FREE eBook

LEARNING zsh

Free unaffiliated eBook created from
Stack Overflow contributors.

#zsh

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: zsh

It is an unofficial and free zsh ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official zsh.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with zsh

Remarks

`zsh` is a POSIX-compatible shell, and a popular alternative to the Bourne shell (`sh`) and `bash`.

Its key feature is a focus on a high level of customization by the user, which has led to an active community of developers creating extensions for `zsh`, including custom, more informative prompt status lines, often integrating with system services.

Many configurations with large sets of sensible defaults and useful extensions exist online, including the popular oh-my-zsh and prezto.

Versions

Version	Release Date
5.3.1	2016-12-21
5.3	2016-12-12
5.2	2015-12-02
5.1.1	2015-09-11
5.1	2015-08-30
5.0.8	2015-05-31
5.0.0	2012-07-24
4.3.17 (beta)	2012-02-23
4.2.7	2007-12-18
4.3.1 (beta)	2006-02-28
4.2.0	2004-03-19
4.0.9	2003-12-19
4.1.1 (beta)	2003-06-19
4.0.1	2001-06-01
3.1.9	2000-06-05
3.0.8	2000-05-16

Version	Release Date
3.1.6 (beta)	1999-08-01
3.0.0	1996-08-15
2.6-beta21	1996-06-19
2.6-beta1	1994-10-16
2.5.0	1994-07-14
2.3.1	1993-02-20
2.2	1992-05-13
2.1	1991-10-24
2.0	1991-04-24
1.0	1990-12-15

Examples

Installation or Setup

Getting `zsh`

`zsh` is available on many UNIX-like platforms via their built-in package management systems. On the Debian and Ubuntu Linux distributions, `zsh` is available in the default package repositories and can be installed using:

```
$ sudo apt-get install zsh
# or, on newer Ubuntu distributions
$ sudo apt install zsh
```

On RPM-based distributions, `zsh` is also often available in the default package archives and can be installed using:

```
$ yum install zsh
```

On Fedora 22 and later:

```
$ dnf install zsh
```

On BSD systems, `zsh` can be installed using `pkg`:

```
$ pkg install zsh
```

On OpenBSD, `zsh` can be installed using `pkg_add`:

```
$ pkg_add zsh
```

On Arch Linux, `zsh` can be installed using `pacman`:

```
$ pacman -S zsh
```

On openSUSE, `zsh` can be installed using `zypper`:

```
$ zypper install zsh
```

On systems running macOS (OS X) `zsh` is already installed by default, although not set as default shell. You can also install newer versions via Homebrew:

```
$ brew install zsh
```

Alternatively, `zsh`'s source code can be obtained from the official website.

From there, the shell can be started by typing `zsh` at the prompt.

Making `zsh` your default shell

On most Linux and BSD systems, `zsh` may be set as the default shell for a user using the `chsh` command:

```
$ chsh -s shell [username]
```

Where

- `username` is a real username (defaults to the current user if left out)
- `shell` is the path to the `zsh` binary. The path should be listed in the `/etc/shells` file, which contains a list of allowed shells for use with `chsh`. Should `zsh` not be listed there - for example because you compiled and installed it from source - you will need to add a line with the absolute path to `zsh` first. You can get this path with `which zsh` (provided it is installed in a directory listed in `PATH`)

In order to see the changes log out once and log in. Open the terminal emulator and use

```
`echo $SHELL`
```

If it displays `/bin/zsh` then you have successfully changed the default shell to zsh.

Configuration

When starting Zsh, it'll source the following files in this order by default:

1. `/etc/zsh/zshenv` Used for setting system-wide environment variables; it should not contain commands that produce output or assume the shell is attached to a tty. This file will always be sourced, this cannot be overridden.

2. `$ZDOTDIR/.zshenv` Used for setting user's environment variables; it should not contain commands that produce output or assume the shell is attached to a tty. This file will always be sourced.

3. `/etc/zsh/zprofile` Used for executing commands at start, will be sourced when starting as a login shell.

   Note that on Arch Linux, by default it contains one line which source the /etc/profile.

   `/etc/profile` This file should be sourced by all Bourne-compatible shells upon login: it sets up `$PATH` and other environment variables and application-specific (/etc/profile.d/*.sh) settings upon login.

4. `$ZDOTDIR/.zprofile` Used for executing user's commands at start, will be sourced when starting as a login shell.

5. `/etc/zsh/zshrc` Used for setting interactive shell configuration and executing commands, will be sourced when starting as a interactive shell.

6. `$ZDOTDIR/.zshrc` Used for setting user's interactive shell configuration and executing commands, will be sourced when starting as a interactive shell.

7. `/etc/zsh/zlogin` Used for executing commands at ending of initial progress, will be sourced when starting as a login shell.

8. `$ZDOTDIR/.zlogin` Used for executing user's commands at ending of initial progress, will be sourced when starting as a login shell.

9. `$ZDOTDIR/.zlogout` Will be sourced when a login shell exits.

10. `/etc/zsh/zlogout` Will be sourced when a login shell exits.

    If $ZDOTDIR is not set, $HOME is used instead.

    For general personal use, it is typical to edit the user's `.zshrc` file for personal preferences

Aliases

To alias a command in you `~/.zshrc` file, you can use the following syntax:

```
alias [alias-name]="[command-to-execute]"
```

For example, it is common to execute the command `ls -a`. You can alias this command as `la` as

such:

```
alias la="ls -a"
```

After reloading the `~/.zshrc` file, you will be able to type `la` and `ls -a` will be executed.

Directory Aliases

It is common to have certain folders that you `cd` into often. If this is the case, you can create aliasses to those directories to make `cd`ing to them easier. For example, the following will alias the Dropbox folder:

```
alias db="cd ~/Dropbox"
```

allowing you to enter `db` and change directories to `~/Dropbox`.

Reload ZSH Configuration

zsh loads configuration from the `~/.zshrc` file on startup. If you make changes to that file, you can either restart zsh or run the following command to reload the configuration.

```
. ~/.zshrc
```

You can alias this useful command in your `~/.zshrc` like this:

```
alias reload=". ~/.zshrc"
```

Read Getting started with zsh online: https://riptutorial.com/zsh/topic/4570/getting-started-with-zsh

Chapter 2: Main differences from bash

Examples

Pipes and subshells

In bash, every command in a pipeline is executed in a subshell. In zsh, the last command in a pipeline is executed in current shell. For example, the following code

```
var="before"
echo "after" | read var
echo $var
```

will print *before* in bash, but *after* in zsh.

Differences in quoting

In bash, you have to quote arguments in order to preserve white space:

```
# bash

function print_first_argument {
    echo "$1"
}

argument="has white space"
print_first_argument "$argument"
```

In Zsh, you don't need the quotes, because of different evaluation order:

```
# zsh

function print_first_argument {
    echo $1
}

argument="has white space"
print_first_argument $argument
```

Wildcard Handling

When nothing matches a wildcard such as `*` in bash, it gets passed on as a literal `*` to the command, as if you had typed `\*`. However, zsh throws an error.

Bash:

```
duncan@K7DXS-Laptop-Arch:~/test$ echo *.txt
*.txt
duncan@K7DXS-Laptop-Arch:~/test$ touch abc.txt
```

```
duncan@K7DXS-Laptop-Arch:~/test$ echo *.txt
abc.txt
duncan@K7DXS-Laptop-Arch:~/test$
```

Zsh:

```
K7DXS-Laptop-Arch% echo *.txt
abc.txt
K7DXS-Laptop-Arch% rm abc.txt
K7DXS-Laptop-Arch% echo *.txt
zsh: no matches found: *.txt
K7DXS-Laptop-Arch%
```

This is most noticeable in programs that use a literal `*`, such as find:

```
duncan@K7DXS-Laptop-Arch:~/test$ ls -R
.:
abc

./abc:
123.txt
```

Bash:

```
duncan@K7DXS-Laptop-Arch:~/test$ find -name *.txt
./abc/123.txt
duncan@K7DXS-Laptop-Arch:~/test$
```

Zsh:

```
K7DXS-Laptop-Arch% find -name *.txt
zsh: no matches found: *.txt
K7DXS-Laptop-Arch% find -name \*.txt
./abc/123.txt
K7DXS-Laptop-Arch% find -name '*.txt' # Notice single rather than double quotes
./abc/123.txt
K7DXS-Laptop-Arch%
```

Aliases

Global aliases

In bash, aliases can only be placed at the beginning of of a command, but zsh supports aliases anywhere. If you place the following line in your `$ZDOTDIR/.zshrc`

```
alias -g G=' | grep -i'
```

You can then run

```
cat haystack.txt G "needle"
```

Suffix aliases (Added in zsh 4.2.x)

Suffix aliases allow you to tell zsh to open files with specify a program to open files with certain extensions. Examples:

```
alias -s c="emacs"
alias -s php="vim"
alias -s java="$EDITOR"
```

Now in your shell, if you have a php file, `file.php`, and you run the command

```
file.php
```

It will automatically open `file.php` in vim.

Read Main differences from bash online: https://riptutorial.com/zsh/topic/5599/main-differences-from-bash

Chapter 3: oh-my-zsh

Introduction

Oh-my-zsh is one of many configuration frameworks for interactive use of zsh (it is not relevant for writing zsh scripts). It can be useful if you'd like to start with an experienced user's configuration.

Examples

Installation

Oh-my-zsh is a community-driven framework for managing your zsh configuration. It contains many plugins, which extend functionality, themes, which manage the appearance of the shell and the prompt, and helpful functions that will make your terminal much more powerful and customizable. You can create your own plugins and themes if you find there's something missing.

You can only install oh-my-zsh after you have installed zsh. To do so, run the following command:

via curl

```
$ sh -c "$(curl -fsSL https://raw.github.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"
```

via wget

```
$ sh -c "$(wget https://raw.github.com/robbyrussell/oh-my-zsh/master/tools/install.sh -O -)"
```

Once that's done, zsh will be your default shell (it will ask for your password to do this). Feel free to edit your `~/.zshrc` configuration to setup plugins and themes. Remember to run `$ source ~/.zshrc` for changes to take effect, or log out and log back in.

Read oh-my-zsh online: https://riptutorial.com/zsh/topic/5121/oh-my-zsh

Credits

S. No	Chapters	Contributors
1	Getting started with zsh	Adaephon, Adaline Valentina Simonian, adarsh, Community, Doryx, Jules, Kronos, Sumner Evans
2	Main differences from bash	Doctor Strange, Duncan X Simpson, Steven Spasbo, user1934428
3	oh-my-zsh	Alphonsus, Duncan X Simpson, Gilles, Kronos