' eftersom '&' matchar mönstret i den negativa sökningen framåt.

(?<=MÖNSTER) (Positiv sökning bakåt)

Sökning bakåt har samma effekt som sökning framåt, men fungerar baklänges. Ett påstående som söker bakåt tittar på strängen som föregår en möjlig matchning. Den positiva sökningen bakåt matchar bara en sträng om den föregås av *MÖNSTER* i påståendet, men texten som matchas av det ingår inte i resultatet.

Uttrycket **(?<=små)** **kaka** matchar vid 'kaka' om det föregås av 'små' (i 'småkaka' men inte av 'ostkaka' eller för bara 'kaka').

(?<!MÖNSTER) (Negativ sökning bakåt)

Den negativa sökningen bakåt förhindrar en möjlig matchning från att bekräftas om den föregående delen av söksträngen matchar sitt *MÖNSTER*.

Uttrycket **(?<![\\w\\.])** **[0-9]+** matchar vid '123' i strängarna '=123' och '-123' medan det inte kan matcha '123' i '.123' eller 'ord123'.

(MÖNSTER) (Spara grupp)

Delmönstret inom parenteser lagras och blir ihågkommet, så att det kan användas i bakåtreferenser. Exempelvis matchar uttrycket **("+)** **[^"]*\\1** **~~~~text~~~~** och **~~~~text~~~~**.

Se avsnittet [Spara text som matchar \(bakåtreferenser\)](#) för mer information.

(?:MÖNSTER) (Spara inte grupp)

Delmönstret inom parenteser lagras inte och blir inte ihågkommet. Det är bättre att alltid använda icke-lagrande grupper om det som platsmarkörerna inte ska användas.

Bilaga B

Sakregister

A
avkommentera, [32](#)

E
ersätt, sed-stil
sök, sed-stil, [36](#)

K
kommentera, [32](#)