

Pizza hier bestellen:

<https://go.uni-wue.de/pizza>



Zum mitmachen vorher eines hiervon installieren:

Vim www.vim.org/download.php

NeoVim github.com/neovim/neovim/releases

Cheat Sheet:

www.viemu.com/a_vi_vim_graphical_cheat_sheet_tutorial.html

Wie man Vim beendet

(und andere grundlegende Befehle)

Alexander Gehrke

13. November 2019

Was ist vi(m) eigentlich? — Geschichtsstunde

vi: (1976) einer der ersten Editoren, die mehrere Zeilen gleichzeitig anzeigen konnten

Vim: (1991) von „Vi IMproved“, aktiv entwickelt, bekannteste vi-Variante

Neovim: (2014) Fork von Vim mit neuer Plugin-Architektur und weniger Unterstützung für Hardware und Betriebssysteme aus den Siebzigern

Was ist vi(m) eigentlich? – Modales Konzept

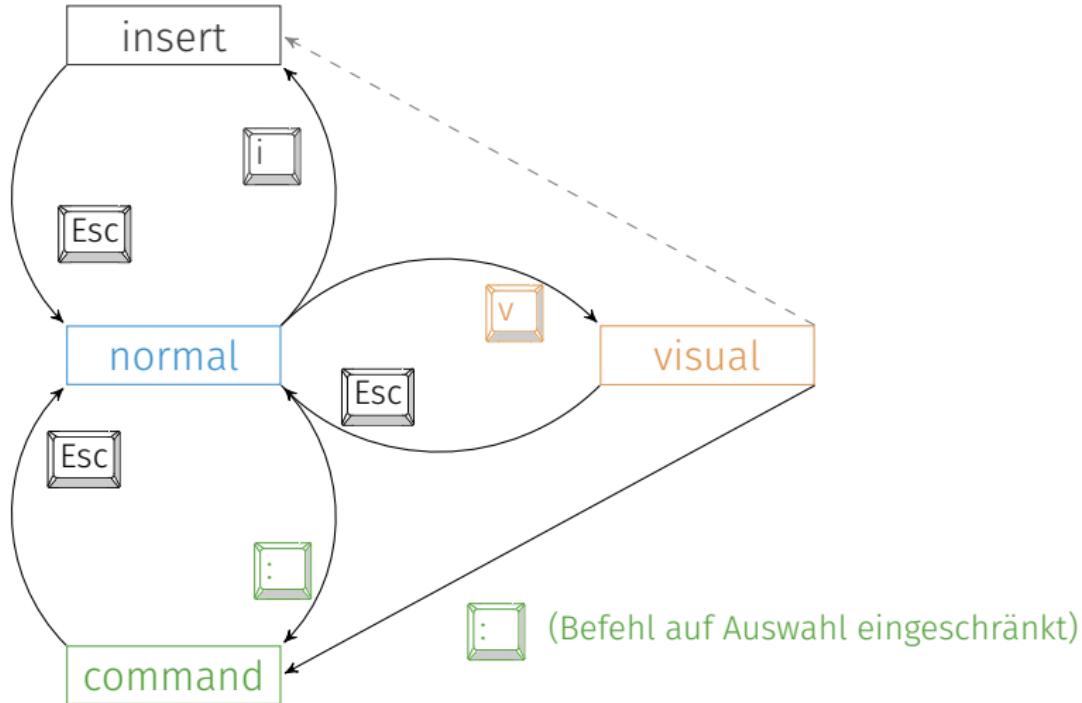
In Vim hängt die Bedeutung von Tastendrücken davon ab, in welchem Modus man sich befindet:

- insert mode Eingabe von Text wie in typischen Editoren
- normal mode Alle Tasten mit Funktionen belegt
- command mode Eingabe von Befehlen auf Kommandozeile
- visual mode Markieren von Text

Der „normal mode“ heißt so, weil man sich in diesem normalerweise aufhält, in anderen Modi nur für bestimmte Aktionen.

(weitere Modi existieren, das sind die wichtigsten)

Moduswechsel



(weitere Varianten zu wechseln existieren, das sind die wichtigsten)

Warum der Vortragstitel?

stack overflow Questions Jobs Documentation BETA Tags Users Search... Log In Sign Up

How to exit the Vim editor?

Ask Question

Download the new Official App LEARN MORE

I'm stuck and cannot escape. It says:
"type :quit<Enter> to quit VIM"
But when I type that it simply appears in the object body.

vim vi

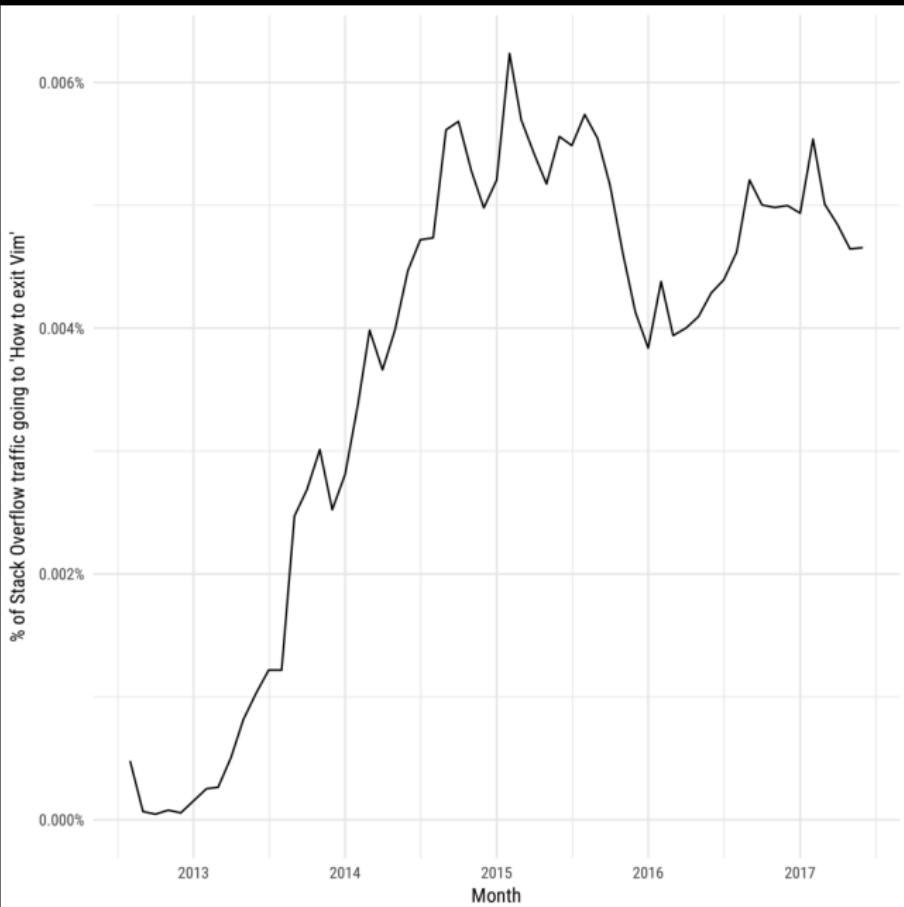
share improve this question edited Nov 3 '16 at 20:26 asked Aug 6 '12 at 12:25
Peter Mortensen |clancy 11.1k 16 76 109 6,784 5 17 26

asked 4 years, 9 months ago viewed 1000082 times active 13 days ago

BLOG How Stack Overflow Flipped the Switch on HTTPS

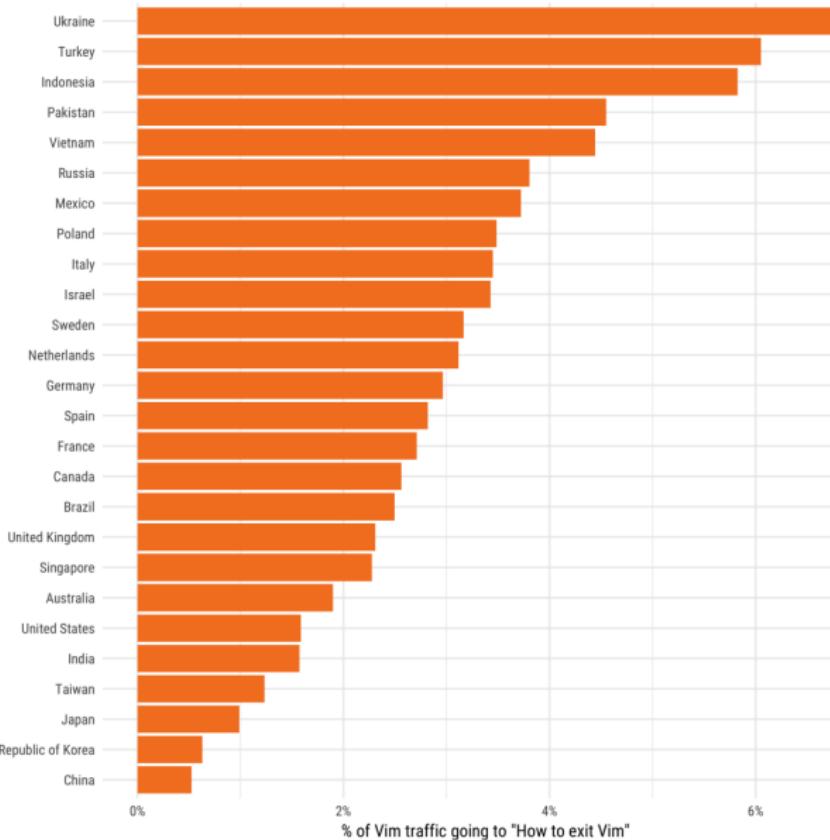
Microsoft Azure Microsoft Got 5 minutes?

(Stackoverflow Blog, 23. Mai 2017, inzwischen > 2 Mio. Views)



What countries are getting stuck in Vim?

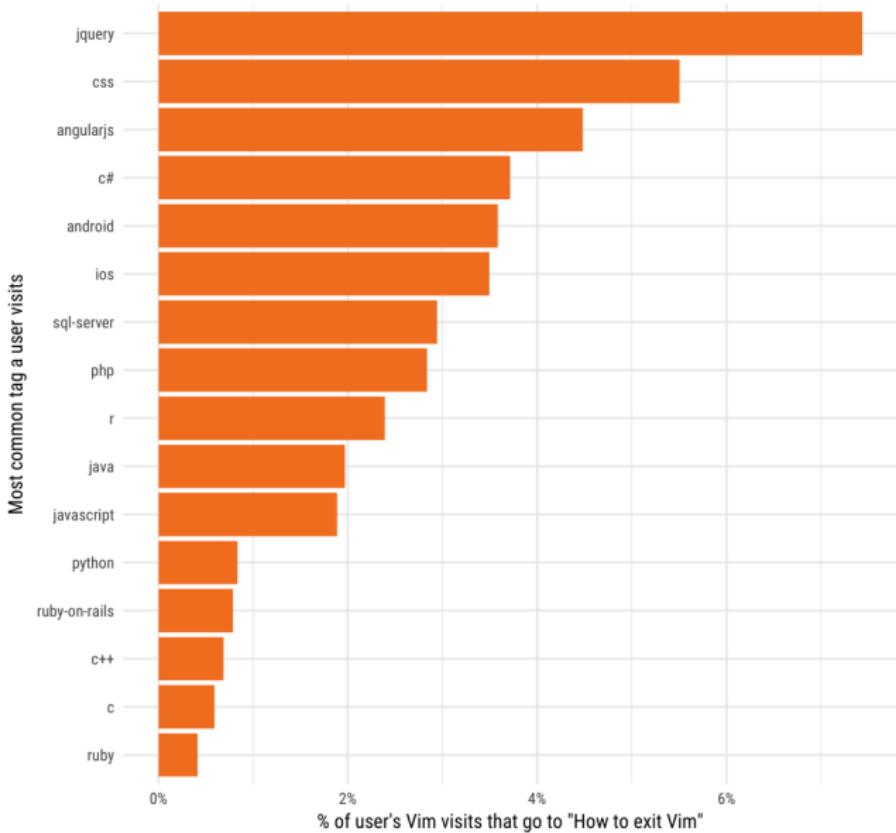
Measured by a % of all Vim visits in one year of traffic. Only countries with >100 million visits in that year.



Korrelation zu Linuxen mit Vim als Standardeditor?

What developers are most likely to get stuck in Vim?

Divided by tag the user visited most frequently, showing the 16 most common such tags.



Und wie beende ich Vim jetzt?

-  in den **Command Mode** wechseln
-  (oder „quit“) eingeben
-  zum Ausführen des Befehls

(einige andere Möglichkeiten später)

DEMO TIME

Normal mode basics

Normal mode basics – Zusammenfassung

- **motions** bewegen den Cursor (grün im Cheatsheet)
- **operators** arbeiten auf dem Text und nutzen eine motion um den Wirkungsbereich anzugeben
- **commands** alles andere

Die meisten Befehle im normal mode können durch vorheriger Eingabe einer Zahl mehrfach ausgeführt werden, z.B. `d3w` für „delete 3 words“.

DEMO TIME
Visual mode

Visual mode – Zusammenfassung

Der **visual mode** entspricht dem Markieren in anderen Editoren. Hier können alle Motions benutzt werden.

▼ „normales“ Markieren

▼ Markieren von ganzen Zeilen

Ctrl+v Markieren eines rechteckigen Bereichs (Block selection)

Operatoren wirken sich auf selektierten Text aus, anstatt eine motion zu erwarten.

DEMO TIME

Command mode

Command mode – Zusammenfassung

Der **command mode** enthält hauptsächlich Befehle, die oft auf mehrere Zeilen angewendet werden oder die in anderen Editoren in Menüs wären, z.B.:

- :e foo.txt Datei foo.txt im aktuellen Fenster öffnen
- :w datei.txt nach datei.txt speichern
(Dateiname optional)
- :s/foo/bar/g alle Vorkommen von foo durch bar ersetzen
(s = substitute, g = global)

Command mode – Zusammenfassung

Bei zeilenweisen Befehlen, wie `s`, kann eine *range* angegeben werden:

`:1,5s/.../` auf Zeilen 1-5 anwenden

`:%s/.../` auf ganze Datei anwenden

`: '<,'>s/.../` auf letzte visual-Markierung anwenden

Weitere mögliche Marker sind `$` (letzte Zeile), `.` (aktuelle Zeile), `/pattern/` (bis zu erster Zeile, die auf das Pattern passt), ...

DEMO TIME

Weitere Motions

Weitere Motions

- f / F** bis zu nächstem / vorherigem Vorkommen des danach gelesenen Zeichens
- t / T** wie **f/F**, aber Zeichen nicht inklusive
- %** zum passenden Begrenzer (z.B. Klammer) springen
 - wenn Cursor nicht auf Klammer zu umgebender Klammer
 - wenn Cursor nicht in Klammer: zu nächster Klammer
- (,), {, }** Satz- bzw. absatzweise springen

Text objects

Wenn gerade eine Motion erwartet wird, **a** oder **i** und anschließend weitere Taste drücken, um ein *text object* auszuwählen:

- | | |
|------|--------------------------------------|
| ' | ganze Klammer bzw. Anführungszeichen |
| w, W | ganzes Wort |
| p | ganzer Absatz (paragraph) |
| s | ganzer Satz (Punkt oder Leerzeile) |
| s | ganzer Satz (Punkt oder Leerzeile) |

Bei **a** immer mit umgebenden Zeichen bzw. Whitespace,
bei **i** ohne.

Siehe :help objects.

Buffer, Windows, Tabs

Buffer: Inhalt der Datei im Speicher

Window: Ein Fenster, das einen Buffer anzeigt. Der selbe Buffer kann in mehreren Fenstern gleichzeitig angezeigt werden

Tab: kann ein oder mehrere Fenster enthalten.

Buffer können „hidden“ sein, das bedeutet, eine Datei ist weiterhin geöffnet, aber kein Window zeigt sie gerade an.

DEMO TIME

Buffers, Windows, Tabs

Weitere Möglichkeiten, Vim zu beenden

Schließt ein Fenster:

`:q` Fehler bei ungespeicherten Änderungen

`:wq` speichert und schließt dann

`:x` oder `ZZ` Wie `:wq`, aber schreibt nur, wenn auch geändert

`:conf q` Nachfrage bei ungespeicherten Änderungen

`:q!` oder `ZQ` immer ohne Speichern beenden

`:cq` immer ohne Speichern beenden, gibt Fehlercode zurück

Um alle Fenster zu schließen: nach `q` bzw `x` ein `a` oder `all` anhängen (geht nicht bei `cq`)

Kaputte Möglichkeiten, Vim zu beenden

Vorschlag von Lukas (:! führt externe Befehle aus):

```
:!killall -9 vim
```

Kaputte Möglichkeiten, Vim zu beenden

Vorschlag von Lukas (:! führt externe Befehle aus):

```
:!killall -9 vim
```

... aber das killt alle Vims.

Kaputte Möglichkeiten, Vim zu beenden

Vorschlag von Lukas (:! führt externe Befehle aus):

```
:!killall -9 vim
```

... aber das killt alle Vims. Gegenvorschlag:

```
:exec "!kill -9 " . getpid()
```

:exec führt einen String als Befehl im command mode aus.
String-Konkatenation in vimscript geht mit Punkt.

(geht vermutlich beides nicht unter Windows, ist aber eh nicht
sinnvoll ;-)

weitere Features

- Register / Macros
- Marks
- Konfiguration
- Plugins
- ...