

fishshell

比 bash 更好用的 shell

安装

```
# Ubuntu 和 Debian 的安装方法。
sudo apt-get install fish
# Mac 的安装方法。
brew install fish
```

启动与帮助

由于 Fish 的语法与 Bash 有很大差异，Bash 脚本一般不兼容。因此，建议不要将 Fish 设为默认 Shell，而是每次手动启动它。

```
# 安装完成后，就可以启动 Fish
$ fish
# 使用过程中，如果需要帮助，可以输入 help 命令
$ help
```

彩色显示

```
# 无效命令为红色
$ mkd
# 有效命令为蓝色
$ mkdir
# 有效路径会有下划线。如果没有下划线，你就知道这个路径不存在。
$ cat ~/somefi
```

自动建议

Fish 会自动在光标后面给出建议，表示可能的选项，颜色为灰色。如果采纳建议，可以按下 $\rightarrow$ 或 $\text{Control} + \text{F}$ 。如果只采纳一部分，可以按下 $\text{Alt} + \rightarrow$

```
$ /bin/hostname # 命令建议
$ grep --ignore-case # 参数建议
$ ls node_modules # 路径建议
```

自动补全

输入命令时，Fish 会自动显示匹配的上一条历史记录。如果没有匹配的历史记录，Fish 会猜测可能的结果，自动补全各种输入。比如，输入 `pyt` 再按下 `Tab`，就会自动补全为 `python` 命令。

Fish 还可以自动补全 Git 分支。

脚本语法

if 语句

```
if grep fish /etc/shells
    echo Found fish
else if grep bash /etc/shells
    echo Found bash
else
    echo Got nothing
end
```

switch 语句

```
switch (uname)
case Linux
    echo Hi Tux!
case Darwin
    echo Hi Hexley!
case FreeBSD NetBSD DragonFly
    echo Hi Beastie!
case '*'
    echo Hi, stranger!
end
```

while 循环

```
while true
    echo "Loop forever"
end
```

for 循环

```
for file in *.txt
    cp $file $file.bak
end
```

函数

Fish 的函数用来封装命令，或者为现有的命令起别名。

```
function ll
    ls -lhG $argv
end
```

上面代码定义了一个 ll 函数。命令行执行这个函数以后，就可以用 ll 命令替代 ls -lhG。其中，变量 \$argv 表示函数的参数。

```
function ls
    command ls -hG $argv
end
```

上面的代码重新定义 `ls` 命令。注意，函数体内的 `ls` 之前，要加上 `command`，否则会因为无限循环而报错。

提示符

`fish_prompt` 函数用于定义命令行提示符 `prompt`

```
function fish_prompt
  set_color purple
  date "+%m/%d/%y"
  set_color FF0
  echo (pwd) '>'
  set_color normal
end
```

执行上面的函数以后，你的命令行提示符就会变成下面这样。

```
02/06/13
/home/tutorial >
```

配置

Fish 的配置文件是 `~/.config/fish/config.fish`，每次 Fish 启动，就会自动加载这个文件。Fish 还提供 Web 界面配置该文件。

```
$ fish_config # 浏览器打开 Web 界面配置
```

Running Commands: 兼容 `bash` 等 shell 的命令执行方式

Getting Help: `help/man cmd -> browser/terminal`

Syntax Highlighting: 实时检查命令是否正确

Wildcards: 支持缩写 * 递归 匹配

Pipes and Redirections: 使用 `^` 代表 `stderr`

Autosuggestions: 自动建议, 可以使用 `Ctrl-f / ->` 来补全

Tab Completions: 更强大的 `tab` 补全

Variables: 使用 `set` 设置

Exit Status: 使用 `echo $status` 替代 `$?`

Exports (Shell Variables)

Lists: all variables in fish are really lists

Command Substitutions: 使用 `(cmd)` 来执行命令, 而不是反引号、`$()`

Combiners (And, Or, Not): 不支持使用符合来表示逻辑运算

Functions 使用 `$argv` 替代 `$1`

Conditionals (If, Else, Switch) / Functions / Loops: 更人性化的写法(参考 `py`)

Prompt: `function fish_prompt` 实现

Startup (Where's `.bashrc`?): `~/.config/fish/config.fish`, 更好的方式是 `autoloading-function` `universal-variables`

Autoloading Functions: `~/.config/fish/functions/.`

Universal Variables a variable whose value is shared across all instances of fish

```
set name 'czl' # 设置变量, 替代 name=czl
echo $name
echo $status # exit status 替代 $?
```

```
env # 环境变量
set -x MyVariable SomeValue # 替代 export
set -e MyVariable
set PATH $PATH /usr/local/bin # 使用 lists 记录 PATH
set -U fish_user_paths /usr/local/bin $fish_user_paths # 永久生效
touch "testing_"(date +%s)".txt" # command substitution 替代 `date +%s`
cp file.txt file.txt.bak; and echo 'back success'; or echo 'back fail' #
combiner
functions # 列出 fish 下定义的函数
```

参考资料

- [fish-shell官网](#)

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

<https://rd.irust.top/doku.php?id=command:fishshell>

Last update: **2021/10/15 14:58**

