

# Neovim workshop

Ivan Evdokimov, University of Essex

March 28, 2025

# Outline

- 1 Introduction
- 2 Modes
- 3 Motions
- 4 Editing
- 5 Visual
- 6 Tips
- 7 Customization
- 8 Conclusion

# Before we begin...

- It is impossible to learn without getting your hands dirty.
- There are a number of challenges that I prepared, download them from github.
- You should be able to just run 'git clone [https://github.com/ivan020/vim\\_workshop.git](https://github.com/ivan020/vim_workshop.git)'

# But what is "Neo" and "VI" and "M"?

- "Vi" is the original editor.
- "M" is the modification (improved).
- "Neo" is the most recent version of this editor.

# Why bother with Neovim?

- Very-very light and therefore fast.
- Async loading of plugins makes it blazingly fast.
- Having both hands on keyboard makes you even faster.
- You really are in control of everything in your IDE.

# Why **not** to use NeoVIM?

- It takes time and effort to build the muscle memory.
- Requires tinkering and setting.
- VIM-motions are supported in other editors (i.e. VScode, nano, sublime, etc.)

# Which VIM?

- We will use Neovim.
- The most recent iteration in the vim-family.
- Community-driven.
- **Lua** – the programming language to support plugins for Neovim.

# Modes

Vim has 4 key modes:

- NORMAL
- COMMAND
- INSERT
- VISUAL



# NORMAL

NORMAL is the default vim mode.

- Can always use ESC or CTRL+C keys to return to NORMAL.
- Commonly used to: redo (u), motions (h,j,k,l).
- Also used for file navigation:
  - M move cursor to middle.
  - L move cursor to the bottom of the display.
  - H move cursor to the head of the display.
  - zz to center a screen.

# COMMAND

COMMAND is used to run commands. You type ':' to start a command, with most used:

- `:w` to write a file.
- `:w filename.txt` write a file under 'filename.txt' filename.
- `:q` to exit a file.
- `:q!` to exit without save.
- `:wq` to save and exit.
- `:e filename2.txt` to open 'filename2.txt' from withing current editor.
- `:vsplit` to split current editor (vertically).
- `:Ex` to open the current tree buffer.

# Motions

- The key movements are:
  - **h** to move left.
  - **l** to move right.
  - **j** to move up.
  - **k** to move down.
- You can combine them to jump several times, try: **2k,2l**, etc.
- TRY: do the challenge `./challenges/basic_motions.txt`.

# Advanced Horizontal Motions

- **w** jump to the **beginning** of the next token.
- **e** jump to the **end** of the next token.
- **b** jump to the **previous** token.
- **\_** jump to the **beginning** of the line.
- **\$** jump to the **end** of the line.
- You can also combine those with numbers, try: **2b**, **5e**.
- TRY: do the challenge './challenges/advanced\_motions.txt'.

# More Advanced Horizontal Motions

- **f** + ';' to jump to the next ';' char (can be replaced with any other char).
- **t** + ';' to jump the char before the ';'.
- in these modes, you can type ';' to jump to the next occurrence of this character.
- **F** and **T** for the backward search.
- TRY: do the challenge './challenges/to\_f\_motions.txt'

# Advanced Vertical Motions

- `Ctrl + u` jump 35 lines up.
- `Ctrl + d` jump 35 lines down.
- `{` to go up to the next blank line.
- `}` to go down to the next blank line.
- `gg` to go to the first line in the document.
- `G` to go to the last line in the document.
- TIP: use `zz` to center screen around cursor.

# Enter the Insert Mode

- **i** to **i**nsert before cursor.
- **a** to insert **a**fter cursor.
- **I** to **I**nsert at the beginnig of the line.
- **A** to insert **a**t the end of the line.
- **o** to start editing from the new line.
- **O** to start editing from the previous line.
- **Ctrl + C** or **Esc** to exit to normal mode.
- **x** to remove a character (backspace).
- TRY: do the challenge `'./challenges/basic_editing.txt'`.

# Visual Mode

- commonly used to highlight something in the text.
- `v` in normal mode to enter the visual mode.
- you can use a number of shortcuts to select a token.
- `viw` to highlight a word.
- `V` to highlight the whole line.



# Copy-Paste

- vim does not use global clipboard for copy-paste operations.
- **y** and **p** will only work within a document.
- if you want to paste from the buffer, you need **Ctrl** + **Shift** + **v** combination.
- TRY: use './challenges/copyPaste.txt' to check out these motions.

# Copy/Cut and paste by token

- There are some very useful vim-combinations that we can use.
- **diw** delete a token, stay in normal mode after.
- **ciw** delete a token and enter the editing mode.
- **daW** deletes a token and its surrounding to the next token.
- **caW** deletes a token and its surrounding to the next token, entering the edit mode.
- **y** will copy under the cursor.
- **p** will past what you copied.
- **/** to find a token in the document.
- TIP: use **'.'** to repeat the previous command.
- TRY: use **'./challenges/editing\_patters.txt'** to checkout these commands.

# Copy/Cut and paste by paragraph

- Often, you need to remove the whole block (paragraph).
- **dip** delete a paragraph, stay in normal mode after.
- **cip** delete a paragraph and enter the editing mode.
- **daP** deletes a paragraph and its surrounding to the next paragraph.
- **caP** deletes a paragraph and its surrounding to the next paragraph, entering the edit mode.
- TRY: use './challenges/blockRemoval.txt' to checkout these commands.

# Search and Replace

- you can search and replace using a combination in the normal mode.
- `:%s/!/!/g` will replace all '!' with '?', globally.
- `:%s/!/!/gc` will replace and ask you whether you want to replace.
- TRY: the './challenges/searchAndReplace.txt' for exercise.

# Search and Replace with Regex

- Regex allows you to make a search more complex.
- You can lookup duplicates, digits, upper/lower/CamelCase, etc.
- Let us try to remove a some duplicates in  
`'./challenges/remove_duplicates.txt'`

# Customization

- Neovim is the canvas, you're the artist.
- You can build your own setup.
- You can download other people's setup (github).
- You can build the pre-installed configurations.

# 0 to IDE

There is a number of pre-set vim configs that you can install:

- LazyVim: <https://www.lazyvim.org/>
- LunarVim: <https://www.lunarvim.org/>
- Kickstart: <https://github.com/nvim-lua/kickstart.nvim>
- nvChad: <https://nvchad.com/>

# Final words...

- Vim is giga-cool.
- It has a ton of different flavours: distro, plugin, standalone app, etc.
- Makes you very efficient once you know this tool.



# Thank You!