

# CREATE BETTER DOTFILES

---



TEXT

---

# ANDREW WOODS

- ❖ Software Engineer at Paramount
- ❖ Formerly of Seattle PHP
- ❖ Live in NYC
- ❖ <https://joind.in/talk/c30ff>



There's no place  
like \$HOME

The universal is  
in the personal

**FOUNDATION**

---

## ORGANIZATION

- ▶ Smart small
- ▶ Divide into files
- ▶ Give them good names
- ▶ Shallow includes
- ▶ Conditional load

bin

share

doc

src

etc

tmp

opt

var

## HOME DIRECTORIES

- ▶ `bin` Custom commands
- ▶ `etc` User-specific system configuration
- ▶ `opt` Third-party packages/libraries
- ▶ `tmp` Temporary files
- ▶ `var` Variable data



`$HOME/.config`

## XDG BASE DIRECTORY SPECIFICATION

- ▶ XDG = Cross-Desktop Group
- ▶ Rebranded as [freedesktop.org](https://freedesktop.org)
- ▶ defines a set of directories for user data
- ▶ cleans up your \$HOME directory
- ▶ Provides a standard for application devs

`XDG_CONFIG_HOME`

`$HOME/.config`

`XDG_CACHE_HOME`

`$HOME/.cache`

`XDG_DATA_HOME`

`$HOME/.local/share`

XDG\_STATE\_HOME

\$HOME/.local/state

`$HOME/dotfiles`

## DOTFILES DIRECTORIES

- ▶ `bin, doc, etc`
- ▶ `config (nvim, tmux)`
- ▶ `shells (bash, zsh)`
- ▶ `git`



# BASH

---

`$HOME/.bash_profile`

```
if [ -f ~/.bashrc ]; then
```

```
    source ~/.bashrc
```

```
fi
```

```
export XDG_CONFIG_HOME=$HOME/.config
```

```
export XDG_CACHE_HOME=$HOME/.cache
```

```
export XDG_DATA_HOME=$HOME/.local/share
```

```
export XDG_STATE_HOME=$HOME/.local/state
```

```
export XDG_CONFIG_HOME=$HOME/.config
```

```
export XDG_CACHE_HOME=$HOME/.cache
```

```
export XDG_DATA_HOME=$HOME/.local/share
```

```
export XDG_STATE_HOME=$HOME/.local/state
```

```
PATH="$PATH:$HOME/bin"
```

```
PATH="$PATH:$HOME/ dotfiles/bin"
```

```
PATH="$PATH:$HOME/.local/bin"
```

```
export BASH_SILENCE_DEPRECATION_WARNING=1
```

`$HOME/.bashrc`



## BASHRC

- ▶ Variables
- ▶ Settings
- ▶ Functions
- ▶ Aliases
- ▶ Main

```
9 # File: ~/.bashrc
8 # Date Created: 2024-04-25
7 # Date Updated: 2024-04-25
6 #
5
4 #####
3 #   Variables
2 #####
1
11 █
1
2 #####
3 #   Settings
4 #####
5
6
7
8 #####
9 #   Functions
10 #####
11
12
13
14 #####
15 #   Aliases
16 #####
17
18
```

TEXT

---

## VARIABLES

- ▶ quick navigation to directories
- ▶ long filenames

## SETTINGS

- ▶ Shell Options (shopt)
- ▶ Settings

```
shopt -s cdspell
```

```
shopt -s checkwinsize
```

```
shopt -s extdebug
```

```
shopt -s histappend
```

```
shopt -s huponexit
```

```
shopt -s nocaseglob
```

```
set -o vi
```

```
set -o posix
```

```
set -o nounset
```

## FUNCTIONS

- ▶ Be more effective
- ▶ Greater consistency
- ▶ Add value to your work

## MAKING FUNCTIONS

- ▶ Prototype on the command line
- ▶ Wrap it
- ▶ Substitute values
- ▶ Add argument handling



## WHAT DO YOU WANT TO MAKE?

- ▶ OS-specific functions
- ▶ App Specific
- ▶ Wrap commands

```
$ mac:db_import_notify
```

```
function mac:db_import_notify {  
    osascript -e 'display notification  
        "Database import complete"  
        with title  
        "PHPTek" '  
}
```

```
$ tek:mk_branch ARCHMAG-0423 \
```

```
$ > 'Edit Community Corner for 2024 June'
```

```
function tek:mk_branch {  
    local title=$(opal:str_slug "$2")  
  
    local branch_name="feature/${1}--${title}"  
    git checkout -b $branch_name  
}
```

```
$ mac:restart_touchbar
```

```
function mac:restart_touchbar {  
    sudo pkill TouchBarServer  
    sudo killall ControlStrip  
}
```

```
$ tek:num_seg ~/dotfiles/mac.bash 190 210
```



```
function tek:num_seg {  
    local filename=$1  
    local -i start=$2  
    local -i end=$3  
  
    nl $filename | awk "NR ≥ $start && NR ≤ $end"  
}
```

declare -F

```
$ tek:show
```

```
function tek:show {  
    if [[ -n "$1" ]]; then  
        shopt -s extdebug  
        declare -F "$1"  
        shopt -u extdebug  
    else  
        declare -F  
    fi  
}
```

```
declare -f opal:swap
declare -f opal:symlink_exists
declare -f opal:term_bg
declare -f opal:today
declare -f opal:touchx
declare -f opal:trace_url
declare -f opal:truncate
declare -f opal:type_file
declare -f opal:type_line
declare -f opal:version
declare -f punch
declare -f say_done
declare -f show_dotfiles
declare -f tabname
declare -f tek:describe
declare -f tek:show
declare -f wget
declare -f whoops
declare -f width_check
declare -f winname
```

```
main
awoods@Aerilon
2024 Apr 24 Wed 22:01 ~/Dropbox/talks/phptek-2024
$
```

```
$ tek:show opał:truncate
```

```
opal:truncate 282 ~/opal/bash/functions.files-  
and-dirs.bash
```

tek:str\_diff phptek2023 phpTek2024



```
function tek:str_diff()  
{  
    diff -u <(echo "$1") <(echo "$2") |  
        diff-so-fancy  
}
```

2024 Apr 24 Wed 23:11 ~

\$ tek:str\_diff phptek2023 phpTek2024

---

renamed: /dev/fd/63 to /dev/fd/62

---

@ /dev/fd/62:1 @

phptek2023

phpTek2024

awoods@Aerilon

2024 Apr 24 Wed 23:11 ~

\$ █

## ALIASES

- ▶ good name
- ▶ map a longer function name
- ▶ fill a gap

```
# Pretty print the value of $PATH
```

```
alias path='echo $PATH | tr -s ":" "\n"'
```

```
# for Linux Systems
```

```
alias pbcopy='xclip -selection clipboard'
```

TEXT

---

## MAIN

- ▶ Prompt Functions
- ▶ Theme Functions
- ▶ Clacks

```
# ~/.bashrc - in the Main Section
```

```
tek:ps2_dark
```

```
tek:ps3
```

```
tek:ps4
```

```
function tek:ps2_dark {  
    PS2=""  
    PS2+="\[\e[1;37m\]"      # Color: Bright White  
    PS2+="\$"                #  
    PS2+="\[\e[1;36m\]"     # Color: Cyan  
    PS2+=" >"                #  
    PS2+="\[\e[0m\]  "  
}
```



```
function tek:ps3 {  
    PS3=""  
    PS3+="Choose a #> "  
}
```

```
function tek:ps4 {  
    PS4="\n"  
    PS4+='source-file: ${BASH_SOURCE}\n'  
    PS4+='Function: ${FUNCNAME[0]:+${FUNCNAME[0]}} \n'  
    PS4+='Line: ${LINENO} \n'  
    PS4+="> "  
    export PS4  
}
```

```
utc; eastern; pacific
```

**TMUX**

---

TEXT

---

## TMUX.CONF

- ▶ Key Bindings
- ▶ Settings

```
$ aerilon.tmux
```



TEXT

# THANK YOU

❖ Andrew Woods

❖ [@awoodsnet@phpc.social](mailto:@awoodsnet@phpc.social)

❖ [andrewwoods.net](http://andrewwoods.net)

❖ <https://joind.in/talk/c30ff>

