

# tldr pages book

Simplified and community-driven man pages

*Generated on Mon Jul 22 16:57:35 2024*

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}}
```

- یک اینتنت رو به یک URI تبدیل میکنه :

```
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

[https://cs.android.com/android/platform/superproject/+main:frameworks/native/cmds/bugreport](https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreport) اطلاعات بیشتر:

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

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

- تولید که گزارش خطای کامل از یک دستگاه اندرویدی :

`bugreportz`

- `bugreportz` نمایش فرایند اجرای دستور :

`bugreportz -p`

- `bugreportz` نمایش نسخه :

`bugreportz -v`

- نمایش راهنمایی :

`bugreportz -h`

# 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 10 ثانیه) :

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



# getprop

نمایش اطلاعات مربوط به مشخصات سیستم اندروید.

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

- نمایش اطلاعات مربوط به مشخصات سیستم اندروید:

```
getprop
```

- نمایش اطلاعات مربوط به یک مشخصه:

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

- نمایش سطح SDK رابط برنامه نویسی:

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

- نمایش نسخه اندروید:

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

- نمایش مدل دستگاه اندروید:

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

- نمایش اطلاعات قفل OEM:

```
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

[https://developer.android.com/reference/android/view/KeyEvent.html#constants\\_1](https://developer.android.com/reference/android/view/KeyEvent.html#constants_1) اطلاعات بیشتر:

- 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 پیا هم های سیستمی، شامل پشته رهگیری زمان وقوع خطا، و اطلاعات پیام های لاگ شده توسط برنامه ها.

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

- 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}}
```

- نمایش لاگ های مربوط به یک PID مشخص :

```
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.

[https://wiki.termux.com/wiki/Package\\_Management](https://wiki.termux.com/wiki/Package_Management) اطلاعات بیشتر:

- 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 ...}}
```

- نمایش مسیر فایل APK یک برنامه مشخص:

```
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://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://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://docs.python.org/3/library/2to3.html>.

- 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.

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

- 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 می‌کند اما چند-سکوپی است.

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

- 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}}
```

- استخراج یک آرشیو در stdout:

```
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 فایل های 7z عملکردی مشابه

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

- 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}}
```

- استخراج یک آرشیو در stdout:

```
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 مقدارها

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

اطلاعات بیشتر: <https://www.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 متغیری 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 متغیری 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://www.gnu.org/software/bash/manual/bash.html#index-005b\\_005b](https://www.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 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} ]]
```

# agate

یک سرور ساده برای پروتکل شبکه Gemini.

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
```



# airpaste

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

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

- 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}}
```

# 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 تعریف شوند. `.bashrc` فایل های پیکربندی شل مربوط نظیر

<https://tldp.org/LDP/abs/html/aliases.html> اطلاعات بیشتر:

- 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.

<https://github.com/pystardust/ani-cli> اطلاعات بیشتر:

- جستجوی انیمه با نام:

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

- دانلود قسمت:

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

- استفاده از VLC به عنوان پخش کننده ویدیو:

```
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
```

# 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}}"
```

- دانلود از FTP با نام کاربری و رمز عبور:

```
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
```

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

```
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.

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

- 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

مدیریت و جابجایی بین نسخه های مختلف Neovim.

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

- نصب و جابجایی به نسخه خاصی از Neovim:

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

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

```
bob list
```

- حذف یک نسخه خاص از Neovim:

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

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

```
bob erase
```

- بازگشت به نسخه شبانه قبلی:

```
bob rollback
```

# boxes

کشیدن، حذف و تعمیر جعبه های هنری اسکی.

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

- 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
```



# calendar

نمایش رویداد های پیش رو با استفاده از یک فایل تقویم.

<https://www.commandlinux.com/man-page/man1/calendar.1.html> اطلاعات بیشتر:

- 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.

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

- 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://github.com/posva/catimg>.

- چاپ یک JPEG, PNG, یا GIF در ترمینال:

```
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.

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

- نمایش تمام کلیپ بوردها:

```
cb
```

- رونوشت یک فایل به کلیپ برد:

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

- کپی متن دلخواه به کلیپ برد:

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

- رونوشت داده ی هدایت شده به داخل در کلیپ برد:

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

- چسباندن محتوای کلیپ برد:

```
cb paste
```

- هدایت محتوای کلیپ برد به خارج:

```
cb | cat
```

- نمایش تاریخچه کلیپ برد:

```
cb history
```

- نمایش اطلاعات کلیپ برد:

```
cb info
```

# cd

تغییر دایرکتوری جاری.

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

- تغییر مسیر به دایرکتوری مشخص شده:

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

- تغییر مسیر به دایرکتوری Home کاربر فعلی:

```
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.

<https://www.gnu.org/software/coreutils/chmod> اطلاعات بیشتر:

- به مالک فایل دسترسی اجرا میدهد:

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

- به کابر 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 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 a+rX {{path/to/directory}}
```

# chown

مالک فایل یا پوشه را تغییر میدهد.

<https://www.gnu.org/software/coreutils/chown> اطلاعات بیشتر:

- 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 {{user}} {{path/to/directory}}
```

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

```
chown -h {{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.

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

- 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/html\\_node/Invoking-cmp.html](https://www.gnu.org/software/diffutils/manual/html_node/Invoking-cmp.html).

- 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 --print-bytes {{path/to/file1}} {{path/to/file2}}
```

- نمایش شماره بایتها و مقادیر تمامی تفاوت ها:

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

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

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

# composer

ابزاری بسته محور برای مدیریت وابستگی های پروژه های PHP.

اطلاعات بیشتر: <https://getcomposer.org/>.

- 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` وارد می کند:

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

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

```
composer install
```

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

```
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.

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

- 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}}
```

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

```
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}}
```

- رونوشت یک عکس با فرمت JPEG در کلیپ بورد:

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

# cp

## کپی فایل ها و دایرکتوری ها.

<https://www.gnu.org/software/coreutils/cp> اطلاعات بیشتر:

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

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

- اجرا روی فایل دلخواه مانند main.cpp:

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

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

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

# csvkit

ابزاری برای کارکردن با فایل های CSV.

همچنین ببینید: `csvclean`, `csvcut`, `csvformat`, `csvgrep`, `csvlook`, `csvpy`, `csvsort`, `csvstat`.

اطلاعات بیشتر: <https://csvkit.readthedocs.io/en/0.9.1/cli.html>.

- اجرای یک دستور روی یک فایل CSV با جداکننده دلخواه:

```
{{command}} -d {{delimiter}} {{path/to/file.csv}}
```

- اجرای یک دستور روی یک فایل CSV با استفاده از tab به عنوان جداکننده:

```
{{command}} -t {{path/to/file.csv}}
```

- اجرای یک دستور روی فایل CSV با کاراکتر نقلی شخصی سازی شده:

```
{{command}} -q {{quote_char}} {{path/to/file.csv}}
```

- اجرای یک دستور روی یک فایل CSV بدون خط هدر:

```
{{command}} -H {{path/to/file.csv}}
```

# csvpy

اجرای یک فایل CSV در شل پایتون.

در csvkit گنجانده شده است.

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

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

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

- CSVKitDictReader اجرای یک فایل CSV در یک آبجت:

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

# curl

انتقال داده از/به سرور.

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

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

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

```
curl {{http://example.com}} --output {{path/to/file}}
```

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

```
curl --remote-name {{http://example.com/filename}}
```

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

```
curl --fail --remote-name --location --continue-at -  
{{http://example.com/filename}}
```

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

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

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

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

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

```
curl --data {'{"name":"bob"}'} --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 --user {{username}} {{http://example.com}}
```

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

```
curl --cert {{client.pem}} --key {{key.pem}} --insecure  
{{https://example.com}}
```



# dd

## تبدیل و کپی یک فایل.

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

- `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}}
```

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

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

- یک درایو را از یک فایل IMG بازیابی کنید:

```
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:

```
?
```

# df

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

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

- 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}}
```

# diff

## مقایسه فایل(ها) و پوشه(ها).

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

- 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 --ignore-all-space {{old_file}} {{new_file}}
```

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

```
diff --side-by-side {{old_file}} {{new_file}}
```

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

```
diff --unified {{old_file}} {{new_file}}
```

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

```
diff --recursive {{old_directory}} {{new_directory}}
```

- نمایش نام فایل های متفاوت مقایسه شده:

```
diff --recursive --brief {{old_directory}} {{new_directory}}
```

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

```
diff --text --unified --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.

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

- 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.

<https://docs.djangoproject.com/en/4.2/ref/django-admin/> اطلاعات بیشتر:

- 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.

<https://www.gnu.org/software/coreutils/echo> اطلاعات بیشتر:

- 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}}
```

- فعال کردن تفسیر کاراکترهای خاص، برای مثال `tab`:

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

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

```
echo $?
```

# export

دستور تغییر متغیرهای محلی سیستم موجود برای پروسه های جدید.

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

- ایجاد و تعیین مقدار یک متغیر جدید:

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

- افزودن یک مسیر به متغیر PATH\$:

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



# false

برگرداندن 1 به عنوان کد خروجی.

<https://www.gnu.org/software/coreutils/false> اطلاعات بیشتر:

• برگرداندن 1 به عنوان کد خروجی:

```
false
```

# fg

اجرای کارها در پیش زمینه.

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

- 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.

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

- 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 {{-r|--recursive}} {{-n|--line-number}} --binary-files={{without-match}} "{{search_pattern}}" {{path/to/directory}}
```

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

```
grep {{-E|--extended-regexp}} {{-i|--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 {{-H|--with-filename}} {{-n|--line-number}} --color=always "{{search_pattern}}" {{path/to/file}}
```

- جستجوی خطوط منطبق، چاپ متن منطبق :

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

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

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

# head

نمایش محتوای ابتدایی یک فایل.

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

- نمایش چند خط اول یک فایل:

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

# hexo

چارچوب وبلاگ سریع، ساده و قدرتمند.

<https://hexo.io/> اطلاعات بیشتر:

- 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
```

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

```
hexo clean
```

# 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
```

- غیرفعال کردن رابط eth0 :

```
ifconfig eth0 down
```

- فعال کردن رابط eth0 :

```
ifconfig eth0 up
```

- اختصاص آدرس ای پی به رابط eth0 :

```
ifconfig eth0 {{آدرس_ای_پی}}
```

# ln

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

اطلاعات بیشتر: <https://www.gnu.org/software/coreutils/ln>.

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

```
ln -s {{/path/to/file_or_directory}} {{path/to/symlink}}
```

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

```
ln -sf {{/path/to/new_file}} {{path/to/symlink}}
```

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

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



# logname

نمایش نام کاربر.

اطلاعات بیشتر: <https://www.gnu.org/software/coreutils/logname>.

- نمایش نام کاربر لاگین شده:

```
logname
```

# ls

نمایش محتویات دایرکتوری.

<https://www.gnu.org/software/coreutils/ls> اطلاعات بیشتر:

- نمایش فایل ها به صورت خطی:

```
ls -l
```

- نمایش کلیه فایل ها، شامل فایل های مخفی:

```
ls -a
```

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

```
ls -F
```

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

```
ls -la
```

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

```
ls -lh
```

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

```
ls -ls
```

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

```
ls -ltr
```

# magick compare

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

بخشی از ImageMagick است.

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

- 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/mkdir>.

- ساخت پوشه مشخص:

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

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

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

- ساخت پوشه با مجوز های خاص:

```
mkdir -m {{rwxrw-r--}} {{path/to/directory1 path/to/directory2 ...}}
```

# nohup

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

اطلاعات بیشتر: <https://www.gnu.org/software/coreutils/nohup>.

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

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

# sleep

ایجاد تاخیر بر اساس زمان.

اطلاعات بیشتر: <https://pubs.opengroup.org/onlinepubs/9699919799/utilities/sleep.html>.

- تاخیر به ثانیه:

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

- تاخیر به دقیقه:

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

- تاخیر به ساعت:

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

# time

نمایش زمان اجرای یک دستور.

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

- command:نمایش زمان اجرای دستور

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

# todo

ابزاری ساده و استاندارد برای مدیریت یادداشت و فهرست وظایف.

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

- 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
```



# tty

نمایش نام ترمینال.

<https://www.gnu.org/software/coreutils/tty> اطلاعات بیشتر:

- 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` استفاده کنید.

<https://www.gnu.org/software/coreutils/uname> اطلاعات بیشتر:

- 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.

<https://www.gnu.org/software/coreutils/users> اطلاعات بیشتر:

- نمایش لیست کاربران لاگین شده:

```
users
```

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

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

# view

نسخه فقط خواندنی vim.

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.

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

- نمایش محل فایل اجرایی یک دستور:

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

# whereis

پیدا کردن فایل اجرایی، سورس، صفحه راهنما برای یک دستور.

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

- پیدا کردن فایل اجرایی، سورس و صفحه راهنما برای SSH:

```
whereis {{ssh}}
```

- پیدا کردن فایل اجرایی و صفحه راهنما برای ls:

```
whereis -bm {{ls}}
```

- پیدا کردن سورس برای gcc و صفحه راهنما برای git:

```
whereis -s {{gcc}} -m {{git}}
```

- پیدا کردن فایل اجرایی برای gcc در مسیر /usr/bin/:

```
whereis -b -B {{/usr/bin/}} -f {{gcc}}
```

- پیدا کردن فایل های اجرایی غیر عادی(برای آنهایی که بیشتر از یک فایل اجرایی در سیستم دارند):

```
whereis -u *
```

- پیدا کردن صفحات راهنمای غیر عادی(برای آنهایی که بیشتر از یک فایل اجرایی در سیستم دارند):

```
whereis -u -m *
```

# yard

ابزار مستند سازی برای زبان برنامه نویسی رویی.

<https://yardoc.org/> اطلاعات بیشتر:

- 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 چند رسانه ای از وب (ویدیو، صوت، عکس).

<https://you-get.org> اطلاعات بیشتر:

- چاپ اطلاعات در مورد 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}}
```

# zm

ابزاری برای مدیریت مقاله های مجلات و وبلاگ ها.

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

- ساخت یک پیش نویس جدید:

```
zm new
```

- ویرایش یک پیش نویس:

```
zm edit
```

- انتشار یک پیش نویس و ثبت در گیت:

```
zm publish
```

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}}
```

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

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

# ac

چاپ مدت زمان اتصال کاربران.

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

- چاپ تعداد ساعات اتصال کاربر کنونی:

```
ac
```

- چاپ تعداد ساعات اتصال کاربران:

```
ac --individual-totals
```

- چاپ تعداد ساعات اتصال یک کاربر خاص:

```
ac --individual-totals {{username}}
```

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

```
ac --daily-totals --individual-totals {{username}}
```

- نمایش اطلاعات بیشتر:

```
ac --compatibility
```

# adduser

## ابزار اضافه کردن کاربر.

اطلاعات بیشتر: <https://manpages.debian.org/latest/adduser/adduser.html>.

- 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}}
```

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

```
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.

[https://wiki.alpinelinux.org/wiki/Alpine\\_Linux\\_package\\_management](https://wiki.alpinelinux.org/wiki/Alpine_Linux_package_management) اطلاعات بیشتر:

- 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 دیان و اوبونتو.

**apt-cache** جستجو در بسته‌ها با استفاده از

اطلاعات بیشتر: <https://manpages.debian.org/latest/apt/apt-get.8.html>.

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

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

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

- حذف یک بسته:

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

- حذف یک بسته و فایل‌های پیکربندی آن:

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

- ارتقاء تمامی بسته‌های نصب شده به آخرین نسخه‌های موجود:

```
apt-get upgrade
```

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

```
apt-get autoclean
```

- حذف تمام بسته‌هایی که دیگر نیازی به آن‌ها نیست:

```
apt-get autoremove
```

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

```
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

مدیریت فایلها و پوشه ها در محیط دسکتاپ MATE.

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

- 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
```

- در نظر گرفتن دوشنبه به عنوان اولین روز هفته:

```
cal --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.

<https://www.gnu.org/software/coreutils/dd> اطلاعات بیشتر:

- (و نمایش `archlinux-xxx.iso` ساخت یک درایو USB قابل بوت از یک فایل iso (مثل 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}}
```

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

```
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}}
```

- به `if` ساخت یک پشتیبان از سامانه و ذخیره آن در یک فایل IMG (می‌توان بعداً با تغییر `of`: 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) ابزار مدیریت بسته‌ها برای RHEL، Fedora و CentOS (جایگزین).

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

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

- 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
```

# pacman

## واحد مدیریت پکیج آرچ لینوکس

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

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

<https://man.archlinux.org/man/pacman.8> اطلاعات بیشتر:

- 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.

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

- 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).

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

- نصب یک بسته:

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

- نصب یک بسته با فرض بر اینکه پاسخ شما برای تمامی سوالات بله است(با گزینه update 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.

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

- پاک کردن صفحه:

```
cls
```

# type

نمایش محتویات فایل.

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

- نمایش محتویات فایل مشخص شده:

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

# ver

نمایش شماره نسخه ویندوز یا داس.

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

- نمایش شماره نسخه:

ver

# where

نمایش محل فایل باینری یا اجرایی یک دستور در لینوکس.

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

- نمایش محل فایل اجرایی یک دستور:

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

