

# CS Essentials

## Session 2: Vim Essentials



# Introduction to Vim

What is Vim?

- just a text editor

What makes Vim better?

- people spend more time editing existing text than writing new text
- one of the key features of Vim is efficiency
- particularly useful for maintaining large pieces of code
- comes with a lot of useful shortcuts and it allows you to create others
- it is everywhere! (Mac, Windows, Linux)

# Vim modes

## 1. Command mode

- the letters from the keyboard work as commands (for deleting, moving the cursor etc.)
- Vim opens in command mode

## 2. Insert mode

- everything you write is added to the document
- exit the insert mode using ESC

## 3. Visual mode

- you can select text

# Vim commands

## 1. Moving the cursor

- h - move the cursor to the left
- j - move the cursor down
- k - move the cursor up
- l - move the cursor to the right

## 2. Exiting the editor

### 2.1 Exiting without saving

- type :q! and you will be sent back to the terminal

### 2.2 Exiting with saving

- type :wq to exit and save the changes
- if you want to save into a different file, type :w filename

# Vim commands

## 3. Deleting

### 3.1 Deleting one character at a time

- press `x` to delete the character under the cursor

For other deleting commands, you need to use: `d` motion.

`d` is the delete command and motion specifies whether you want to delete a word, up to the end of the sentence etc., more precisely:

# Vim commands

## 3.2 Deleting a word

3.2.1 Delete the word from the cursor to the beginning of the next word, excluding its first character

- move the cursor to the first letter of the word you want to delete and type `dw`

3.2.2 Delete the word from the cursor to the end of the current word, including its last character

- move the cursor to the first letter of the word you want to delete and type `de`

# Vim commands

## 3.3 Delete to the end of the line

- move the cursor to the first letter from where you want to delete and type `d$`

## 3.4 Delete a whole line

- move the cursor to the line you want to delete and type `dd`

# Vim commands

## 3.5 Deleting with a count for the motion

- ex1: type `d2w` to delete two words and move the cursor to the beginning of the third word
- ex2: type `d3e` to delete three words and move the cursor to the end of the last deleted word

Aside: You can use these commands to just navigate through words:

- `w` moves the cursor to the beginning of the next word
- `e` moves the cursor to the end of the current word; if it is already at the end, it places it at the end of the next one
- `$` moves the cursor to the end of the line
- `0` takes you to the beginning of the line



# Vim commands

## 4. Inserting

Reminder: After pressing any of the following commands, you are in insert mode. To go back to command mode after each change you make, you need to press ESC (you need to do this before you can navigate again).

Note: You can see what mode you are on at the bottom of the file

### 4.1 Inserting a character

- type i to insert a character before the character where the cursor is currently one

# Vim commands

## 4.2 Appending after the cursor

- type `a` to append text on the position after the character where the cursor is currently on

## 4.3 Appending at the end of the file

- type `A` to append text at the end of the line on which you currently are

## 4.4 Opening a new line

- type `o` to open a line after the position of the cursor
- type `O` to open a line before the position of the cursor

# Vim commands

## 5. Undoing and redoing

- type `u` to undo the last command and `U` to undo all the changes made on the line you are on
- press `Ctrl + R` to redo

## 6. The put command

- type `p` to put the previously deleted (or yanked) text where the cursor is

## 7. Replacing

- type `r` to replace the character the cursor is currently on
- type `R` to replace the characters starting from the one the cursor is currently on

# Vim commands

## 8. The change command

- the command for change is `c` and it works with the same motions as delete (`d`) : `w`, `e` and `$`

### 8.1 Changing a word

- type `ce` (or `cw`) to change the word starting with the character the cursor is currently on

### 8.2 Changing to the end of the line

- type `c$` to change the words starting with the character the cursor is currently on

Note: this change command is a shortcut for delete + insert

# Vim commands

9. Cursor information and checking the status of the file
  - type `Ctrl + g` and at the bottom of the file will appear the filename and the position of the cursor
  - type `G` to move to the bottom of the file
  - type `gg` to move to the top of the file
  - to go to a certain line, type line number + `G`

# Vim commands

## 10. Searching

- type / + the phrase you want to search + Enter
- to search for the same thing again, press n; to go in the backward direction, press N
- you can use the "ignore case" command by typing :set ic after you start searching for a word
- Note: when you type / + phrase, it searches for a phrase from the line you are on forward; if you want to search in the other direction, type ? + phrase

Useful aside: Press Ctrl + O to go to the previous position of the cursor and Ctrl + I to go forward

# Vim commands

## 11. The substitute command

### 11.1 Substitute one word in a line

- substitute word1 with word2 by typing `:s/word1/word2`
- this only changes one occurrence of word1 with word2
- to substitute the word in the whole line, type `:s/word1/word2/g` (g comes from globally)
- you can also substitute a word between two lines by typing `:line1,line2s/word1/word2/g`

# Vim commands

## 11.2 Substitute in the whole file

- to substitute word1 with word2 in the whole file, type `:%s/word1/word2/g`
- a more cautious version is `:%s/word1/word2/gc`, which finds word1 in the file and prompts whether to substitute or not
- you will get: replace with word2 (y/n/a/q/l/Ê/ÿ)? , where those mean:
  - y - replace this match
  - n - skip this match
  - q - quit
  - l - replace this match then quit (think of it as coming from last)
  - Ê - actually means Ctrl + E, scroll down the page
  - ÿ - actually means Ctrl + Y, scroll up the page



# Vim commands

## 12. Matching parentheses

- if you have the cursor on any type of parentheses, typing % will find its match
- Note: very useful when coding

## 13. Executing commands from the terminal

- type `:!command` + Enter
- to return to vim after doing so, type Enter again

# Vim commands

## 14. The visual mode

- type `v` to enter visual mode
- if you move the cursor in this mode you select the text
- after that you can use any of the commands previously introduced
- Reminder: Press `ESC` to exit visual mode

## 15. Retrieving files

- if you want to paste the content of a different file into this one, place the cursor where you want it and type `:r filename`
- this also works with external commands, type `:r! command`

# Vim commands

## 16. Copy & Paste

- type y to copy (yank) the text you selected and p to paste it
- Reminder: we learned p as the put command, and indeed it is; now we can see how it works in more than one way
- Note: when you want to copy text, you first have to enter the visual mode and move the cursor to select the text; typing y takes you back to normal mode with the text you yanked in the buffer
- y works as an operator too, in that you can use motions and counters the same as we saw for deleting

# Vim commands

## 17. Getting help

- you can access the help page by typing `:help`, pressing F1 or the help button
- to exit help mode, type `:q`
- you can use `Ctrl + w` to navigate through the windows

Thank you!