



VIM Editor Introduction

Getting Started

VIM – Entering and Exiting

- To enter VIM, at the prompt, type: **vi YourFileName**
or: **vim YourFileName**
- On *athena*, **vi** takes us to **vim**. This is not true on all installations.
- Technically, **vi** is the older version. **vim** is an updated version.

VIM Modes

- **Command Mode.**

- For entering commands, usually two letters and number.
- For moving around a file
- Commands that start with a colon require a Enter/Return key

- **Insert Mode.**

- To insert, type: **i**
- Now you can start typing code or other information.
- To leave Insert Mode, press: **EscapeKey**

VIM – Entering and Exiting

Type: **i** to enter insert mode

EscapeKey to exit insert mode

:q! to exit without saving

:wq to exit and save work Or **:x**

ZZ to exit and save work, version 2, & be sure to use
caps.

:q to quit

:w YourFileName if you forgot to use a filename at the start,

Basic Cursor Movement

Lowercase Command	Arrow Key equivalent
l	Right arrow
h	Left arrow
j	Down arrow
k	Up arrow

Our version of **vim** allows the use of the arrow keys.



Basic Corrections

- While in Insert Mode:
 - Delete characters by using the back space key
- While in Command Mode:
 - Move Cursor to the first character to be deleted. Press: **x**
 - To delete multiple characters
 - Example: If you want to delete 6 characters in a row, move the cursor on the first one, and type: **6x**



Cursor Movement. 1 of 6

- **Spacebar** – Forward one character position
- **l** - Right (forward) one character position
- **h** - Left (backward) one character position
- **j** - Down to the same position in line below;
moves left to last position
- **k** - Up to the same position in the line above;
moves left to the last position



Cursor Movement. 2 of 6

- **w** - Forward to first letter of next word
- **W** - Forward to first letter of next blank-delimited word
- **b** - Backward to first letter of previous word
- **B** - Backward to first letter of previous blank-delimited word



Cursor Movement. 3 of 6

- **Return** - Forward to beginning of next line
- **0** - Back to beginning of current line (zero)
- **\$** - End of current line



Cursor Movement. 4 of 6

- (- Back to beginning of current sentence
-) - Ahead to beginning of next sentence
- { - Back to beginning of current paragraph
- } - Ahead to beginning of next paragraph



Cursor Movement. 5 of 6

- **H** - Home, or left end of the top line on screen
- **M** - Middle, or left end of middle line on screen
- **L** - Lower, or left end of lowest line on screen
- **G** - Last line in work buffer
- ***n*G** - Indicated relative line *n* in buffer.



Cursor Movement. 6 of 6

- **Ctrl-U** - Up half screen
- **Ctrl-D** - Down half screen

- **Ctrl-F** - Forward (down) almost a full screen
- **Ctrl-B** - Backward (up) almost a full screen



Delete Commands. 1 of 3

- **x** - Character at cursor
- **X** - Character following cursor
- **dw** - To end of word
- **dW** - To end of blank-delimited word
- **db** - To beginning of word
- **dB** - To beginning of blank-delimited word

Delete Commands. 2 of 3

- **d** then **Return** - Two lines; current and following
- **X** - Character following cursor
- **dw** - To end of word
- **dW** - To end of blank-delimited word
- **db** - To beginning of word
- **dB** - To beginning of blank-delimited word



Delete Commands. 3 of 3

- **d)** - To end of sentence
- **d(** - To beginning of sentence
- **d}** - To end of paragraph
- **d{** - To beginning of paragraph



Delete Commands. 3 of 3

- **d)** - To end of sentence
- **d(** - To beginning of sentence
- **d}** - To end of paragraph
- **d{** - To beginning of paragraph

HINTS: Yank & Put (Copy & Paste)

Copying and pasting in **vim** are accomplished with the commands **yank** and **put**.

<i>Command Syntax</i>	<i>What It Accomplishes</i>
y2w	Yanks two words, starting at the current cursor position, going to the right
4yb	Yanks four words, starting at the current cursor position, going to the left
yy or Y	Yanks the current line
p	Puts the yanked text after the current cursor position (lower case p)
P	Puts the yanked text before the current cursor position (upper case P)
5p	Puts the yanked text in the buffer five times after the current cursor position

Setting up *tabs* & *line numbers* in vim.

Go to your **home** directory

Type: **vim .vimrc** *which is a settings-for-vim file*

Once in the file, add to the Vim default file (if you have one):

type: **:set tabstop=4** *//corrected line*

type: **:set smartindent**

type: **:set expandtab**

type: **:set number**



More Help

At the prompt on athena, type **vimtutor** for more instruction.

Go to Google.

Enter “**VIM Tutorial**”

You will find lots of choices, some practical, even one that sells itself as “Zelda meets VIM”.



VIM Editor Introduction

Getting Started

The End