

Console Workshop  
Make the terminal great again.

Anne Steenbeek

## ■ Topics

- Dotenv Management
  - Make your stuff persistent
- ZSH
  - Plugins
  - Themes
  - Config
- TMUX
  - Config
- Usefull Tips & Tricks

## Requirements

### ■ Make sure you have

- ZSH shell installed
- Git installed
- TMUX installed

### Ubuntu

```
$ sudo apt update
$ sudo apt install curl git zsh tmux -y
$ chsh -s $(which zsh) # make ZSH default shell (optional)
```

### OSX

```
$ brew install tmux git curl
```

### ■ Check out the repo!

It is important to be able to manage your dotfiles correctly.  
This allows you to use the same configuration accross all your devices.  
Use version control to keep them in sync.

- Navigate to [https://github.com/annesteenbeek/console\\_workshop](https://github.com/annesteenbeek/console_workshop)
- Use this template (or just clone it if you don't have a github account)
- Name it something like dotfiles
- Clone it in your home directory: `$ cd ~ && git clone https://github.com/YOU/dotfiles`
- Open it in your favorite code editor (like vim)
- backup your `~/.zshrc` and `~/.tmux.conf` if you already have them
- Run it: `$ ./dotfiles/install`

### ■ How it works

- It pulls the dotbot submodule
- It runs dotbot script to run install scripts
- It creates symlinks to your home directory

## ZSH Plugin Manager

■ We use antigen to handle plugins

- webpage: <https://github.com/zsh-users/antigen>
- `$ mkdir ~/.antigen`
- `$ curl -L git.io/antigen > ~/.antigen/antigen.zsh`
- ✨ Magic ✨

~/.zshrc

```
source ~/.antigen/antigen.zsh

# Load oh-my-zsh library
antigen use oh-my-zsh

# load all your plugins here

# done
antigen apply
```

Apply: `$ source ~/.zshrc`

## SpaceShip Theme 🚀

### ■ ZSH allows for awesome themes

There are a lot of built in themes with oh-my-zsh But none are as complete as SpaceShip Prompt

### ▣ Configuration (you can skip this)

It has a lot of configuration options Webpage:  
<https://spaceship-prompt.sh/config/intro/>

Example `$ touch ~/.spaceshiprc.zsh`

```
SPACESHIP_CHAR_SYMBOL="🚀"

# Display time
SPACESHIP_TIME_SHOW=true

# Display username always
SPACESHIP_USER_SHOW=always

# Do not truncate path in repos
SPACESHIP_DIR_TRUNC_REPO=false
```

~/.zshrc

```
source ~/.antigen/antigen.zsh

# Load oh-my-zsh library
antigen use oh-my-zsh

# load all your plugins here
antigen theme spaceship-prompt/spaceship-prompt

# done
antigen apply
```

Apply: `$ source ~/.zshrc`

## ZSH Autosuggestions

■ Never double type a command again

- Provides autosuggestions
- Accept suggestion with `ctrl+e`
- webpage: <https://github.com/zsh-users/zsh-autosuggestions>

~/.zshrc

```
source ~/.antigen/antigen.zsh

# Load oh-my-zsh library
antigen use oh-my-zsh

# load all your plugins here
antigen theme spaceship-prompt/spaceship-prompt
antigen bundle zsh-users/zsh-autosuggestions

# done
antigen apply
```

Apply: `$ source ~/.zshrc`

## FZF

### A command line fuzzy finder

FZF is a better, fuzzy finding, autocompleter

### Install

```
$ git clone --depth 1 https://github.com/junegunn/fzf.git ~/.fzf
$ ~/.fzf/install --bin
```

### Usage

- Try it with `ctrl+r` history search
- Type letters you remember: eg. apt install
- Use arrows or `ctrl+n`/`ctrl+p` to move

~/.zshrc

```
source ~/.antigen/antigen.zsh

# Load oh-my-zsh library
antigen use oh-my-zsh

# load all your plugins here
antigen theme spaceship-prompt/spaceship-prompt
antigen bundle zsh-users/zsh-autosuggestions
antigen bundle fzf

# done
antigen apply

setopt HIST_IGNORE_DUPS # no double history
```

Apply: `$ source ~/.zshrc`



## FASD

### ■ Move around the terminal faster

- Use fuzzy finding to jump to directories in your history
- Use tab style completion with fzf

### ■ Example

- `$ take -p ~/this/is/a/nested/directory` Create nested directory and cd into it
- `$ cd` go back home
- `$ z nest` fasd into the directory
- Try it with tab

~/.zshrc

```
source ~/.antigen/antigen.zsh

# Load oh-my-zsh library
antigen use oh-my-zsh

# load all your plugins here
antigen theme spaceship-prompt/spaceship-prompt
antigen bundle zsh-users/zsh-autosuggestions
antigen bundle fzf
antigen bundle clvv/fasd fasd
antigen bundle wookayin/fzf-fasd

# done
antigen apply

setopt HIST_IGNORE_DUPS # no double history

# initialize fasd
eval "$(fasd --init auto)"
```

Apply: `$ source ~/.zshrc`

## ZSH tips and general tips

### Command line tips

- `..` instead of `cd ..`
- `...` instead of `cd ../../`
- `!!` to run previous command. For example: `sudo !!` if you forgot sudo.
- `ctrl+x e` to edit current command (remap with: bindkey '^x' edit-command-line)
- `curl ifconfig.me` to get current public IP
- `wormhole send` and `wormhole receive` to easily transfer files (`sudo apt install python-wormhole`)
- You can fuzzy autocomplete paths with tab, example: `cd /l/s/s -` will autocomplete to `cd /lib/systemd/system/`

### Other nice ZSH plugins:

Check them out at: <https://github.com/ohmyzsh/ohmyzsh/wiki/Plugins> Or google for awesome lists. My tips:

- zsh-users/zsh-completions (more autocomplete suggestions)
- arzzen/calc.plugin.zsh (calculator in your terminal, try: `$ = 1 + 1`)
- per-directory-history (toggle history only for this directory)
- zsh-users/zsh-syntax-highlighting (highlight working commands)
- command-not-found (suggest package on command not found)

### Other cool tools:

- Ranger: a file explorer for in your terminal <https://github.com/ranger/ranger>
- LSD: an alternative and better ls <https://github.com/lsd-rs/lsd>
- bat: a better cat <https://github.com/sharkdp/bat>

### General tips

- `tab` is your friend, use it!
- `ctrl-` is your friend
- Try to not use your mouse, it WILL become easier
- Learn a little bit vim bindings (hjkl gg G :q! /), it works everywhere

## TMUX

Tmux is a terminal multiplexer It allows you to run multiple shells inside a single terminal

### ■ Tmux vs iTerm2/Terminator

- Session handling: sessions are separate from terminal
- If your terminal dies, your work lives on!
- Platform agnostic: works on server, raspberry PI, OSX or linux with same configuration
- You don't have to be there for it to work, start on server, attach to inspect logs later
- Can be automated and scripted
- Customizability!

## ■ Configuring

- Tmux has unreasonable defaults
- Lets make them a bit more reasonable

~/tmux.conf

```
# change prefix to backtick
unbind-key C-b
set-option -g prefix `
bind-key ` send-prefix

# split panes using | and -
bind | split-window -h
bind - split-window -v
unbind '"'
unbind %

# bind reload conf to prefix-r
bind r source-file ~/.tmux.conf \; display "Reloaded config"

# enable mouse
set -g mouse on
```

To apply: restart Tmux, or thereafter **prefix-r**

## Controlling Tmux

Tmux has 3 basic levels of splitting

### Levels

1. Sessions
2. Windows
3. Panes

### Important keybindings

Key	Command
<code>` + arrows</code>	Move between panes
<code>` + c</code>	New window
<code>` + number</code>	move to window number
<code>` + z</code>	Toggle Maximize pane
<code>` + x</code>	Kill pane
<code>` + [</code>	Scroll mode
<code>q</code>	Exit scroll mode
<code>` + s</code>	Browse sessions

### How to:

1. `$ tmux` creates a new session in your current shell
2. `prefix+c` creates a new window
3. `splitting` creates new panes

### Important commands

<code>\$ command</code>	Command
<code>tmux</code>	create new tmux session
<code>tmux ls</code>	list sessions
<code>tmux a -t #</code>	attach to tmux session #

## Tmux Plugin Manager

Ofcourse Tmux also needs plugins!

For this we use TPM:

```
git clone https://github.com/tmux-plugins/tpm ~/.tmux/plugins/tpm
```

You can find more awesome tmux plugins and themes here:

<https://github.com/rothgar/awesome-tmux>

Some of these might require NerdFonts <https://www.nerdfonts.com/> in order to look pretty. (Try Hack Font)

Then try a theme like `set -g @plugin "nordtheme/tmux"`

`~/.tmux.conf`

```
# List of plugins
set -g @plugin 'tmux-plugins/tpm'
set -g @plugin 'tmux-plugins/tmux-sensible'

# change prefix to backtick
unbind-key C-b
set-option -g prefix `
bind-key ` send-prefix

# split panes using | and -
bind | split-window -h
bind - split-window -v
unbind '"'
unbind %

# bind reload conf to prefix-r
bind r source-file ~/.tmux.conf \; display "Reloaded config"

# enable mouse
set -g mouse on
```

```
run '~/.tmux/plugins/tpm/tpm'
```

To apply: restart Tmux, or thereafter `prefix-r` Install plugins: `prefix-I`

```
# Initialize TMUX plugin manager (keep this line at the very bottom of tmux.conf)
```

Thank you

Thank you for your time.

Try stuff out, experiment, everything is possible.





