

# type

显示指定命令的类型。

## 概要

```
type [-afptP] name [name ...]
```

## 主要用途

- 显示要查找的命令的信息。
- 控制查找范围和行为。
- 显示要查找的命令优先级最高的类型。

## 选项

```
-a[]在环境变量PATH中查找并显示所有包含name的可执行文件路径；当'-p'选项没有同时给出时，如果在别名、关键字，函数，内建的信息中存在name[]则一并显示。
-f[]排除对shell函数的查找。
-p[]如果name在执行'type -t name'返回的不是'file'[]那么什么也不返回；否则会在环境变量PATH中查找并返回可执行文件路径。
-P[]即使要查找的name是别名、内建、函数中的一个，仍然会在环境变量PATH中查找并返回可执行文件路径。
-t[]根据name的类型返回一个单词（别名，关键字，函数，内建，文件），否则返回空值。
```

## 参数

name[]要查找的命令，可以为多个。

## 返回值

当指定的命令可以找到时返回成功，如果有没找到的返回失败。

## 例子

接下来要用到的例子假设 '~/.bashrc' 文件定义了以下的内容：

```
alias ls='ls --color=auto'
mybash(){ vim ~/.bashrc; }
```

而且执行环境里没有使用enable禁用内建命令。

```
type -a mybash
# 输出
mybash is a function
mybash ()
{
    vim ~/.bashrc
}
```

```
type -a -f mybash
# 输出 (因为排除了函数, 所以报错)
bash: type: mybash: not found

type -a -p mybash
# 输出为空 (因为排除了函数, 所以什么也不返回)

type -a ls
# 输出
ls is aliased to `ls --color=suto'
ls is /usr/bin/ls
ls is /bin/ls

type -a -p ls
# 输出
/usr/bin/ls
/bin/ls
```

```
# '-f' 不会影响 '-P' 的范围, '-f' 不建议和 '-p' 使用。
# 注意 printf 同时是内建命令以及可执行文件 GNU coreutils 优先作为内建处理。
```

```
type -p printf
# 输出为空

type -P printf
# 输出
/usr/bin/printf
/bin/printf
```

```
# 如果有多个类型, 那么输出优先级最高的类型。
```

```
type -t ls
# 输出
alias

type -t for
# 输出 (bash 关键字)
keyword

type -t mybash
# 输出
function

type -t -f mybash
# 输出空值

type -t printf
# 输出 (bash 内建优先级高)
builtin

type -t chmod
```

```
# 输出  
file
```

## 注意

1. 该命令是bash内建命令，相关的帮助信息请查看help命令。
2. 命令优先级问题请查看builtin命令。

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

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

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

