

tldr pages book

Simplified and community-driven man pages

Generated on Tue Nov 18 08:36:56 2025

Website: <https://tldr.sh>

GitHub: <https://github.com/tldr-pages/tldr>

Android

am

مدیر فعالیت های اندروید

اطلاعات بیشتر: <https://developer.android.com/tools/adb#am>.

- یک فعالیت خاص رو شروع کن:

```
am start -n {{com.android.settings/.Settings}}
```

- یک فعالیت خاص رو شروع کن و داده به آن ارسال کن:

```
am start -a {{android.intent.action.VIEW}} -d {{tel:123}}
```

- فعالیتی که با دسته بندی و عمل خاصی مطابقت داره رو شروع کن:

```
am start -a {{android.intent.action.MAIN}} -c  
{{android.intent.category.HOME}}
```

- تبدیل میکنه URI یک اینترنت رو به یک:

```
am to-uri -a {{android.intent.action.VIEW}} -d {{tel:123}}
```

bugreport

نمایش یک گزارش خطا اندروید.

قابل اجراست `adb shell` این دستور فقط از طریق

اطلاعات بیشتر: <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreport>.

: یک گزارش خطای کامل از دستگاه مورد نظر نمایش میدهد •

`bugreport`

bugreportz

تولید یک گزارش خطای اندروید فشرده شده

قابل اجراست adb shell این دستور فقط از طریق

اطلاعات بیشتر: <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreportz>.

- تولید که گزارش خطای کامل از یک دستگاه اندرویدی :

```
bugreportz
```

- bugreportz نمایش فرایند اجرای دستور :

```
bugreportz -p
```

- نمایش راهنمایی :

```
bugreportz -h
```

- bugreportz نمایش نسخه :

```
bugreportz -v
```

cmd

مدیر سرویس اندروید.

اطلاعات بیشتر: <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/cmd/>.

- فهرست تمام سرویس های در حال اجرا :

```
cmd -l
```

- فراخوان یک سرویس :

```
cmd {{service}}
```

- فراخوان یک سرویس با مقادیر ورودی :

```
cmd {{service}} {{argument1 argument2 ...}}
```

dalvikvm

ماشین مجازی اندروید.

اطلاعات بیشتر: <https://source.android.com/docs/core/runtime>.

• اجرای یک برنامه جاوا:

```
dalvikvm -classpath {{path/to/file.jar}} {{classname}}
```

dumpsys

ارائه اطلاعات درباره سیستم سرویس اندروید.

قابل اجراست adb shell این دستور فقط از طریق

اطلاعات بیشتر: <https://developer.android.com/tools/dumpsys>.

- دریافت اطلاعات عیب بای تمامی سرویس های سیستمی :

```
dumpsys
```

- دریافت اطلاعات عیب یابی یک سرویس سیستمی مشخص :

```
dumpsys {{service}}
```

- میتواند اطلاعات بدهد dumpsys فهرستی از تمامی سرویس هایی که :

```
dumpsys -l
```

- ورودی های یک سرویس مشخص را فهرست میکند :

```
dumpsys {{service}} -h
```

- حذف یک سرویس مشخص از خروجی عیب یابی :

```
dumpsys --skip {{service}}
```

- تعیین فرصت زمانی بر مبنای ثانیه (پیش فرض 10ثانیه) :

```
dumpsys -t {{8}}
```


getprop

نمایش اطلاعات مربوط به مشخصات سیستم اندروید

اطلاعات بیشتر: <https://manned.org/getprop>.

- نمایش اطلاعات مربوط به مشخصات سیستم اندروید :

```
getprop
```

- نمایش اطلاعات مربوط به یک مشخصه :

```
getprop {{property}}
```

- رابط برنامه نویسی SDK نمایش سطح :

```
getprop {{ro.build.version.sdk}}
```

- نمایش نسخه اندروید :

```
getprop {{ro.build.version.release}}
```

- نمایش مدل دستگاه اندروید :

```
getprop {{ro.vendor.product.model}}
```

- OEM نمایش اطلاعات قفل :

```
getprop {{ro.oem_unlock_supported}}
```

- نمایش آدرس مک کارت وای فای اندروید :

```
getprop {{ro.boot.wifimacaddr}}
```

input

ارسال کد رویداد یا ورودی صفحه نمایش به یک دستگاه اندروید.

قابل اجراست `adb shell` این دستور فقط از طریق

اطلاعات بیشتر: <https://developer.android.com/reference/android/view/KeyEvent.html#constants> 1.

- ارسال کد رویداد یک کاراکتر به یک دستگاه اندروید :

```
input keyevent {{event_code}}
```

- (نمایانگر فاصله است %s) ارسال یک متن به یک دستگاه اندروید :

```
input text "{{text}}"
```

- ارسال یک ضربه به یک دستگاه اندروید :

```
input tap {{x_position}} {{y_position}}
```

- ارسال حرکت افقی عمودی به یک دستگاه اندروید :

```
input swipe {{x_start}} {{y_start}} {{x_end}} {{y_end}}  
{{duration_in_ms}}
```

- ارسال یک نگهداشتن بلند مدت به یک دستگاه اندرویدی از طریق حرکت افقی عمودی :

```
input swipe {{x_position}} {{y_position}} {{x_position}}  
{{y_position}} {{duration_in_ms}}
```

logcat

تخلیه یک لاگ از پیاپی سیستمی، شامل پشته رهگیری زمان وقوع خطا، و اطلاعات پیام های لاگ شده توسط برنامه ها.

اطلاعات بیشتر: <https://developer.android.com/tools/logcat>.

- نمایش لاگ سیستمی :

```
logcat
```

- نوشتن لاگ سیستمی به یک فایل :

```
logcat -f {{path/to/file}}
```

- نمایش خطاهای منطبق با الگوی ورودی :

```
logcat --regex {{regular_expression}}
```

- مشخص PID نمایش لاگ های مربوط به یک :

```
logcat --pid {{pid}}
```

- نمایش لاگ های پروسه های مربوط به یک بسته مشخص :

```
logcat --pid $(pidof -s {{package}})
```

pkg

ابزار مدیر بسته ها برای ترماکس

اطلاعات بیشتر: https://wiki.termux.com/wiki/Package_Management.

- ارتقا تمامی بسته های نصب شده :

```
pkg upgrade
```

- نصب یک بسته :

```
pkg install {{package}}
```

- حذف یک بسته :

```
pkg uninstall {{package}}
```

- نصب دوباره یک بسته :

```
pkg reinstall {{package}}
```

- جستجوی یک بسته :

```
pkg search {{package}}
```

pm

نمایش اطلاعات مربوط به برنامه های یک دستگاه اندروید.

اطلاعات بیشتر: <https://developer.android.com/tools/adb#pm>.

- فهرست تمامی برنامه های نصب شده :

```
pm list packages
```

- فهرست تمامی برنامه های سیستمی نصب شده :

```
pm list packages -s
```

- فهرست تمامی برنامه های نصب شده شخص ثالث :

```
pm list packages -3
```

- فهرست برنامه های منطبق با کلید واژه(ها) :

```
pm list packages {{keyword1 keyword2 ...}}
```

- یک برنامه مشخص APK نمایش مسیر فایل :

```
pm path {{app}}
```

screencap

ضبط یک اسکرین شات از صفحه یک موبایل.

قابل اجراست adb shell این دستور فقط از طریق

اطلاعات بیشتر: <https://developer.android.com/tools/adb#screencap>.

• ضبط یک اسکرین شات :

```
screencap {{path/to/file}}
```

settings

دریافت اطلاعات مربوط به سیستم عامل اندروید.

اطلاعات بیشتر: <https://web.archive.org/web/20240525010124/https://adbinstaller.com/commands/adb-shell-settings-5b670d5ee7958178a2955536>.

- global نمایش یک فهرست از تنظیمات داخل فضای نام :

```
settings list {{global}}
```

- دریافت مقدار یک تنظیم مشخص :

```
settings get {{global}} {{airplane_mode_on}}
```

- انتصاب یک مقدار مشخص به یک تنظیم :

```
settings put {{system}} {{screen_brightness}} {{42}}
```

- حذف یک تنظیم مشخص :

```
settings delete {{secure}} {{screensaver_enabled}}
```

wm

نمایش اطلاعات مربوط به صفحه نمایش یک دستگاه اندروید.

قابل اجراست `adb shell` این دستور فقط از طریق

اطلاعات بیشتر: <https://web.archive.org/web/20240420064706/https://adbinstaller.com/commands/adb-shell-wm-5b672b17e7958178a2955538>.

• نمایش اندازه فیزیکی صفحه نمایش یک دستگاه اندروید :

```
wm size
```

• نمایش فشردگی پیکسل فیزیکی صفحه نمایش یک دستگاه اندروید :

```
wm density
```


Common

2to3

تبدیل خودکار کد پایتون نسخه 2 به 3

اطلاعات بیشتر: <https://manned.org/2to3>.

- نمایش تغییرات که در حال حاضر انجام میشود بدون اجرای آن (dry-run):

```
2to3 {{path/to/file.py}}
```

- تبدیل فایل پایتون نسخه 2 به 3:

```
2to3 --write {{path/to/file.py}}
```

- تبدیل قابلیت های خاص پایتون نسخه 2 به 3:

```
2to3 --write {{path/to/file.py}} --fix {{raw_input}} --fix {{print}}
```

- تبدیل تمامی قابلیت های نسخه 2 به 3 بغیر از ویژگی های معین شده:

```
2to3 --write {{path/to/file.py}} --nofix {{has_key}} --nofix {{isinstance}}
```

- نمایش لیست قابلیت های زبان پایتون نسخه 2 که قابلیت تبدیل به نسخه 3 را دارند:

```
2to3 --list-fixes
```

- تبدیل تمامی فایل های پایتون نسخه 2 به 3 در یک مسیر:

```
2to3 --output-dir {{path/to/python3_directory}} --write-unchanged-files --nobackups {{path/to/python2_directory}}
```

- اجرای همزمان (چند رشته ای) دستور 2 به 3:

```
2to3 --processes {{4}} --output-dir {{path/to/python3_directory}} --write --nobackups --no-diff {{path/to/python2_directory}}
```

7z

آرشیو کننده فایل با قدرت فشرده سازی بالا

اطلاعات بیشتر: <https://manned.org/7z>.

- اضافه کردن یک فایل یا پوشه به آرشیو موجود:

```
7z a {{path/to/archive.7z}} {{path/to/file_or_directory}}
```

- رمزنگاری آرشیو موجود (همراه با اسم فایلها):

```
7z a {{path/to/encrypted.7z}} -p{{password}} -mhe=on {{path/to/archive.7z}}
```

- استخراج یک آرشیو درحالی که ساختار پوشه اصلی حفظ می شود:

```
7z x {{path/to/archive.7z}}
```

- استخراج یک آرشیو در پوشه مورد نظر:

```
7z x {{path/to/archive.7z}} -o{{path/to/output}}
```

- استخراج stdout یک آرشیو در:

```
7z x {{path/to/archive.7z}} -so
```

- آرشیو کردن با نوع آرشیو دلخواه:

```
7z a -t{{7z|bzip2|gzip|lzip|tar|zip}} {{path/to/archive}}  
{{path/to/file_or_directory}}
```

- فهرست کردن محتواهای یک آرشیو:

```
7z l {{path/to/archive.7z}}
```

- تنظیم میزان فشرده سازی (مقادیر بیشتر به معنی فشرده سازی بیشتر اما آهسته تر است):

```
7z a {{path/to/archive.7z}} -mx={{0|1|3|5|7|9}} {{path/to/  
file_or_directory}}
```

7za

آرشیو کننده فایل با قدرت فشرده سازی بالا

با این تفاوت که دستور مزبور از نوع فایل کمتری پشتیبانی می 7z عملکرد مشابه کند اما چند-سکویی است.

اطلاعات بیشتر: <https://manned.org/7za>.

- آرشیو یک فایل یا پوشه:

```
7za a {{path/to/archive.7z}} {{path/to/file_or_directory}}
```

- رمزنگاری یک آرشیو موجود (شامل نام فایلها):

```
7za a {{path/to/encrypted.7z}} -p{{password}} -mhe={{on}}  
{{path/to/archive.7z}}
```

- استخراج یک آرشیو درحالی که ساختار پوشه اصلی حفظ می شود:

```
7za x {{path/to/archive.7z}}
```

- استخراج یک آرشیو در پوشه موردنظر:

```
7za x {{path/to/archive.7z}} -o{{path/to/output}}
```

- stdout استخراج یک آرشیو در:

```
7za x {{path/to/archive.7z}} -so
```

- آرشیو کردن با نوع آرشیو دلخواه:

```
7za a -t{{7z|bzip2|gzip|lzip|tar|...}} {{path/to/archive.7z}}  
{{path/to/file_or_directory}}
```

- فهرست کردن محتواهای یک آرشیو:

```
7za l {{path/to/archive.7z}}
```

- تنظیم میزان فشرده سازی (مقادیر بیشتر به معنی فشرده سازی بیشتر اما آهسته تر است):

```
7za a {{path/to/archive.7z}} -mx={{0|1|3|5|7|9}} {{path/to/  
file_or_directory}}
```

7zr

آرشیو کننده فایل با قدرت فشرده سازی بالا

پشتیبانی نمی کند 7z. با این تفاوت که از فایل‌های 7z عملکردی مشابه

اطلاعات بیشتر: <https://manned.org/7zr>.

- آرشیو کردن یک فایل یا پوشه:

```
7zr a {{path/to/archive.7z}} {{path/to/file_or_directory}}
```

- رمزنگاری آرشیو موجود (همراه با اسم فایلها):

```
7zr a {{path/to/encrypted.7z}} -p{{password}} -mhe={{on}}  
{{path/to/archive.7z}}
```

- استخراج یک آرشیو درحالی که ساختار پوشه اصلی حفظ می شود:

```
7zr x {{path/to/archive.7z}}
```

- استخراج یک آرشیو در پوشه مورد نظر:

```
7zr x {{path/to/archive.7z}} -o{{path/to/output}}
```

- stdout استخراج یک آرشیو در:

```
7zr x {{path/to/archive.7z}} -so
```

- فهرست کردن محتواهای یک آرشیو:

```
7zr l {{path/to/archive.7z}}
```

- تنظیم میزان فشرده سازی (مقادیر بیشتر به معنی فشرده سازی بیشتر اما آهسته تر است):

```
7zr a {{path/to/archive.7z}} -mx={{0|1|3|5|7|9}} {{path/to/  
file_or_directory}}
```

[

بررسی نوع فایل و مقایسه مقدار ها

عدد 0 برمیگرداند اگر شرط درست باشد و 1 اگر شرط نادرست باشد

اطلاعات بیشتر: <https://gnu.org/software/bash/manual/bash.html#index-test>.

- بررسی میکند که آیا یک متغیر با رشته معین برابر است یا نابرابر:

```
[ "${variable}" {=!} "${string}" ]
```

- اینکه تغییری برابر/ نابرابر/ بزرگتر از/ کوچکتر از/ برابر یا بزرگتر/ کوچکتر یا برابر با : عددی است را بررسی میکند :

```
[ "${variable}" -{eq|ne|gt|lt|ge|le} integer ]
```

- اینکه متغیر با مقدار معین دارد یا مقدار نا معین دارد را بررسی میکند :

```
[ -n "${variable}" ]
```

- اینکه تغییری مقدار تهی دارد را بررسی میکند :

```
[ -z "${variable}" ]
```

- اینکه فایل معینی وجود دارد یا خیر را بررسی میکند :

```
[ -f {path/to/file} ]
```

- اینکه دایرکتوری معینی وجود دارد یا خیر را بررسی میکند :

```
[ -d {path/to/directory} ]
```

- اینکه فایل یا دایرکتوری معینی وجود دارد یا خیر را بررسی میکند :

```
[ -e {path/to/file_or_directory} ]
```

[[

نوع فایل و مقادیر را بررسی میکند.

عدد 0 برمیگرداند اگر حاصل عبارت شرط درست باشد و 1 اگر نادرست باشد.

اطلاعات بیشتر: https://gnu.org/software/bash/manual/bash.html#index-005b_005b.

- اینکه متغیری برابر/نا برابر با رشته ای معین است را بررسی میکند :

```
[[ ${variable} {==|!=} "{string}" ]]
```

- اینکه متغیری از الگوی معین شده پیروی میکند را بررسی میکند :

```
[[ ${variable} {==|=~} {pattern} ]]
```

- اینکه متغیری برابر/نا برابر/بزرگتر از/کوچکتر از/ برابر یا بزرگتر/کوچکتر یا برابر با عددی است را بررسی میکند :

```
[[ ${variable} -{eq|ne|gt|lt|ge|le} {integer} ]]
```

- اینکه متغیری مقدار غیرتهی دارد را بررسی میکند :

```
[[ -n ${variable} ]]
```

- اینکه متغیری مقدار تهی دارد را بررسی میکند :

```
[[ -z ${variable} ]]
```

- وجود فایلی را بررسی میکند :

```
[[ -f {path/to/file} ]]
```

- وجود پوشه ای را بررسی میکند :

```
[[ -d {path/to/directory} ]]
```

- وجود پوشه یا فایلی را بررسی میکند :

```
[[ -e {path/to/file_or_directory} ]]
```

ac

آماري درمورد اينكه کاربران چه مدت متصل بوده اند چاپ مي کند

اطلاعات بیشتر: <https://manned.org/ac.8>.

- مدت زمان اتصال کاربر فعلي را به ساعت چاپ مي کند:

```
ac
```

- مدت زمان اتصال کاربران را به ساعت چاپ مي کند:

```
ac -p
```

- مدت زمان اتصال يك کاربر خاص را به ساعت چاپ مي کند:

```
ac -p {{username}}
```

- مدت زمان اتصال يك کاربر به ساعت در روز را چاپ مي کند (همراه با مجموع):

```
ac -dp {{username}}
```


agate

Gemini. یک سرور ساده برای پروتوکل شبکه

اطلاعات بیشتر: <https://github.com/mbrubeck/agate>.

- اجرا و ساخت یک کلید خصوصی و مجوز:

```
agate --content {{path/to/content}}/ --addr {{[:]:1965}} --  
addr {{0.0.0.0:1965}} --hostname {{example.com}} --lang {{en-  
US}}
```

- اجرا کردن سرور:

```
agate {{path/to/file}}
```

- نمایش راهنما:

```
agate {{[-h|--help]}}
```

airmon-ng

فعالسازی مانیتور مود بر روی دستگاه های شبکه بیسیم

aircrack-ng بخشی از

اطلاعات بیشتر: <https://www.aircrack-ng.org/doku.php?id=airmon-ng>.

- فهرست کردن دستگاه های بی سیم و وضعیت آنها:

```
sudo airmon-ng
```

- فعالسازی مانیتور مود برای یک دستگاه خاص:

```
sudo airmon-ng start {{wlan0}}
```

- متوقف کردن پروسه های مزاحم که از دستگاه های بی سیم استفاده می کنند:

```
sudo airmon-ng check kill
```

- خاموش کردن مانیتور مود برای یک اینترفیس شبکه خاص:

```
sudo airmon-ng stop {{wlan0mon}}
```

airpaste

mDNS. اشتراک گذاری پیام و فایل از طریق شبکه مشترک با استفاده از

اطلاعات بیشتر: <https://github.com/mafintosh/airpaste>.

- منتظر پیام می ماند و هنگام دریافت آن را نشان می دهد:

```
airpaste
```

- ارسال متن:

```
echo {{text}} | airpaste
```

- ارسال فایل:

```
airpaste < {{path/to/file}}
```

- دریافت فایل:

```
airpaste > {{path/to/file}}
```

- ساخت یا ورود به یک کانال:

```
airpaste {{channel_name}}
```

alex

نوشته های نادرست و مشکل دار پیدا کنید.

به شما کمک می کند نوشته های حاوی تعصب جنسیتی، دوقطبی کننده جامعه، نژاد پرستانه، توهین به مذاهب و سایر جملات مشابه را پیدا کنید.

اطلاعات بیشتر: <https://github.com/get-alex/alex>.

- بررسی کردن متن ورودی از:

```
echo {{His network looks good}} | alex --stdin
```

- بررسی تمام فایل های موجود در این دایرکتوری:

```
alex
```

- بررسی یک فایل خاص:

```
alex {{path/to/file.md}}
```

- بررسی تمام فایل های نشانه گذاری به جز example.md:

```
alex *.md !{{example.md}}
```

alias

ایجاد نام مستعار -- کلمه ای که جایگزین یک دستور می باشد

نام های مستعار طول عمری برابر با جلسه جاری شل مربوطه دارند مگر اینکه در `~/.bashrc` فایل های پیکربندی شل مربوط نظیر

اطلاعات بیشتر: <https://www.gnu.org/software/bash/manual/bash.html#index-alias>.

- نمایش تمامی نام های مستعار:

```
alias
```

- ایجاد یک نام مستعار:

```
alias {{word}}="{{command}}"
```

- نمایش نام مستعار مرتبط با کلمه مشخص شده:

```
alias {{word}}
```

- حذف یک نام مستعار:

```
unalias {{word}}
```

- به نسخه تعاملی با تعریف نام مستعار `rm` تغییر:

```
alias {{rm}}="{{rm -i}}"
```

- `ls -a` به عنوان میانبری برای `la` تعریف:

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

ani-cli

یک ابزار خط فرمان برای کاوش و تماشای انیمه

اطلاعات بیشتر: <https://manned.org/ani-cli>.

- جستجوی انیمه با نام:

```
ani-cli "{{anime_name}}"
```

- دانلود قسمت:

```
ani-cli -d "{{anime_name}}"
```

- به عنوان پخش کننده ویدیو VLC استفاده از:

```
ani-cli -v "{{anime_name}}"
```

- مشخص کردن قسمتی که میخواهید پخش شود:

```
ani-cli -e {{episode_number}} "{{anime_name}}"
```

- ادامه دادن تماشای انیمه از تاریخچه موجود:

```
ani-cli -c
```

- ani-cli بروزرسانی:

```
ani-cli -U
```

apg

ساخت پسوردهای دلخواه، پیچیده و تصادفی.

اطلاعات بیشتر: <https://manned.org/apg>.

- ساخت پسوردهای تصادفی (با طول پیشفرض 8 کاراکتر):

```
apg
```

- و یک حرف (C) یک حرف بزرگ، (N) یک عدد، (S) ساخت یک پسورد با حداقل یک علامت و یک (L) کوچک:

```
apg -M SNCL
```

- ساخت یک پسورد با 16 کاراکتر:

```
apg -m {{16}}
```

- ساخت یک پسورد با حداکثر طول 16 کاراکتر:

```
apg -x {{16}}
```

- ساخت پسوردهای که در یک دیکشنری دیده نمی شود (فایل دیکشنری باید ارائه شود):

```
apg -r {{path/to/dictionary_file}}
```

apktool

APK. مهندسی معکوس فایل های

اطلاعات بیشتر: <https://ibotpeaches.github.io/Apktool/>.

- APK رمزگشایی یک فایل:

```
apktool d {{path/to/file.apk}}
```

- از یک دایرکتوری APK ساخت یک فایل:

```
apktool b {{path/to/directory}}
```

- نصب و ذخیره یک فریمورک:

```
apktool if {{path/to/framework.apk}}
```


aria2c

ابزاری برای دانلود سریع.

قابلیت پشتیبانی از Http(s), FTP, SFTP, BitTorrent, Metalink.

اطلاعات بیشتر: <https://aria2.github.io>.

- دانلود لینک موردنظر و ذخیره در فایل:

```
aria2c "{{url}}"
```

- دانلود یک فایل از لینک موردنظر با اسم خروجی دلخواه:

```
aria2c --out {{path/to/file}} "{{url}}"
```

- دانلود چند فایل مختلف به صورت همزمان:

```
aria2c --force-sequential {{false}} "{{url1 url2 ...}}"
```

- دانلود از چند لینک مختلف:

```
aria2c "{{url1 url2 ...}}"
```

- دانلود لینک های لیست شده در یک فایل همراه با تنظیم تعداد دانلود های همزمان:

```
aria2c --input-file={{path/to/file}} --max-concurrent-downloads={{number_of_downloads}}
```

- دانلود با چندین اتصال مختلف:

```
aria2c --split {{number_of_connections}} "{{url}}"
```

- با نام کاربری و رمز عبور FTP دانلود از:

```
aria2c --ftp-user={{username}} --ftp-passwd={{password}}  
"{{url}}"
```

- محدود کردن سرعت دانلود با واحد بایت بر ثانیه:

```
aria2c --max-download-limit={{speed}} "{{url}}"
```

atq

batch یا at نمایش کارهای زمانبندی شده توسط دستورهای

اطلاعات بیشتر: <https://manned.org/atq>.

- نمایش کارهای زمانبندی شده کاربر کنونی:

```
atq
```

- (صف ها اسم های یک کاراکتری دارند) 'a' نمایش کارها از صفی به اسم:

```
atq -q {{a}}
```

- نمایش کارهای تمام کاربرها (با دسترسی سوپریوزر اجرا کنید):

```
sudo atq
```

atuin

ذخیره سازی تاریخچه شل شما در یک دیتابیس قابل جستجو

به صورت اختیاری می توانید تاریخچه رمزنگاری شده را بین دستگاه هایتان هماهنگ کنید.

اطلاعات بیشتر: <https://docs.atuin.sh/>.

- نصب آتوین بر روی شل شما:

```
eval "$(atuin init {{bash|zsh|fish}})"
```

- وارد کردن تاریخچه از فایل مربوط به تاریخچه پیشفرض شل:

```
atuin import auto
```

- جستجوی تاریخچه شل برای یک دستور خاص:

```
atuin search {{command}}
```

- ثبت نام یک حساب بر روی سرور پیشفرض با استفاده از نام کاربری و ایمیل و رمز عبور:
دلخواه:

```
atuin register -u {{username}} -e {{email}} -p {{password}}
```

- ورود به سرور پیشفرض:

```
atuin login -u {{username}} -p {{password}}
```

- هماهنگ کردن تاریخچه با سرور:

```
atuin sync
```

bob

Neovim. مدیریت و جابجایی بین نسخه های مختلف

اطلاعات بیشتر: <https://github.com/MordechaiHadad/bob>.

- Neovim نصب و جابجایی به نسخه خاصی از:

```
bob use {{nightly|stable|latest|version_string|commit_hash}}
```

- فهرست نسخه های نصب شده و نسخه ای که اکنون استفاده می شود:

```
bob list
```

- Neovim حذف یک نسخه خاص از:

```
bob uninstall {{nightly|stable|latest|version_string|commit_hash}}
```

- اعمال کرده است bob و پاک کردن هر تغییری که Neovim حذف:

```
bob erase
```

- بازگشت به نسخه شبانه قبلی:

```
bob rollback
```

boxes

کشیدن، حذف و تعمیر جعبه های هنری اسکی

اطلاعات بیشتر: <https://boxes.thomasjensen.com/boxes-man-1.html>.

- کشیدن یک جعبه در اطراف یک رشته:

```
echo "{{string}}" | boxes
```

- حذف جعبه از یک رشته:

```
echo "{{string}}" | boxes -r
```

- کشیدن یک جعبه با طرح دلخواه در اطراف یک رشته:

```
echo "{{string}}" | boxes -d {{parchment}}
```

- کشیدن یک جعبه به طول 10 و ارتفاع 5:

```
echo "{{string}}" | boxes -s {{10}}x{{5}}
```

- کشیدن یک جعبه با متن در وسط:

```
echo "{{string}}" | boxes -a c
```

bpkg

یک پکیج منیجر برای بش اسکریپت

اطلاعات بیشتر: <https://github.com/bpkg/bpkg>.

- بروزرسانی فهرست محلی:

```
bpkg update
```

- نصب یک بسته به صورت گلوبال:

```
bpkg install --global {{package}}
```

- نصب یک بسته در یک زیرپوشه در پوشه ی کنونی:

```
bpkg install {{package}}
```

- نصب یک نسخه خاص از یک بسته به صورت گلوبال:

```
bpkg install {{package}}@{{version}} -g
```

- نمایش جزئیات یک بسته خاص:

```
bpkg show {{package}}
```

- اجرای یک دستور، آرگومان ها به صورت اختیاری نوشته شده اند:

```
bpkg run {{command}} {{argument1 argument2 ...}}
```

bun

برای جاوا اسکریپت toolkit و runtime یک

package manager یک ابزار تست و یک bundler، شامل یک

اطلاعات بیشتر: <https://bun.com/docs>.

- package.json اجرای یک فایل جاوااسکریپت یا یک اسکریپت در:

```
bun run {{path/to/file|script_name}}
```

- اجرای تست‌های واحد:

```
bun test
```

- package.json دانلود و نصب تمام بسته‌های ذکر شده در:

```
bun install
```

- package.json افزودن یک وابستگی به:

```
bun add {{module_name}}
```

- package.json حذف یک وابستگی از:

```
bun remove {{module_name}}
```

- در دایرکتوری فعلی Bun ایجاد یک پروژه جدید:

```
bun init
```

- (پوسته تعاملی) REPL شروع یک:

```
bun repl
```

- به آخرین نسخه موجود Bun ارتقا:

```
bun upgrade
```

calendar

نمایش رویداد های پیش رو با استفاده از یک فایل تقویم

اطلاعات بیشتر: <https://manned.org/calendar>.

- نمایش رویداد های امروز و فردا (یا آخر هفته یعنی روز جمعه) با استفاده از تقویم پیشفرض:

```
calendar
```

- نگاهی به آینده، نمایش رویداد های 30 روز آینده:

```
calendar -A {{30}}
```

- نگاهی به گذشته، نمایش رویداد های هفت روز گذشته:

```
calendar -B {{7}}
```

- نمایش رویداد ها با استفاده از یک فایل تقویم دلخواه:

```
calendar -f {{path/to/file}}
```


cat

چاپ و ترکیب کردن فایل ها

اطلاعات بیشتر: <https://manned.org/cat.1posix>.

- چاپ محتویات فایل بر روی صفحه نمایش:

```
cat {{path/to/file}}
```

- ادغام چند فایل با هم و ایجاد فایل جدید:

```
cat {{path/to/file1 path/to/file2 ...}} > {{target_file}}
```

- ادغام چند فایل با هم و اضافه کردن آن به فایل مقصد:

```
cat {{path/to/file1 path/to/file2 ...}} >> {{target_file}}
```

catimg

چاپ عکس در ترمینال

موارد مشابه: **pixterm**, **chafa**.

اطلاعات بیشتر: <https://manned.org/catimg>.

- در ترمینال GIF یا JPEG, PNG چاپ یک:

```
catimg {{path/to/file}}
```

- دوبرابر کردن وضوح یک تصویر:

```
catimg -r 2 {{path/to/file}}
```

- غیرفعال سازی رنگ 24 بیتی برای پشتیبانی بهتر ترمینال:

```
catimg -t {{path/to/file}}
```

- مشخص کردن طول و عرض دلخواه:

```
catimg {{-w|-H}} {{40}} {{path/to/file}}
```

cb

بریدن، رونوشت و چسباندن هر چیزی در ترمینال

اطلاعات بیشتر: <https://github.com/Slackadays/Clipboard>.

- نمایش تمام کلیپ بوردها:

```
cb
```

- رونوشت یک فایل به کلیپ بورد:

```
cb copy {{path/to/file}}
```

- کپی متن دلخواه به کلیپ بورد:

```
cb copy "{{Some example text}}"
```

- رونوشت داده‌ی هدایت شده به داخل در کلیپ بورد:

```
echo "{{Some example text}}" | cb
```

- چسباندن محتوای کلیپ بورد:

```
cb paste
```

- هدایت محتوای کلیپ بورد به خارج:

```
cb | cat
```

- نمایش تاریخچه کلیپ بورد:

```
cb history
```

- نمایش اطلاعات کلیپ بورد:

```
cb info
```

cd

تغییر دایرکتوری جاری

اطلاعات بیشتر: <https://www.gnu.org/software/bash/manual/bash.html#index-cd>.

- تغییر مسیر به دایرکتوری مشخص شده:

```
cd {{path/to/directory}}
```

- کاربر فعلی Home تغییر مسیر به دایرکتوری:

```
cd
```

- تغییر مسیر به دایرکتوری بالاتر از مسیر جاری:

```
cd ..
```

- تغییر مسیر به دایرکتوری قبلی:

```
cd -
```

chmod

تغییر مجوز(ها)ی دسترسی به یک فایل یا پوشه.

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/chmod-invocation.html.

- به مالک فایل دسترسی اجرا میدهد:

```
chmod u+x {{path/to/file}}
```

- به کاربر مالک دسترسی خواندن|نوشتن یک فایل|پوشه را میدهد:

```
chmod u+rw {{path/to/file_or_directory}}
```

- دسترسی اجرا را از گروه سلب میکند:

```
chmod g-x {{path/to/file}}
```

- به تمامی کاربرها دسترسی خواندن و اجرا میدهد:

```
chmod a+rx {{path/to/file}}
```

- به دیگران(کاربرانی که صاحب فایل نیستند) دسترسی های گروه را میدهد:

```
chmod o=g {{path/to/file}}
```

- به همگان همه دسترسی(ها) را میدهد:

```
chmod o= {{path/to/file}}
```

- به صورت بازگشتی به گروه و دیگران دسترسی نوشتن میدهد:

```
chmod {{[-R|--recursive]]} g+w,o+w {{path/to/directory}}
```

- به صورت بازگشتی در پوشه و زیرپوشه(ها) دسترسی اجرا و خواندن فایل(ها) را میدهد:

```
chmod {{[-R|--recursive]]} a+rX {{path/to/directory}}
```

chown

مالک فایل یا پوشه را تغییر میدهد.

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/chown-invocation.html.

- تغییر مالک به فایل یا پوشه:

```
chown {{user}} {{path/to/file_or_directory}}
```

- تغییر کاربر و گروه مالک فایل:

```
chown {{user}}:{{group}} {{path/to/file_or_directory}}
```

- تغییر بازگشتی مالک به پوشه و محتویات آن:

```
chown {[[-R|--recursive]]} {{user}} {{path/to/directory}}
```

- تغییر مالک یک فایل میانبر(به فایل دیگری اشاره میکند):

```
chown {[[-h|--no-dereference]]} {{user}} {{path/to/symlink}}
```

- تغییر مالک یک فایل/پوشه برای همسان شدن با فایل مرجع:

```
chown --reference {{path/to/reference_file}} {{path/to/file_or_directory}}
```

clear

پاک کردن صفحه ترمینال

اطلاعات بیشتر: <https://manned.org/clear>.

- پاک کردن صفحه (معادل فشردن کلیدهای کنترل+ال در پوسته بش):

```
clear
```

- پاک کردن صفحه ترمینال با نگاه داشتن حافظه:

```
clear -x
```

- (Term) مشخص نمودن نوع ترمینال برای پاکسازی (مقدار پیشفرض در متغیر عمومی):

```
clear -T {{type_of_terminal}}
```

- clear استفاده شده در ncurses نمایش نسخه:

```
clear -V
```

cmp

مقایسه بایت به بایت دو فایل.

اطلاعات بیشتر: <https://www.gnu.org/software/diffutils/manual/diffutils.html#Invoking-cmp>.

- نمایش کارکتر و خطی که اولین تفاوت دو فایل در آن یافت شد:

```
cmp {{path/to/file1}} {{path/to/file2}}
```

- نمایش اطلاعات اولین تفاوت پیدا شده: کاراکتر، شماره خط، بایت ها، و مقادیر آنها:

```
cmp {[ -b|--print-bytes]} {{path/to/file1}} {{path/to/file2}}
```

- نمایش شماره بایتها و مقادیر تمامی تفاوت ها:

```
cmp {[ -l|--verbose]} {{path/to/file1}} {{path/to/file2}}
```

- مقایسه فایلها در حالت خاموش، تنها مقدار خروجی برنامه در ترمینال در دسترس است:

```
cmp {[ -s|--quiet]} {{path/to/file1}} {{path/to/file2}}
```


composer

PHP ابزاری بسته محور برای مدیریت وابستگی های پروژه های

اطلاعات بیشتر: <https://getcomposer.org/doc/03-cli.md>.

- به صورت کنشگرا `composer.json` ساخت یک فایل:

```
composer init
```

- اضافه کردن یک بسته به عنوان وابستگی به این پروژه، همچنین یک ورودی به `composer.json` می کند:

```
composer require {{user/package}}
```

- `composer.lock` هستند و `composer.json` نصب تمام وابستگی های این پروژه که در را ایجاد می کند:

```
composer install
```

- و `composer.json` حذف یک بسته از این پروژه، وابستگی مربوط به آنرا از `composer.lock` می کند حذف:

```
composer remove {{user/package}}
```

- هستند و یادداشت `composer.json` بروزرسانی تمام وابستگی های این پروژه که در `composer.lock` کردن نسخه های جدید در فایل:

```
composer update
```

- را به `composer.json` بروزرسانی می کند بعد از این که `composer.lock` فقط صورت دستی بروزرسانی کردید:

```
composer update --lock
```

- اطلاعات بیشتری درباره دلیل نصب نشدن یک وابستگی ارائه می دهد:

```
composer why-not {{user/package}}
```

- بروزرسانی کامپوزر به آخرین نسخه اش:

```
composer self-update
```

copyq

مدیریت کلیپ بورد با قابلیت های پیشرفته.

اطلاعات بیشتر: <https://copyq.readthedocs.io/en/latest/command-line.html>.

- اجرای کپی کیو برای ذخیره تاریخچه کلیپ بورد:

```
copyq
```

- نمایش محتوای کنونی کلیپ بورد:

```
copyq clipboard
```

- وارد کردن متن خام به تاریخچه کلیپ بورد:

```
copyq add -- {{text1}} {{text2}} {{text3}}
```

- در تاریخچه کلیپ بورد (\n, \t) وارد کردن متن شامل رشته های فاصله انداز مانند:

```
copyq add {{firstline\nsecondline}}
```

- چاپ محتوای سه مورد اول در تاریخچه کلیپ بورد:

```
copyq read 0 1 2
```

- رونوشت محتوای یک فایل در کلیپ بورد:

```
copyq copy < {{path/to/file.txt}}
```

- در کلیپ بورد JPEG رونوشت یک عکس با فرمت:

```
copyq copy image/jpeg < {{path/to/image.jpg}}
```

cp

کپی فایل ها و دایرکتوری ها

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/cp-invocation.html.

- کپی فایل از مبدا به مقصد مشخص شده:

```
cp {{path/to/source_file.ext}} {{path/to/target_file.ext}}
```

- کپی فایل به دایرکتوری مشخص شده با حفظ نام فایل:

```
cp {{path/to/source_file.ext}} {{path/to/  
target_parent_directory}}
```

- کپی یک دایرکتوری به صورت کامل به مقصد جدید(اگر در مقصد دایرکتوری وجود داشت (دایرکتوری مبدا در داخل دایرکتوری مقصد کپی می شود):

```
cp -R {{path/to/source_directory}} {{path/to/  
target_directory}}
```

- کپی یک دایرکتوری به صورت کامل با نمایش جزئیات (نمایش فایل های کپی شده):

```
cp -vR {{path/to/source_directory}} {{path/to/  
target_directory}}
```

- به دایرکتوری مقصد در حالت تعاملی (قبل از بازنویسی txt کپی کلیه فایل های با پسوند (تاییده توسط کاربر نیاز است):

```
cp -i {{*.txt}} {{path/to/target_directory}}
```

- کپی لینک به مقصد بدون ارجاع به فایل اصلی:

```
cp -L {{link}} {{path/to/target_directory}}
```

cppclean

پیدا کردن کد های بدون استفاده در پروژه های سی پلاس پلاس.

اطلاعات بیشتر: <https://github.com/myint/cppclean>.

- اجرا در پوشه ی پروژه:

```
cppclean {{path/to/project}}
```

- قرار دارند inc1/ و inc2/ اجرا روی پروژه درحالی که هدرها در پوشه های:

```
cppclean {{path/to/project}} --include-path {{inc1}} --  
include-path {{inc2}}
```

- main.cpp اجرا روی فایل دلخواه مانند:

```
cppclean {{main.cpp}}
```

- "build" اجرا روی پوشه کنونی به استثنای پوشه:

```
cppclean {{.}} --exclude {{build}}
```

cradle

cradle. فریمورک پی اچ پی

مستندات خاص خود را دارند `install` بعضی از زیردستور ها مانند

اطلاعات بیشتر: <https://cradlephp.github.io>.

- اتصال به یک سرور:

```
cradle connect {{server_name}}
```

- Cradle اجرای یک دستور:

```
cradle {{command}}
```

- نمایش راهنما:

```
cradle help
```

- نمایش راهنما برای یک دستور خاص:

```
cradle {{command}} help
```

createdb

ساخت یک دیتابیس پستگرس.

اطلاعات بیشتر: <https://www.postgresql.org/docs/current/app-createdb.html>.

- ساخت یک دیتابیس که صاحب آن کاربر فعلی باشد:

```
createdb {{database_name}}
```

- ساخت یک دیتابیس که صاحب آن یک کاربر خاص باشد و توضیحاتی درمورد آن نیز ارائه شده باشد:

```
createdb --owner {{username}} {{database_name}}  
'{{description}}'
```

- ساخت یک دیتابیس از روی یک قالب:

```
createdb --template {{template_name}} {{database_name}}
```

csvpy

در شل پایتون CSV اجرای یک فایل

گنجانده شده است csvkit در

اطلاعات بیشتر: <https://csvkit.readthedocs.io/en/latest/scripts/csvpy.html>.

- CSVKitReader در یک آجست از نوع CSV بارگذاری یک فایل:

```
csvpy {{data.csv}}
```

- CSVKitDictReader در یک آجکت CSV اجرای یک فایل:

```
csvpy --dict {{data.csv}}
```

curl

انتقال داده از/به سرور.

پشتیبانی می‌کند POP3 و FTP، HTTP از اکثر پروتکل‌ها از جمله

اطلاعات بیشتر: <https://curl.se/docs/manpage.html>.

- و ذخیره آن در یک فایل (با نام دلخواه) URL دانلود محتوای یک:

```
curl {{http://example.com}} {[ -o|--output ]} {{path/to/
file}}
```

- URL دانلود یک فایل و ذخیره آن با نام فایل مشخص شده توسط:

```
curl {[ -O|--remote-name ]} {{http://example.com/filename}}
```

- و ادامه خودکار (از location redirects) دانلود یک فایل، با دنبال کردن تغییرمسیرهای لینک (سرگیری) انتقال فایل قبلی. در صورت بروز دادن خطای سرور، خطا نمایش داده خواهد شد:

```
curl {[ -f|--fail ]} {[ -O|--remote-name ]} {[ -L|--
location ]} {[ -C|--continue-at ]} - {{http://example.com/
filename}}
```

- استفاده از نوع POST ارسال داده (فرم) رمزگذاری شده (درخواست application/x-www-form-urlencoded). برای خواندن از stdin، از --data @file_name یا --data @'-' استفاده کنید:

```
curl {[ -d|--data ]} {'name=bob'} {{http://example.com/
form}}
```

- اضافی (header) دلخواه و هدرهای HTTP ارسال یک درخواست با استفاده از متود:

```
curl {[ -H|--header ]} {'X-My-Header: 123'} {[ -X|--
request ]} {PUT} {{http://example.com}}
```

- مناسب content-type با مشخص کردن JSON، ارسال داده به صورت:

```
curl {[ -d|--data ]} {'{"name":"bob"}'} {[ -H|--header ]}
{'Content-Type: application/json'} {{http://example.com/
users/1234}}
```

- مشخص کردن یک نام کاربری و درخواست وارد کردن رمز عبور از سرور، به منظور احراز هویت:


```
curl {[[-u|--user]]} {{username}} {{http://example.com}}
```

- عبور از گواهی و کلید کاربر یک منبع (رد شدن از اعتبارسنجی گواهی):

```
curl {[[-E|--cert]]} {{client.pem}} --key {{key.pem}} {[[-k|--insecure]]} {{https://example.com}}
```

dd

تبدیل و کپی یک فایل

اطلاعات بیشتر: <https://manned.org/dd.1p>.

- `archlinux-xxx.iso` یک حافظه قابل حمل با قابلیت بوت شدن میسازد، برای مثال:

```
dd if={{path/to/file.iso}} of={{/dev/usb_drive}}
```

- محتویات یک درایو را در مکانی دیگر با بلوک های 4 مگابایتی کپی و همچنین از خطاها صرف نظر میکند:

```
dd bs=4194304 conv=noerror if={{/dev/source_drive}} of={{/dev/dest_drive}}
```

- یک فایل ۱۰۰ بایتی تصادفی با استفاده از درایور تصادفی هسته بسازید:

```
dd if=/dev/urandom of={{path/to/random_file}} bs=100  
count={{1}}
```

- عملکرد نوشتن دیسک را بسنجید:

```
dd if=/dev/zero of={{path/to/file_1GB}} bs={{1024}}  
count={{1000000}}
```

- میسازد IMG یک پشتیبان از سیستم را در یک فایل:

```
dd if={{/dev/drive_device}} of={{path/to/file.img}}
```

- بازیابی کنید IMG یک درایو را از یک فایل:

```
dd if={{path/to/file.img}} of={{/dev/drive_device}}
```

ddgr

جستجو در موتور جستجوی داک داک گو با تریمنال

اطلاعات بیشتر: <https://github.com/jarun/ddgr>.

- شروع در حالت تعاملی:

```
ddgr
```

- جستجو برای یک کلیدواژه:

```
ddgr {{keyword}}
```

- محدود کردن تعداد نتایج به مقدار دلخواه:

```
ddgr -n {{N}} {{keyword}}
```

- نمایش لینک کامل در نتایج به دست آمده:

```
ddgr -x {{keyword}}
```

- جستجو برای یک کلیدواژه و باز کردن اولین نتیجه در مرورگر:

```
ddgr !w {{keyword}}
```

- اجرای جستجوی منحصر به یک وبسایت خاص:

```
ddgr -w {{site}} {{keyword}}
```

- جستجو برای یک نوع فایل خاص:

```
ddgr {{keyword}} filetype:{{filetype}}
```

- نمایش راهنما در حالت تعاملی:

```
<?>
```

deno

TypeScript و JavaScript **امن** برای runtime یک

اطلاعات بیشتر: <https://docs.deno.com/runtime/reference/cli>.

- اجرای یک فایل JavaScript یا TypeScript:

```
deno run {{path/to/file.ts}}
```

- (پوسته تعاملی) REPL شروع یک:

```
deno
```

- اجرای یک فایل با دسترسی به شبکه فعال:

```
deno run --allow-net {{path/to/file.ts}}
```

- URL اجرای یک فایل از یک:

```
deno run {{https://deno.land/std/examples/welcome.ts}}
```

- URL نصب یک اسکریپت اجرایی از یک:

```
deno install {{https://deno.land/std/examples/colors.ts}}
```

df

یک نمای کلی از میزان استفاده فضای دیسک و سیستم فایل ارائه می‌دهد.

اطلاعات بیشتر: <https://manned.org/df.1posix>.

- نمایش تمامی سیستم‌های فایل و فضای دیسک آن‌ها:

```
df
```

- نمایش تمامی سیستم‌های فایل و استفاده از حالت خوانا برای فضای دیسک آن‌ها:

```
df -h
```

- نمایش سیستم‌فایل و فضای دیسک آن شامل فایل یا دایرکتوری داده شده:

```
df {{path/to/file_or_directory}}
```

- نمایش آمار تعداد درایوهای آزاد:

```
df -i
```

- نمایش سیستم‌های فایل با حذف انواع مشخص شده:

```
df -x {{squashfs}} -x {{tmpfs}}
```

dict

DICT دیکشنری تحت خط فرمان با استفاده از پروتکل

اطلاعات بیشتر: <https://github.com/cheusov/dictd>.

- فهرست کردن دیتابیس های موجود:

```
dict {[ -D|--dbs]}
```

- دریافت اطلاعات درمورد یک دیتابیس:

```
dict {[ -i|--info]} {{database_name}}
```

- جستجوی یک کلمه در یک دیتابیس:

```
dict {[ -d|--database]} {{database_name}} {{word}}
```

- جستجوی یک کلمه در تمام دیتابیس های موجود:

```
dict {{word}}
```

- DICT نمایش اطلاعات درمورد سرور:

```
dict {[ -I|--serverinfo]}
```

diff

مقایسه فایل(ها) و پوشه(ها).

اطلاعات بیشتر: <https://manned.org/diff>.

- مقایسه فایل ها (فهرست تغییرات فایل های قدیمی به جدید):

```
diff {{old_file}} {{new_file}}
```

- مقایسه فایل ها، با صرف نظر از فاصله های خالی:

```
diff {[ -w|--ignore-all-space]} {{old_file}} {{new_file}}
```

- مقایسه فایل ها، با نمایش تفاوت ها در کنار هم:

```
diff {[ -y|--side-by-side]} {{old_file}} {{new_file}}
```

- (git diff) مقایسه فایل ها، به نمایش تفاوت ها به صورت یکپارچه (همانند):

```
diff {[ -u|--unified]} {{old_file}} {{new_file}}
```

- مقایسه بازگشتی پوشه ها (نمایش اسامی متفاوت فایل ها و پوشه ها و همچنین تغییرات فایل ها):

```
diff {[ -r|--recursive]} {{old_directory}} {{new_directory}}
```

- نمایش نام فایل های متفاوت مقایسه شده:

```
diff {[ -r|--recursive]} {[ -q|--brief]} {{old_directory}}  
{{new_directory}}
```

- : از تفاوت دو فایل متنی یک بروزرسانی میسازد، فایل های ناموجود را خالی فرض میکند

```
diff {[ -a|--text]} {[ -u|--unified]} {[ -N|--new-file]}  
{{old_file}} {{new_file}} > {{diff.patch}}
```

dive

ابزاری برای کاوش یک فایل داکر، لایه بندی محتواها و پیدا کردن راه هایی که می توان آن را کاهش داد.

اطلاعات بیشتر: <https://github.com/wagoodman/dive>.

- بررسی یک ایمج داکر:

```
dive {{your_image_tag}}
```

- ساخت یک ایمج و شروع بررسی آن:

```
dive build -t {{some_tag}}
```


django-admin

جعبه ابزار فریمورک جنگو برای کارهای مدیریتی.

اطلاعات بیشتر: <https://docs.djangoproject.com/en/stable/ref/django-admin/>.

- ساخت یک پروژه جنگو جدید:

```
django-admin startproject {{project_name}}
```

- ساخت یک برنامه جدید برای این پروژه:

```
django-admin startapp {{app_name}}
```

- نسخه کنونی جنگو:

```
django-admin --version
```

- نمایش اطلاعات بیشتر برای دستور موردنظر:

```
django-admin help {{command}}
```

echo

چاپ ورودی داده شده.

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/echo-invocation.html.

- چاپ پیام ورودی. نکته: استفاده از علامت نقل قول انتخابی است:

```
echo "{{سلام دنیا}}"
```

- چاپ یک پیام حاوی متغیرهای سیستمی:

```
echo "{{است $PATH مسیر سیستم من}}"
```

- چاپ یک پیام بدون انتقال به خط جدید:

```
echo -n "{{سلام دنیا}}"
```

- افزودن یک پیام به انتهای یک فایل:

```
echo "{{سلام دنیا}}" >> {{file.txt}}
```

- tab فعال کردن تفسیر کاراکترهای خاص، برای مثال:

```
echo -e "{{Column 1\tColumn 2}}"
```

- چاپ وضعیت خروج آخرین دستور اجرا شده (نکته: معادل های این دستور در ویندوز به (echo %errorlevel% و \$lastexitcode هستند):

```
echo $?
```

export

دستور تغییر متغیرهای محلی سیستم موجود برای پروسه های جدید

اطلاعات بیشتر: <https://manned.org/export.1posix>.

- ایجاد و تعیین مقدار یک متغیر جدید:

```
export {{VARIABLE}}={{value}}
```

- \$PATH افزودن یک مسیر به متغیر:

```
export PATH=$PATH:{{path/to/append}}
```

false

برگرداندن 1 به عنوان کد خروجی.

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/false-invocation.html.

- برگرداندن 1 به عنوان کد خروجی:

false

fg

اجرای کارها در پیش زمینه

اطلاعات بیشتر: <https://www.gnu.org/software/bash/manual/bash.html#index-fg>.

- آوردن آخرین کار اجرا شده در پس زمینه به پیش زمینه

```
fg
```

- آوردن کار با آی دی مشخص شده به پیش زمینه

```
fg %{{job_id}}
```

fortune

چاپ نقل قول های تصادفی.

اطلاعات بیشتر: <https://manned.org/fortune>.

- چاپ یک نقل قول:

```
fortune
```

- چاپ یک نقل قول توهین آمیز:

```
fortune -o
```

- چاپ یک نقل قول طولانی:

```
fortune -l
```

- چاپ یک نقل قول کوتاه:

```
fortune -s
```

- فهرست فایل های پایگاه داده نقل قول های موجود:

```
fortune -f
```

- فهرست شده `fortune -f` چاپ یک نقل قول از یکی از فایل های پایگاه داده که توسطه اند:

```
fortune {{path/to/file}}
```

grep

یافتن الگو در فایل ها به کمک عبارات با قاعده.

اطلاعات بیشتر: <https://www.gnu.org/software/grep/manual/grep.html>.

- جستجو یک الگو در یک فایل :

```
grep "{{search_pattern}}" {{path/to/file}}
```

- جستجو یک عبارت خاص (معادل مقایسه رشته ای) :

```
grep {[ -F|--fixed-strings]} "{{exact_string}}" {{path/to/file}}
```

- جستجو بازگشتی یک الگو در تمامی فایل های یک پوشه، نمایش تمامی خطوط منطبق، فایل های باینری را رد میکند:

```
grep {[ -rnI|--recursive --line-number --binary-files=without-match]} "{{search_pattern}}" {{path/to/directory}}
```

- در حالت `|`، `()`، `{}`، `+`، `?` استفاده از عبارات با قاعده توسعه یافته (با پشتیبانی از حساس به بزرگی کوچکی کاراکترها :

```
grep {[ -Ei|--extended-regexp --ignore-case]} "{{search_pattern}}" {{path/to/file}}
```

- چاپ 3 خط از قبل و بعد محل انطباق:

```
grep {[ --context|--before-context|--after-context]} 3  
"{{search_pattern}}" {{path/to/file}}
```

- چاپ نام فایل و شماره خط برای هر انطباق با رنگبندی :

```
grep {[ -Hn|--with-filename --line-number]} --color=always  
"{{search_pattern}}" {{path/to/file}}
```

- جستجوی خطوط منطبق، چاپ متن منطبق :

```
grep {[ -o|--only-matching]} "{{search_pattern}}" {{path/to/file}}
```

- رو برای الگوهایی که منطبق نیستند جستجو میکند (stdin) ورودی استاندارد :

```
cat {{path/to/file}} | grep {[ -v|--invert-match]}  
"{{search_pattern}}"
```

hash

مشاهده مکان‌های ذخیره شده فایل‌های اجرایی در حافظه پنهان

اطلاعات بیشتر: <https://www.gnu.org/software/bash/manual/bash.html#index-hash>.

- مشاهده مکان‌های دستورهای ذخیره شده برای پوسته فعلی:

```
hash
```

- پاک کردن جدول هش:

```
hash -r
```

- حذف یک دستور خاص از جدول هش:

```
hash -d {{command}}
```

- نمایش مسیر کامل command:

```
hash -t {{command}}
```


head

نمایش محتوای ابتدایی یک فایل.

اطلاعات بیشتر: <https://manned.org/head.1p>.

- نمایش چند خط اول یک فایل:

```
head -n {{count}} {{path/to/file}}
```

hexo

.چارچوب وبلاگ سریع، ساده و قدرتمند

اطلاعات بیشتر: <https://hexo.io/docs/commands>.

- ساخت اولیه یک وبسایت:

```
hexo init {{path/to/directory}}
```

- ایجاد یک مقاله جدید:

```
hexo new {{layout}} {{title}}
```

- تولید فایل های ایستا:

```
hexo generate
```

- شروع یک سرور محلی:

```
hexo server
```

- پیاده سازی وبسایت:

```
hexo deploy
```

- (public/)و تولید فایل های پابلیک (db.json) پاک کردن فایل کش:

```
hexo clean
```

hostname

نمایش یا تنظیم نام میزبان سیستم.

اطلاعات بیشتر: <https://manned.org/hostname>.

- نمایش نام میزبان فعلی:

```
hostname
```

- نمایش آدرس شبکه نام میزبان:

```
hostname -i
```

- نمایش تمام آدرس‌های شبکه میزبان:

```
hostname -I
```

- نام دامنه کامل (FQDN) نمایش:

```
hostname --fqdn
```

- تنظیم نام میزبان فعلی:

```
hostname {{new_hostname}}
```

ifconfig

تنظیم کننده رابط های شبکه.

اطلاعات بیشتر: <https://net-tools.sourceforge.io/man/ifconfig.8.html>.

- نمایش تنظیمات شبکه یک کارت شبکه :

```
ifconfig eth0
```

- نمایش جزئیات تمامی رابط ها، مشمول رابط های غیرفعال میشود :

```
ifconfig -a
```

- eth0 غیرفعال کردن رابط :

```
ifconfig eth0 down
```

- eth0 فعال کردن رابط :

```
ifconfig eth0 up
```

- eth0 اختصاص آدرس ای پی به رابط :

```
ifconfig eth0 {{آدرس_ای_پی}}
```

ln

استفاده می (Directories) به فایل ها و پوشه ها (link) این دستور برای ایجاد ارتباط شود.

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/ln-invocation.html.

- به یک فایل یا پوشه (symbolic link) ایجاد یک ارتباط نمادین:

```
ln {[ -s|--symbolic]} {/path/to/file_or_directory} {path/to/symlink}
```

- جایگزینی یک ارتباط نمادین موجود، برای اشاره به یک فایل متفاوت:

```
ln {[ -sf|--symbolic --force]} {/path/to/new_file} {path/to/symlink}
```

- به یک فایل (hard link) ایجاد یک لینک سخت:

```
ln {/path/to/file} {path/to/hardlink}
```

logname

نمایش نام کاربر

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/logname-invocation.html.

• نمایش نام کاربر لاگین شده:

```
logname
```

ls

نمایش محتویات دایرکتوری.

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/ls-invocation.html.

- نمایش فایل ها به صورت خطی:

```
ls -l
```

- نمایش کلیه فایل ها، شامل فایل های مخفی:

```
ls {[[-a|--all]]}
```

- نمایش فایل ها، با این تفاوت که / به نام دایرکتوری ها اضافه می شود:

```
ls {[[-F|--classify]]}
```

- نمایش فایل ها به همراه مجوزها، مالک، اندازه و تاریخ تغییرات:

```
ls {[[-la|--all -l]]}
```

- نمایش فایل ها به همراه مجوزها، مالک، اندازه و تاریخ تغییرات، اندازه ها به صورت قابل درک برای انسان:

```
ls {[[-lh|-l --human-readable]]}
```

- نمایش فایل ها به همراه مجوزها، مالک، اندازه و تاریخ تغییرات، مرتب شده با اندازه فایل به صورت نزولی:

```
ls {[[-lsR|-ls --recursive]]}
```

- نمایش فایل ها به همراه مجوزها، مالک، اندازه و تاریخ تغییرات، مرتب شده با تاریخ تغییر فایل به صورت صعودی:

```
ls {[[-ltr|-lt --reverse]]}
```

magick compare

ایجاد یک تصویر مقایسه ای برای مشخص کردن تفاوت‌های دو عکس به صورت بصری.

است ImageMagick بخشی از

اطلاعات بیشتر: <https://imagemagick.org/script/compare.php>.

- مقایسه دو عکس:

```
magick compare {{path/to/image1.png}} {{path/to/image2.png}}  
{{path/to/diff.png}}
```

- مقایسه دو عکس با استفاده از معیار دلخواه:

```
magick compare -verbose -metric {{PSNR}} {{path/to/  
image1.png}} {{path/to/image2.png}} {{path/to/diff.png}}
```


mkdir

ساخت پوشه ها و تنظیم مجوز آنها.

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/mkdir-invocation.html.

- ساخت پوشه مشخص:

```
mkdir {{path/to/directory1 path/to/directory2 ...}}
```

- ساخت پوشه های مشخص به همراه پوشه های والد در صورت نیاز:

```
mkdir {[[-p|--parents]]} {{path/to/directory1 path/to/directory2 ...}}
```

- ساخت پوشه با مجوز های خاص:

```
mkdir {[[-m|--mode]]} {{rwxrw-r--}} {{path/to/directory1 path/to/directory2 ...}}
```

node

(Node.js). برای اجرای جاوا اسکریپت runtime یک

اطلاعات بیشتر: <https://nodejs.org>.

- اجرای یک فایل جاوا اسکریپت:

```
node {{path/to/file}}
```

- (پوسته تعاملی) REPL شروع یک:

```
node
```

- اجرای فایل مشخص شده و راه اندازی مجدد فرآیند هنگام تغییر در یک فایل وارد شده (Node.js 18.11+ نیازمند نسخه):

```
node --watch {{path/to/file}}
```

- ارزیابی کد جاوا اسکریپت با ارسال آن به عنوان یک آرگومان:

```
node {[ -e|--eval]} "{code}"
```

- Node: ارزیابی و چاپ نتیجه، مفید برای چاپ نسخه های وابستگی های

```
node {[ -p|--print]} "process.versions"
```

- توقف اجرای برنامه تا زمانی که دیباگر به کد متصل شود و، Inspector فعال کردن حالت: کد کاملاً تحلیل شود

```
node --no-lazy --inspect-brk {{path/to/file}}
```

nohup

اجرای یک پردازش در پس زمینه حتی زمانی که ترمینال بسته شود

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/nohup-invocation.html.

- اجرای پردازش در پس زمینه فارغ از اجرا بودن ترمینال:

```
nohup {{command}} {{command_options}}
```

pdf-parser

عناصر اساسی یک فایل پی دی اف را بدون رندر کردن آن مشخص می کند.

اطلاعات بیشتر: <https://blog.didierstevens.com/programs/pdf-tools>.

- نمایش آمار مربوط به یک فایل پی دی اف:

```
pdf-parser {[ -a|--stats ]} {[path/to/file.pdf]}
```

- در یک فایل پی دی اف (/Font, /URI, ...) نمایش اشیاء از یک نوع خاص:

```
pdf-parser {[ -t|--type ]} {[object_type]} {[path/to/file.pdf]}
```

- جستجوی رشته ها در اشیاء غیرمستقیم:

```
pdf-parser {[ -s|--search ]} {[search_string]} {[path/to/file.pdf]}
```

sleep

ایجاد تاخیر بر اساس زمان

اطلاعات بیشتر: <https://manned.org/sleep>.

- تاخیر به ثانیه:

```
sleep {{seconds}}
```

- تاخیر به دقیقه:

```
sleep {{minutes}}m
```

- تاخیر به ساعت:

```
sleep {{hours}}h
```

tar

ابزار آرشیو کردن.

ترکیب می‌شود **bzip2** یا **gzip** اغلب با یک روش فشرده‌سازی مانند

اطلاعات بیشتر: <https://www.gnu.org/software/tar/manual/tar.html>.

- فایل [f] ایجاد یک آرشیو و نوشتن آن در یک [c]:

```
tar cf {{path/to/target.tar}} {{path/to/file1 path/to/
file2 ...}}
```

- فایل [f] شده و نوشتن آن در یک gzip ایجاد یک آرشیو [c]:

```
tar czf {{path/to/target.tar.gz}} {{path/to/file1 path/to/
file2 ...}}
```

- شده (فشرده) از یک پوشه با استفاده از مسیرهای نسبی gzip ایجاد یک آرشیو [c]:

```
tar czf {{path/to/target.tar.gz}} {[[-C|--directory]]}
{{path/to/directory}} .
```

- کامل [v] فایل در پوشه فعلی به صورت [f] یک آرشیو (فشرده) [X] استخراج:

```
tar xvf {{path/to/source.tar[.gz|.bz2|.xz]}}
```

- فایل در پوشه مقصد [f] یک آرشیو (فشرده) [X] استخراج:

```
tar xf {{path/to/source.tar[.gz|.bz2|.xz]}} {[[-C|--
directory]]} {{path/to/directory}}
```

- فایل، با استفاده از پسوند فایل برای [f] ایجاد یک آرشیو فشرده و نوشتن آن در یک [c]:
خودکار برنامه فشرده‌سازی [a] تعیین:

```
tar caf {{path/to/target.tar.xz}} {{path/to/file1 path/to/
file2 ...}}
```

- کامل [v] به صورت tar [f] محتویات یک فایل [t] نمایش:

```
tar tvf {{path/to/source.tar}}
```

- فایل [f] فایل‌های منطبق با یک الگو از یک آرشیو [X] استخراج:

```
tar xf {{path/to/source.tar}} --wildcards "{{*.html}}"
```

time

نمایش زمان اجرای یک دستور.

اطلاعات بیشتر: <https://manned.org/time>.

• command: نمایش زمان اجرای دستور.

```
time {{command}}
```

todo

ابزاری ساده و استاندارد برای مدیریت یادداشت و فهرست وظایف.

اطلاعات بیشتر: <https://todoman.readthedocs.io>.

- لیست کارهای آغاز نشده:

```
todo list --startable
```

- اضافه کردن یک وظیفه به فهرست کارها:

```
todo new {{thing_to_do}} --list {{work}}
```

- اضافه کردن مکان به یک وظیفه با آیدی:

```
todo edit --location {{location_name}} {{task_id}}
```

- نمایش جزییات یک وظیفه:

```
todo show {{task_id}}
```

- علامت زدن وظیفه ها با آیدی مشخص شده به عنوان تکمیل شده:

```
todo done {{task_id1 task_id2 ...}}
```

- حذف یک وظیفه:

```
todo delete {{task_id}}
```

- حذف وظایف انجام شده و بازشماری آیدی وظایف باقی مانده:

```
todo flush
```


touch

ایجاد فایل‌ها و تنظیم زمان‌های دسترسی/تغییر

اطلاعات بیشتر: <https://manned.org/touch>.

- ایجاد فایل‌های مشخص:

```
touch {{path/to/file1 path/to/file2 ...}}
```

- ایجاد فایل در صورت [c] تغییر فایل به زمان فعلی و عدم [m] دسترسی یا [a] تنظیم زمان عدم وجود:

```
touch {{[-c|--no-create]}} -{{a|m}} {{path/to/file1 path/to/file2 ...}}
```

- ایجاد فایل در صورت عدم وجود [c] زمان فایل به یک مقدار مشخص و عدم [t] تنظیم:

```
touch {{[-c|--no-create]}} -t {{YYYYMMDDHHMM.SS}} {{path/to/file1 path/to/file2 ...}}
```

- ایجاد فایل در صورت عدم وجود [c] مرجع، و عدم [r] تنظیم زمان فایل‌ها به زمان فایل:

```
touch {{[-c|--no-create]}} {{[-r|--reference]}} {{path/to/reference_file}} {{path/to/file1 path/to/file2 ...}}
```

tty

نمایش نام ترمینال

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/tty-invocation.html.

- نمایش نام فایل ترمینال جاری

tty

uname

نمایش اطلاعاتی درباره سخت افزار و سیستم عامل.

نکته: برای دستیابی به اطلاعات اضافه در رابطه با سیستم عامل از دستور `lsb_release` کنید.

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/uname-invocation.html.

- نمایش اطلاعات مربوط به سخت افزار و پردازنده سیستم:

```
uname -mp
```

- نمایش اطلاعات مربوط به نرم افزار از جمله: سیستم عامل، شماره انتشار و نسخه:

```
uname -srv
```

- نمایش نام سیستم:

```
uname -n
```

- نمایش تمامی اطلاعات سیستم (سخت افزار، نرم افزار، نام سیستم):

```
uname -a
```

unclutter

مخفی کردن نشان ماوس.

اطلاعات بیشتر: <https://manned.org/unclutter.1x>.

• مخفی کردن نشان ماوس بعد از 3 ثانیه:

```
unclutter -idle {{3}}
```

users

نمایش لیست کاربران لاگین شده

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/users-invocation.html.

- نمایش لیست کاربران لاگین شده

```
users
```

- نمایش لیست کاربران لاگین شده بر اساس یک لاگ فایل خاص

```
users {{/var/log/wtmp}}
```

view

vim. نسخه فقط خواندنی

vim -R. معادل

اطلاعات بیشتر: <https://www.vim.org>.

- باز کردن فایل:

```
view {{file}}
```

where

نمایش محل فایل باینری یا اجرایی یک دستور در لینوکس.

اطلاعات بیشتر: <https://zsh.sourceforge.io/Doc/Release/Shell-Builtin-Commands.html>.

• نمایش محل فایل اجرایی یک دستور:

```
where {{command}}
```

yard

ابزار مستند سازی برای زبان برنامه نویسی روبی.

اطلاعات بیشتر: <https://rubydoc.info/gems/yard/file/docs/GettingStarted.md#using-yard-to-generate-documentation>.

- ساخت مستندات:

```
yard
```

- ساخت مستندات و ذخیره آن در یک فایل:

```
yard --one-file
```

- فهرست تمام اشیاء مستند نشده:

```
yard stats --list-undoc
```


you-get

دانلود محتوای چندرسانه ای از وب (ویدیو، صوت، عکس)

اطلاعات بیشتر: <https://you-get.org/#getting-started>.

- چاپ اطلاعات درمورد یک رسانه خاص در سطح وب:

```
you-get --info {{https://example.com/video?id=value}}
```

- دانلود رسانه از لینک موردنظر:

```
you-get {{https://example.com/video?id=value}}
```

- جستجو در ویدیوهای گوگل و دانلود:

```
you-get {{keywords}}
```

- دانلود یک رسانه و ذخیره در محل ذخیره شده:

```
you-get --output-dir {{path/to/directory}} --output-filename  
{{filename}} {{https://example.com/watch?v=value}}
```

- دانلود یک رسانه با استفاده از پروکسی:

```
you-get --http-proxy {{proxy_server}} {{https://example.com/  
watch?v=value}}
```

zip

Zip بسته‌بندی و فشرده‌سازی (آرشیو) فایل‌ها در یک آرشیو

unzip: همچنین ببینید

اطلاعات بیشتر: <https://manned.org/zip>.

- افزودن فایل‌ها/پوشه‌ها به یک آرشیو مشخص (به صورت بازگشتی):

```
zip {[[-r|--recurse-paths]]} {{path/to/compressed.zip}}  
{{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- حذف فایل‌ها/پوشه‌ها از یک آرشیو مشخص:

```
zip {[[-d|--delete]]} {{path/to/compressed.zip}} {{path/to/  
file_or_directory1 path/to/file_or_directory2 ...}}
```

- آرشیو فایل‌ها/پوشه‌ها با مستثنی کردن موارد مشخص شده:

```
zip {[[-r|--recurse-paths]]} {{path/to/compressed.zip}}  
{{path/to/file_or_directory1 path/to/file_or_directory2 ...}}  
{{[-x|--exclude]]} {{path/to/excluded_files_or_directories}}
```

- (بیشترین - 9، کمترین - 0) آرشیو فایل‌ها/پوشه‌ها با سطح فشرده‌سازی مشخص:

```
zip {[[-r|--recurse-paths]]} -{{0..9}} {{path/to/  
compressed.zip}} {{path/to/file_or_directory1 path/to/  
file_or_directory2 ...}}
```

- ایجاد یک آرشیو رمزگذاری شده با رمز عبور مشخص:

```
zip {[[-re|--recurse-paths --encrypt]]} {{path/to/  
compressed.zip}} {{path/to/file_or_directory1 path/to/  
file_or_directory2 ...}}
```

- آرشیو فایل‌ها/پوشه‌ها در یک آرشیو چند بخشی تقسیم شده (مثلاً بخش‌های ۳ گیگابایتی):

```
zip {[[-rs|--recurse-paths --split-size]]} {{3g}} {{path/to/  
compressed.zip}} {{path/to/file_or_directory1 path/to/  
file_or_directory2 ...}}
```

- نمایش محتویات یک آرشیو مشخص:

```
zip {[[-sf|--split-size --freshen]]} {{path/to/  
compressed.zip}}
```

zm

ابزاری برای مدیریت مقاله های مجلات و وبلاگ ها

اطلاعات بیشتر: <https://github.com/ZERMZeitung/zm2>.

- ساخت یک پیش نویس جدید:

```
zm new
```

- ویرایش یک پیش نویس:

```
zm edit
```

- انتشار یک پیش نویس و ثبت در گیت:

```
zm publish
```

~

به یک دایرکتوری گسترش می یابد.

اطلاعات بیشتر: <https://gnu.org/software/bash/manual/bash.html#Tilde-Expansion>.

- محتوای دایرکتوری خانه کاربر فعلی را فهرست می کند:

```
ls ~
```

- محتوای دایرکتوری خانه یک کاربر دیگر را فهرست می کند:

```
ls ~{{username}}
```

- محتوای دایرکتوری قبلی که شما در آن بودید را فهرست می کند:

```
ls ~-
```

Linux

abbr

مدیریت مخفف های fish shell

جایگزین کردن کلمات وارد شده توسط کاربر با جملات طولانی

اطلاعات بیشتر: <https://fishshell.com/docs/current/cmds/abbr.html>.

- اضافه کردن مخفف جدید:

```
abbr --add {{abbreviation_name}} {{command}}  
{{command_arguments}}
```

- تغییر نام یک مخفف موجود:

```
abbr --rename {{old_name}} {{new_name}}
```

- پاک کردن یک مخفف موجود:

```
abbr --erase {{abbreviation_name}}
```

- SSH: وارد کردن یک مخفف وارد شده در یک میزبان دیگر از طریق:

```
ssh {{host_name}} abbr --show | source
```

ac

چاپ مدت زمان اتصال کاربران

اطلاعات بیشتر: <https://www.gnu.org/software/acct/manual/accounting.html#ac>.

- چاپ تعداد ساعات اتصال کاربر کنونی:

```
ac
```

- چاپ تعداد ساعات اتصال کاربران:

```
ac {[[-p|--individual-totals]]}
```

- چاپ تعداد ساعات اتصال یک کاربر خاص:

```
ac {[[-p|--individual-totals]]} {{username}}
```

- چاپ تعداد ساعات اتصال یک کاربر خاص (به همراه مجموع آن):

```
ac {[[-d|--daily-totals]]} {[[-p|--individual-totals]]}  
{{username}}
```

- نمایش اطلاعات بیشتر:

```
ac --compatibility
```

adduser

ابزار اضافه کردن کاربر.

اطلاعات بیشتر: <https://manned.org/adduser>.

- ایجاد یک کاربر جدید با دایرکتوری خانگی پیش فرض و درخواست از کاربر برای تنظیم رمز عبور:

```
adduser {{username}}
```

- ایجاد یک کاربر جدید بدون دایرکتوری خانگی:

```
adduser --no-create-home {{username}}
```

- ایجاد یک کاربر جدید با دایرکتوری خانگی در مسیر مشخص:

```
adduser --home {{path/to/home}} {{username}}
```

- مشخص به عنوان پوسته ورود (shell) ایجاد یک کاربر جدید با تنظیم پوسته:

```
adduser --shell {{path/to/shell}} {{username}}
```

- ایجاد یک کاربر جدید که به گروه مشخصی تعلق دارد:

```
adduser --ingroup {{group}} {{username}}
```


apk

ابزار مدیریت بسته آلباین لینوکس.

اطلاعات بیشتر: https://wiki.alpinelinux.org/wiki/Alpine_Package_Keeper.

- آپدیت فهرست مخزن ها از تمام مخازن ریموت:

```
apk update
```

- نصب یک بسته جدید:

```
apk add {{package}}
```

- حذف یک بسته:

```
apk del {{package}}
```

- تعمیر یک بسته یا ارتقا آن بدون تغییر دادن وابستگی های اصلی:

```
apk fix {{package}}
```

- جستجوی یک بسته با کلمات کلیدی:

```
apk search {{keywords}}
```

- نمایش اطلاعات درمورد بسته مورد نظر:

```
apk info {{package}}
```

apt-get

ابزار مدیریت بسته‌های دبیان و اوبونتو.

`apt-cache` جستجو در بسته‌ها با استفاده از

اطلاعات بیشتر: <https://manned.org/apt-get.8>.

- به‌روزرسانی لیست بسته‌ها و نسخه‌های موجود (توصیه می‌شود که این دستور را قبل از اجرا کنید `apt-get` دیگر دستورات):

```
sudo apt-get update
```

- نصب یک بسته یا به‌روزرسانی آن به آخرین نسخه موجود:

```
sudo apt-get install {{package}}
```

- حذف یک بسته:

```
sudo apt-get remove {{package}}
```

- حذف یک بسته و فایل‌های پیکربندی آن:

```
sudo apt-get purge {{package}}
```

- ارتقاء تمامی بسته‌های نصب شده به آخرین نسخه‌های موجود:

```
sudo apt-get upgrade
```

- از دانلودهای متوقف شده که دیگر (`.deb`) پاکسازی مخزن محلی - حذف فایل‌های بسته قابل دانلود نیستند:

```
sudo apt-get autoclean
```

- حذف تمام بسته‌هایی که دیگر نیازی به آن‌ها نیست:

```
sudo apt-get autoremove
```

- اما با حذف بسته‌های قدیمی و نصب (`upgrade` ارتقاء بسته‌های نصب شده (مانند) بسته‌های اضافی برای برآورده کردن وابستگی‌های جدید:

```
sudo apt-get dist-upgrade
```

bitwise

ابزار تبدیل مبنا چند منظوره با قابلیت پشتیبانی از مبنای پویا و دستکاری بیت.

اطلاعات بیشتر: <https://github.com/mellowcandle/bitwise>.

- اجرا در حالت هم کنشی:

```
bitwise
```

- تبدیل از دسیمال:

```
bitwise {{12345}}
```

- تبدیل از هگزادسیمال:

```
bitwise {{0x563d}}
```

- تبدیل یک محاسبه با قالب زبان سی:

```
bitwise "{{0x123 + 0x20 - 30 / 50}}"
```

caja

MATE. مدیریت فایلها و پوشه ها در محیط دسکتاپ

اطلاعات بیشتر: <https://manned.org/caja>.

- باز کردن پوشه خانگی کاربر کنونی:

```
caja
```

- بازکردن پوشه های مشخص شده در پنجره جداگانه:

```
caja {{path/to/directory1 path/to/directory2 ...}}
```

- بازکردن پوشه های مشخص شده در تب ها:

```
caja --tabs {{path/to/directory1 path/to/directory2 ...}}
```

- بازکردن یک پوشه در یک پنجره با اندازه مشخص:

```
caja --geometry={{600}}x{{400}} {{path/to/directory}}
```

- بستن همه پنجره ها:

```
caja --quit
```

cal

چاپ اطلاعات تقویم همراه با هایلایت کردن روز جاری

اطلاعات بیشتر: <https://manned.org/cal>.

- نمایش تقویم ماه جاری:

```
cal
```

- نمایش ماه قبلی، جاری و بعدی:

```
cal {[ -3 | --three ]}
```

- در نظر گرفتن دوشنبه به عنوان اولین روز هفته:

```
cal {[ -m | --monday ]}
```

- نمایش تقویم سال مورد نظر (چهار رقمی):

```
cal {{year}}
```

- نمایش تقویم ماه و سال مورد نظر:

```
cal {{month}} {{year}}
```

dd

تبدیل و کپی یک فایل

اطلاعات بیشتر: https://www.gnu.org/software/coreutils/manual/html_node/dd-invocation.html.

- و نمایش (مثل archlinux-xxx.iso) iso قابل بوت از یک فایل USB ساخت یک درایو:
پیشرفت:

```
dd if={{path/to/file.iso}} of={{/dev/usb_drive}}  
status=progress
```

- کلون کردن یک درایو به یک درایو دیگر با اندازه بلوک ۴ مگابایت و اعمال چیزهای نوشته شده پیش از خاتمه دستور:

```
dd bs=4M conv=fsync if={{/dev/source_drive}} of={{/dev/  
dest_drive}}
```

- کرنل random ایجاد یک فایل با تعداد مشخصی بایت تصادفی با استفاده از درایور:

```
dd bs={{100}} count={{1}} if=/dev/urandom of={{path/to/  
random_file}}
```

- ارزیابی عملکرد نوشتن روی یک دیسک:

```
dd bs={{1M}} count={{1024}} if=/dev/zero of={{path/to/  
file_1GB}}
```

- of به if می‌توان بعداً با تغییر IMG ساخت یک پشتیبان از سامانه و ذخیره آن در یک فایل (آن را بازسازی کرد):

```
dd if={{/dev/drive_device}} of={{path/to/file.img}}  
status=progress
```

- این دستور را از یک پوسته دیگر اجرا) dd بررسی پیشرفت یک عملکرد در حال اجرای (کنید:

```
kill -USR1 $(pgrep -x dd)
```

dnf

(yum جایگزین) CentOS و Fedora، RHEL ابزار مدیریت بسته‌ها برای

برای دستورات معادل در دیگر مدیران بسته، به <https://wiki.archlinux.org/title/Pacman/Rosetta> مراجعه کنید.

اطلاعات بیشتر: <https://dnf.readthedocs.io>.

- ارتقاء بسته‌های نصب شده به جدیدترین نسخه‌های موجود:

```
sudo dnf upgrade
```

- جستجوی بسته‌ها بر اساس کلمات کلیدی:

```
dnf search {{keyword1 keyword2 ...}}
```

- نمایش جزئیات یک بسته:

```
dnf info {{package}}
```

- (برای تأیید اتوماتیک تمام پنجره‌ها استفاده کنید y - نصب یک بسته جدید (از):

```
sudo dnf install {{package1 package2 ...}}
```

- حذف یک بسته:

```
sudo dnf remove {{package1 package2 ...}}
```

- لیست بسته‌های نصب شده:

```
dnf list --installed
```

- یافتن بسته‌هایی که دستور مشخصی را ارائه می‌دهند:

```
dnf provides {{command}}
```

- مشاهده تاریخچه تمام عملیات‌های گذشته:

```
dnf history
```

fdisk

مدیریت جدول‌های پارتیشن و پارتیشن‌ها بر روی یک درایو ذخیره‌سازی.

partprobe, parted, cfdisk. این را هم ببینید

اطلاعات بیشتر: <https://manned.org/fdisk>.

- لیست کردن پارتیشن‌ها:

```
sudo fdisk {{[-l|--list]}}
```

- ورود به حالت مدیریت پارتیشن به صورت تعاملی:

```
sudo fdisk {{/dev/sdX}}
```

- نمایش منوی راهنما ([m]enu):

<m>

- چاپ جدول پارتیشن ([p]rint):

<p>

- ایجاد یک پارتیشن جدید ([n]ew):

<n>

- حذف یک پارتیشن ([d]elete):

<d>

- ذخیره تغییرات در جدول ([w]rite):

<w>

- خروج بدون ذخیره‌ی تغییرات ([q]uit):

<q>

lscpu

(CPU). نمایش اطلاعات درباره‌ی معماری پردازنده

اطلاعات بیشتر: <https://manned.org/lscpu>.

- نمایش اطلاعات درباره‌ی تمام پردازنده‌ها:

```
lscpu
```

- نمایش اطلاعات در قالب جدول:

```
lscpu {[ -e|--extended]}
```

- نمایش اطلاعات فقط درباره‌ی پردازنده‌های آنلاین (فعال) در قالب جدول:

```
lscpu {[ -e|--extended]} {[ -b|--online]}
```

- نمایش اطلاعات فقط درباره‌ی پردازنده‌های آفلاین (غیرفعال) در قالب جدول:

```
lscpu {[ -e|--extended]} {[ -c|--offline]}
```

- پردازنده (Cache) نمایش جزئیات درباره‌ی حافظه‌های پنهان:

```
lscpu {[ -C|--caches]}
```

- JSON نمایش اطلاعات در فرمت:

```
lscpu {[ -J|--json]}
```

lspci

PCI لیست کردن تمام دستگاه‌های

اطلاعات بیشتر: <https://manned.org/lspci>.

- نمایش یک لیست مختصر از دستگاه‌ها:

```
lspci
```

- برای جزئیات بیشتر قابل تکرار -v (نکته: فلگ) ([v]erbose) نمایش اطلاعات با جزئیات کامل (است):

```
lspci -v
```

- مربوط به هر دستگاه ([k]ernel) نمایش درایورها و ماژول‌های کرنل:

```
lspci -k
```

- انتخاب و نمایش اطلاعات یک دستگاه خاص ([s]elect):

```
lspci -s {{00:18.3}}
```

- ([m]achine) نمایش خروجی در فرمت قابل خواندن برای ماشین :

```
lspci -vm
```

- ([n]umbers) هم به صورت عددی (device) و دستگاه (vendor) نمایش کدهای فروشنده و ([n]ames) هم نام:

```
lspci -nn
```

pacman

واحد مدیریت پکیج آرچ لینوکس

همچنین: `pacman-database`, `pacman-deptest`, `pacman-files`, `pacman-key`, `pacman-mirrors`, `pacman-query`, `pacman-remove`, `pacman-sync`, `pacman-upgrade`.

برای دیدن دستور های معادل در سایر پکیج منیجر ها <https://wiki.archlinux.org/title/Pacman/Rosetta>.

اطلاعات بیشتر: <https://manned.org/pacman.8>.

- همگام سازی و بروز رسانی تمام پکیج ها:

```
sudo pacman -Syu
```

- نصب پکیج جدید:

```
sudo pacman -S {{package}}
```

- حذف یک پکیج به همراه وابستگی هاش:

```
sudo pacman -Rs {{package}}
```

- جستجو در دیتابیس برای پکیج هایی که با یک فایل خاص تعارض دارند:

```
pacman -F "{{file_name}}"
```

- لیست کردن پکیج های نصب شده با نسخه آنها:

```
pacman -Q
```

- لیست کردن تنها پکیج هایی که مستقیماً نصب شده اند به همراه نسخه آنها:

```
pacman -Qe
```

- لیست کردن پکیج هایی که به عنوان وابستگی نصب شده اند اما توسط هیچ پکیجی استفاده نمیشوند:

```
pacman -Qtdq
```

- `pacman` خالی کردن کل کش:

```
sudo pacman -Scc
```

whatis

نمایش توضیحات یک خطی از صفحات راهنما

اطلاعات بیشتر: <https://manned.org/whatis>.

- نمایش توضیحات یک دستور از صفحات راهنما:

```
whatis {{command}}
```

- توضیحات در آخر خط ترمینال برش نمی خورد:

```
whatis --long {{command}}
```

- نمایش توضیحات تمامی دستورات مطابق با الگو:

```
whatis --wildcard {{net*}}
```

- جستجو در توضیحات صفحات راهنما با عبارات منظم:

```
whatis --regex '{{wish[0-9]\.[0-9]}}'
```

yum

ابزار مدیریت بسته برای ردهت، فدورا و سنت اواس (برای نسخه های قدیمی)

اطلاعات بیشتر: <https://manned.org/yum>.

- نصب یک بسته:

```
yum install {{package}}
```

- update نصب یک بسته با فرض بر اینکه پاسخ شما برای تمامی سوالات بله است (با گزینه (هم می توان از این روش استفاده کرد، مناسب برای به روز رسانی خودکار

```
yum -y install {{package}}
```

- پیدا کردن بسته ای که دستور مورد نظر را فراهم می کند:

```
yum provides {{command}}
```

- حذف یک بسته:

```
yum remove {{package}}
```

- نمایش به روز رسانی ها برای بسته های نصب شده

```
yum check-update
```

- به روز رسانی بسته های نصب شده به آخرین نسخه موجود

```
yum upgrade
```

Windows

cls

پاک کردن صفحه

اطلاعات بیشتر: <https://learn.microsoft.com/windows-server/administration/windows-commands/cls>.

• پاک کردن صفحه:

```
cls
```

type

نمایش محتویات فایل.

اطلاعات بیشتر: <https://learn.microsoft.com/windows-server/administration/windows-commands/type>.

• نمایش محتویات فایل مشخص شده •

```
type {{path/to/file}}
```


ver

نمایش شماره نسخه ویندوز یا داس

اطلاعات بیشتر: <https://learn.microsoft.com/windows-server/administration/windows-commands/ver>.

- نمایش شماره نسخه •

ver

where

نمایش محل فایل باینری یا اجرایی یک دستور در لینوکس

اطلاعات بیشتر: <https://learn.microsoft.com/windows-server/administration/windows-commands/where>.

• نمایش محل فایل اجرایی یک دستور:

```
where {{command}}
```

