# X forwarding

- X-windows is Linux's system for displaying graphical programs.

# X forwarding

- X-windows is Linux's system for displaying graphical programs.

- X server: Program that manages what is displayed on the screen.
- X client: Program that wants to display something.

# X forwarding

- ▶ X-windows is Linux's system for displaying graphical programs.

- ▶ X server: Program that manages what is displayed on the screen.
- ▶ X client: Program that wants to display something.

- ▶ Windows: Run Xming to start the server, then use PuTTY like normal.
- ▶ Mac: Install X11.app, then use `ssh -X <hostname>`
- ▶ Linux: You are already running an X server! Just use `ssh -X <hostname>`

# X forwarding

- ▶ X-windows is Linux's system for displaying graphical programs.

- ▶ X server: Program that manages what is displayed on the screen.
- ▶ X client: Program that wants to display something.

- ▶ Windows: Run Xming to start the server, then use PuTTY like normal.
- ▶ Mac: Install X11.app, then use `ssh -X <hostname>`
- ▶ Linux: You are already running an X server! Just use `ssh -X <hostname>`

- ▶ Use `xeyes` to check if X forwarding is set up correctly.

# What is 'text'?

- 'Plain text' is ASCII-encoded data. (Or UTF-8 encoded data, or EBCDIC encoded data...)

# What is 'text'?

- 'Plain text' is ASCII-encoded data. (Or UTF-8 encoded data, or EBCDIC encoded data...)
- Most programming languages and network protocols use plain text because it is easy for developers to understand.

# What is 'text'?

- 'Plain text' is ASCII-encoded data. (Or UTF-8 encoded data, or EBCDIC encoded data...)
- Most programming languages and network protocols use plain text because it is easy for developers to understand.
- Text editors are just tools for editing plain text files. You'll be using one a lot, so you should know how to use your editor of choice.

# Notepad++: notepad-plus-plus.org

Features:

- Document Map

# Notepad++: notepad-plus-plus.org

Features:

- Document Map
- Multiple Cursors

## Atom: atom.io

- Terminology
  - Buffer: Contents of an open file.
  - Pane: A visual section of the editor. Holds buffers, status bars, etc.

# Atom: atom.io

- Terminology
  - Buffer: Contents of an open file.
  - Pane: A visual section of the editor. Holds buffers, status bars, etc.
- Shortcuts
  - `Ctrl` + `⇧` + `o` Show a directory in the tree view
  - `Ctrl` + `⇧` + `p` Command palette you can search through
  - Searching for files:
    - `Ctrl` + `t`, `Ctrl` + `p` Find a file in the current open directories
    - `Ctrl` + `b` Jump to an open buffer

# Atom: atom.io

- Terminology
    - Buffer: Contents of an open file.
    - Pane: A visual section of the editor. Holds buffers, status bars, etc.
- Shortcuts
    - `Ctrl` + `⇧` + `o` Show a directory in the tree view
    - `Ctrl` + `⇧` + `p` Command palette you can search through
    - Searching for files:
        - `Ctrl` + `t` , `Ctrl` + `p` Find a file in the current open directories
        - `Ctrl` + `b` Jump to an open buffer
- Packages
    - **minimap** Shows a scrollable overview of your file
    - **autocomplete-clang** C++ Autocompletion
    - **vim-mode** Some vim shortcuts

# Atom: atom.io

- Terminology
    - Buffer: Contents of an open file.
    - Pane: A visual section of the editor. Holds buffers, status bars, etc.
- Shortcuts
    - `Ctrl`+`⇧`+`o` Show a directory in the tree view
    - `Ctrl`+`⇧`+`p` Command palette you can search through
    - Searching for files:
        - `Ctrl`+`t`, `Ctrl`+`p` Find a file in the current open directories
        - `Ctrl`+`b` Jump to an open buffer
- Packages
    - **minimap** Shows a scrollable overview of your file
    - **autocomplete-clang** C++ Autocompletion
    - **vim-mode** Some vim shortcuts
- Documentation

# Vim Terminology: vim.org

- Buffer: Opened file
- Window: Visual section displaying a buffer
- (Atom borrowed this terminology from vim!)

# Vim Terminology: vim.org

- Buffer: Opened file
- Window: Visual section displaying a buffer
- (Atom borrowed this terminology from vim!)
- Composability: You can connect simple commands together into complex ones

# Vim Terminology: vim.org

- ▶ Buffer: Opened file
- ▶ Window: Visual section displaying a buffer
- ▶ (Atom borrowed this terminology from vim!)
- ▶ Composability: You can connect simple commands together into complex ones
- ▶ `vimtutor` can teach you some basic vim stuff

# Vim Terminology: vim.org

- Buffer: Opened file
- Window: Visual section displaying a buffer
- (Atom borrowed this terminology from vim!)
- Composability: You can connect simple commands together into complex ones
- `vimtutor` can teach you some basic vim stuff

**Tip:** `:set mouse=a`

# Modes: Keys do different things in different modes

- Normal: Navigation/commands. `Esc`, `Ctrl` + `c`

# Modes: Keys do different things in different modes

- Normal: Navigation/commands. `Esc`, `Ctrl`+`c`
- Insert: Writing text.
    - `i`: Insert at cursor
    - `I`: Insert at beginning of line
    - `a`: Insert after cursor (append at cursor)
    - `A`: Insert at end of line (Append to line)
    - `o`: Insert on new line below cursor
    - `O`: Insert on new line above cursor
    - `c`: Change text at cursor

# Modes: Keys do different things in different modes

- ▶ Normal: Navigation/commands. `Esc`, `Ctrl`+`c`
- ▶ Insert: Writing text.
    - ▶ `i`: Insert at cursor
    - ▶ `I`: Insert at beginning of line
    - ▶ `a`: Insert after cursor (append at cursor)
    - ▶ `A`: Insert at end of line (Append to line)
    - ▶ `o`: Insert on new line below cursor
    - ▶ `O`: Insert on new line above cursor
    - ▶ `c`: Change text at cursor
- ▶ Replace: Overwriting text. `R`

# Modes: Keys do different things in different modes

- Normal: Navigation/commands. `Esc`, `Ctrl`+`c`
- Insert: Writing text.
    - `i`: Insert at cursor
    - `I`: Insert at beginning of line
    - `a`: Insert after cursor (append at cursor)
    - `A`: Insert at end of line (Append to line)
    - `o`: Insert on new line below cursor
    - `O`: Insert on new line above cursor
    - `c`: Change text at cursor
- Replace: Overwriting text. `R`
- Visual: Selecting text.
    - `v`: Character select
    - `V`: Line select
    - `Ctrl`+`v`: Block select

# Moving Around

- `j`/`k`/`r`/`l`: up/down/left/right
- `^`/`$`: Beginning/end of line
- `w`/`e`/`b`: Next word/Next word end/Back one word
- `%`: Matching brace or parenthesis
- `gg`/`G`: Top/bottom of document

**Tip:** Repeat commands by specifying a number first: `2w`

# Moving Around

- `j`/`k`/`r`/`l`: up/down/left/right
- `^`/`$`: Beginning/end of line
- `w`/`e`/`b`: Next word/Next word end/Back one word
- `%`: Matching brace or parenthesis
- `gg`/`G`: Top/bottom of document

**Tip:** Repeat commands by specifying a number first: `2w`
**Tip:** Compose with insert modes: `c4w`

# Miscellany

- `:w` Save
- `:q` Quit
- `:wq`/`ZZ` Save and quit

# Miscellany

- `:w` Save
- `:q` Quit
- `:wq`/`ZZ` Save and quit
- `u`/`Ctrl`+`r`: undo/redo

# Miscellany

- `:w` Save
- `:q` Quit
- `:wq`/`ZZ` Save and quit
- `u`/`Ctrl`+`r`: undo/redo
- Clipboards
    - `y`/`yy`: Copy/'yank' (line)
    - `d`/`dd`: Cut/'delete' (line)
    - `p`/`P`: Paste after cursor/Paste before cursor

# Miscellany

- `:w` Save
- `:q` Quit
- `:wq`/`ZZ` Save and quit
- `u`/`Ctrl`+`r`: undo/redo
- Clipboards
  - `y`/`yy`: Copy/'yank' (line)
  - `d`/`dd`: Cut/'delete' (line)
  - `p`/`P`: Paste after cursor/Paste before cursor
  - `"+y`: Yank to system clipboard

# Miscellany

- `:w` Save
- `:q` Quit
- `:wq`/`ZZ` Save and quit
- `u`/`Ctrl`+`r`: undo/redo
- Clipboards
    - `y`/`yy`: Copy/'yank' (line)
    - `d`/`dd`: Cut/'delete' (line)
    - `p`/`P`: Paste after cursor/Paste before cursor
    - `"+y`: Yank to system clipboard
- `>`/`<`: Indent/Outdent
- `=`: Autoformat

# Miscellany

Configuration

- `.vimrc` in your home directory
- Example config on the website

## Miscellany

Configuration

- ► `.vimrc` in your home directory
- ► Example config on the website

Plugins

- ► Airline: Pretty status bar
- ► Fugitive: Git integration
- ► Syntastic: Syntax checking

# Miscellany

Configuration

- `.vimrc` in your home directory
- Example config on the website

Plugins

- Airline: Pretty status bar
- Fugitive: Git integration
- Syntastic: Syntax checking

Links

- Vim Cheat Sheet
- Another cheat sheet

# Emacs: GNU Emacs

Use `emacs -nw` to launch in the terminal (instead of a GUI)

- ▶ Frame: Everything you can see (you can have multiple frames!)
- ▶ Buffer, Window: As with vim and atom
- ▶ Instead of `Ctrl`+`n`, emacs writes this as `C-n`
- ▶ Meta key (written `M-x`): Use `Alt`+`x` or `Esc` `x`
- ▶ In Emacs: `C-h t` starts a tutorial

# Moving

- `C-n` / `C-p` next/previous line
- `C-f` / `C-b` forward/back one character
- `M-f` / `M-b` forward/back one word

# Moving

- `C-n` / `C-p` next/previous line
- `C-f` / `C-b` forward/back one character
- `M-f` / `M-b` forward/back one word
- `C-a` / `C-e` Beginning/end of line
- `C-v` / `M-v` Down/Up one page

# Moving

- `C-n` / `C-p` next/previous line
- `C-f` / `C-b` forward/back one character
- `M-f` / `M-b` forward/back one word
- `C-a` / `C-e` Beginning/end of line
- `C-v` / `M-v` Down/Up one page

# Cut/Paste

- Deleting: Backspace( `<DEL>` )/ `C-d`
- Kill (cut) a word: `M-<DEL>` / `M-d`
- Kill to the end of a line: `C-k`
- Unkill (paste): `C-y`
- `M-y` to cycle through kills

# Miscellany

- Saving a file: `C-x C-s`
- Quitting: `C-x C-c`
- Stop a command: `C-g` or hit `Esc` three times

# Miscellany

- Saving a file: `C-x C-s`
- Quitting: `C-x C-c`
- Stop a command: `C-g` or hit `Esc` three times
- Undo: `C-_`

## Miscellany

- Saving a file: `C-x C-s`
- Quitting: `C-x C-c`
- Stop a command: `C-g` or hit `Esc` three times
- Undo: `C-_`
- Windows:
    - Make 2 windows: `C-x 3`
    - Jump between windows: `C-x o`

# Miscellany

- Saving a file: `C-x C-s`
- Quitting: `C-x C-c`
- Stop a command: `C-g` or hit `Esc` three times
- Undo: `C-_`
- Windows:
    - Make 2 windows: `C-x 3`
    - Jump between windows: `C-x o`
- Reference Card
- Emacs Tour