



Vim-Konfiguration für Perl

Autor: Rüdiger Pretzlaff

E-Mail: ruediger AT ruhr.pm.org

Datum: 22. November 2007

<http://ruhr.pm.org/>



Ruhr . pm

Vorbereitungen

- Anlegen eines neuen Benutzers 'neo' für diese Beispiele
 - Vim-Konfiguration in `/etc/vimrc` wird benutzt
 - Benutzerspezifische Konfiguration in `~/vimrc`
- Verzeichnisse anlegen:

```
[neo@matrix ~]$ pwd
/home/neo
[neo@matrix ~]$ # User-specific Vim configuration
[neo@matrix ~]$ touch .vimrc
[neo@matrix ~]$ mkdir .vim
[neo@matrix ~]$ mkdir .vim/colors
[neo@matrix ~]$ mkdir .vim/compiler
[neo@matrix ~]$ mkdir .vim/doc
[neo@matrix ~]$ mkdir .vim/ftplugin
[neo@matrix ~]$ mkdir .vim/plugin
[neo@matrix ~]$ mkdir .vim/syntax
```



Grundlegende Konfiguration

- `:set wrap`
 - Beim Erreichen des rechten Bildschirmrandes wird der Text umgebrochen und in der nächsten Zeile weiter geschrieben.
- `:set nowrap`
 - Bei nowrap wird die Zeile nach rechts verschoben und die Eingabe wird in der gleichen Zeile hinzugefügt.
- Der Doppelpunkt kann in den Konfigurationsdateien weggelassen werden



Ruhr . pm

Flexible Tabs

- `:set tabstop=4`
 - Anzahl Leerzeichen, die einem Tab entsprechen
- `:set autoindent`
 - Bewirkt, dass der Cursor nach Drücken der Eingabetaste wieder an die gleiche Stelle gesetzt wird, wie in der vorherigen Zeile.
 - Autoindent ist immer dann sinnvoll, wenn Schleifen oder Funktionen programmiert werden.
 - Mehr dazu: `:help autoindent`



Ruhr .pm

Zeilennummerierung

- `:set nu`
 - Um Zeilennummern anzuzeigen wird in der `.vimrc` “`nu`” bzw. “`number`” gesetzt.
- `:set nonu`
 - Deaktivieren der Option Zeilennummerierung



Ruhr . pm

Syntax Highlighting

- `:syntax enable`
 - Aktiviert die Hervorhebung der Syntax
- `:set bg=dark`
 - Bei der Voreinstellung werden Kommentare dunkelblau angezeigt und sind auf dunklen Hintergründen schwer zu lesen. Diese Option zeigt Kommentare heller an.



Ruhr . pm

Textcompletion

- Ein weiteres Feature ist Textcompletion. Beim Schreiben kann mit Ctrl+n/Ctrl+p durch die Liste der Möglichkeiten gesprungen werden.
- :ctags *.p?
 - Generiert ctags aus allen Perl-Scripten und Perl-Modulen



Ruhr . pm

Window Splitting

- Manchmal ist es nötig, andere Dateien gleichzeitig in Vim geöffnet zu haben. Hierfür gibt es die Möglichkeit, mehrere Dateien im sogenannten Splitmode zu öffnen. Um waagerecht zu splitten, gibt man im Commandmode ein:

- `:split <dateiname>`
 - horizontal

Für senkrechtes Splitting benutzt man:

- `:vsplit <dateiname>`
 - vertikal



Ruhr . pm

Window Splitting ctd.

- Zwischen den Fenstern kann nun mit `ctrl+w w` gewechselt werden.
- Ein nicht mehr benötigtes Fenster wird mit `:close` geschlossen.
- Alle Fenster schließen, bis auf das aktuelle ausgewählte, erledigt `:only`



Ruhr . pm

Diff Mode

Zwei Dateien lassen sich mit Hilfe von “vimdiff” bzw. “vim -d” miteinander vergleichen:

- vimdiff file1.pl file2.pl
- vim -d file1.pl file2.pl
- Vim zeigt beide Dateien im Splitmode an und markiert Unterschiede farblich.
- Mit Hilfe von "do" und "dp" kann man die Änderungen im aktuellen Fenster in den Buffer schreiben bzw. aus dem Buffer ins aktuelle Fenster.
- Mit "do" werden Änderungen in den Buffer geschrieben. Mit "dp" werden die Zeilen aus dem Buffer ins aktuelle Fenster übernommen.



Ruhr . pm

Textfolding

- Viele Editoren und IDEs unterstützen Folding. Es macht langen Quellcode übersichtlich, da es Code gruppiert.
- Vim unterstützt 6 Methoden beim Folding.
 - Indent
 - Expression
 - Syntax
 - Changes in the text
 - Marker
 - Manual



Ruhr .pm

Textfolding ctd.

- Als Beispiel wird in die `.vimrc` folgendes eingetragen:

```
" Folding configuration"
set foldmethod=marker
let perl_fold=1
let perl_fold_blocks=2

" Edit and uncomment next line if you want non-default marker
"set foldmarker="{{{,}}}"
hi Folded ctermbg=black ctermfg=cyan
set expandtab
```



Ruhr . pm

Marks

- Mit Hilfe von Marks können Zeilen oder Bereiche in einer oder auch mehreren Dateien markiert werden. So kann zwischen diesen markierten Stellen hin und her gesprungen werden.
- In der Hilfe wird beschrieben, wie man Marks setzen und zwischen diesen Stellen hin und her springen kann.



Ruhr .pm

Vertical Indent Display

- Manche Benutzer wollen sich die Tabstops anzeigen lassen. Dies kann mit der Einstellung "list" und "listchars" in der .vimrc gesetzt werden.

```
" This is for vertical indenting"  
set list  
  
" set listchars=tab:|
```



Ruhr . pm

Color Schemes

- Auf der Vim Homepage gibt es eine große Auswahl an Color Schemes. Einfach von http://www.vim.org/scripts/script.php?script_id=625 das SampleColorPack herunterladen und Entpacken. Nach dem Entpacken ist der Ordner "colors" in das Verzeichnis `~/.vim` zu kopieren.
- Anschliessend muss noch "dos2unix" im Verzeichnis "colors" ausgeführt werden.

```
cd ~/.vim/  
colors  
dos2unix *  
cd -
```

- Jetzt kann im Vim eines der Colorschemes ausgewählt werden mit:
 - `:colorscheme`
- Hat man einen Favoriten gefunden kann das Schema in die `~/.vimrc` eingetragen werden mit:
 - `:colorscheme matrix`



Ruhr .pm

Block Comment

- Von Chris Russel
(http://www.vim.org/scripts/script.php?script_id=473)

```
1 #!/usr/bin/perl -w
2
3 use Debug;
4 use strict;
5
6 # Words that I want to be printed out.
7 my @words = ("one", "two", "three", "four");
8 my $debug = new Debug;
9
10
11 -----
12 # foreach my $word (@words) {
13 # | $debug->print_message(message=>"Inside loop");
14 # | print_word($word) or die "Could not print $word";
15 # }
16 #-----
17
18 =head1 print_word()
19
20 Output a single line containing specified word.
21
22 =cut
23
```

11.1

Top



Ruhr .pm

Embperl Syntax Highlighting

- Von Lukas Zapletal
(http://www.vim.org/scripts/script.php?script_id=61)

```
1 [-
2     my $page_title = "Foo Bar";
3     my $user_name = "";
4     if (defined($fdat{user_name})) {
5         $user_name = $fdat{user_name};
6     }
7     else {
8         $user_name = "Blah";
9     }
10 -]
11 <HTML>
12     <HEAD>
13         <TITLE>[+ $page_title +]</TITLE>
14     </HEAD>
15     <BODY BGCOLOR="white">
16         Hello, [+ $user_name +]
17         This is my test Embperl file<BR>
18     </BODY>
19 </HTML>
```

~/src/perl-project/test.epl" 19L, 321C 19.7 All



Ruhr.pm

Perl Syntax Compiler

- Von Lukas Zapletal
(http://www.vim.org/scripts/script.php?script_id=56)

```
4 use strict;
5
6 # Words that I want to be printed out.
7 my @words = ("one", "two", "three", "four");
8 my $debug = new Debug;
9
10
11 foreach my $word (@words) {
12     $debug->print_message(message=>"Inside loop")
13     print_word($word) or die "Could not print $word";
14 }
15
16 =head1 print_word()
17
18 Output a single line containing specified word.
19
20 =cut
21
22 sub print_word {
23     my $word_to_print = shift;
24 }
25
:clist
1 main.pl:13: Bareword found where operator expected
2 main.pl:13: syntax error
Hit ENTER or type command to continue
```

