

 eBook Gratuit

APPRENEZ

zsh

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#zsh

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec zsh.....	2
Remarques.....	2
Versions.....	2
Exemples.....	3
Installation ou configuration.....	3
Obtenir zsh.....	3
Faire de zsh votre shell par défaut.....	4
Configuration.....	5
Alias.....	5
Alias d'annuaire.....	6
Recharger la configuration ZSH.....	6
Chapitre 2: oh-my-zsh.....	7
Introduction.....	7
Exemples.....	7
Installation.....	7
via curl.....	7
via wget.....	7
Chapitre 3: Principales différences de bash.....	8
Exemples.....	8
Tuyaux et sous-coquilles.....	8
Différences dans la cotation.....	8
Manipulation générique.....	8
Alias.....	9
Alias globaux.....	9
Alias de suffixe (Ajouté dans zsh 4.2.x).....	9
Crédits.....	11

À propos

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.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

Chapitre 1: Démarrer avec zsh

Remarques

`zsh` est un shell compatible POSIX, une alternative populaire au shell Bourne (`sh`) et `bash`.

Sa principale caractéristique est de mettre l'accent sur un niveau élevé de personnalisation par l'utilisateur, ce qui a conduit une communauté active de développeurs à créer des extensions pour `zsh`, notamment des lignes d'état d'invite personnalisées et plus informatives, souvent intégrées aux services système.

De nombreuses configurations avec de grands ensembles de valeurs par défaut et d'extensions utiles existent en ligne, y compris le populaire [oh-my-zsh](#) et [prezto](#).

Versions

Version	Date de sortie
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 (bêta)	2012-02-23
4.2.7	2007-12-18
4.3.1 (bêta)	2006-02-28
4.2.0	2004-03-19
4.0.9	2003-12-19
4.1.1 (bêta)	2003-06-19
4.0.1	2001-06-01
3.1.9	2000-06-05

Version	Date de sortie
3.0.8	2000-05-16
3.1.6 (bêta)	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

Exemples

Installation ou configuration

Obtenir zsh

`zsh` est disponible sur de nombreuses plates-formes UNIX via leurs systèmes de gestion de paquets intégrés. Sur les distributions Linux Debian et Ubuntu, `zsh` est disponible dans les référentiels de packages par défaut et peut être installé avec:

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

Sur les distributions basées sur RPM, `zsh` est également souvent disponible dans les archives du package par défaut et peut être installé à l'aide de:

```
$ yum install zsh
```

Sur Fedora 22 et versions ultérieures:

```
$ dnf install zsh
```

Sur les systèmes BSD, `zsh` peut être installé avec `pkg` :

```
$ pkg install zsh
```

Sous OpenBSD, `zsh` peut être installé avec `pkg_add` :

```
$ pkg_add zsh
```

Sur Arch Linux, `zsh` peut être installé avec `pacman` :

```
$ pacman -S zsh
```

Sur openSUSE, `zsh` peut être installé avec `zypper` :

```
$ zypper install zsh
```

Sur les systèmes exécutant macOS (OS X), `zsh` est déjà installé par défaut, mais pas comme shell par défaut. Vous pouvez également installer des versions plus récentes via Homebrew:

```
$ brew install zsh
```

Alternativement, le code source de `zsh` peut être obtenu sur le [site officiel](#) .

De là, le shell peut être démarré en tapant `zsh` à l'invite.

Faire de `zsh` votre shell par défaut

Sur la plupart des systèmes Linux et BSD, `zsh` peut être défini comme shell par défaut pour un utilisateur utilisant la commande `chsh` :

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

Où

- `username` est un nom d'utilisateur réel (par défaut pour l'utilisateur actuel si omis)
- `shell` est le chemin du binaire `zsh` . Le chemin doit être répertorié dans le fichier `/etc/shells` , qui contient une liste des shells autorisés à utiliser avec `chsh` . Si `zsh` n'y figure pas - par exemple parce que vous l'avez compilé et installé depuis le code source - vous devrez d'abord ajouter une ligne avec le chemin absolu vers `zsh` . Vous pouvez obtenir ce chemin avec `which zsh` (à condition qu'il soit installé dans un répertoire répertorié dans `PATH`)

Pour voir les modifications, déconnectez-vous une fois et connectez-vous. Ouvrez l'émulateur de terminal et utilisez

```
`echo $SHELL`
```

S'il affiche `/bin/zsh` vous avez modifié le shell par défaut en `zsh`.

Configuration

Lors du démarrage de `Zsh`, les fichiers suivants sont classés par défaut dans cet ordre:

1. `/etc/zsh/zshenv` Utilisé pour définir des variables d'environnement à l'échelle du système; il ne doit pas contenir de commandes produisant une sortie ou supposant que le shell est attaché à un `tty`. Ce fichier sera toujours généré, cela ne peut pas être remplacé.
2. `~/.zshenv` Utilisé pour définir les variables d'environnement de l'utilisateur; il ne doit pas contenir de commandes produisant une sortie ou supposant que le shell est attaché à un `tty`. Ce fichier sera toujours source.
3. `/etc/zsh/zprofile` Utilisé pour exécuter des commandes au démarrage, il sera généré au démarrage en tant que shell de connexion.

Notez que sur Arch Linux, il contient par défaut une ligne qui source le fichier `/etc/profile`.

`/etc/profile` Ce fichier doit être généré par tous les shells compatibles Bourne lors de la connexion: il configure `$PATH` et d'autres variables d'environnement et paramètres spécifiques à l'application (`/etc/profile.d/*.sh`) lors de la connexion.

4. `~/.zprofile` Utilisé pour exécuter les commandes de l'utilisateur au démarrage, il sera généré au démarrage en tant que shell de connexion.
5. `/etc/zsh/zshrc` Utilisé pour définir la configuration interactive du shell et exécuter des commandes, il sera généré au démarrage en tant que shell interactif.
6. `~/.zshrc` Utilisé pour définir la configuration du shell interactif de l'utilisateur et exécuter des commandes, sera généré au démarrage en tant que shell interactif.
7. `/etc/zsh/zlogin` Utilisé pour exécuter des commandes à la fin de la progression initiale, sera généré au démarrage en tant que shell de connexion.
8. `~/.zlogin` Utilisé pour exécuter les commandes de l'utilisateur à la fin de la progression initiale, sera généré au démarrage en tant que shell de connexion.
9. `~/.zlogout` Sera généré lorsqu'un shell de connexion se `~/.zlogout`.
10. `/etc/zsh/zlogout` Sera généré lorsqu'un shell de connexion se `/etc/zsh/zlogout`.

Si `$ZDOTDIR` n'est pas défini, `$HOME` est utilisé à la place.

Pour un usage personnel général, il est `.zshrc` d'éditer le fichier `.zshrc` l'utilisateur pour des préférences personnelles.

Alias

Pour alias une commande dans votre fichier `~/.zshrc` , vous pouvez utiliser la syntaxe suivante:

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

Par exemple, il est courant d'exécuter la commande `ls -a` . Vous pouvez créer un alias de cette commande comme `la` en tant que tel:

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

Après avoir rechargé le fichier `~/.zshrc` , vous pourrez taper `la` et `ls -a` sera exécuté.

Alias d'annuaire

Il est courant d'avoir certains dossiers dans lesquels vous `cd` souvent. Si tel est le cas, vous pouvez créer alias à ces répertoires pour faire `cd` ING pour les faciliter. Par exemple, ce qui suit alias le dossier Dropbox:

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

vous permettant de saisir la `db` et de changer de répertoire en `~/Dropbox` .

Recharger la configuration ZSH

zsh charge la configuration à partir du fichier `~/.zshrc` au démarrage. Si vous modifiez ce fichier, vous pouvez redémarrer zsh ou exécuter la commande suivante pour recharger la configuration.

```
./ . ~/.zshrc
```

Vous pouvez alias cette commande utile dans votre `~/.zshrc` comme ceci:

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

Lire Démarrer avec zsh en ligne: <https://riptutorial.com/fr/zsh/topic/4570/demarrer-avec-zsh>

Chapitre 2: oh-my-zsh

Introduction

Oh-my-zsh est l'un des [nombreux frameworks de configuration](#) pour l'utilisation interactive de zsh (il n'est pas pertinent pour l'écriture de scripts zsh). Cela peut être utile si vous souhaitez démarrer avec la configuration d'un utilisateur expérimenté.

Exemples

Installation

Oh-my-zsh est un framework piloté par la communauté pour gérer votre configuration zsh. Il contient de nombreux plugins, qui étendent les fonctionnalités, les thèmes, qui gèrent l'apparence du shell et l'invite, ainsi que des fonctions utiles qui rendront votre terminal beaucoup plus puissant et personnalisable. Vous pouvez créer vos propres plugins et thèmes si vous constatez qu'il manque quelque chose.

Vous ne pouvez installer oh-my-zsh qu'après avoir installé zsh. Pour ce faire, exécutez la commande suivante:

via curl

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

via wget

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

Une fois cela fait, zsh sera votre shell par défaut (il vous demandera votre mot de passe pour cela). N'hésitez pas à modifier votre configuration `~/.zshrc` pour configurer des plugins et des thèmes. N'oubliez pas d'exécuter `$ source ~/.zshrc` pour que les modifications prennent effet ou déconnectez-vous et `$ source ~/.zshrc`.

Lire oh-my-zsh en ligne: <https://riptutorial.com/fr/zsh/topic/5121/oh-my-zsh>

Chapitre 3: Principales différences de bash

Exemples

Tuyaux et sous-coquilles

En bash, chaque commande dans un pipeline est exécutée dans un sous-shell. Dans zsh, la dernière commande d'un pipeline est exécutée dans le shell actuel. Par exemple, le code suivant

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

va imprimer *avant* dans bash, mais *après* dans zsh.

Différences dans la cotation

En bash, vous devez citer des arguments afin de préserver l'espace blanc:

```
# bash

function print_first_argument {
    echo "$1"
}

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

Dans Zsh, vous n'avez pas besoin des guillemets, car l'ordre d'évaluation est différent:

```
# zsh

function print_first_argument {
    echo $1
}

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

Manipulation générique

Lorsque rien ne correspond à un caractère générique tel que `*` dans bash, il est transmis comme un littéral `*` à la commande, comme si vous aviez tapé `*`. Cependant, zsh génère une erreur.

Frapper:

```
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%
```

Cela se remarque surtout dans les programmes utilisant un littéral `*`, tels que `find`:

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

./abc:
123.txt
```

Frapper:

```
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%
```

Alias

Alias globaux

Dans `bash`, les alias ne peuvent être placés qu'au début d'une commande, mais `zsh` prend en charge les alias n'importe où. Si vous placez la ligne suivante dans votre `~/.zshrc`

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

Vous pouvez alors courir

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

Alias de suffixe (Ajouté dans zsh 4.2.x)

Les alias de suffixes vous permettent d'indiquer à zsh d'ouvrir des fichiers en spécifiant un programme pour ouvrir des fichiers avec certaines extensions. Exemples:

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

Maintenant, dans votre shell, si vous avez un fichier php, `file.php`, et que vous exécutez la commande

```
file.php
```

Cela ouvrira automatiquement `file.php` dans vim.

Lire Principales différences de bash en ligne: <https://riptutorial.com/fr/zsh/topic/5599/principales-differences-de-bash>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec zsh	Adaephon , Adaline Valentina Simonian , adarsh , Community , Doryx , Jules , Kronos , Sumner Evans
2	oh-my-zsh	Alphonsus , Duncan X Simpson , Gilles , Kronos
3	Principales différences de bash	Doctor Strange , Duncan X Simpson , Steven Spasbo , user1934428