

# Vim workshop

Joan V. and Guillem R.

LinuxUPC

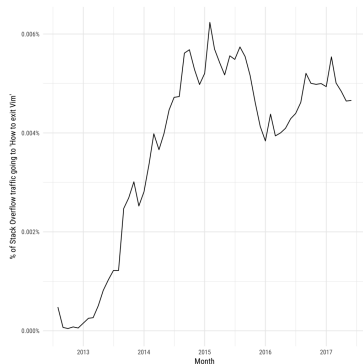
*linuxupc AT linuxupc.upc.edu*

November 1, 2018

- 1 Background
  - Exiting Vim
  - Some data
- 2 Introduction to Vim
  - What is Vim
  - Vim features
  - Modes
- 3 Moving in Vim
  - Basic moving
  - Moving keystrokes
- 4 Insert mode

- 1 Background
  - Exiting Vim
  - Some data
- 2 Introduction to Vim
  - What is Vim
  - Vim features
  - Modes
- 3 Moving in Vim
  - Basic moving
  - Moving keystrokes
- 4 Insert mode

# Exiting Vim



Source: <https://stackoverflow.blog/2017/05/23/stack-overflow-helping-one-million-developers-exit-vim/>

# Who uses Vim?

- Most popular editor in 2006 in Linux Journal readers.
- Third most popular editor in 2016 in Stack Overflow
- Fifth most popular editor nowadays in Stack Overflow.
- 40% of sysadmins stackoverflow users use Vim.

Vim is not just for nosaltgic nerds!

- 1 Background
  - Exiting Vim
  - Some data
- 2 Introduction to Vim
  - What is Vim
  - Vim features
  - Modes
- 3 Moving in Vim
  - Basic moving
  - Moving keystrokes
- 4 Insert mode

# What is VIM?

Vim is a terminal-based text editor oriented to programmers with highly efficiency writting.

```

\begin{document}
\begin{frame}
\titlepage
\end{frame}

\begin{frame}
\frametitle{Overview}
\tableofcontents
\end{frame}

-----
\PRESENTATION SLIDES
-----

\section{Introduction to VIM}
\subsection{What is VIM?}

\begin{frame}
\frametitle{What is VIM?}
Vim is a terminal-based text editor with just keyboard functionality.
\end{frame}

-----

\begin{frame}
\frametitle{Bullet Points}
\begin{itemize}
\item Lorem ipsum dolor sit amet, consectetur adipiscing elit
\item Aliquam blandit faucibus nisi, sit amet dapibus enim tempus eu
\item Nulla commodo, erat quis gravida posuere, elit lacus lobortis est, quis porttitor odio mauris at libero
\item Nam cursus est eget velit posuere pellentesque
\item Vestibulum faucibus velit a augue condimentum quis convallis nulla gravida
\end{itemize}
\end{frame}

! change; before #16 5 seconds ago

```

# Vim features

- Keyboard focused.
- Text autocompletion
- Terminal multiplexed
- More than 200 default key commands.
- Highly configurable
- Programmer focused
- Modes for implementing different functions



# Modes

Vim has 3 main modes with its variations:

## Insert mode

Mainly we will enter this mode with the key `i` from normal mode. In this mode any character key we pressed will be inserted on the place of the cursor.

## Normal mode

This is the default mode of vim. When you start up vim you are in this mode. In this mode you can type several commands to change to other modes, move efficiently the cursor, paste, etc..

## Visual mode

In this mode, which have several variations you can select text to cut, copy, replace. Its variations consist in the way you select code, for ex: Visual line will be selecting line by line while visual will be selecting character by character.

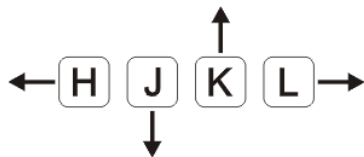
- 1 Background
  - Exiting Vim
  - Some data
- 2 Introduction to Vim
  - What is Vim
  - Vim features
  - Modes
- 3 Moving in Vim
  - Basic moving
  - Moving keystrokes
- 4 Insert mode

# Moving in vim (intro)

Moving the cursor around the screen in Normal mode can be performed in many ways.

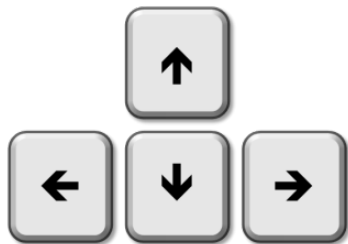


# Basic moving



Vim moving

VS



Classic moving

# Moving keystrokes

The most important commands for moving are:

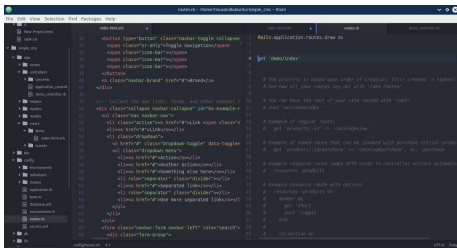
Command	Explanation
<code>h j k l</code>	moving around as arrow keys
<code>w b</code>	next and last word
<code>( )</code>	moving around paragraphs
<code>0 \$</code>	first and last character of line
<code>%</code>	move to matching character (ex: <code>[] () {}</code> )
<code>gg G</code>	move to the top and bottom of the file

For more commands, take a look at the cheat-sheet!

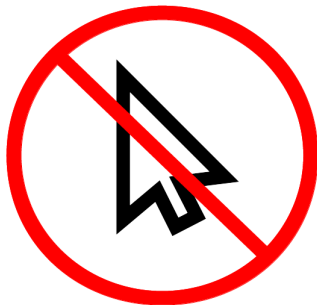
# Outline

- 1 Background
  - Exiting Vim
  - Some data
- 2 Introduction to Vim
  - What is Vim
  - Vim features
  - Modes
- 3 Moving in Vim
  - Basic moving
  - Moving keystrokes
- 4 Insert mode

# What is insert mode?



## Modern editors



## Vim insert mode

# Entering insert mode

There are different ways to enter insert mode which will position the cursor differently:

Command	Explanation
i	insert before the cursor
I	insert at the beginning of the line
a	insert after the cursor
A	insert at the end of the line
o	insert at the line bellow (ex: [] () {})
O	insert at the line above
ea	insert at the end of the current word
Esc	exit insert mode