Navigating the Shell

A brownbag workshop at

MX®

by Seth House

@whiteinge
seth@eseth.com

Presentation Scope: Just the Basics

How not to look like your parents trying to use a computer.

Cursor

Archaic mappings and semantics for things you already know how to do.

Key	Description

Cursor

Archaic mappings and semantics for things you already know how to do.

Key	Description
`^a`	Home

Cursor

Archaic mappings and semantics for things you already know how to do.

Key	Description
`^a`	Home
`^e`	End

Cursor

Archaic mappings and semantics for things you already know how to do.

Key	Description
`^a`	Home
`^e`	End
`^w`	kill word

Cursor

Archaic mappings and semantics for things you already know how to do.

Key	Description
`^a`	Home
`^e`	End
`^w`	kill word
`^u`	kill line

Cursor

Archaic mappings and semantics for things you already know how to do.

Key	Description
`^a`	Home
`^e`	End
`^w`	kill word
`^u`	kill line
`^y`	yank last kill (i.e., paste)

Navigation

There and back again: A UNIX Tale.

Command	Description

Navigation

There and back again: A UNIX Tale.

Command	Description
`cd`	Go to `$HOME`

Navigation

There and back again: A UNIX Tale.

Command	Description
`cd`	Go to `$HOME`
`cd -`	Toggle between `$OLDPWD` and `$PWD`

Navigation

There and back again: A UNIX Tale.

Command	Description
`cd`	Go to `$HOME`
`cd -`	Toggle between `$OLDPWD` and `$PWD`
`..`	Up one directory

Navigation

There and back again: A UNIX Tale.

Command	Description
`cd`	Go to `$HOME`
`cd -`	Toggle between `$OLDPWD` and `$PWD`
`..`	Up one directory
`git checkout -`	Toggle between branches

History

How to never, ever, ever type (or remember) any command after the first time.

Tip	Description

History

How to never, ever, ever type (or remember) any command after the first time.

Tip	Description
`^r`	Start searching history

History

How to never, ever, ever type (or remember) any command after the first time.

Tip	Description
`^r`	Start searching history
`^r` (again)	Next match

History

How to never, ever, ever type (or remember) any command after the first time.

Tip	Description
`^r`	Start searching history
`^r` (again)	Next match
`^s`	Prev match (but won't work without `stty -ixon`)

History

How to never, ever, ever type (or remember) any command after the first time.

Tip	Description
`^r`	Start searching history
`^r` (again)	Next match
`^s`	Prev match (but won't work without `stty -ixon`)
`!!`	Substitute the last command

History

How to never, ever, ever type (or remember) any command after the first time.

Tip	Description
`^r`	Start searching history
`^r` (again)	Next match
`^s`	Prev match (but won't work without `stty -ixon`)
`!!`	Substitute the last command
`!$`	Substitute the last argument from the last command

History

How to never, ever, ever type (or remember) any command after the first time.

Tip	Description
`^r`	Start searching history
`^r` (again)	Next match
`^s`	Prev match (but won't work without `stty -ixon`)
`!!`	Substitute the last command
`!$`	Substitute the last argument from the last command

(We'll come back to this one.)

Completion

How to never blindly stumble around the filesystem.
(And self-discover some program arguments too.)

Tip	Description

Completion

How to never blindly stumble around the filesystem.
(And self-discover some program arguments too.)

Tip	Description
`Tab`	Show completion options

Completion

How to never blindly stumble around the filesystem.
(And self-discover some program arguments too.)

Tip	Description
`Tab`	Show completion options

(Sorta...)

Configuration

How to ruin your ability to ever use anyone else's shell.
(And ruin their ability to use your shell too.)

`$HOME/.inputrc`

`$HOME/.inputrc`

- Readline

`$HOME/.inputrc`

- Readline

  (Works in `irb` too!)

  ```
  $if Ruby
  ...ruby-specific stuff here...
  $endif
  ```

`$HOME/.inputrc`

- Readline

  (Works in `irb` too!)

  ```
  $if Ruby
  ...ruby-specific stuff here...
  $endif
  ```

- Temporary override (like on a shared server):

  `INPUTRC=/path/to/.inputrc-youruser exec bash`

```
# Complete by showing an ephemeral menu instead of printing matches.
TAB: menu-complete
# Enable shift-tab to complete backward.
"\e[Z": menu-complete-backward

# Show completion options on first tab; begin cycling on second tab.
set menu-complete-display-prefix on

# Show all available completion options on second tab.
set show-all-if-ambiguous on
set show-all-if-unmodified on

# Avoid duplicating text when completing an option mid-word.
set skip-completed-text on

# Use colors in the completion options.
set colored-stats on

# Treat symbols as word delimiters too.
set bind-tty-special-chars off

# Expand C-w to use more than just whitespace as a delimeter.
"\C-w": backward-kill-word

# Complete the current line from the shell history.
"\C-p": history-search-backward
"\C-n": history-search-forward
```

Extras

`$HOME/.bashrc`

$HOME/.bashrc

```
PS1='\w \$ '

alias ls='ls -F --color'
alias ll='ls -l'
```

Git Prompt Status & Completion

Homebrew, XCode, Apple's "Command Line Tools":
It's probably already on your system somewhere.

Git Prompt Status & Completion

Homebrew, XCode, Apple's "Command Line Tools":
It's probably already on your system somewhere.

```
% find / -name 'git-prompt\.sh' 2>/dev/null
/usr/local/etc/bash_completion.d/git-prompt.sh
/Library/Developer/CommandLineTools/usr/share/git-core/git-prompt.sh
/Applications/Xcode.app/Contents/Developer/usr/share/git-core/git-pr
```

```
% find / -name 'git-completion\.bash' 2>/dev/null
/usr/local/etc/bash_completion.d/git-completion.bash
/Library/Developer/CommandLineTools/usr/share/git-core/git-completio
/Applications/Xcode.app/Contents/Developer/usr/share/git-core/git-co
```

$HOME/.bashrc

```
source /usr/local/etc/bash_completion.d/git-prompt.sh
source /usr/local/etc/bash_completion.d/git-completion.bash

GIT_PS1_SHOWCOLORHINTS=1
GIT_PS1_SHOWDIRTYSTATE=1
GIT_PS1_SHOWSTASHSTATE=1
GIT_PS1_SHOWUPSTREAM="verbose"

function __git_ps1_sha() { git rev-parse --short HEAD 2>/dev/null; }

PROMPT_COMMAND='__git_ps1 "\w" " \$ " " ($(__git_ps1_sha) %s)"'
```

Discussion!