

BEGINNING BASH

TONY WILLIAMS

X WORLD 2016

SETTING UP TERMINAL

LOOKING GOOD

- A Good Profile 'Solarized'
- Setting Font and Size
- Setting Window Size

BEFORE WE BEGIN

- Homebrew
- coreutils
- dockutil
- source-highlight

SOURCE A FILE

- `bash_profile.sh`

DEFAULTS READ

```
defaults read com.apple.dock | less
```

DEFAULTS

```
defaults write com.apple.dock autohide -bool YES ; killall Dock
defaults write com.apple.dock autohide -bool NO ; killall Dock
defaults write com.apple.dock orientation "right" ; killall Dock
defaults write com.apple.dock orientation "bottom" ; killall Dock
```

SHELL PROGRAMMING

VARIABLES

```
WORD="Foobar" ; echo $WORD
```

PRETTY PROMPT

```
PS1="\[\033[34m\]\h:\w \u\$\[\033[0m\] "
```

ALIAS FUN

- Commonly used options
- Hard to remember options

ALIAS EXAMPLES

```
LS_OPTIONS="--color=auto -F -G"  
# standard ls coloured  
alias ls='gls $LS_OPTIONS'  
# standard plus dot files  
alias la='gls $LS_OPTIONS -A'  
# long ls  
alias ll='gls $LS_OPTIONS -l'  
alias l='gls $LS_OPTIONS -lA'
```

FIRST STEPS - JUST A LITTLE FUNCTION

```
# function to send man page to preview  
manp()  
{  
man -t $* | open -f -a /Applications/Preview.app/  
}
```

DECISIONS, DECISIONS

- If, else, then
- Case
- Using '&&' and '||'

EXAMPLE DECISION

```
# make root red  
if [ `id -u` = 0 ]  
then  
    PS1="\[\033[31m\]\h:\W \u\$\[\033[0m\] "  
fi
```

COLOUR LS

```
# colours for the Gnu ls (from coreutils)  
if [ "$TERM" != "dumb" ]; then  
    export LS_OPTIONS='--color=auto -F -G -h'  
    eval `gdircolors ~/.dircolors`  
fi
```


MAKE A CASE

```
#!/bin/bash

echo -n "Word: " ; read WORD

case $WORD in
    ( "Foo" )      echo "Bar" ;;
    ( "Bar" )      echo "Foo" ;;
    ( "FooBar" )   echo "No Way" ;;
    ( * )          echo "FooBar" ;;
esac
```

USING && AND ||

A AND B OR C

```
[ `id -u` = 0 ] && PS1="\[\033[31m\]\h:\w \u\$\[\033[0m\] " \  
|| PS1="\[\033[34m\]\h:\w \u\$\[\033[0m\] "
```

CHECKING THE RESULT CODE

```
if ls mysillyfilename ; then
    echo "File exists."
fi

# checking result code variable
ls mysillyfilename
if [ $? = 0 ] ; then
    echo "File exists."
fi
```

CHECKING A FILE

```
if [ -e README.md ] ; then  
    echo "Readme exists"  
fi
```

FILE CHECKS

-a if file exists

-d if file is directory

-e if file exists

-r if file is readable

-w if file is writeable

-x if file is executable

-O is owned by the user

-G is owned by the group

ROUND AND ROUND

- For
- While
- Done

WHILE LOOP

```
#!/bin/bash

echo -n "Word: " ; read WORD

while [ $WORD -ne "" ] ; do
    case $WORD in
        ( "Foo" )      echo "Bar" ;;
        ( "Bar" )      echo "Foo" ;;
        ( "FooBar" )   echo "No Way" ;;
        ( * )          echo "FooBar" ;;
    esac
done
```

FOR EVER

```
for file in *.sh ; do  
    echo $file  
done
```


EVEN MORE FOR

```
for (( i = 1 ; i <= $1 ; i++ )) ; do  
    echo "I is $i"  
done
```

EXPANDING VARIABLES

```
LIST="Foo Bar Baz"  
for i in $LIST ; do  
    echo $i  
done
```

FIELD SEPARATOR

```
IFS=":"  
LIST="a:b:c d"  
for i in $LIST ; do  
    echo $i  
done
```

SPECIAL CHARACTERS IN STRINGS

```
IFS=$'\t'  
LIST=$'a\tb\tc d'  
for i in $LIST ; do  
    echo $i  
done
```

REDIRECTION

HOW IT'S DONE

- Fresh output `ls > files.txt`
- Append to the file `ls >> files.txt`

USING TEE

- Fresh file `ls | tee files.txt`
- Append to the file `ls | tee -a files.txt`

BOTH OUT AND ERR

If you want to log to the same file:

```
command1 >> log_file 2>&1
```

If you want different files:

```
command1 >> log_file 2>> err_file
```


HERE DOCUMENTS

```
logger -t SYSADMIN <<EOM  
I am going to reboot the system  
I'm doing it because I want to  
EOM
```

MATHS USING EXPR

```
#!/bin/bash
```

```
WEEKS=$1
```

```
DAYS=`expr $WEEKS '*' '7'`
```

```
date -v +${DAYS}d
```

MATHS USING EXPANSION

```
#!/bin/bash
```

```
WEEKS=$1
```

```
DAYS=$(( $WEEKS*7 ))
```

```
date -v +${DAYS}d
```

Or even echo $((7 * (4 + 2)))$ Though the spaces are not required echo $((7 * (4 + 2)))$

IMPROVE THE LOOK

```
WEEKS=$1
DAYS=`expr $WEEKS '*' '7'`
date -v +${DAYS}d | awk '{ print $1 " " $2 " " $3 " " $4 }'
```

LAUNCHAGENTS & LAUNCHDAEMONS

- LaunchAgents run at user login
- definition found in ~/Library/LaunchAgents/
- LaunchDaemons run at boot
- definition found in /System/Library/LaunchDaemons/

CONTROLLING LAUNCHAGENTS

- plist file
- named 'something.command.plist' e.g.
- com.honestpuck.command.plist

GIVING IT A TRY

- `agent_test.sh`
- `com.honestpuck.test.plist`

A COMPLEX EXAMPLE

- local.job.plist

PUTTING IT ALL TOGETHER

- `dock_setup.sh`

GETTING A PROFILE

- `bash_profile.sh`

PLACES TO GO

- This presentation <http://bit.ly/xw2016-9>
- Apple have a good bash tutorial <http://apple.co/1MFChLE>
- Good bash tutorial <http://bit.ly/xw2016-2>
- Comprehensive awk tutorial <http://bit.ly/xw2016-5>
- Effective AWK Programming <http://bit.ly/xw2016-7>
- All About Redirection <http://bit.ly/xw2016-1>

MORE PLACES TO GO

- Apple's page on Launch Daemons and Agents
<http://bit.ly/xw2016-8>
- Lingon X <http://bit.ly/xw2016-6>
- LaunchControl <http://bit.ly/xw2016-11>
- Good tutorial on LaunchAgent and Daemon format
<http://launchd.info>
- My home directory on git <http://bit.ly/xw2016-10>
- MacAdmins Slack channel
<https://macadmins.herokuapp.com>

A FINAL LIST

- Input font <http://input.fontbureau.com>
- Solarized colour theme
<http://ethanschoonover.com/solarized>
- bash completion <http://bit.ly/xw2016-12>
- shellcheck, a script checker <http://www.shellcheck.net>