Contents

1 Bash best practices

An attempt to bring order in good advice on writing Bash scripts I collected from several sources.

1.1 General

- Always check for syntax errors and use ShellCheck. Integrate this in your text editor (e.g. Syntastic plugin in Vim)
- The principles of Clean Code apply to Bash as well
- Always use long parameter notation when available

```
# Avoid:
rm -rf "${dir}"

# Good:
rm --recursive --force "${dir}"
```

1.2 Variables

- Prefer local variables within functions over global variables
- If you need global variables, make them readonly
- Variable should always be referred to in the `${var}` form (as opposed to `$var`.
- Variables should always be quoted: `"${var}"`
- Capitalization:
  - Readonly/environment variables: `${ALL_CAPS}`
  - Local/mutable variables: `${lower_case}`
- Declare variables with a meaningful name for positional parameters of functions

```
foo() {
  local first_arg="${1}"
  local second_arg="${2}"
  [...]
}
```

- Positional parameters of the script should be checked, those of functions should not

1.3 Substitution

- Always use `$(cmd)` for command substitution (as opposed to backquotes)

1.4 Functions

Bash can be hard to read and interpret. Using functions can greatly improve readability. Principles from Clean Code apply here.

- Apply the Single Responsibility Principle: a function does one thing.
- Create functions with a meaningful name for complex tests

```
# Don't do this
if [ "$#" -ge "1" ] && [ "$1" = '-h' -o "$1" = '--help' -o "$1" = "-?" ]; then
  usage
  exit 0
fi

# Do this
help_wanted() {
  [ "$#" -ge "1" ] && [ "$1" = '-h' -o "$1" = '--help' -o "$1" = "-?" ]
}

if help_wanted "$@"; then
  usage
  exit 0
fi
```

- Don't mix levels of abstraction
- Describe the usage of each function: number of arguments, return value, output

1.5 Cleanup code

An idiom for tasks that need to be done before the script ends (e.g. removing temporary files, etc.).

```
function finish {
  # Your cleanup code here
}
trap finish EXIT
```

Source: Aaron Maxwell, How “Exit Traps” can make your Bash scripts way more robust and reliable.

1.6 Shell script template

An annotated template for Bash shell scripts:

For now, see https://github.com/bertvv/dotfiles/blob/master/.vim/templates/sh

1.7 Resources

- Free Software Foundation (2014). Bash Reference Manual, https://www.gnu.org/software/bash/manual/bashref.html
- Brian Fox and Chet Ramey (2009). bash(1) man page, http://linux.die.net/man/1/bash
- Mendel Cooper (2014). The Advanced Bash Scripting Guide (ABS) http://www.tldp.org/LDP/abs/html/
- Defensive Bash Programming (2012), http://www.kfirlavi.com/blog/2012/11/14/defensive-bash-programming
- Aaron Maxwell (2014), *Use the Unofficial Bash Strict Mode (Unless You Looove Debugging)*. http://redsymbol.net/articles/unofficial-bash-strict-mode/

- Yoann Bentz (2016), Good practices for writing shell scripts http://www.yoone.eu/articles/2-good-practices-for-writing-shell-scr
  html
- Paul Armstrong (s.d.), Shell Style Guide https://google.github.io/styleguide/shell.xml
- Pádraig Brady (2008), Common Shell Script Mistakes http://www.pixelbeat.org/programming/shell_script_mistakes.
  html
- Avery Pennarun (2011). Insufficiently known POSIX shell features http://apenwarr.ca/log/?m=201102#28
- BashFAQ http://mywiki.wooledge.org/BashFAQ
- Simon Sheppard (s.d.). Bash Keyboard Shortcuts. http://ss64.com/bash/syntax-keyboard.html
- M.Tim Jones (2011) Evolution of shells in Linux: From Bourne to Bash and beyond. https://www.ibm.com/
  developerworks/library/l-linux-shells/
- Donny Berkholz (2011) Bash shell-scripting libraries. https://dberkholz.com/2011/04/07/bash-shell-scripting-libraries/
  ### Templates
- bash-script-template https://github.com/ralish/bash-script-template
- Bash3 Boilerplate http://bash3boilerplate.sh/

1.7.1 Portable shell scripts

- https://wiki.ubuntu.com/DashAsBinSh
- http://pubs.opengroup.org/onlinepubs/009695399/utilities/contents.html
- http://sites.harvard.edu/~lib113/reference/unix/portable_scripting.html
- https://www.gnu.org/software/autoconf/manual/autoconf.html#Portable-Shell