

Texteditor

Ein Texteditor ist eines der grundlegendsten Werkzeuge eines Programmierers. Die Entscheidung, ob für unterschiedliche Sprachen jeweils spezialisierte Editoren verwendet werden sollen bzw. ein möglichst mit allen Sprachen zurechtkommender Editor verwendet wird, sei jedem selbst überlassen. Wählt man letztere Alternative, so ist besondere Sorgfalt bei der Entscheidung angebracht. Im Folgenden schildere ich einige Funktionen des Editors meiner Wahl - GNU Emacs. Die Schilderungen dienen als Follow-up eines Vortrages, in welchem ich im Zuge der Grazer Linxstage 2011 ein ausgewähltes Subset an Features des Editors vorgestellt habe. Das Thema des Vortrags war Emacs als IDE, wobei besonders auf Webentwicklung inklusive Python und Django eingegangen wurde. Dazu wurden 4 hilfreiche Pakete vorgestellt, welche hier kurz erklärt werden.

Gnuserv und Gnuclient

Beschreibung

Ermöglichen einen schnellen Start des Editors, wenn Emacs als Serverprozess läuft.

Installation und Konfiguration

Falls nicht ohnehin in der verwendeten Version von Emacs vorhanden, können die relevanten Pakete mittels

```
sudo aptitude install gnuserv
```

installiert werden. Danach ist ein aktivieren des Servermodus in `.emacs` erforderlich

```
;; Start GNUServe process when starting up. This lets us send new files
;; to previously spawned emacs process.
(load "gnuserv-compat")
(load-library "gnuserv")
(gnuserv-start)
```

Um Emacs danach automatisch nur dann als Client zu starten, wenn bereits ein Server läuft, hilft folgendes Mini-Script

```
#!/bin/bash

RUNNING=`lsof /usr/bin/emacs | grep $USER`

if [ -z "$RUNNING" ]; then
  echo "starting emacs server"
  /usr/bin/emacs $* &
else
  echo "starting emacs client"
  /usr/bin/gnuclient $* &
fi
```

Autocomplete

Beschreibung

Ermöglicht automatische Vervollständigung von Code, die auch auf noch nicht in der Datei vorhandene Schlüsselwörter etc. anwendbar ist.

Installation und Konfiguration

Da das Paket sehr aktiv entwickelt wird, empfiehlt es sich, es nicht aus dem Paketmanager zu installieren, sondern direkt von der Projekthomepage herunterzuladen: <http://cx4a.org/software/auto-complete/> . Informationen zur Konfiguration sind in den Hilfedateien des Download-Archivs vorhanden. Weitere Informationen und Tips sowie ein Vergleich zu Alternativen findet sich auf <http://www.emacswiki.org/emacs/AutoComplete> .

Yasnipped

Beschreibung

Yasnipped - yet another snippet extension for Emacs. Hervorragendes Snippetsystem, welches viel Zeit und Tipparbeit erspart.

Installation und Konfiguration

Die im Paketmanager vorhandene Version ist in gut verwendbarem Zustand, also reicht

```
sudo aptitude install yasnipped
```

Erstellen eigener Snippets

Um neben den im Paket vorhandenen auch eigene Snippets definieren zu können, muss deren Verzeichnis mittels `yas/load-directory` geladen werden. Speichert man eigene Snippets zum Beispiel in `~/yasnippet`, so ist `.emacs` wie folgt zu editieren

```
;; yasnippet
(setq yas/root-directory (cons yas/root-directory ('("~/yasnippet")))
(mapc 'yas/load-directory yas/root-directory)
```

Möchte man ein neues Snippet für das HTML `input` Element erstellen, so ist dieses innerhalb des mit `yas/load-directory` geladenen Verzeichnisses unter `text-mode/html-mode/input` zu speichern. Eine tolle Referenz für das Erstellen von neuen Snippets sind die mit yasnippet mitgelieferten Beispiele. Deren Speicherort ist mit

```
dpkg -L yasnippet | grep snippets
```

einfach zu ermitteln.

Ropemacs

Beschreibung

Rope ist eine Refactoring Bibliothek für Python, welche von unterschiedlichen Editoren verwendet werden kann. Rope ermöglicht unter anderem automatische Code-Vervollständigung via Code-Introspection. In Emacs kann Rope mittels Ropemacs verwendet werden.

Installation und Konfiguration

Leider ist Ropemacs erst in der Version 0.1 und daher noch nicht ausgereift - diese steht aber bereits als Debian Packet zur Verfügung.

```
sudo aptitude install python-ropemacs
```

Die notwendige Aktivierung erfolgt wieder in `.emacs`

```
;; enable pymacs and ropemacs
(require 'pymacs)
(pymacs-load "ropemacs" "rope-")
```

Shortcuts

Code-Vervollständigung wird mittels `M-/` ausgelöst. Darüberhinaus bietet Ropemacs auch andere Shortcuts, wovon

- C-c d zeige den Docstring für das Element unter dem Cursor
- C-c g springe zur Definition des Elements unter dem Cursor
- C-c f durchsucht das gesamte Projekt nach dem Element unter dem Cursor, wobei Strings, Docstrings und Kommentare ignoriert werden

die wichtigsten sind.