# Keyboard Cribsheet

- Although harder to learn, some text-based programs (starting with the command line shell itself) may have more interactivity than you might have expected

- Not surprisingly, tapping this interactivity does involve a lot of keyboard shortcuts

- This is a partial (starter) list of these shortcuts—some of them have good mnemonics, some don't—but all of them let you work a little more efficiently once you've learned enough of them

*emphasis on starter!*

# bash

| | |
|---|---|
| `ctrl-d` | End input |
| `tab` | Autocomplete |
| `ctrl-c` | Stop currently-running program |
| `ctrl-z` | Suspend currently-running program |
| `up/down` | Traverse command-line history |
| `ctrl-r` | Search through command-line history (`ctrl-r` again to move further back) |
| `left/right` `ctrl-a/ctrl-f` | Move backward/forward along the current command by character |
| `ctrl-a` | (while editing a command) Jump to the beginning of the current command |
| `ctrl-e` | Jump to the end of the current command |
| `meta-b/meta-f` | Move backward/forward along the current command by word |
| `Del/ctrl-d` | Delete the current character |
| `meta-d` | Delete the current word |
| `ctrl-k` | Delete to the end of the current command |

"meta" stands for "the other modifier key"—alt on PCs, option on Macs; by default it is a tap on Esc

# more/less

`more` and `less` are called <u>pagers</u>—programs whose sole job is to navigate a long block of text

| | | |
|---|---|---|
| `page down/space` | Forward by a screenful | These also apply to `man`, because (surprise surprise) the paging mechanism you see there is simply `man` piping its output into the configured pager |
| `page up/b` | Backward by a screenful | |
| `enter/return/k` | Forward by a line | |
| `j` | Backward by a line | |
| `/` | Begin a search (hit `Enter` or `Return` to trigger) | |
| `p/n` | (while searching) Move to the previous/next search hit | |
| `q` | Quit the pager | |

# top

| | |
|---|---|
| `?` | Display keyboard commands |
| `q` | Quit |

- Why so short, you might ask—`top` is one of those command line programs which vary <u>significantly</u> between the two Unix-like platforms that you are most likely to use: Linux and Mac OS X (long story)

- Instead, the above two are the common ones…and just look up the rest

# vi/vim

vi uses <u>modes</u>: initially, you are in text navigation/revision mode; some keys (e.g., i, a, o) enter text entry mode; : switches to command entry mode

| | |
|---|---|
| arrow keys | Navigate through text |
| ^/$ | Jump to the beginning/end of the line |
| x | Delete the current character |
| / | Begin a search (hit Enter or Return to trigger) |
| i/a/o | Switch to text entry, either inserting, appending, or on a new line, respectively |
| Esc | Get out of text entry mode (when in doubt, hit Esc multiple times) |
| : | Enter a command, like… |
| :number | Jump to that line number |
| :w | Save (write) the current file |
| :q | Quit |

Preceding a keyboard shortcut with a number performs that command that many times

You can concatenate commands to do them in sequence, (e.g., :wq); append a ! to skip confirmation (e.g., :q!)

# emacs

| | |
|---|---|
| arrow keys | Traverse the text file |
| meta–</meta–> | Jump to the beginning/end of the file |
| meta–x goto–line | Jump to a particular line number |
| tab | "Smart" indent (depends on file type) |
| ctrl–x ctrl–s | Save |
| ctrl–x ctrl–c | Quit |
| ctrl–s | Begin a search (hit Enter or Return to trigger) |
| ctrl–space | "Mark" your place—move your cursor and then… |
| ctrl–w/meta–w | "Cut"/"copy" from your mark to the cursor, respectively |
| ctrl–y | "Paste" (actually, "yank") what you cut/copied to the current location |
| ctrl–a | Jump to the beginning of the current line |
| ctrl–e | Jump to the end of the current line |
| meta–b/meta–f | Move backward/forward along the current line <u>by word</u> |
| Del/ctrl–d | Delete the current character |
| meta–d | Delete the current word |
| ctrl–k | Delete to the end of the line |

meta–x is actually a generic entry point for additional commands, like : in vi—e.g., meta–x tetris

If the last few keyboard shortcuts seem familiar, you'd be right—bash uses them too