

# tldr pages book

Simplified and community-driven man pages

*Generated on Mon Oct 21 05:10:31 2024*

Website: <https://tldr.sh>

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

# Android

# am

Gestore delle Attività di Android.

Maggiori informazioni: <https://developer.android.com/tools/adb#am>.

- Inizia un'attività:

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

- Inizia un'attività e e invia dati:

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

- Inizia un'attività corrispondente ad un'azione, e in una categoria delineato:

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

- Converti un intent in un URI:

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

# bugreport

Mostra un report dei bug del dispositivo Android.

Questo comando può essere usato solamente attraverso la **adb shell**.

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

- Mostra un report completo dei bug riscontrati nel dispositivo Android:

```
bugreport
```

# cmd

Gestore del servizio Android.

Maggiori informazioni: <https://cs.android.com/android/platform/superproject/main/frameworks/native/cmds/cmd/>.

- Elenca tutti i servizi in esecuzione:

```
cmd -l
```

- Chiama un servizio specifico:

```
cmd {{alarm}}
```

- Chiama un servizio con argomenti:

```
cmd {{vibrator}} {{vibrate 300}}
```

# dalvikvm

Macchina virtuale Java per Android.

Maggiori informazioni: <https://source.android.com/docs/core/runtime>.

- Lancia un programma Java:

```
dalvikvm -classpath {{percorso/del/file.jar}} {{nomeclasse}}
```

# getprop

Visualizza informazioni sulle proprietà del sistema Android.

Maggiori informazioni: <https://manned.org/getprop>.

- Visualizza informazioni sulle proprietà del sistema Android:

```
getprop
```

- Visualizza informazioni su una specifica proprietà del sistema Android:

```
getprop {{prop}}
```

- Visualizza il livello della API SDK:

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

- Visualizza la versione Android:

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

- Visualizza il modello del dispositivo Android:

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

- Visualizza lo stato dello sblocco OEM:

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

- Visualizza l'indirizzo MAC della scheda Wi-Fi di Android:

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

# input

Invia codici evento o gesture touchscreen a un dispositivo Android.

Questo comando può essere usato solo attraverso **abd shell**.

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

- Invia un codice evento per un singolo carattere a un dispositivo Android:

```
input keyevent {{codice_evento}}
```

- Invia un testo a un dispositivo Android (%s rappresenta lo spazio):

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

- Invia un singolo tap a un dispositivo Android:

```
input tap {{pos_x}} {{pos_y}}
```

- Invia una gesture di scorrimento a un dispositivo Android:

```
input swipe {{inizio_x}} {{inizio_y}} {{fine_x}} {{fine_y}}  
{{durata_in_ms}}
```

- Invia un tap lungo a un dispositivo Android usando una gesture di scorrimento:

```
input swipe {{pos_x}} {{pos_y}} {{pos_x}} {{pos_y}}  
{{durata_in_ms}}
```



# logcat

Scarica il registro dei messaggi di sistema, comprese le stack traces quando si verifica un errore, e i messaggi di log delle applicazioni.

Maggiori informazioni: <https://developer.android.com/tools/logcat>.

- Mostra il log di sistema:

```
logcat
```

- Scrivi il log di sistema su file:

```
logcat -f {{percorso/del/file}}
```

- Mostra le righe corrispondenti ad una specifica espressione regolare:

```
logcat --regex {{espressione_regolare}}
```

# pkg

Strumento di gestione pacchetti per Termux.

Maggiori informazioni: [https://wiki.termux.com/wiki/Package\\_Management](https://wiki.termux.com/wiki/Package_Management).

- Aggiorna tutti i pacchetti installati:

```
pkg upgrade
```

- Installa un pacchetto:

```
pkg install {{pacchetto}}
```

- Disinstalla un pacchetto:

```
pkg uninstall {{pacchetto}}
```

- Reinstalla un pacchetto:

```
pkg reinstall {{pacchetto}}
```

- Cerca un pacchetto:

```
pkg search {{pacchetto}}
```

# settings

Acquisisci informazioni su Android OS.

Maggiori informazioni: <https://adbinstaller.com/commands/adb-shell-settings-5b670d5ee7958178a2955536>.

- Visualizza una lista di impostazioni nel namespace `global`:

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

- Ottieni il valore di un'impostazione specifica:

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

- Imposta il valore di un'impostazione:

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

- Elimina un'impostazione specifica:

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

# wm

Visualizza informazioni dello schermo del dispositivo Android.

Questo comando può essere usato solo attraverso **adb shell**.

Maggiori informazioni: <https://adbinstaller.com/commands/adb-shell-wm-5b672b17e7958178a2955538>.

- Visualizza le dimensioni fisiche dello schermo di un dispositivo Android:

```
wm size
```

- Visualizza la densità fisica dello schermo di un dispositivo Android:

```
wm density
```

Common

# Exclamation mark

Comando bash integrato per sostituire con un comando trovato nella cronologia.

Maggiori informazioni: <https://gnu.org/software/bash/manual/bash.html#Event-Designators>.

- Sostituisci con il comando precedente ed esegilo con sudo:

```
sudo !!
```

- Sostituisci con un comando in base al numero di riga trovato con `history`:

```
!{{numero}}
```

- Sostituisci con un comando che è stato utilizzato un numero specificato di righe indietro:

```
!-{{numero}}
```

- Sostituisci con il comando più recente che inizia con stringa:

```
!{{stringa}}
```

- Sostituisci con gli argomenti dell'ultimo comando:

```
{{comando}} !*
```

# 7z

Archiviatore di file con alto fattore di compressione.

Maggiori informazioni: <https://manned.org/7z>.

- Archivia un file o una directory:

```
7z a {{archivio.7z}} {{percorso/del/file_o_directory}}
```

- Cripta un archivio esistente (inclusi gli header):

```
7z a {{archivio_criptato.7z}} -p{{password}} -mhe=on  
{{archivio.7z}}
```

- Estrai un archivio mantenendo la gerarchia delle directory:

```
7z x {{archivio.7z}}
```

- Estrai un archivio in una specifica directory:

```
7z x {{archivio.7z}} -o{{percorso/della/directory}}
```

- Estrai un archivio sullo standard output:

```
7z x {{archivio.7z}} -so
```

- Archivia utilizzando uno specifico tipo di archivio:

```
7z a -t {{7z|bzip2|gzip|lzip|tar|zip}} {{archivio}}  
{{percorso/del/file_o_directory}}
```

- Elenca i contenuti in un archivio:

```
7z l {{archivio.7z}}
```

# 7za

Archiviatore di file con alto fattore di compressione.

Versione standalone di **7z** con supporto a meno tipi di archivi.

Maggiori informazioni: <https://manned.org/7za>.

- Archivia un file o una directory:

```
7za a {{archivio.7z}} {{percorso/del/file_o_directory}}
```

- Estrai un archivio mantenendo la gerarchia delle directory:

```
7za x {{archivio.7z}}
```

- Archivia utilizzando uno specifico tipo di archivio:

```
7za a -t {{zip|gzip|bzip2|tar}} {{archivio.7z}} {{percorso/del/file_o_directory}}
```

- Elenca i contenuti in un archivio:

```
7za l {{archivio}}
```



# 7zr

Archiviatore di file con alto fattore di compressione.

Versione standalone di **7z** che supporta solo file .7z.

Maggiori informazioni: <https://manned.org/7zr>.

- Archivia un file o una directory:

```
7zr a {{archivio.7z}} {{percorso/del/file_o_directory}}
```

- Estrai un archivio mantenendo la gerarchia delle directory:

```
7zr x {{archivio.7z}}
```

- Elenca i contenuti in un archivio:

```
7zr l {{archivio}}
```

[

Controlla i tipi di file e confronta i valori.

Restituisce uno stato pari a 0 se la condizione risulta vera, 1 se risulta falsa.

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

- Verifica se una determinata variabile è uguale/diversa dalla stringa specificata:

```
[ "${variabile}" {=!|=} "${stringa}" ]
```

- Verifica se una variabile data è [eq]uguale/[n]on [e]uguale/[g]maggiore [t]di/[l]minore [t]di/[g]maggiore di o [e]uguale/[l]minore di o [e]uguale al numero specificato:

```
[ "${variabile}" -{eq|ne|gt|lt|ge|le} {{intero}} ]
```

- Verifica se la variabile specificata ha un valore [n]on-empty :

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

- Verifica se la variabile specificata ha un valore vuoto:

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

- Verifica se il [f]ile specificato esiste:

```
[ -f {{percorso/del/file}} ]
```

- Verifica se la [d]irectory specificata esiste:

```
[ -d {{percorso/della/directory}} ]
```

- Verifica se la directory o file specificata/o [e]siste:

```
[ -e {{percorso/del/file_o_directory}} ]
```

# ab

Strumento di benchmarking di Apache. Il più semplice modo per eseguire un test sul carico del server.

Maggiori informazioni: <https://httpd.apache.org/docs/current/programs/ab.html>.

- Esegui 100 richieste HTTP GET ad un dato URL:

```
ab -n 100 {{url}}
```

- Esegui 100 richieste HTTP GET ad un dato URL, processandone fino a 10 contemporaneamente:

```
ab -n 100 -c 10 {{url}}
```

- Esegui 100 richieste HTTP POST a un dato URL, utilizzando un payload JSON tramite file:

```
ab -n 100 -T {{application/json}} -p {{percorso/del/
file.json}} {{url}}
```

- Usa HTTP [K]eep Alive, ovvero esegui richieste multiple in una stessa sessione HTTP:

```
ab -k {{url}}
```

- Setta il massimo numero di secondi per il benchmarking:

```
ab -t {{secondi}} {{url}}
```

# abduco

Gestore di sessioni di terminale.

Maggiori informazioni: <https://www.brain-dump.org/projects/abduco/>.

- List sessioni:

```
abduco
```

- Entra in una sessione, creandola se non esiste già:

```
abduco -A {{nome_sessione}} {{bash}}
```

- Entra in una sessione con `dvtm`, creandola se non esiste già:

```
abduco -A {{nome_sessione}}
```

- Esci da una sessione:

```
<Ctrl> + \
```

- Entra in una sessione in modalità sola lettura:

```
abduco -Ar {{nome_sessione}}
```

# ack

Un tool di ricerca simile a **grep**, ottimizzato per programmatori.

Vedi anche: **rg**, che è molto più veloce.

Maggiori informazioni: <https://beyondgrep.com/documentation>.

- Cerca ricorsivamente file contenenti una stringa o un'espressione regolare nella directory corrente:

```
ack "{{pattern_di_ricerca}}"
```

- Cerca un pattern in modalità case-insensitive:

```
ack --ignore-case "{{pattern_di_ricerca}}"
```

- Cerca righe di testo contenenti un pattern, mostrando solo il testo corrispondente e non il resto della riga:

```
ack -o "{{pattern_di_ricerca}}"
```

- Limita la ricerca ai file di un tipo specifico:

```
ack --type {{ruby}} "{{pattern_di_ricerca}}"
```

- Non cercare nei file di un tipo specifico:

```
ack --type no{{ruby}} "{{pattern_di_ricerca}}"
```

- Conta il numero totale di corrispondenze trovate:

```
ack --count --no-filename "{{pattern_di_ricerca}}"
```

- Mostra i nomi dei file e il numero di corrispondenze per ogni singolo file:

```
ack --count --files-with-matches "{{pattern_di_ricerca}}"
```

- Mostra la lista di tutti i valori che possono essere usati con `--type`:

```
ack --help-types
```

# adb install

Android Debug Bridge Install: Invia pacchetti ad un emulatore Android od ad un dispositivo Android connesso.

Maggiori informazioni: <https://developer.android.com/tools/adb>.

- Invia un'applicazione Android ad un emulatore emulatore/Android:

```
adb install {{percorso/del/file.apk}}
```

- Reinstalla una applicazione esistente, preservandone i dati:

```
adb install -r {{percorso/del/file.apk}}
```

- Fornisce tutti i permessi elencati nel manifest dell'applicazione:

```
adb install -g {{percorso/del/file.apk}}
```

- Aggiorna rapidamente un pacchetto installato aggiornando solamente le parti dell'APK che sono cambiate:

```
adb install --fastdeploy {{percorso/del/file.apk}}
```

# adb shell

Android Debug Bridge Shell: Esegue un comando remoto sull'emulatore o dispositivo Android connesso.

Maggiori informazioni: <https://developer.android.com/tools/adb>.

- Avvia un interprete di comandi iterativo remoto sull'emulatore/dispositivo:

```
adb shell
```

- Fornisce tutte le proprietà dell'emulatore o dispositivo:

```
adb shell getprop
```

- Ripristina tutti i permessi di esecuzione ai loro valori predefiniti:

```
adb shell pm reset-permissions
```

- Revoca un permesso pericoloso da un'applicazione:

```
adb shell pm revoke {{pacchetto}} {{permesso}}
```

- Attiva un evento chiave:

```
adb shell input keyevent {{keycode}}
```

- Pulisce i dati di un'applicazione sull'emulatore o dispositivo:

```
adb shell pm clear {{pacchetto}}
```

- Avvia un'attività sull'emulatore/dispositivo:

```
adb shell am start -n {{pacchetto}}/{{attività}}
```

- Avvia la schermata iniziale sull'emulatore o dispositivo:

```
adb shell am start -W -c android.intent.category.HOME -a  
android.intent.action.MAIN
```

# adb

Android Debug Bridge: comunica con un'istanza di un emulatore Android o con un dispositivo android connesso.

Alcuni comandi aggiuntivi, come **shell**, hanno la propria documentazione.

Maggiori informazioni: <https://developer.android.com/tools/adb>.

- Controlla se il processo server adb è attivo ed avvialo:

```
adb start-server
```

- Termina il processo server adb:

```
adb kill-server
```

- Avvia una shell remota nell'emulatore o dispositivo target:

```
adb shell
```

- Installa un'applicazione Android nell'emulatore o dispositivo target:

```
adb install -r {{percorso/del/file.apk}}
```

- Copia file o directory dal dispositivo target:

```
adb pull {{percorso/del/file_o_directory_dispositivo}}  
{{percorso/del/file_o_directory_locale}}
```

- Copia file/directory sul dispositivo target:

```
adb push {{percorso/del/file_o_directory_locale}} {{percorso/  
della/directory_destinazione_dispositivo}}
```

- Mostra una lista dei dispositivi connessi:

```
adb devices
```



# ag

The Silver Searcher. Come **ack**, ma più veloce.

Maggiori informazioni: [https://github.com/ggreer/the\\_silver\\_searcher](https://github.com/ggreer/the_silver_searcher).

- Trova file contenenti "foo" e mostra le corrispondenti linee contenenti il termine:

```
ag {{foo}}
```

- Trova file contenenti "foo" in una specifica directory:

```
ag {{foo}} {{percorso/della/directory}}
```

- Trova file contenenti "foo" elencandone solamente i nomi:

```
ag -l {{foo}}
```

- Trova file contenenti "FOO" senza distinguere tra maiuscole e minuscole, e stampa solo il termine trovato piuttosto che l'intera linea:

```
ag -i -o {{F00}}
```

- Trova "foo" in file il quale nome contiene "bar":

```
ag {{foo}} -G {{bar}}
```

- Trova file il quale contenuto soddisfi una determinata espressione regolare:

```
ag '{{espressione_regolare}}'
```

- Trova file il quale nome contiene "foo":

```
ag -g {{foo}}
```

# agate

Un semplice server per il protocollo di rete Gemini.

Maggiori informazioni: <https://github.com/mbrubeck/agate>.

- Esegui e genera una chiave privata e un certificato:

```
agate --content {{percorso/del/contenuto}} --addr {[::]:  
1965}} --addr {{0.0.0.0:1965}} --hostname {{esempio.it}} --  
lang {{it-IT}}
```

- Avvia server:

```
agate {{percorso/del/file}}
```

- Visualizza la guida:

```
agate -h
```

# age

Uno strumento semplice, moderno e sicuro per la cifratura di file.

Maggiori informazioni: <https://github.com/FiloSottile/age>.

- Generare un file cifrato che può essere decifrato con una passphrase:

```
age --passphrase --output {{percorso/del/file_cifrato}}  
{{percorso/del/file_non_cifrato}}
```

- Generare una coppia di chiavi, salvando la chiave privata in un file non cifrato e stampando sullo stdout la chiave pubblica:

```
age-keygen --output {{percorso/del/file}}
```

- Cifrare un file con una o più chiavi pubbliche inserite come letterali:

```
age --recipient {{chiave_pubblica_1}} --recipient  
{{chiave_pubblica_2}} {{percorso/del/file_non_cifrato}} --  
output {{percorso/del/file_cifrato}}
```

- Cifrare un file con una o più chiavi pubbliche specificate in un file di destinatari:

```
age --recipients-file {{percorso/del/file_di_destinatari}}  
{{percorso/del/file_non_cifrato}} --output {{percorso/del/  
file_cifrato}}
```

- Decifrare un file con una passphrase:

```
age --decrypt --output {{percorso/del/file_decifrato}}  
{{percorso/del/file_cifrato}}
```

- Decifrare un file con il file di una chiave privata:

```
age --decrypt --identity {{percorso/del/file_chiave_privata}}  
--output {{percorso/del/file_decifrato}} {{percorso/del/  
file_cifrato}}
```

# airpaste

Condividi messaggi e file nella stessa rete.

Maggiori informazioni: <https://github.com/mafintosh/airpaste>.

- Aspetta un messaggio e mostralo una volta ricevuto:

```
airpaste
```

- Invia un messaggio di testo:

```
echo {{messaggio}} | airpaste
```

- Invia un file:

```
airpaste < {{percorso/del/file}}
```

- Ricevi un file:

```
airpaste > {{percorso/del/file}}
```

- Crea o entra in un canale:

```
airpaste {{nome_canale}}
```

# alacritty

Multiplatforma, GPU-accelerato emulatore di terminale.

Maggiori informazioni: <https://github.com/alacritty/alacritty>.

- Apri un nuovo finestra di Alacritty:

```
alacritty
```

- Esegui in una directory specifica:

```
alacritty --working-directory {{percorso/della/directory}}
```

- Esegui un comando in una nuova finestra di Alacritty:

```
alacritty -e {{comando}}
```

- Specifica un file di configurazione alternativo (predefinito a  
\$XDG\_CONFIG\_HOME/alacritty/alacritty.toml):

```
alacritty --config-file {{percorso/di/config.toml}}
```

- Esegui con ricaricamento configurazione live (può anche essere acceso in  
alacritty.toml):

```
alacritty --live-config-reload --config-file {{percorsi/al/  
config.toml}}
```

# alex

Uno strumento per individuare frasi scritte in modo insensibile o sconsiderato.

Aiuta a trovare termini che favoriscono un certo genere, legati alla razza, religiosamente inappropriati, o simili termini non equi in un testo.

Maggiori informazioni: <https://github.com/get-alex/alex>.

- Analizza testo da standard input:

```
echo {{Frase da analizzare}} | alex --stdin
```

- Analizza tutti i file nella directory corrente:

```
alex
```

- Analizza uno specifico file:

```
alex {{file.md}}
```

- Analizza tutti i file Markdown eccetto `esempio.md`:

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

# alias

Crea alias -- parole che sono sostituite da stringhe di comandi.

Gli alias vengono persi alla chiusura della shell corrente, a meno che non siano definiti nel file di configurazione della shell (ad esempio `~/.bashrc`).

Maggiori informazioni: <https://tldp.org/LDP/abs/html/aliases.html>.

- Crea un alias:

```
alias {{parola}}="{{comando}}"
```

- Mostra il comando associato ad un dato alias:

```
alias {{parola}}
```

- Rimuovi un alias:

```
unalias {{parola}}
```

- Rendi il comando rm interattivo:

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

- Crea un alias la come scorciatoia per il comando ls -a:

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

# ansible-galaxy

Crea e gestisci ruoli di Ansible.

Maggiori informazioni: <https://docs.ansible.com/ansible/latest/cli/ansible-galaxy.html>.

- Installa un ruolo:

```
ansible-galaxy install {{nome_utente}}.{{ruolo}}
```

- Rimuovi un ruolo:

```
ansible-galaxy remove {{nome_utente}}.{{ruolo}}
```

- Elenca i ruoli installati:

```
ansible-galaxy list
```

- Cerca un determinato ruolo:

```
ansible-galaxy search {{nome_ruolo}}
```

- Crea un nuovo ruolo:

```
ansible-galaxy init {{nome_ruolo}}
```

- Acquisisci informazioni su un ruolo di un utente:

```
ansible-galaxy role info {{nome_utente}}.{{nome_ruolo}}
```

- Acquisisci informazioni su una collection:

```
ansible-galaxy collection info {{nome_utente}}.  
{{nome_raccolta}}
```



# ansible-playbook

Esegui task definiti nel playbook di un computer remoto via SSH.

Maggiori informazioni: <https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html>.

- Esegui tasks nel playbook:

```
ansible-playbook {{playbook}}
```

- Esegui task nel playbook con un inventory personalizzato:

```
ansible-playbook {{playbook}} -i {{inventory}}
```

- Esegui task nel playvook con variabili aggiuntive definite da linea di comando:

```
ansible-playbook {{playbook}} -e "{{variabile1}}={{valore1}}  
{{variabile2}}={{valore2}} ..."
```

- Esegui task nel playbook con variabili aggiuntive definite in un file JSON:

```
ansible-playbook {{playbook}} -e "@{{variabili.json}}"
```

# ansible

Gestisce gruppi di computer da remoto via SSH.

Usa il file `/etc/ansible/hosts` per aggiungere nuovi gruppi/host.

Alcuni comandi aggiuntivi, come **galaxy**, hanno la propria documentazione.

Maggiori informazioni: <https://www.ansible.com/>.

- Elenca gli host appartenenti ad un gruppo:

```
ansible {{gruppo}} --list-hosts
```

- Invia un ping ad un gruppo di host invocando il modulo "ping":

```
ansible {{gruppo}} -m ping
```

- Mostra informazioni su un gruppo di host invocando il modulo "setup":

```
ansible {{gruppo}} -m setup
```

- Esegui un comando su un gruppo di host invocando il modulo "command" con degli argomenti:

```
ansible {{gruppo}} -m command -a '{{comando_da_eseguire}}'
```

- Esegui un comando con privilegi di amministratore:

```
ansible {{gruppo}} --become --ask-become-pass -m command -a  
'{{comando}}'
```

- Esegui un comando usando un file di inventory personalizzato:

```
ansible {{gruppo}} -i {{file_inventory}} -m command -a  
'{{comando}}'
```

- Elenca i gruppi in un inventory:

```
ansible localhost -m debug -a '{{var=groups.keys()}}'
```

# ansiweather

Uno script per mostrare le attuali condizioni meteo nel tuo terminale.

Maggiori informazioni: <https://github.com/fcambus/ansiweather>.

- Mostra una previsione usando unità SI per i prossimi cinque giorni in Rzeszow (Polonia):

```
ansiweather -u metric -f 7 -l {{Rzeszow,PL}}
```

- Mostra una previsione con simboli e informazioni sulla luce solare per la tua posizione attuale:

```
ansiweather -F -s true -d true
```

- Mostra una previsione con informazioni su vento ed umidità per la tua posizione attuale:

```
ansiweather -w true -h true
```

# apg

Crea password randomiche arbitrariamente complesse.

Maggiori informazioni: <https://manned.org/apg>.

- Genera password randomiche (la lunghezza predefinita è 8):

```
apg
```

- Crea una password con almeno 1 simbolo (S), 1 numero (N), 1 lettera maiuscola (C) e una minuscola (L):

```
apg -M SNCL
```

- Crea una password di 16 caratteri:

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

- Crea una password di massimo 16 caratteri:

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

- Crea una password che non è già presente in un dizionario (file dizionario fornito come argomento):

```
apg -r {{percorso/del/dizionario}}
```

# apm

Manager di pacchetti per l'editor di testo Atom.

Vedi **atom**.

Maggiori informazioni: <https://github.com/atom/apm>.

- Installa pacchetti da <http://atom.io/packages> e temi da <http://atom.io/themes>:

```
apm install {{nome_pacchetto}}
```

- Rimuovi pacchetti/temi:

```
apm remove {{nome_pacchetto}}
```

- Aggiorna pacchetti/temi:

```
apm upgrade {{nome_pacchetto}}
```

# apropos

Cerca nelle pagine di manuale, ad esempio per trovare un nuovo comando.

Maggiori informazioni: <https://manned.org/apropos>.

- Cerca una parola chiave:

```
apropos {{qualcosa}}
```

- Cerca senza limitare l'output alla larghezza del terminale:

```
apropos -l {{qualcosa}}
```

# ar

Crea, modifica ed estrai da archivi (.a, .so, .o).

Maggiori informazioni: <https://manned.org/ar>.

- Estrai ([x]) tutti i membri da un archivio:

```
ar x {{percorso/del/file.a}}
```

- Lis[t]a tutti i membri di un archivio:

```
ar t {{percorso/del/file.ar}}
```

- Sostituisci ([r]) o aggiungi file ad un archivio:

```
ar r {{percorso/del/file.deb}} {{percorso/del/debian-binary  
percorso/del/control.tar.gz percorso/del/data.tar.xz ...}}
```

- In[s]erisci o sostituisci un indice in un archivio (equivalente ad usare `ranlib`):

```
ar s {{percorso/del/file.a}}
```

- Crea un archivio con dei file creando anche il relativo indice:

```
ar rs {{percorso/del/file.a}} {{percorso/del/file1.o  
percorso/del/file2.o ...}}
```

# arch

Mostra il nome dell'architettura del sistema.

Vedi anche **uname**.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/arch>.

- Mostra l'architettura del sistema:

arch



# aria2

Questo comando è un alias per **aria2c**.

- Consulta la documentazione del comando originale:

```
tlDR aria2c
```

# aria2c

Veloce utilità di download.

Supporta HTTP(S), FTP, SFTP, BitTorrent, e Metalink.

Maggiori informazioni: <https://aria2.github.io>.

- Scarica un file da un URI:

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

- Scarica più file da diverse sorgenti:

```
aria2c {{url_1}} {{url_2}}
```

- Scarica gli URI elencati in un file:

```
aria2c -i {{filename}}
```

- Esegui il download con connessioni multiple:

```
aria2c -s {{numero_connessioni}} {{url}}
```

- Scarica via FTP con username e password:

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

- Limita la velocità di download (in byte al secondo):

```
aria2c --max-download-limit={{velocità}} {{url}}
```

# arp

Mostra e gestisci la cache ARP di sistema.

Maggiori informazioni: <https://manned.org/arp>.

- Mostra la tabella ARP corrente:

```
arp -a
```

- Elimina una specifica voce:

```
arp -d {{indirizzo}}
```

- Crea una nuova voce:

```
arp -s {{indirizzo}} {{indirizzo_mac}}
```

# arping

Scopri e interroga host in una rete utilizzando il protocollo ARP.

Utile per scoprire indirizzi MAC.

Maggiori informazioni: <https://github.com/ThomasHabets/arping>.

- Invia un ping ad un host inviando pacchetti ARP request:

```
arping {{ip_host}}
```

- Invia un ping ad un host su una specifica interfaccia:

```
arping -I {{interfaccia}} {{ip_host}}
```

- Invia un ping ad un host e termina all prima risposta:

```
arping -f {{ip_host}}
```

- Invia uno specifico numero di ping:

```
arping -c {{count}} {{ip_host}}
```

- Invia pacchetti ARP request in broadcast per aggiornare la cache ARP dei vicini:

```
arping -U {{ip_da_inviare_in_broadcast}}
```

- Rileva indirizzi IP duplicati nella rete inviando richieste ARP con un timeout di 3 secondi:

```
arping -D -w {{3}} {{ip_da_controllare}}
```

# asar

Gestore di archivi per la piattaforma Electron.

Maggiori informazioni: <https://github.com/electron/asar>.

- Archivia un file o una directory:

```
asar pack {{percorso/del/file}} {{archivio.asar}}
```

- Estrai un archivio:

```
asar extract {{archivio.asar}}
```

- Estrai uno specifico file da un archivio:

```
asar extract-file {{archivio.asar}} {{nome_file}}
```

- Elenca i contenuti in un archivio:

```
asar list {{archivio.asar}}
```

# asciinema

Registra e riproduci sessioni di terminale, condividendole opzionalmente su [asciinema.org](https://asciinema.org).

Maggiori informazioni: <https://docs.asciinema.org/manual/cli/usage>.

- Associa l'installazione locale di `asciiname` ad un account di `asciinema.org`:

```
asciinema auth
```

- Avvia una nuova registrazione (una volta terminata, verrà chiesto se caricarla o salvarla localmente):

```
asciinema rec
```

- Avvia una nuova registrazione e salvala come file locale:

```
asciinema rec {{percorso/del/registrazione.cast}}
```

- Riproduci una sessione da un file locale:

```
asciinema play {{percorso/del/registrazione.cast}}
```

- Riproducei una sessione da `asciinema.org`:

```
asciinema play https://asciinema.org/a/{{id_registrazione}}
```

- Avvia una nuova registrazione, limitando qualsiasi periodo di [i]nattività a 2.5 secondi:

```
asciinema rec {{-i|--idle-time-limit}} 2.5
```

- Stampa l'output completo di una sessione locale:

```
asciinema cat {{percorso/del/registrazione.cast}}
```

- Carica una sessione locale su `asciinema.org`:

```
asciinema upload {{percorso/del/registrazione.cast}}
```

# asdf

Interfaccia da linea di comando per gestire le versioni di diversi pacchetti.

Maggiori informazioni: <https://asdf-vm.com>.

- Elenca tutti i plugin disponibili:

```
asdf plugin list all
```

- Installa un plugin:

```
asdf plugin add {{nome}}
```

- Elenca tutte le versioni disponibili per un pacchetto:

```
asdf list all {{nome}}
```

- Installa una specifica versione di un pacchetto:

```
asdf install {{nome}} {{versiona}}
```

- Imposta la versione globale per un pacchetto:

```
asdf global {{nome}} {{versiona}}
```

- Imposta la versione locale per un pacchetto:

```
asdf local {{nome}} {{versiona}}
```

# assimp

Client da linea di comando per la Open Asset Import Library.

Supporta il caricamento di 40+ formati di file per modelli 3D, e l'esportazione di diversi formati 3D popolari.

Maggiori informazioni: <https://assimp-docs.readthedocs.io/>.

- Elenca tutti i formati supportati:

```
assimp listtext
```

- Elenca tutti i formati supportati per l'esportazione:

```
assimp listexport
```

- Converti un file a uno dei tipi supportati, utilizzando i parametri predefiniti:

```
assimp export {{file_input.stl}} {{file_output.obj}}
```

- Converti un file utilizzando parametri personalizzati (il file `dox_cmd.h` nel source code di assimp contiene una lista di parametri disponibili):

```
assimp export {{file_input.stl}} {{file_output.obj}}  
{{parametri}}
```

- Mostra un riepilogo del contenuto di un file:

```
assimp info {{percorso/del/file}}
```

- Elenca tutti i sottocomandi disponibili (detti "verbs"):

```
assimp help
```

- Chiedi aiuto per uno specifico sottocomando (ad esempio riguardo i suoi parametri):

```
assimp {{sottocomando}} --help
```



# astronomer

Strumento per individuare star illegittime da account bot su progetti GitHub.

Maggiori informazioni: <https://github.com/Ullaakut/astronomer>.

- Scannerizza una repository:

```
astronomer {{tldr-pages/tldr-node-client}}
```

- Scannerizza le star di una repository fino ad un massimo di 50:

```
astronomer {{tldr-pages/tldr-node-client}} --stars {{50}}
```

- Scannerizza una repository includendo report comparativi:

```
astronomer {{tldr-pages/tldr-node-client}} --verbose
```

# astyle

Indentatore, formattatore e beautifier di codice sorgente per i linguaggi C, C++, C# e Java.

Quando eseguito, una copia del file originale è creata con l'estensione ".orig" aggiunta come suffisso.

Maggiori informazioni: <https://astyle.sourceforge.net>.

- Applica lo stile di default di 4 spazi per livello di indentazione e nessun cambiamento alla formattazione:

```
astyle {{file_sorgente}}
```

- Applica lo stile Java con parentesi graffe aperte sulla stessa riga (attached braces):

```
astyle --style=java {{percorso/del/file}}
```

- Applica lo stile allman per parentesi graffe su linee separate (broken braces):

```
astyle --style=allman {{percorso/del/file}}
```

- Applica un'indentazione personalizzata utilizzando spazi. Scegli tra 2 e 20 spazi:

```
astyle --indent=spaces={{numero_spazi}} {{percorso/del/file}}
```

- Applica un'indentazione personalizzata utilizzando tab. Scegli tra 2 e 20 tab:

```
astyle --indent=tab={{numero_tab}} {{percorso/del/file}}
```

# at

Programma l'esecuzione di comandi nel futuro.

Il servizio atd (o atrun) deve essere attivo per eseguire i comandi.

Maggiori informazioni: <https://manned.org/at>.

- Esegui i comandi inseriti standard input tra 5 minuti (premere `Ctrl + D` dopo aver inserito i comandi):

```
at now + 5 minutes
```

- Esegui un comando passato da standard input alle 10:00 di mattina:

```
echo "{{./mio_script.sh}}" | at 1000
```

- Esegui comandi contenuti in un dato file il prossimo martedì alle 9:30 di sera:

```
at -f {{percorso/del/file}} 9:30 PM Tue
```

# atom

Un editor di testo cross-platform personalizzabile.

I plugin sono gestiti da **apm**.

Maggiori informazioni: <https://atom.io/>.

- Apri un file o una directory:

```
atom {{percorso/del/file_o_directory}}
```

- Apri un file o una directory in una nuova finestra:

```
atom -n {{percorso/del/file_o_directory}}
```

- Apri un file o una directory in una finestra esistente:

```
atom --add {{percorso/del/file_o_directory}}
```

- Avvia Atom in safe mode (non carica nessun pacchetto):

```
atom --safe
```

- Impedisci ad Atom di creare un nuovo processo in background, lasciandolo in esecuzione nel terminale:

```
atom --foreground
```

# atoum

Un semplice, moderno ed intuitivo framework PHP per unit testing.

Maggiori informazioni: <https://atoum.org>.

- Inizializza un file di configurazione:

```
atoum --init
```

- Esegui tutti i test:

```
atoum
```

- Esegui test utilizzando uno specifico file di configurazione:

```
atoum -c {{percorso/del/file}}
```

- Esegui uno specifico file di test:

```
atoum -f {{percorso/del/file}}
```

- Esegui una specifica directory di test:

```
atoum -d {{percorso/della/directory}}
```

- Esegui tutti i test sotto uno specifico namespace:

```
atoum -ns {{namespace}}
```

- Esegui tutti i test con uno specifico tag:

```
atoum -t {{tag}}
```

- Carica un file di bootstrap personalizzato prima di eseguire i test:

```
atoum --bootstrap-file {{percorso/del/file}}
```

# atq

Mostra job programmati dai comandi **at** o **batch**.

Maggiori informazioni: <https://manned.org/atq>.

- Mostra i job programmati per l'utente corrente:

```
atq
```

- Mostra i job della coda 'a' (le code hanno nomi di un singolo carattere):

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

- Mostra i job di tutti gli utenti (da eseguire come super user):

```
sudo atq
```

# atrm

Rimuovi job programmati dai comandi **at** o **batch**.

Per trovare i numeri dei job utilizzare **atq**.

Maggiori informazioni: <https://manned.org/atrm>.

- Rimuovi il job numero 10:

```
atrm {{10}}
```

- Rimuovi più job, separati da spazi:

```
atrm {{15}} {{17}} {{22}}
```

# autoflake

Uno strumento per rimuovere import e variabili inutilizzati da codice Python.

Maggiori informazioni: <https://github.com/myint/autoflake>.

- Rimuovi le variabili inutilizzate da un file e mostra la differenza:

```
autoflake --remove-unused-variables {{percorso/del/file.py}}
```

- Rimuovi gli import inutilizzati da multipli file mostrando le differenze:

```
autoflake --remove-all-unused-imports {{percorso/del/file1.py  
percorso/del/file2.py ...}}
```

- Rimuovi le variabili inutilizzate da un file, sovrascrivendolo:

```
autoflake --remove-unused-variables --in-place {{percorso/  
del/file.py}}
```

- Rimuovi le variabili inutilizzate da tutti i file in una directory, ricorsivamente e sovrascrivendoli:

```
autoflake --remove-unused-variables --in-place --recursive  
{{percorso/della/directory}}
```



# autojump

Salta velocemente tra le directory che usi più spesso.

Alias come `j` o `jc` sono disponibili per una maggiore velocità di scrittura.

Maggiori informazioni: <https://github.com/wting/autojump>.

- Muoviti in una directory il cui nome contiene una parola chiave:

```
j {{parola_chiave}}
```

- Salta in una sotto-directory della directory corrente il quale nome contiene una parola chiave:

```
jc {{parola_chiave}}
```

- Apri una directory il cui nome contiene una parola chiave nel gestore file di sistema:

```
jo {{parola_chiave}}
```

- Rimuovi directory non esistenti dal database di autojump:

```
j --purge
```

- Mostra le voci nel database di autojump:

```
j -s
```

# autossh

Esegue, monitora e riavvia connessioni SSH.

Si riconnette automaticamente per tenere attivi tunnel di port forwarding. Accetta tutte le flag di SSH.

Maggiori informazioni: <https://www.harding.motd.ca/autossh>.

- Apri una sessione SSH, riavviandola quando una porta monitorata smette di rispondere:

```
autossh -M {{porta_monitorata}} "{{comando_ssh}}"
```

- Apri una sessione SSH che forwarda una porta locale verso una remota, riavviandola se necessario:

```
autossh -M {{porta_monitorata}} -L  
{{porta_locale}}:localhost:{{porta_remota}} {{utente}}  
@{{host}}
```

- Forka prima di eseguire il comando SSH (si avvia in background) e non aprire una shell remota:

```
autossh -f -M {{porta_monitorata}} -N "{{comando_ssh}}"
```

- Esegui autossh in background, senza una porta da monitorare, utilizzando i keep-alive di SSH ogni 10 secondi per rilevare una disconnessione:

```
autossh -f -M 0 -N -o "ServerAliveInterval 10" -o  
"ServerAliveCountMax 3" "{{comando_ssh}}"
```

- Esegui autossh in background, senza una porta da monitorare, senza una shell remota, uscendo se il port forwarding fallisce:

```
autossh -f -M 0 -N -o "ServerAliveInterval 10" -o  
"ServerAliveCountMax 3" -o ExitOnForwardFailure=yes -L  
{{porta_locale}}:localhost:{{porta_remota}} {{utente}}  
@{{host}}
```

- Esegui autossh in background con output di debug su un file e output verboso di ssh su un altro file:

```
AUTOSSH_DEBUG=1 AUTOSSH_LOGFILE={{file_log}} autossh -f -M  
{{porta_monitorata}} -v -E {{file_log_ssh}} {{comando_ssh}}
```

# avrdude

Driver per il programmatore di microcontrollori Atmel AVR.

Maggiori informazioni: <https://www.nongnu.org/avrdude/>.

- Leggi dal microcontrollore AVR:

```
avrdude -p {{dispositivo_AVR}} -c {{id_programmatore}} -U  
flash:r:{{file.hex}}:i
```

- Scrivi sul microcontrollore AVR:

```
avrdude -p {{dispositivo_AVR}} -c {{id_programmatore}} -U  
flash:w:{{file.hex}}
```

- Elenca dispositivi AVR disponibili:

```
avrdude -p \?
```

- Elenca programmatori AVR disponibili:

```
avrdude -c \?
```

# awk

Un versatile linguaggio di programmazione per elaborare file.

Maggiori informazioni: <https://github.com/onetrueawk/awk>.

- Stampa la quinta colonna (field) in un file di linee separate da spazi:

```
awk '{print $5}' {{nome_file}}
```

- Stampa la seconda colonna delle linee contenenti "qualcosa":

```
awk '/{{qualcosa}}/ {print $2}' {{nome_file}}
```

- Stampa l'ultima colonna di ogni linea di un file, utilizzando la virgola (invece dello spazio) come separatore di colonne:

```
awk -F ',' '{print $NF}' {{nome_file}}
```

- Somma i valori nella prima colonna di un file e stampa il totale:

```
awk '{s+=$1} END {print s}' {{nome_file}}
```

- Somma i valori nella prima colonna e stampali seguiti dal totale:

```
awk '{s+=$1; print $1} END {print "-----"; print s}' {{nome_file}}
```

- Stampa una linea ogni tre iniziando dalla prima:

```
awk 'NR%3==1' {{nome_file}}
```

# aws s3

CLI per AWS S3 - fornisce spazio di archiviazione tramite le interfacce di Amazon Web Services.

Maggiori informazioni: <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/index.html>.

- Mostra file in un bucket:

```
aws s3 ls {{nome_bucket}}
```

- Sincronizza file e directory locali su un bucket:

```
aws s3 sync {{percorso/dei/file}} s3://{{nome_bucket}}
```

- Sincronizza file e directory da un bucket in locale:

```
aws s3 sync s3://{{nome_bucket}} {{percorso/del/
file_o_cartella}}
```

- Sincronizza file e directory escludendo alcuni file o directory:

```
aws s3 sync {{percorso/dei/file}} s3://{{nome_bucket}} --
exclude {{percorso/del/file}} --exclude {{directory}}/*
```

- Rimuovi un file dal bucket:

```
aws s3 rm s3://{{bucket}}/{{percorso/del/file}}
```

- Mostra solo un'anteprima dei cambiamenti:

```
aws s3 {{qualsiasi_comando}} --dryrun
```

# aws

Il tool da linea di comando ufficiale per Amazon Web Services.

Alcuni comandi aggiuntivi, come **s3**, hanno la propria documentazione.

Maggiori informazioni: <https://aws.amazon.com/cli>.

- Lista tutti gli utenti IAM (Identity and Access Management):

```
aws iam list-users
```

- Lista tutte le istanze EC2 per una specifica regione:

```
aws ec2 describe-instances --region {{us-east-1}}
```

- Ricevi un messaggio da una specifica coda SQS:

```
aws sqs receive-message --queue-url {{https://  
queue.amazonaws.com/546123/Test}}
```

- Pubblica un messaggio SNS su uno specifico argomento:

```
aws sns publish --topic-arn {{arn:aws:sns:us-  
east-1:54633:Agomento}} --message "Message"
```

- Mostra la pagina di aiuto per uno specifico comando AWS:

```
aws {{comando}} help
```

# axel

Downloader accelerato.

Supporta HTTP, HTTPS e FTP.

Maggiori informazioni: <https://github.com/axel-download-accelerator/axel>.

- Scarica un file da un URL:

```
axel {{url}}
```

- Scarica specificando il nome del file da creare:

```
axel {{url}} -o {{filename}}
```

- Scarica sfruttando connessioni multiple:

```
axel -n {{numero_connessioni}} {{url}}
```

- Cerca dei mirror:

```
axel -S {{numero_mirror}} {{url}}
```

- Limita la velocità di download (in bytes al secondo):

```
axel -s {{limite_velocità}} {{url}}
```

# az

Strumento ufficiale da linea di comando per Microsoft Azure.

Alcuni comandi aggiuntivi, come **login**, hanno la propria documentazione.

Maggiori informazioni: <https://learn.microsoft.com/cli/azure>.

- Effettua il log in ad Azure:

```
az login
```

- Gestisci la tua iscrizione ad Azure:

```
az account
```

- Elenca tutti i dischi Azure Managed Disks:

```
az disk list
```

- Elenca tutte le virtual machine Azure:

```
az vm list
```

- Gestisci i servizi Azure Kubernetes:

```
az aks
```

- Gestisci le risorse della rete di Azure:

```
az network
```



# b2sum

Calcola checksum BLAKE2.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/b2sum>.

- Calcola il checksum BLAKE2 per un file:

```
b2sum {{file1}}
```

- Calcola checksum BLAKE2 per più file:

```
b2sum {{file1}} {{file2}}
```

- Leggi un file di checksum BLAKE2 e nomi di file e verifica che tutti i file abbiano lo stesso checksum:

```
b2sum -c {{elenco_checksum.b2}}
```

- Calcola il checksum BLAKE2 da standard input:

```
{{comando}} | b2sum
```

# babel

Un transpiler che converte codice JavaScript da sintassi ES6/ES7 ad ES5.

Maggiori informazioni: <https://babeljs.io/>.

- Transpila uno specifico file e stampa il risultato su `stdout`:

```
babel {{percorso/del/file}}
```

- Transpila un file e scrivi il risultato su uno specifico file di output:

```
babel {{percorso/del/file_input}} --out-file {{percorso/del/file_output}}
```

- Transpila un file ogni volta che viene modificato:

```
babel {{percorso/del/file}} --watch
```

- Transpila un'intera directory di file:

```
babel {{percorso/della/directory_input}}
```

- Transpila un'intera directory ignorando specifici file separati da virgola:

```
babel {{percorso/della/directory_input}} --ignore {{file_ignorati}}
```

- Transpila minimizzando il codice JavaScript in output:

```
babel {{percorso/del/file_input}} --minified
```

- Scegli un insieme di preset per formattare l'output:

```
babel {{percorso/del/file_input}} --presets {{preset}}
```

- Mostra tutte le opzioni disponibili:

```
babel --help
```

# badblocks

Cerca blocchi corrotti in un dispositivo.

Alcuni utilizzi di badblocks possono avere esiti non reversibili, come perdita dei dati o anche della tabella delle partizioni di un disco.

Maggiori informazioni: <https://manned.org/badblocks>.

- Cerca blocchi corrotti in un disco utilizzando un test non distruttivo in sola lettura:

```
sudo badblocks {{/dev/sda}}
```

- Cerca blocchi corrotti in un disco non montato con un test non distruttivo in lettura e scrittura:

```
sudo badblocks -n {{/dev/sda}}
```

- Cerca blocchi corrotti in un disco non montato con un test distruttivo in scrittura:

```
sudo badblocks -w {{/dev/sda}}
```

# banner

Stampa il testo fornito per argomento come un grande banner in ASCII art.

Maggiori informazioni: <https://manned.org/banner>.

- Stampa il testo come un grande banner (le virgolette sono opzionali):

```
banner "{{Hello World}}"
```

- Stampa il testo come un banner con una larghezza di 50 caratteri:

```
banner -w 50 "{{Hello World}}"
```

- Leggi testo da `stdin`:

```
banner
```

# base32

Codifica o decodifica file o standard input in Base32 su standard output.

Maggiori informazioni: <https://manned.org/base32>.

- Codifica un file:

```
base32 {{percorso/del/file}}
```

- Avvolgi l'output codificato a una larghezza specifica (0 disabilita l'avvolgimento):

```
base32 {{-w|--wrap}} {{0|76|...}} {{percorso/del/file}}
```

- Decodifica un file:

```
base32 {{-d|--decode}} {{percorso/del/file}}
```

- Codifica da stdin:

```
{{comando}} | base32
```

- Decodifica da stdin:

```
{{comando}} | base32 {{-d|--decode}}
```

# base64

Codifica o decodifica file o standard input in Base64 su standard output.

Maggiori informazioni: <https://manned.org/base64>.

- Codifica un file:

```
base64 {{percorso/del/file}}
```

- Avvolgi l'output codificato a una larghezza specifica (0 disabilita l'avvolgimento):

```
base64 {{-w|--wrap}} {{0|76|...}} {{percorso/del/file}}
```

- Decodifica un file:

```
base64 {{-d|--decode}} {{percorso/del/file}}
```

- Codifica da stdin:

```
{{comando}} | base64
```

- Decodifica da stdin:

```
{{comando}} | base64 {{-d|--decode}}
```

# basename

Restituisce la parte finale un percorso.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/basename>.

- Mostra solo il nome del file da un percorso:

```
basename {{percorso/del/file}}
```

- Mostra solo il nome di un file da un percorso, rimuovendo un suffisso:

```
basename {{percorso/del/file}} {{suffisso}}
```

# bash

Bourne-Again SHell.

Interprete da linea di comando compatibile con **sh**.

Maggiori informazioni: <https://www.gnu.org/software/bash/>.

- Avvia una shell interattiva:

```
bash
```

- Esegui un comando:

```
bash -c "{{comando}}"
```

- Esegui dei comandi da un file:

```
bash {{file.sh}}
```

- Esegui dei comandi da un file, loggando tutti i comandi eseguiti nel terminale:

```
bash -x {{file.sh}}
```

- Esegui comandi da standard input:

```
bash -s
```

- Stampa informazioni sulla versione di Bash (usa `echo $BASH_VERSION` per mostrare solo la versione):

```
bash --version
```



# bashmarks

Salva e salta a directory comunemente utilizzate utilizzando comandi di un carattere.

Maggiori informazioni: <https://github.com/huyng/bashmarks>.

- Elenca i segnalibri disponibili:

```
l
```

- Salva la directory corrente come segnalibro:

```
s {{nome_segnalibro}}
```

- Vai ad una directory salvata:

```
g {{nome_segnalibro}}
```

- Lista i contenuti di una directory salvata:

```
p {{nome_segnalibro}}
```

- Elimina un segnalibro:

```
d {{nome_segnalibro}}
```

# bastet

Clone di Tetris nel Terminale.

Maggiori informazioni: <https://fph.altervista.org/prog/bastet.html>.

- Invia un gioco tetris:

```
bastet
```

- Sposta il pezzo a sinistra o a destra:

```
{{tasto freccia sinistra|destra}}
```

- Ruota in senso orario o inverso:

```
{{Spazio|tasto freccia su}}
```

- Caduta lenta:

```
<tasto freccia giù>
```

- Caduta veloce:

```
<Invio>
```

- Pausa:

```
p
```

- Lasciare il gioco:

```
<Ctrl> + C
```

# bat

Stampa e concatena file.

Un clone di **cat** con syntax highlighting e integrazione Git.

Maggiori informazioni: <https://github.com/sharkdp/bat>.

- Stampa i contenuti di un file su standard output:

```
bat {{file}}
```

- Concatena diversi file in un unico file:

```
bat {{file1}} {{file2}} > {{file_finale}}
```

- Aggiungi il contenuto di diversi file alla fine di un file:

```
bat {{file1}} {{file2}} >> {{file_finale}}
```

- Numera tutte le linee stampate:

```
bat --number {{file}}
```

- Evidenzia la sintassi di un file JSON:

```
bat --language json {{file.json}}
```

- Mostra tutti i linguaggi supportati:

```
bat --list-languages
```

# batch

Esegui comandi nel futuro quando il carico di lavoro del sistema lo permette.

Il servizio atd (o atrun) deve essere attivo per eseguire i comandi.

Maggiori informazioni: <https://manned.org/batch>.

- Esegui i comandi inseriti standard input (premere `Ctrl + D` dopo aver inserito i comandi):

```
batch
```

- Esegui un comando da standard input:

```
echo "{{./mio_script.sh}}" | batch
```

- Esegui comandi contenuti in un dato file:

```
batch -f {{percorso/del/file}}
```

# bc

Calcolatore.

Maggiori informazioni: <https://manned.org/bc>.

- Esegui in modalità interattiva utilizzando la libreria math della standard library:

```
bc -l
```

- Calcola il risultato di un'espressione:

```
bc <<< "(1 + 2) * 2 ^ 2"
```

- Calcola un'espressione forzando il numero di decimali usati a 10:

```
bc <<< "scale=10; 5 / 3"
```

- Calcola un'espressione con seno e coseno utilizzando mathlib:

```
bc -l <<< "s(1) + c(1)"
```

# beanstalkd

Un semplice e generico gestore di code di lavoro.

Maggiori informazioni: <https://beanstalkd.github.io/>.

- Avvia beanstalkd, ascoltando sulla porta 11300:

```
beanstalkd
```

- Avvia beanstalkd ascoltando su porta ed un indirizzo dati:

```
beanstalkd -l {{indirizzo_ip}} -p {{numero_porta}}
```

- Rendi le code di lavoro persistenti salvandole su disco:

```
beanstalkd -b {{percorso/della/directory_persistente}}
```

- Sincronizza con una directory persistente ogni 500 millisecondi:

```
beanstalkd -b {{percorso/della/directory_persistente}} -f  
{{500}}
```

# bedtools

Un coltellino svizzero di strumenti per una vasta gamma di operazioni di analisi genomica.

Usato per intersecare, raggruppare, convertire e contare dati in formato BAM, BED, GFF/GTF, VCF.

Maggiori informazioni: <https://bedtools.readthedocs.io>.

- Interseca il file [a] ed il/i file [b] in base alla sequenza del filamento [s] e salva il risultato in un file specifico:

```
bedtools intersect -a {{percorso/del/file_A}} -b {{percorso/del/file_B1 percorso/del/file_B2 ...}} -s > {{percorso/del/file_output}}
```

- Interseca 2 file in base a una [l]eft [o]uter [j]oin ovvero una unione d'insieme di dati ordinati in colonne che restituisce i dati della tabella di sinistra. Es: riporta ogni proprietà presente nel file1 e NULL dove non c'è sovrapposizione con file2:

```
bedtools intersect -a {{percorso/del/file1}} -b {{percorso/del/file2}} -loj > {{percorso/del/file_output}}
```

- Usa un algoritmo più efficiente per intersecare due file precedentemente ordinati:

```
bedtools intersect -a {{percorso/del/file1}} -b {{percorso/del/file2}} -sorted > {{percorso/del/file_output}}
```

- Seleziona in un file le prime tre colonne e la quinta [c]olonna utilizzando la sesta colonna per ra[g]gruppare i dati al fine di poter calcolare tramite un'[o]perazione di addizione la somma delle colonne 1,2,3 e 5 per ciascun gruppo:

```
bedtools groupby -i {{percorso/del/file}} -c 1-3,5 -g 6 -o sum
```

- Converti un file in [i]nput formattato bam in un file formattato bed:

```
bedtools bamtobed -i {{percorso/del/file.bam}} > {{percorso/del/file.bed}}
```

- Trova per tutte le proprietà presenti nel `file1.bed` la più vicina nel `file2.bed` e aggiunge la loro [d]istanza in una ulteriore colonna al risultato finale (i file in input devono essere ordinati):

```
bedtools closest -a {{percorso/del/file1.bed}} -b {{percorso/del/file2.bed}} -d
```



# behat

Un framework PHP per lo sviluppo guidato dal comportamento.

Maggiori informazioni: <https://behat.org>.

- Inizializza un nuovo progetto Behat:

```
behat --init
```

- Esegui tutti i test:

```
behat
```

- Esegui tutti i test di una specifica suite:

```
behat --suite {{nome_suite}}
```

- Esegui i test con uno specifico formato di output:

```
behat --format {{pretty|progress}}
```

- Esegui i test e scrivi i risultati in un file:

```
behat --out {{percorso/del/file}}
```

- Mostra una lista delle definizioni nelle suite di test:

```
behat --definitions
```

# berks

Gestore di dipendenze per Chef cookbooks.

Maggiori informazioni: <https://docs.chef.io/berkshelf.html>.

- Installa dipendenze cookbook in una repo locale:  
`berks install`
- Aggiorna uno specifico cookbook e le sue dipendenze:  
`berks update {{cookbook}}`
- Carica un cookbook sul server di Chef:  
`berks upload {{cookbook}}`
- Mostra le dipendenze di un cookbook:  
`berks contingent {{cookbook}}`

# bg

Riprende job che sono stati sospesi (e.g. usando **Ctrl + Z**) mettendoli in esecuzione in background.

Maggiori informazioni: <https://manned.org/bg>.

- Riprendi il job sospeso più recentemente ed esegilo in background:

```
bg
```

- Riprendi uno specifico job (usa `jobs -l` per trovare l'ID) ed esegilo in background:

```
bg %{{id_job}}
```

# binwalk

Strumento per l'analisi di file binari.

Maggiori informazioni: <https://github.com/ReFirmLabs/binwalk>.

- Scansiona un file binario:

```
binwalk {{percorso/del/file}}
```

- Estrae file da un binario, specificando la directory di output:

```
binwalk --extract --directory {{directory_di_output}}  
{{percorso/del/file}}
```

- Estrae file in maniera ricorsiva a partire da un binario, limitando la profondità di ricorsione a 2 livelli:

```
binwalk --extract --matryoshka --depth {{2}} {{percorso/del/  
file}}
```

- Estrae file da un binario utilizzando una particolare firma (ad esempio il MIME Type):

```
binwalk --dd '{{png image:png}}' {{percorso/del/file}}
```

- Analizza l'entropia di un binario e salva il grafico con lo stesso filename del binario, con l'estensione .png in fondo:

```
binwalk --entropy --save {{percorso/del/file}}
```

- Combina analisi di entropia, firme e opcode in un unico comando:

```
binwalk --entropy --signature --opcodes {{percorso/del/file}}
```

# bitcoin-cli

Client da linea di comando per interagire con il demone Bitcoin tramite chiamate RPC.

Utilizza la configurazione definita in **bitcoin.conf**.

Maggiori informazioni: [https://en.bitcoin.it/wiki/Running\\_Bitcoin#Command-line\\_arguments](https://en.bitcoin.it/wiki/Running_Bitcoin#Command-line_arguments).

- Invia una transazione ad un dato indirizzo:

```
bitcoin-cli sendtoaddress "{{indirizzo}}" {{importo}}
```

- Genera uno o più blocchi:

```
bitcoin-cli generate {{numero_blocchi}}
```

- Mostra informazioni di alto livello sul portafogli:

```
bitcoin-cli getwalletinfo
```

- Elenca tutti gli output di transazioni precedenti disponibili per finanziare una nuove transazioni:

```
bitcoin-cli listunspent
```

- Esporta le informazioni sul portafogli in un file di testo:

```
bitcoin-cli dumpwallet "{{percorso/del/file}}"
```

# black

Un formattatore automatico di codice Python.

Maggiori informazioni: [https://black.readthedocs.io/en/stable/usage\\_and\\_configuration/the\\_basics.html](https://black.readthedocs.io/en/stable/usage_and_configuration/the_basics.html).

- Auto-formatta un file o un'intera directory:

```
black {{percorso/del/file_o_directory}}
```

- Formatta il codice che gli viene passato come stringa:

```
black -c "{{codice}}"
```

- Verifica se i file necessitano di auto-formattazione senza modificare nulla:

```
black --check {{percorso/del/file_o_directory}}
```

- Mostra i cambiamenti che verrebbero applicati a ciascun file:

```
black --diff {{percorso/del/file_o_directory}}
```

- Auto-formatta un file o una directory senza produrre output:

```
black --quiet {{percorso/del/file_o_directory}}
```

- Auto-formatta un file o una directory senza sostituire gli apici con le doppie virgolette:

```
black --skip-string-normalization {{percorso/del/file_o_directory}}
```

# blackfire

Strumento di profilazione da linea di comando per PHP.

Maggiori informazioni: <https://blackfire.io>.

- Inizializza e configura il client Blackfire:

```
blackfire config
```

- Lancia l'agent Blackfire:

```
blackfire agent
```

- Lancia l'agent Blackfire su uno specifico socket:

```
blackfire agent --socket="{{tcp://127.0.0.1:8307}}"
```

- Lancia il profiler su uno specifico programma:

```
blackfire run {{php percorso/del/file.php}}
```

- Lancia il profiler e raccogli 10 campioni:

```
blackfire --samples 10 run {{php percorso/del/file.php}}
```

- Lancia il profiler e mostra i risultati in output come JSON:

```
blackfire --json run {{php percorso/del/file.php}}
```

- Carica un file del profiler sul servizio web di Blackfire:

```
blackfire upload {{percorso/del/file}}
```

- Mostra lo stato dei profili sul servizio web di Blackfire:

```
blackfire status
```

# blender

Interfaccia da linea di comando per il programma di grafica Blender 3D.

Gli argomenti sono eseguiti nell'ordine in cui sono dati.

Maggiori informazioni: [https://docs.blender.org/manual/en/latest/advanced/command\\_line/arguments.html](https://docs.blender.org/manual/en/latest/advanced/command_line/arguments.html).

- Renderizza tutti i frame di una animazione in background, senza caricare l'interfaccia grafica (l'output è salvato in /tmp):

```
blender --background {{nome_file.blend}} --render-anim
```

- Renderizza un'animazione usando uno specifico pattern, in un percorso relativo (//) al file .blend:

```
blender --background {{nome_file.blend}} --render-output //{{render/frame_###.png}} --render-anim
```

- Renderizza il decimo frame di un'animazione come singola immagine, salvandolo in una directory esistente (percorso assoluto):

```
blender --background {{nome_file.blend}} --render-output {{/percorso/della/directory_output}} --render-frame {{10}}
```

- Renderizza il penultimo frame di un'animazione come immagine JPEG, salvandolo in una directory esistente (path relativa al file):

```
blender --background {{nome_file.blend}} --render-output //{{directory_output}} --render-frame {{JPEG}} --render-frame {{-2}}
```

- Renderizza l'animazione di una specifica scena, dal frame 10 al 500:

```
blender --background {{nome_file.blend}} --scene {{nome_scena}} --frame-start {{10}} --frame-end {{500}} --render-anim
```

- Renderizza un'animazione ad una specifica risoluzione, attraverso l'utilizzo di uno script python:

```
blender --background {{nome_file.blend}} --python-expr '{{import bpy;
```



```
bpy.data.scenes[0].render.resolution_percentage = 25}}' --  
render-anim
```

- Avvia una sessione interattiva di Blender nel terminale con una console python (esegui `import bpy` dopo l'avvio):

```
blender --background --python-console
```

# bmaptool

Crea o copia blockmap intelligentemente (e quindi più velocemente di **cp** o **dd**).

Maggiori informazioni: <https://source.tizen.org/documentation/reference/bmaptool>.

- Crea una blockmap da un file immagine:

```
bmaptool create -o {{blockmap.bmap}} {{sorgente.img}}
```

- Copia un file immagine su sdb:

```
bmaptool copy --bmap {{blockmap.bmap}} {{sorgente.img}} {/dev/sdb}}
```

- Copia un file immagine compresso su sdb:

```
bmaptool copy --bmap {{blockmap.bmap}} {{sorgente.img.gz}} {/dev/sdb}}
```

- Copia un file immagine su sdb senza utilizzare una blockmap:

```
bmaptool copy --nobmap {{sorgente.img}} {/dev/sdb}}
```

# boot

Strumenti di implementazione per il linguaggio di programmazione Clojure.

Maggiori informazioni: <https://github.com/boot-clj/boot>.

- Avvia una sessione REPL con il progetto o autonomamente (standalone):

```
boot repl
```

- Implementa un singolo uberjar:

```
boot jar
```

- Genera lo scheletro di un nuovo progetto basato su un modello di codice esistente:

```
boot --dependencies boot/new new --template  
{{nome_del_modello}} --name {{nome_del_progetto}}
```

- Implementa l'ambiente di sviluppo (se si sta utilizzando il modello di codice boot/new):

```
boot dev
```

- Implementa l'ambiente di produzione (se si sta utilizzando il modello di codice boot/new):

```
boot prod
```

- Mostra la descrizione di uno specifico comando:

```
boot {{task}} --help
```

# borg

Strumento di backup con deduplicazione.

Crea backup locali o remoti che sono montabili come filesystem.

Maggiori informazioni: <https://borgbackup.readthedocs.io/en/stable/usage/general.html>.

- Inizializza una repository (locale):

```
borg init {{/percorso/del/repo_o_directory}}
```

- Esegui il backup di una directory nella repository, creando un archivio chiamato "Lunedì":

```
borg create --progress {{/percorso/del/repo_o_directory}}::{{Lunedì}} {{/percorso/della/directory_sorgente}}
```

- Lista tutti gli archivi in una repository:

```
borg list {{/percorso/del/repo_o_directory}}
```

- Estrai una specifica directory dall'archivio "Lunedì" in una repository remota, escludendo tutti i file .ext:

```
borg extract {{utente}}@{{host}}:{{/percorso/del/repo_o_directory}}:{{Lunedì}} {{percorso/della/directory_destinazione}} --exclude '{{*.ext}}'
```

- Riduci una repository eliminando tutti gli archivi più vecchi di 7 giorni, elencando i cambiamenti:

```
borg prune --keep-within {{7d}} --list {{/percorso/del/repo_o_directory}}
```

- Monta una repository come filesystem FUSE:

```
borg mount {{/percorso/del/repo_o_directory}}:{{Lunedì}} {{/percorso/del/mountpoint}}
```

- Mostra aiuto sul come creare archivi:

```
borg create --help
```

# bosh

Strumento da linea di comando per distribuire e gestire director BOSH.

Maggiori informazioni: <https://bosh.io/docs/cli-v2/>.

- Crea un alias locale per un director:

```
bosh alias-env {{nome_ambiente}} -e {{indirizzo_ip|url}} --  
ca-cert {{certificato_ca}}
```

- Elenca ambienti:

```
bosh environments
```

- Esegui il login al director:

```
bosh login -e {{ambiente}}
```

- Elenca deployment (distribuzioni):

```
bosh -e {{ambiente}} deployments
```

- Elenca le macchine virtuali in un ambiente:

```
bosh -e {{ambiente}} vms -d {{deployment}}
```

- Avvia una sessione SSH in una macchina virtuale:

```
bosh -e {{ambiente}} ssh {{macchina_virtuale}} -d  
{{deployment}}
```

- Carica una stemcell:

```
bosh -e {{ambiente}} upload-stemcell {{file_stemcell|url}}
```

- Mostra la configurazione cloud attuale:

```
bosh -e {{ambiente}} cloud-config
```

# bower

Un manager di pacchetti ottimizzato per sviluppo web front-end.

Un pacchetto può essere una abbreviazione utente/repo GitHub, un endpoint Git, un URL o un pacchetto registrato.

Maggiori informazioni: <https://bower.io/>.

- Installa le dipendenze di un progetto, listate nel suo file `bower.json`:

```
bower install
```

- Installa pacchetti nella directory `bower_components`:

```
bower install {{pacchetto}} {{pacchetto}}
```

- Disinstalla pacchetti localmente rimuovendolo dalla directory `bower_components`:

```
bower uninstall {{pacchetto}} {{pacchetto}}
```

- Elenca pacchetti locali e possibili aggiornamenti:

```
bower list
```

- Crea un file `bower.json` per i tuoi pacchetti:

```
bower init
```

- Installa una specifica versione di una dipendenza ed aggiungila al file `bower.json`:

```
bower install {{nome_locale}}={{pacchetto}}#{{versione}} --save
```

- Mostra aiuto per un comando di bower:

```
bower help {{comando}}
```

# box

Una applicazione PHP per creare e gestire Phars.

Maggiori informazioni: <https://github.com/box-project/box>.

- Crea un nuovo file Phar:

```
box compile
```

- Crea un nuovo file Phar usando uno specifico file di configurazione:

```
box compile -c {{percorso/della/configurazione}}
```

- Mostra informazioni sulla estensione PHP PHAR:

```
box info
```

- Mostra informazioni su di uno specifico file Phar:

```
box info {{percorso/del/file_phar}}
```

- Valida il primo file di configurazione trovato nella directory corrente:

```
box validate
```

- Verifica la firma di uno specifico file Phar:

```
box verify {{percorso/del/file_phar}}
```

- Mostra tutti i comandi ed opzioni disponibili:

```
box help
```

# brew bundle

Bundler per Homebrew, Homebrew Cask e per il Mac App Store.

Maggiori informazioni: <https://github.com/Homebrew/homebrew-bundle>.

- Installa un pacchetto da un Brewfile nel percorso corrente:

```
brew bundle
```

- Installa pacchetti da un Brewfile specifico in un percorso specifico:

```
brew bundle --file {{percorso/del/file}}
```

- Crea un Brewfile con tutti i pacchetti installati:

```
brew bundle dump
```

- Disinstalla tutti i pacchetti non specificati nel Brewfile:

```
brew bundle cleanup --force
```

- Controlla se c'è qualcosa da installare o da aggiornare nel Brewfile:

```
brew bundle check
```

- Mostra una lista di tutte le righe presenti nel Brewfile:

```
brew bundle list --all
```



# brew

Gestore di pacchetti per macOS.

Maggiori informazioni: <https://docs.brew.sh/Manpage>.

- Installa l'ultima versione stabile di una formula (usa `--devel` per le versioni in sviluppo):

```
brew install {{formula}}
```

- Mostra tutte le formule installate:

```
brew list
```

- Aggiorna una formula installata (se non viene fornito il nome di nessuna formula, tutte le formule installate verranno aggiornate):

```
brew upgrade {{formula}}
```

- Trova la versione più aggiornata di Homebrew e di tutte le formule da GitHub:

```
brew update
```

- Cerca formule e cask:

```
brew search {{testo}}
```

- Mostra le informazioni su una specifica formula (versione, percorso di installazione, dipendenze, ecc...):

```
brew info {{formula}}
```

- Verifica se la versione installata di Homebrew presenta dei problemi:

```
brew doctor
```

# browser-sync

Avvia un web-server locale che si aggiorna al cambiamento dei file.

Maggiori informazioni: <https://browsersync.io/docs/command-line>.

- Avvia un server da una specifica directory:

```
browser-sync start --server {{percorso/della/directory}} --files {{percorso/della/directory}}
```

- Avvia un server da una directory locale, monitorando tutti i file CSS:

```
browser-sync start --server --files '{{percorso/della/directory/*.css}}'
```

- Crea un file di configurazione:

```
browser-sync init
```

- Avvia browser-sync da un file di configurazione:

```
browser-sync start --config {{file_di_configurazione}}
```

# bundle

Gestore di dipendenze per il linguaggio di programmazione Ruby.

Maggiori informazioni: <https://bundler.io/man/bundle.1.html>.

- Installa tutte le gem definite nel gemfile della directory corrente:

```
bundle install
```

- Aggiorna tutte le gem secondo le regole definite nel gemfile e genera un `gemfile.lock`:

```
bundle update
```

- Aggiorna una specifica gem definita nel gemfile:

```
bundle update --source {{nome_gem}}
```

- Crea un scheletro per una nuova gem:

```
bundle gem {{nome_gem}}
```

# bup

Sistema di backup basato sul formato dei packfile Git, fornendo salvataggi incrementali veloci e deduplicazione globale.

Maggiori informazioni: <https://github.com/bup/bup>.

- Inizializza una repository di backup nella directory locale specificata:

```
bup -d {{percorso/della/repository}} init
```

- Prepara una certa directory prima di fare un backup:

```
bup -d {{percorso/della/repository}} index {{percorso/della/directory}}
```

- Esegui il backup di una directory in una repository locale:

```
bup -d {{percorso/della/repository}} save -n {{nome_backup}}  
{{percorso/della/directory}}
```

- Elenca i di backup attualmente nella repository:

```
bup -d {{percorso/della/repository}} ls
```

- Ripristina uno specifico backup in una determinata directory locale:

```
bup -d {{percorso/della/repository}} restore -C {{percorso/della/destinazione}} {{nome_backup}}
```

# bw

CLI per accedere e gestire vault Bitwarden.

Maggiori informazioni: <https://help.bitwarden.com/article/cli/>.

- Esegui il login ad un account Bitwarden:

```
bw login
```

- Esegui il logout da un account Bitwarden:

```
bw logout
```

- Cerca e mostra oggetti in un vault Bitwarden:

```
bw list items --search {{github}}
```

- Mostra un particolare oggetto contenuto in un vault Bitwarden:

```
bw get item {{github}}
```

- Crea una directory in un vault bitwarden:

```
{{echo -n '{"name":"Nome directory"}' | base64}} | bw create  
folder
```

# bzip2

Compressore di file a blocchi ordinati.

Maggiori informazioni: <https://manned.org/bzip2>.

- Comprimi un file:

```
bzip2 {{percorso/del/file}}
```

- Decomprimi un file:

```
bzip2 -d {{percorso/del/file_compresso.bz2}}
```

- Decomprimi un file e mostrane il contenuto su standard output:

```
bzip2 -dc {{percorso/del/file_compresso.bz2}}
```

# c99

Compila programmi C secondo lo standard ISO C.

Maggiori informazioni: <https://manned.org/c99>.

- Compila file sorgente/i e crea un eseguibile:

```
c99 {{file.c}}
```

- Compila file sorgente/i e crea un eseguibile con un nome personalizzato:

```
c99 -o {{nome_eseguibile}} {{file.c}}
```

- Compila file sorgente/i e crea file oggetto:

```
c99 -c {{file.c}}
```

- Compila file sorgente/i linkando con file oggetto e crea un eseguibile:

```
c99 {{file.c}} {{file.o}}
```

# cabal

Interfaccia da linea di comando per l'infrastruttura di compilazione di Haskell (Cabal).

Gestisce progetti Haskell e pacchetti Cabal dal repository di pacchetti Hackage.

Maggiori informazioni: <https://cabal.readthedocs.io/en/latest/getting-started.html>.

- Cerca ed elenca pacchetti da Hackage:

```
cabal list {{termine_di_ricerca}}
```

- Mostra informazioni su di un pacchetto:

```
cabal info {{nome_pacchetto}}
```

- Scarica ed installa un pacchetto:

```
cabal install {{nome_pacchetto}}
```

- Crea un nuovo progetto Haskell nella directory corrente:

```
cabal init
```

- Compila il progetto nella directory corrente:

```
cabal build
```

- Esegui i test del progetto nella directory corrente:

```
cabal test
```



# cake

Strumento da linea di comando per il framework CakePHP.

Maggiori informazioni: <https://cakephp.org>.

- Mostra informazioni sull'attuale app ed i comandi disponibili:

```
cake
```

- Elenca le rotte disponibili:

```
cake routes
```

- Pulisci le cache di configurazione:

```
cake cache clear_all
```

- Costruisci la cache dei metadati:

```
cake schema_cache build --connection {{connessione}}
```

- Pulisci la cache dei metadati:

```
cake schema_cache clear
```

- Pulisci una tabella di cache:

```
cake schema_cache clear {{nome_tabella}}
```

- Avvia un web server di sviluppo (porta predefinita 8765):

```
cake server
```

- Avvia una shell REPL interattiva:

```
cake console
```

# calibre-server

Un'applicazione server che può essere usata per distribuire e-book in una rete.

Gli e-book devono prima essere importati nella libreria usando la GUI o calibredb.

Parte del manager di e-book Calibre.

Maggiori informazioni: <https://manual.calibre-ebook.com/generated/en/calibre-server.html>.

- Avvia un server per distribuire e-book. Accesso a <http://localhost:8080>:

```
calibre-server
```

- Avvia il server su una specifica porta. Accesso a <http://localhost:porta>:

```
calibre-server --port {{porta}}
```

- Proteggi il server con username e password:

```
calibre-server --username {{username}} --password {{password}}
```

# calibredb

Strumento per gestire il tuo database di e-book.

Parte del manager di e-book Calibre.

Maggiori informazioni: <https://manual.calibre-ebook.com/generated/en/calibredb.html>.

- Elenca gli e-book nella libreria con informazioni aggiuntive:

```
calibredb list
```

- Cerca tra gli e-book mostrando informazioni aggiuntive:

```
calibredb list --search {{termine_di_ricerca}}
```

- Cerca mostrando solamente gli ID degli e-book:

```
calibredb search {{termine_di_ricerca}}
```

- Aggiungi uno o più e-book alla libreria:

```
calibredb add {{percorso/del/file1 percorso/del/file2 ...}}
```

- Aggiungere ricorsivamente tutti gli e-book in una directory alla libreria:

```
calibredb add {{-r|--recurse}} {{percorso/della/directory}}
```

- Rimuovi uno o più e-book dalla libreria. Sono necessari gli ID (vedi sopra):

```
calibredb remove {{id1 id2 ...}}
```

# cargo

Gestore di pacchetti di Rust.

Gestisce progetti Rust ed i moduli dai quali sono dipendenti (detti crate).

Alcuni comandi aggiuntivi, come **build**, hanno la propria documentazione.

Maggiori informazioni: <https://doc.rust-lang.org/cargo>.

- Cerca una crate:

```
cargo search {{termine_di_ricerca}}
```

- Installa una crate:

```
cargo install {{nome_crate}}
```

- Elenca crate installate:

```
cargo install --list
```

- Crea un nuovo progetto Rust binario o di libreria nella directory corrente:

```
cargo init --{{bin|lib}}
```

- Crea un nuovo progetto Rust binario o di libreria nella directory specificata:

```
cargo new {{percorso/della/directory}} --{{bin|lib}}
```

- Builda il progetto Rust nella directory corrente:

```
cargo build
```

- Builda utilizzando più job (thread) paralleli:

```
cargo build --jobs {{numero_job}}
```

# case

Esegui branch diversi in base al valore di un'espressione.

Maggiori informazioni: <https://www.gnu.org/software/bash/manual/bash.html#index-case>.

- Esegui il match di una variabile su diverse stringhe per decidere che comando eseguire:

```
case {{${metrica}} in {{parole}}) {{wc -w README}}; ;;  
{{linee}}) {{wc -l README}}; ;; esac
```

- Combina pattern con |, utilizzando \* come pattern di fallback:

```
case {{${metrica}} in {[pP]|parole}}) {{wc -w README}}; ;;  
{{[lL]|linee}}) {{wc -l README}}; ;; *) {{echo "cosa?"}}; ;;  
esac
```

# cat

Stampa e concatena file.

Maggiori informazioni: <https://manned.org/cat.1posix>.

- Stampa i contenuti di un file su standard output:

```
cat {{file}}
```

- Concatena più file in un unico file:

```
cat {{file1}} {{file2}} > {{file_finale}}
```

- Aggiungi il contenuto di diversi file alla fine di un file:

```
cat {{file1}} {{file2}} >> {{file_finale}}
```

- Numera tutte le linee stampate:

```
cat -n {{file}}
```

# cd

Cambia la directory corrente.

Maggiori informazioni: <https://manned.org/cd>.

- Vai alla directory specificata:

```
cd {{percorso/della/directory}}
```

- Vai alla directory home dell'utente corrente:

```
cd
```

- Vai alla directory madre della corrente:

```
cd ..
```

- Vai alla directory precedentemente scelta:

```
cd -
```

# chgrp

Cambia il gruppo proprietario di file e directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/chgrp>.

- Cambia il gruppo proprietario di un file/directory:

```
chgrp {{gruppo}} {{percorso/del/file}}
```

- Cambia ricorsivamente il gruppo proprietario di una directory e dei suoi contenuti:

```
chgrp -R {{gruppo}} {{percorso/della/directory}}
```

- Cambia il gruppo proprietario di un link simbolico:

```
chgrp -h {{gruppo}} {{percorso/del/symlink}}
```

- Cambia il gruppo proprietario di un file/directory rendendolo uguale a quello di un altro file di riferimento:

```
chgrp --reference {{percorso/del/file_riferimento}}  
{{percorso/del/file}}
```



# chmod

Cambia i permessi di accesso di file o directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/chmod>.

- Dai il permesso di eseguire (x) un file al suo proprietario (u):

```
chmod u+x {{percorso/del/file}}
```

- Dai permessi di lettura (r) e scrittura (w) per un file/directory al suo proprietario:

```
chmod u+rw {{percorso/del/file_o_directory}}
```

- Rimuovi i permessi di esecuzione al [g]ruppo proprietario del file:

```
chmod g-x {{percorso/del/file}}
```

- Dai a tutti gli utenti (a) diritti di lettura ed esecuzione:

```
chmod a+rx {{percorso/del/file}}
```

- Dai ad altri utenti (non nel gruppo proprietario) gli stessi diritti del gruppo:

```
chmod o=g {{percorso/del/file}}
```

- Cambia permessi ricorsivamente dando al [g]ruppo e agli altri utenti (o) diritto di scrittura:

```
chmod -R g+w,o+w {{percorso/della/directory}}
```

# chown

Cambia utente e gruppo proprietario di file e directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/chown>.

- Cambia l'utente proprietario di un file/directory:

```
chown {{utente}} {{percorso/del/file_o_directory}}
```

- Cambia utente e gruppo proprietari di un file/directory:

```
chown {{utente}}:{{gruppo}} {{percorso/del/file_o_directory}}
```

- Cambia ricorsivamente il proprietario di una directory ed i suoi contenuti:

```
chown -R {{utente}} {{percorso/della/directory}}
```

- Cambia il proprietario di un link simbolico:

```
chown -h {{utente}} {{percorso/del/link_simbolico}}
```

- Cambia il proprietario di un file/directory rendendolo uguale a quello di un altro file di riferimento:

```
chown --reference={{percorso/del/file_riferimento}}  
{{percorso/del/file_o_directory}}
```

# chromium

Browser web open-source di Google.

Maggiori informazioni: <https://www.chromium.org/developers/how-tos/run-chromium-with-flags/>.

- Apri un file:

```
chromium {{percorso/del/file.html}}
```

- Apri un URL:

```
chromium {{esempio.com}}
```

- Apri in modalità incognito:

```
chromium --incognito {{esempio.com}}
```

- Apri in una nuova finestra:

```
chromium --new-window {{esempio.com}}
```

- Apri in modalità app (senza barre degli strumenti, URL, bottoni, etc.):

```
chromium --app='{{https://esempio.com}}'
```

- Usa un server proxy:

```
chromium --proxy-server="{{socks5://hostname:66}}"  
{{esempio.com}}
```

# chroot

Esegui un comando o una shell interattiva con una speciale directory root.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/chroot>.

- Esegui un comando con una diversa directory root:

```
chroot {{/percorso/della/nuova/root}} {{comando}}
```

- Specifica utente e gruppo (ID o nome) da usare:

```
chroot --userspec={{utente:gruppo}}
```

# chsh

Cambia shell di login dell'utente.

Maggiori informazioni: <https://manned.org/chsh>.

- Cambia shell:

```
chsh -s {{percorso/dell/eseguibile_della_shell}}  
{{nome_utente}}
```

# cksum

Calcola checksum CRC e conta i byte di un file.

Nota: in vecchi sistemi UNIX l'implementazione di CRC potrebbe essere diversa.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/cksum>.

- Calcola e mostra un checksum di 32 bit, dimensione in byte e nome del file:

```
cksum {{percorso/del/file}}
```

# clamscan

Scanner antivirus da linea di comando.

Maggiori informazioni: <https://docs.clamav.net/manual/Usage/Scanning.html#clamscan>.

- Analizza un file cercando vulnerabilità:

```
clamscan {{percorso/del/file}}
```

- Analizza ricorsivamente tutti i file in una specifica directory:

```
clamscan -r {{percorso/della/directory}}
```

- Analizza dati da standard input:

```
{{comando}} | clamscan -
```

- Specifica un file o directory di file da usare come database virus:

```
clamscan --database {{percorso/del/file_o_directory}}
```

- Analizza la directory corrente e mostra in output solo i file infetti:

```
clamscan --infected
```

- Scrivi il risultato di uno scan in un file di log:

```
clamscan --log {{percorso/del/file_log}}
```

- Sposta i file infetti in una specifica directory:

```
clamscan --move {{percorso/della/directory_quarantena}}
```

- Elimina i file infetti:

```
clamscan --remove yes
```

# clang-cpp

Questo comando è un alias per **clang++**.

- Consulta la documentazione del comando originale:

```
tldr clang++
```



# clang

Compilatore per sorgenti C, C++ ed Objective-C. Può essere usato come alternativa a GCC.

Maggiori informazioni: <https://clang.llvm.org/docs/ClangCommandLineReference.html>.

- Compila un file sorgente in un binario eseguibile:

```
clang {{sorgente_input.c}} -o {{eseguibile_output}}
```

- Attiva l'output di tutti gli errori ed i warning:

```
clang {{sorgente_input.c}} -Wall -o {{eseguibile_output}}
```

- Includi librerie contenute in un percorso differente da quello del file di sorgente:

```
clang {{sorgente_input.c}} -o {{eseguibile_output}} -  
I{{percorso_header}} -L{{percorso_librerie}} -  
l{{nome_libreria}}
```

- Compila codice sorgente in IR LLVM (Intermediate Representation):

```
clang -S -emit-llvm {{file.c}} -o {{file.ll}}
```

# clear

Pulisce lo schermo del terminale.

Maggiori informazioni: <https://manned.org/clear>.

- Pulisci lo schermo (equivalente a Control+L se si utilizza la shell Bash):

```
clear
```

# clementine

Un moderno player e gestore di librerie musicali.

Maggiori informazioni: <https://github.com/clementine-player/Clementine/wiki>.

- Avvia l'interfaccia grafica oppure lo mette in evidenza:

```
clementine
```

- Avvia la riproduzione di un file musicale:

```
clementine {{url|percorso/del/file/music.ext}}
```

- Pausa o riprende la riproduzione:

```
clementine --play-pause
```

- Ferma la riproduzione:

```
clementine --stop
```

- Passa alla traccia successiva o precedente:

```
clementine --{{next|previous}}
```

- Crea una nuova playlist con uno o più file musicali oppure URL:

```
clementine --create {{url1 url2 ... | percorso/del/file/  
music1.ext percorso/del/file/music2.ext ...}}
```

- Carica una playlist:

```
clementine --load {{percorso/del/file/playlist.ext}}
```

- Riproduce una specifica traccia nella playlist caricata:

```
clementine --play-track {{5}}
```

# cloc

Conta e calcola le differenze di linee di codice sorgente e commenti.

Maggiori informazioni: <https://github.com/AlDanial/cloc>.

- Conta tutte le linee di codice in una directory:

```
cloc {{percorso/della/directory}}
```

- Conta tutte le linee di codice in una directory, mostrando una barra di avanzamento durante l'operazione:

```
cloc --progress=1 {{percorso/della/directory}}
```

- Compara i file sorgente in 2 diverse directory e conta le differenze tra di essi:

```
cloc --diff {{percorso/della/directory1}} {{percorso/della/directory2}}
```

# clockwork-cli

Una interfaccia da linea di comando per il framework PHP Clockwork.

Maggiori informazioni: <https://github.com/ptrofimov/clockwork-cli>.

- Monitora i log di Clockwork per il progetto corrente:

```
clockwork-cli
```

- Monitora i log di Clockwork per uno specifico progetto:

```
clockwork-cli {{percorso/della/directory_progetto}}
```

- Monitora i log di Clockwork per più progetti:

```
clockwork-cli {{percorso/della/directory1 percorso/della/directory2 ...}}
```

# clojure

Questo comando è un alias per `clj`.

- Consulta la documentazione del comando originale:

```
tldr clj
```

# cmake

Generatore di ambienti di compilazione multiplatforma.

Genera Makefile, progetti Visual Studio o altro, in base al sistema operativo.

Maggiori informazioni: <https://cmake.org/cmake/help/latest/manual/cmake.1.html>.

- Genera un Makefile ed usalo per compilare un progetto nella stessa directory dei sorgenti:

```
cmake && make
```

- Genera un makefile ed usalo per compilare un progetto in una directory "build" separata (out-of-source build):

```
cmake -H. -B{{build}} && make -C {{build}}
```

- Esegui cmake in modalità interattiva (chiederà i valori di ogni variabile invece di usare i predefiniti):

```
cmake -i
```

# cmark

Converte testo CommonMark Markdown in altri formati.

Maggiori informazioni: <https://github.com/commonmark/cmark>.

- Converti un file Markdown in HTML:

```
cmark --to html {{file.md}}
```

- Converti in LaTeX da standard input:

```
cmark --to latex
```

- Converti apici semplici in apici intelligenti:

```
cmark --smart --to html {{file.md}}
```

- Converti validando i caratteri UTF-8:

```
cmark --validate-utf8 {{file.md}}
```



# cmatrix

Genera lettere che cadono, come in Matrix.

Maggiori informazioni: <https://github.com/abishekvashok/cmatrix>.

- Attivare lo scorrimento asincrono:

```
cmatrix -a
```

- Rendi le lettere rosse:

```
cmatrix -C {{red}}
```

- Cambia colore a arcobaleno:

```
cmatrix -r
```

- Imposta il dilazione di aggiornamento a 100 centisecondi (10 secondi):

```
cmatrix -u 100
```

# cmp

Compara due file.

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

- Trova l'indice del primo byte e della prima riga differente tra due file:

```
cmp {{file1}} {{file2}}
```

- Trova ogni coppia di byte differenti ed il relativo indice:

```
cmp -l {{file1}} {{file2}}
```

# code

Visual Studio Code.

Maggiori informazioni: <https://github.com/microsoft/vscode>.

- Apri VS Code:

```
code
```

- Apri la directory corrente in VS Code:

```
code .
```

- Apri un file o una directory in VS Code:

```
code {{percorso/del/file_o_directory}}
```

- Apri un file o una directory nella finestra attualmente aperta di VS Code:

```
code --reuse-window {{percorso/del/file_o_directory}}
```

- Confronta due file in VS Code:

```
code -d {{file1}} {{file2}}
```

# coffee

Esegui script CoffeeScript o compilali in JavaScript.

Maggiori informazioni: <https://coffeescript.org#cli>.

- Esegui uno script:

```
coffee {{percorso/del/file.coffee}}
```

- Compila in JavaScript e salva lo script con lo stesso nome:

```
coffee --compile {{percorso/del/file.coffee}}
```

- Compila in JavaScript e salva lo script specificandone il nome:

```
coffee --compile {{percorso/del/file.coffee}} --output  
{{percorso/del/file.js}}
```

- Esegui una console REPL interattiva:

```
coffee --interactive
```

- Monitora cambiamenti in uno script rieseguendolo ogni volta:

```
coffee --watch {{percorso/del/file.coffee}}
```

# cola

Questo comando è un alias per **git-cola**.

- Consulta la documentazione del comando originale:

`tldr git-cola`

# colordiff

Un'utilità per aggiungere colore all'output diff.

Colordiff è un wrapper scritto in Perl per **diff** e produce lo stesso output, ma con una bella evidenziazione della sintassi. I colori possono essere personalizzati.

Maggiori informazioni: <https://github.com/kimmel/colordiff>.

- Analisi di due file:

```
colordiff {{file1}} {{file2}}
```

- Output in due colonne:

```
colordiff -y {{file1}} {{file2}}
```

- Ignora differenze di maiuscole in file:

```
colordiff -i {{file1}} {{file2}}
```

- Notifica se file identici:

```
colordiff -s {{file1}} {{file2}}
```

- Ignora spazio vuoto (white space):

```
colordiff -w {{file1}} {{file2}}
```

# column

Formatta standard input o un file in più colonne.

Le colonne sono riempite prima delle righe; il separatore predefinito è lo spazio.

Maggiori informazioni: <https://manned.org/column>.

- Formatta l'output per uno schermo largo 30 caratteri:

```
printf "intestazione1 intestazione2\nbar foo\n" | column --  
output-width {{30}}
```

- Separa colonne ed allinea automaticamente in un formato tabulare:

```
printf "intestazione1 intestazione2\nbar foo\n" | column --  
table
```

- Specifica un diverso separatore di colonna (e.g. "," per CSV) (il predefinito è lo spazio):

```
printf "intestazione1 intestazione2\nbar foo\n" | column --  
table --separator {{,}}
```

- Riempi le righe prima delle colonne:

```
printf "intestazione1\nbar\nfoobar\n" | column --output-width  
{{30}} --fillrows
```

# comm

Seleziona o ignora linee comuni a due file. Entrambi i file devono essere ordinati.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/comm>.

- Produci tre colonne separate da tab: linee solo nel primo file, linee solo nel secondo file, e linee comuni ad entrambi:

```
comm {{file1}} {{file2}}
```

- Stampa solo le linee comune ad entrambi i file:

```
comm -12 {{file1}} {{file2}}
```

- Stampa solo le lin comuni ad entrambi i file, leggendone uno da standard input:

```
cat {{file1}} | comm -12 - {{file2}}
```

- Filtra le linee trovate solo nel primo file, salvando il risultato in un terzo file:

```
comm -23 {{file1}} {{file2}} > {{file3}}
```

- Filtra le linee trovate solo nel secondo file, con due file che non sono ordinati:

```
comm -13 <(sort {{file1}}) <(sort {{file2}})
```



# command

Command forza la shell ad eseguire programmi ignorando qualsiasi funzione, builtin e alias con lo stesso nome.

Maggiori informazioni: <https://manned.org/command>.

- Esegui il comando `ls` letteralmente, anche se esiste un alias `ls`:

```
command {{ls}}
```

# complete

Fornisce autocompletamento per argomenti dei comandi della shell.

Maggiori informazioni: [https://www.gnu.org/software/bash/manual/html\\_node/Programmable-Completion-Builtins.html](https://www.gnu.org/software/bash/manual/html_node/Programmable-Completion-Builtins.html).

- Applica ad un comando una funzione per gestirne l'autocompletamento:  
`complete -F {{funzione}} {{comando}}`
- Applica ad un comando un altro comando per gestirne l'autocompletamento:  
`complete -C {{comando_per_autocompletamento}} {{comando}}`
- Applica l'autocompletamento senza aggiungere uno spazio dopo la parola completata:  
`complete -o nospace -F {{function}} {{comando}}`

# composer

Un gestore di dipendenze a pacchetti per progetti PHP.

Maggiori informazioni: <https://getcomposer.org/>.

- Aggiungi un pacchetto come dipendenza per questo progetto, aggiungendolo a `composer.json`:

```
composer require {{user/nome_pacchetto}}
```

- Installa tutte le dipendenze listate nel `composer.json` di questo progetto:

```
composer install
```

- Disinstalla un pacchetto da questo progetto, rimuovendolo come dipendenza da `composer.json`:

```
composer remove {{user/nome_pacchetto}}
```

- Aggiorna tutte le dipendenze nel `composer.json` di questo progetto:

```
composer update
```

- Aggiorna composer alla versione più recente:

```
composer self-update
```

# conda

Gestione pacchetti, dipendenze ed ambiente per qualsiasi linguaggio di programmazione.

Alcuni comandi aggiuntivi, come **create**, hanno la propria documentazione.

Maggiori informazioni: <https://github.com/conda/conda>.

- Crea un nuovo ambiente, installandovi alcuni pacchetti:

```
conda create --name {{nome_ambiente}} {{python=3.9  
matplotlib}}
```

- Elenca tutti gli ambienti:

```
conda info --envs
```

- Attiva un ambiente:

```
conda activate {{nome_ambiente}}
```

- Disattiva un ambiente:

```
conda deactivate
```

- Elimina un ambiente rimuovendo anche tutti i pacchetti:

```
conda remove --name {{nome_ambiente}} --all
```

- Installa pacchetti nell'ambiente corrente:

```
conda install {{python=3.4 numpy}}
```

- Elenca i pacchetti attualmente installati nell'ambiente corrente:

```
conda list
```

- Elimina pacchetti inutilizzati e cache:

```
conda clean --all
```

# consul-kv

Rete distribuita per gestire e configurare servizi tramite database chiave-valore.

Maggiori informazioni: <https://learn.hashicorp.com/consul/getting-started/kv>.

- Leggi il valore di una chiave da un database chiave-valore:

```
consul kv get {{chiave}}
```

- Memorizza una nuova coppia chiave-valore:

```
consul kv put {{chiave}} {{valore}}
```

- Elimina una coppia chiave-valore:

```
consul kv delete {{chiave}}
```

# consul

Rete distribuita per gestire e configurare servizi tramite database chiave-valore.

Alcuni comandi aggiuntivi, come **kv**, hanno la propria documentazione.

Maggiori informazioni: <https://www.consul.io/commands>.

- Mostra informazioni di aiuto generali:

```
consul --help
```

- Mostra aiuto per un sottocomando:

```
consul {{sottocomando}} --help
```

- Controlla la versione di Consul:

```
consul --version
```

# convmv

Conversione dei nomi dei file (NON del contenuto) da un encoding ad un altro.

Maggiori informazioni: <https://www.j3e.de/linux/convmv/man/>.

- Controlla la conversione di encoding (non rinomina realmente il file):

```
convmv -f {{encoding_originale}} -t {{encoding_finale}}  
{{file_input}}
```

- Converti l'encoding del nome di un file e rinominalo utilizzando il nuovo encoding:

```
convmv -f {{encoding_originale}} -t {{encoding_finale}} --  
notest {{file_input}}
```

# cordova

Applicazioni mobile con HTML, CSS e JavaScript.

Maggiori informazioni: <https://cordova.apache.org/docs/en/latest/guide/cli/>.

- Crea un progetto Cordova:

```
cordova create {{percorso}} {{nome_pacchetto}}  
{{nome_progetto}}
```

- Mostra lo stato della workspace corrente:

```
cordova info
```

- Aggiungi una piattaforma Cordova:

```
cordova platform add {{piattaforma}}
```

- Rimuovi una piattaforma Cordova:

```
cordova platform remove {{piattaforma}}
```

- Aggiungi un plugin Cordova:

```
cordova plugin add {{id_plugin}}
```

- Rimuovi un plugin Cordova:

```
cordova plugin remove {{id_plugin}}
```



# couchdb

Interfaccia da linea di comando per il server di database Apache CouchDB.

Maggiori informazioni: <https://couchdb.apache.org>.

- Avvia couchdb:

```
couchdb
```

- Avvia una shell couchdb interattiva:

```
couchdb -i
```

- Avvia couchdb come processo in background:

```
couchdb -b
```

- Termina il processo in background (nota: si riavvierà da solo se necessario):

```
couchdb -k
```

- Disattiva il processo in background:

```
couchdb -d
```

# cowsay

Genera un personaggio ASCII (di default una mucca) che dice o pensa qualcosa.

Maggiori informazioni: <https://github.com/tlalnpgge/rank-amateur-cowsay>.

- Stampa una mucca ASCII che dice "Hello world":

```
cowsay "Hello world"
```

- Usa il testo da standard input per il fumetto:

```
echo "Ciao" | cowsay
```

- Elenca tutti i personaggi disponibili:

```
cowsay -l
```

- Stampa un drago ASCII che dice "Ciao":

```
cowsay -f dragon "Ciao"
```

- Stampa una mucca ASCII sballata che pensa:

```
cowthink -s "Sono solo una mucca, non un grande pensatore..."
```

# cp

Copia file e directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/cp>.

- Copia un file in un'altra posizione:

```
cp {{percorso/del/file}} {{percorso/della/copia}}
```

- Copia un file in una directory mantenendo il nome:

```
cp {{percorso/del/file}} {{percorso/della/directory}}
```

- Copia una directory ricorsivamente in un'altra posizione:

```
cp -r {{percorso/della/directory}} {{percorso/della/copia}}
```

- Copia una directory ricorsivamente in modo verboso (mostra a schermo ogni file copiato):

```
cp -vr {{percorso/della/directory}} {{percorso/della/copia}}
```

- Copia i contenuti di una directory in una seconda directory:

```
cp -r {{percorso/della/directory/*}} {{percorso/della/  
seconda/directory}}
```

- Copia tutti i file di testo in una seconda directory in modo interattivo (chiede conferma prima di sovrascrivere):

```
cp -i {{*.txt}} {{percorso/della/directory}}
```

# cpio

Copia file da/a archivi.

Supporta i seguenti formati di archivio: cpio binario, vecchio ASCII, nuovo ASCII, crc, HPUX binario, HPUX vecchio ASCII, vecchio tar, e tar POSIX.1.

Maggiori informazioni: <https://www.gnu.org/software/cpio>.

- Accetta una lista di nomi di file da standard input ed aggiungili ad un archivio in formato binario cpio:

```
echo "{{file1}} {{file2}} {{file3}}" | cpio -o >
{{archivio.cpio}}
```

- Copia tutti i file e le directory in una directory ed aggiungili ad un archivio, in modalità verbosa:

```
find {{percorso/della/directory}} | cpio -ov >
{{archivio.cpio}}
```

- Estrai file da un archivio, generando le directory necessarie, in modalità verbosa:

```
cpio -idv < {{archivio.cpio}}
```

# cppcheck

Strumento di analisi statica per codice C/C++.

Piuttosto che sugli errori di sintassi, si concentra su tipi di bug che normalmente non vengono rilevati dai compilatori.

Maggiori informazioni: <https://cppcheck.sourceforge.net>.

- Controlla la directory corrente ricorsivamente, mostrando il progresso a schermo e loggando i messaggi di errore in un file:

```
cppcheck . 2> cppcheck.log
```

- Controlla una determinata directory ricorsivamente, senza stampare informazioni sul progresso:

```
cppcheck --quiet {{percorso/della/directory}}
```

- Controlla un determinato file, specificando quali test eseguire (di default, solo gli errori sono mostrati):

```
cppcheck --enable {{error|warning|style|performance|portability|information|all}} {{percorso/del/file.cpp}}
```

- Elenca i test disponibili:

```
cppcheck --errorlist
```

- Controlla un determinato file, ignorando specifici test:

```
cppcheck --suppress {{id_test1}} --suppress {{it_test2}} {{percorso/del/file.cpp}}
```

- Controlla la directory corrente, fornendo percorsi da includere per file esterni (e.g. librerie esterne):

```
cppcheck -I {{include/directory_1}} -I {{include/directory_2}} .
```

- Controlla un progetto Microsoft Visual Studio (\*.vcxproj) o file solution (\*.sln):

```
cppcheck --project {{percorso/del/progetto.sln}}
```

# cppclean

Trova codice inutilizzato in progetti C++.

Maggiori informazioni: <https://github.com/myint/cppclean>.

- Esegui nella directory di un progetto:

```
cppclean {{percorso/della/directory_progetto}}
```

- Esegui su di un progetto dove gli header sono nella directory "inc1" ed "inc2":

```
cppclean {{percorso/della/directory_progetto}} --include-path  
{{inc1}} --include-path {{inc2}}
```

- Esegui su di uno specifico file `main.cpp`:

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

- Esegui della directory corrente, escludendo la directory "build":

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

# cradle deploy

Gestisci distribuzioni Cradle.

Maggiori informazioni: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#deploy>.

- Distribuisci Cradle su un server:

```
cradle deploy production
```

- Distribuisci assets statici ad Amazon S3:

```
cradle deploy s3
```

- Distribuisci assets statici, inclusa la directory "components" di Yarn:

```
cradle deploy s3 --include-yarn
```

- Distribuisci assets statici, includendo la directory "upload":

```
cradle deploy s3 --include-upload
```

# cradle elastic

Gestisci le istanze Elasticsearch per un'istanza Cradle.

Maggiori informazioni: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#elastic>.

- Svuota l'indice Elasticsearch:

```
cradle elastic flush
```

- Svuota l'indice Elasticsearch per uno specifico pacchetto:

```
cradle elastic flush {{nome_pacchetto}}
```

- Carica lo schema Elasticsearch:

```
cradle elastic map
```

- Carica lo schema Elasticsearch per uno specifico pacchetto:

```
cradle elastic map {{nome_pacchetto}}
```

- Popola gli indici Elasticsearch per tutti i pacchetti:

```
cradle elastic populate
```

- Popola gli indici Elasticsearch per uno specifico pacchetto:

```
cradle elastic populate {{nome_pacchetto}}
```



# cradle install

Installa i componenti del framework Cradle per PHP.

Maggiori informazioni: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#install>.

- Installa i componenti di Cradle (maggiori dettagli verranno richiesti all'utente):

```
cradle install
```

- Sovrascrivi i file forzatamente:

```
cradle install --force
```

- Salta l'esecuzione di migrazioni SQL:

```
cradle install --skip-sql
```

- Salta l'esecuzione di aggiornamenti dei pacchetti:

```
cradle install --skip-versioning
```

- Utilizza specifici dettagli per il database:

```
cradle install -h {{hostname}} -u {{nome_utente}} -p {{password}}
```

# cradle package

Gestisci pacchetti per un'istanza Cradle.

Maggiori informazioni: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#package>.

- Mostra una lista dei pacchetti disponibili:

```
cradle package list
```

- Cerca un pacchetto:

```
cradle package search {{pacchetto}}
```

- Installa un pacchetto da Packagist:

```
cradle package install {{pacchetto}}
```

- Installa una specifica versione di un pacchetto:

```
cradle package install {{pacchetto}} {{versione}}
```

- Aggiorna un pacchetto:

```
cradle package update {{pacchetto}}
```

- Aggiorna un pacchetto ad una specifica versione:

```
cradle package update {{pacchetto}} {{versione}}
```

- Rimuovi uno specifico pacchetto:

```
cradle package remove {{pacchetto}}
```

# cradle sql

Gestisci database SQL di Cradle.

Maggiori informazioni: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#sql>.

- Ricostruisci lo schema del database:

```
cradle sql build
```

- Ricostruisci lo schema del database per uno specifico pacchetto:

```
cradle sql build {{nome_pacchetto}}
```

- Svuota l'intero database:

```
cradle sql flush
```

- Svuota le tabelle del database per uno specifico pacchetto:

```
cradle sql flush {{nome_pacchetto}}
```

- Popola le tabelle per tutti i pacchetti:

```
cradle sql populate
```

- Popola le tabelle per uno specifico pacchetto:

```
cradle sql populate {{nome_pacchetto}}
```

# cradle

Il framework Cradle per PHP.

Vedi **cradle-install**, **cradle-deploy** e altre pagine per maggiori informazioni.

Maggiori informazioni: <https://cradlephp.github.io>.

- Connetti ad un server:

```
cradle connect {{nome_server}}
```

- Esegui un comando Cradle:

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

- Mostra informazioni di aiuto generali:

```
cradle help
```

- MOstra aiuto per uno specifico comando:

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

# crontab

Programma cron job per essere eseguiti a determinati intervalli di tempo per l'utente corrente.

Formato definizione di un job: "(minuto) (ora) (giorno\_del\_mese) (mese) (giorno\_della\_settimana) comando\_da\_eseguire".

Maggiori informazioni: <https://crontab.guru/>.

- Modifica il file crontab per l'utente corrente:

```
crontab -e
```

- Elenca i cron job esistenti per l'utente corrente:

```
crontab -l
```

- Rimuovi tutti i cron job per l'utente corrente:

```
crontab -r
```

- Esempio di un job eseguito alle 10:00 ogni giorno (\* vuol dire qualsiasi valore):

```
0 10 * * * {{comando_da_eseguire}}
```

- Esempio di un job eseguito ogni minuto il 3 Aprile:

```
* * 3 Apr * {{comando_da_eseguire}}
```

- Esempio di un job che esegue un determinato script alle 02:30 ogni venerdì:

```
30 2 * * Fri {{/percorso/assoluto/dello/script.sh}}
```

# crystal

Strumento per gestire codice sorgente Crystal.

Maggiori informazioni: [https://crystal-lang.org/reference/using\\_the\\_compiler](https://crystal-lang.org/reference/using_the_compiler).

- Esegui un file Crystal:

```
crystal {{percorso/del/file.cr}}
```

- Compila un file e tutte le sue dipendenze in un unico eseguibile:

```
crystal build {{percorso/del/file.cr}}
```

- Avvia un server locale interattivo per testare il linguaggio:

```
crystal play
```

- Crea una directory di progetto per un'applicazione Crystal:

```
crystal init app {{nome_applicazione}}
```

- Mostra tutte le opzioni di aiuto:

```
crystal help
```

# CSC

Compilatore per Microsoft C#.

Maggiori informazioni: <https://learn.microsoft.com/dotnet/csharp/language-reference/compiler-options/command-line-building-with-csc-exe>.

- Compila uno o più file C# in un eseguibile da command-line:

```
csc {{percorso/del/file_input_a.cs}} {{percorso/del/
file_input_b.cs}}
```

- Specifica il nome del file output:

```
csc /out:{{percorso/del/nome_file_output}} {{percorso/del/
file_input.cs}}
```

- Compila in una libreria .dll invece che in un eseguibile:

```
csc /target:library {{percorso/del/file_input.cs}}
```

- Referenzia un altro assembly:

```
csc /reference:{{percorso/della/libreria.dll}} {{percorso/
del/file_input.cs}}
```

- Includi una risorsa:

```
csc /resource:{{percorso/del/file_risorsa}} {{percorso/del/
file_input.cs}}
```

- Genera una documentazione XML automaticamente:

```
csc /doc:{{percorso/della/documentazione.xml}} {{percorso/
del/file_input.cs}}
```

- Specifica un'icona:

```
csc /win32icon:{{percorso/dell/icona.ico}} {{percorso/del/
file_input.cs}}
```

- Firma un assembly con un nome sicuro utilizzando una chiave:

```
csc /keyfile:{{percorso/della/chiave.snk}} {{percorso/del/
file_input.cs}}
```

# csslint

Un linter per codice CSS.

Maggiori informazioni: <https://github.com/CSSLint/csslint/wiki/Command-line-interface>.

- Esegui il linting di un singolo file CSS:

```
csslint {{file.css}}
```

- Esegui il linting di file CSS multipli:

```
csslint {{file1.css file2.css ...}}
```

- Elenca tutte le possibili regole di stile:

```
csslint --list-rules
```

- Specifica certe regole come errori (che risulteranno in un codice d'uscita diverso da zero):

```
csslint --errors={{errors,universal-selector,imports}}  
{{file.css}}
```

- Specifica certe regole come warning:

```
csslint --warnings={{box-sizing,selector-max,floats}}  
{{file.css}}
```

- Specifica certe regole da essere completamente ignorate:

```
csslint --ignore={{ids,rules-count,shorthand}} {{file.css}}
```



# csvclean

Trova e corregge errori di sintassi comuni in file CSV.

Incluso in csvkit.

Maggiori informazioni: <https://csvkit.readthedocs.io/en/latest/scripts/csvclean.html>.

- Correggi un file CSV:

```
csvclean {{file.csv}}
```

- Elenca le posizioni di errori di sintassi in un file CSV:

```
csvclean -n {{file.csv}}
```

# csvcut

Filtra e tronca file CSV. Come il comando Unix **cut**, ma per dati tabellari.

Incluso in csvkit.

Maggiori informazioni: <https://csvkit.readthedocs.io/en/latest/scripts/csvcut.html>.

- Stampa indici e nomi di tutte le colonne:

```
csvcut -n {{dati.csv}}
```

- Estrai la prima e terza colonna:

```
csvcut -c {{1,3}} {{dati.csv}}
```

- Estrai tutte le colonne eccetto la quarta:

```
csvcut -C {{4}} {{dati.csv}}
```

- Estrai le colonne "id" e "nome di battesimo" (in quest'ordine):

```
csvcut -c {{id,"nome di battesimo"}} {{dati.csv}}
```

# csvformat

Converti un file CSV in un formato di output personalizzato.

Incluso in csvkit.

Maggiori informazioni: <https://csvkit.readthedocs.io/en/latest/scripts/csvformat.html>.

- Converti in un file delimitato da tab (TSV):

```
csvformat -T {{dati.csv}}
```

- Converti i delimitatori in un carattere personalizzato:

```
csvformat -D "{{carattere_personalizzato}}" {{dati.csv}}
```

- Converti caratteri newline a carriage return (^M) + newline:

```
csvformat -M "{{\r\n}}" {{dati.csv}}
```

- Minimizza l'utilizzo delle virgolette:

```
csvformat -U 0 {{dati.csv}}
```

- Massimizza l'utilizzo delle virgolette:

```
csvformat -U 1 {{dati.csv}}
```

# csvgrep

Filtra righe CSV con stringhe e pattern matching.

Incluso in csvkit.

Maggiori informazioni: <https://csvkit.readthedocs.io/en/latest/scripts/csvgrep.html>.

- Trova righe contenenti una certa stringa nella colonna 1:

```
csvgrep -c {{1}} -m {{stringa}} {{data.csv}}
```

- Trova righe per le quali le colonne 3 e 4 soddisfano una certa espressione regolare:

```
csvgrep -c {{3,4}} -r {{espressione_regolare}} {{data.csv}}
```

- Trova righe dove la colonna "nome" NON include la stringa "Mario Rossi":

```
csvgrep -i -c {{nome}} -m "{{Mario Rossi}}" {{data.csv}}
```

# csvlook

Visualizza un file CSV nella console come tabella a larghezza fissa.

Incluso in csvkit.

Maggiori informazioni: <https://csvkit.readthedocs.io/en/latest/scripts/csvlook.html>.

- Visualizza un file CSV:

```
csvlook {{data.csv}}
```

# csvpy

Carica un file CSV in una shell Python.

Incluso in csvkit.

Maggiori informazioni: <https://csvkit.readthedocs.io/en/latest/scripts/csvpy.html>.

- Carica un file CSV in un oggetto CSVKitReader:

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

- Carica un file CSV in un oggetto CSVKitDictReader:

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

# csvsort

Ordina le righe di di file CSV.

Incluso in csvkit.

Maggiori informazioni: <https://csvkit.readthedocs.io/en/latest/scripts/csvsort.html>.

- Ordina un file CSV secondo la colonna 9:

```
csvsort -c {{9}} {{data.csv}}
```

- Ordina un file CSV secondo la colonna "nome" in ordine decrescente:

```
csvsort -r -c {{nome}} {{data.csv}}
```

- Ordina un file CSV secondo la colonna 2 e secondo la 4:

```
csvsort -c {{2,4}} {{data.csv}}
```

- Ordina un file CSV senza inferire il tipo dei dati:

```
csvsort --no-inference -c {{colonne}} {{data.csv}}
```

# csvstat

Stampa statistiche descrittive per tutte le colonne di un file CSV.

Incluso in csvkit.

Maggiori informazioni: <https://csvkit.readthedocs.io/en/latest/scripts/csvstat.html>.

- Mostra tutte le statistiche per tutte le colonne:

```
csvstat {{dati.csv}}
```

- Mostra tutte le statistiche per le colonne 2 e 4:

```
csvstat -c {{2,4}} {{dati.csv}}
```

- Mostra la somma per tutte le colonne:

```
csvstat --sum {{dati.csv}}
```

- Mostra la lunghezza massima dei valori della colonna 3:

```
csvstat -c {{3}} --len {{dati.csv}}
```

- Mostra il numero di valori unici nella colonna "nome":

```
csvstat -c {{nome}} --unique {{dati.csv}}
```



# ctest

Programma per eseguire test in progetti CMake.

Maggiori informazioni: <https://gitlab.kitware.com/cmake/community/wikis/doc/ctest/Testing-With-CTest>.

- Esegui tutti i test definiti nel progetto CMake, eseguendo 4 job allo stesso tempo in parallelo:

```
ctest -j{{4}} --output-on-failure
```

- Mostra una lista dei test disponibili:

```
ctest -N
```

- Esegui un singolo test in base al suo nome, o filtrando con una espressione regolare:

```
ctest --output-on-failure -R '^{{nome_test}}$'
```

# curl

Trasferisci dati da o ad un server.

Supporta la maggior parte dei protocolli, tra cui HTTP, HTTPS, FTP, SCP, etc.

Maggiori informazioni: <https://curl.se/docs/manpage.html>.

- Esegue una richiesta HTTP GET e stampa il contenuto in `stdout`:

```
curl {{https://example.com}}
```

- Esegue una richiesta HTTP GET, [L] segue eventuali reindirizzamenti 3xx e [D] stampa il contenuto e la intestazione della risposta su `stdout`:

```
curl --location --dump-header - {{https://example.com}}
```

- Scarica un file salvando l'[O]utput con lo stesso nome indicato nell'URL:

```
curl --remote-name {{https://example.com/nome_file.zip}}
```

- Invia [d]ati form-encoded (con una richiesta POST di tipo `application/x-www-form-urlencoded`). Utilizza `--data @file_name` oppure `--data @' - '` per leggere da `stdin`:

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

- Invia una richiesta con una intestazione aggiuntiva utilizzando un metodo HTTP personalizzato attraverso un pro[x]y (come BurpSuite) ignorando i certificati autofirmati non sicuri:

```
curl -k --proxy {{http://127.0.0.1:8080}} --header {'Authorization: Bearer token'} --request {{GET|PUT|POST|DELETE|PATCH|...}} {{https://example.com}}
```

- Invia dati in formato JSON, specificando la [H] intestazione `content-type` appropriata:

```
curl --data {'{"name":"mario"}'} --header {'Content-Type: application/json'} {{http://example.com/users/1234}}
```

- Invia il certificato client e la chiave per una risorsa omettendo la validazione del certificato:

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

- Risolve un hostname con un indirizzo IP personalizzato con un output [v] dettagliato (simile alla modifica del file `/etc/hosts` per la risoluzione di un DNS personalizzato):

```
curl --verbose --resolve {{example.com}}:{{80}}:{{127.0.0.1}}  
{{http://example.com}}
```

# cut

Taglia dividendo in campi **stdin** o file.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/cut>.

- Estrai i primi 16 caratteri di ogni riga da **stdin**:

```
cut -c {{1-16}}
```

- Estrai i primi 16 caratteri di ogni riga da un dato file:

```
cut -c {{1-16}} {{file}}
```

- Estrai tutto dal terzo carattere fino alla fine di ogni riga:

```
cut -c {{3-}}
```

- Estrai il quinto campo di ogni linea, utilizzando i due punti come separatore di campo (il default è tab):

```
cut -d'{{:}}' -f{{5}}
```

- Estrai il secondo e decimo campo di ogni linea, utilizzando il punto e virgola come delimitatore:

```
cut -d'{{;}}' -f{{2,10}}
```

- Estrai i campi dal terzo in poi di ogni linea, utilizzando lo spazio come delimitatore:

```
cut -d'{{ }}' -f{{3-}}
```

# darkhttpd

Web server Darkhttpd.

Maggiori informazioni: <https://unix4lyfe.org/darkhttpd>.

- Avvia il server utilizzando la directory specificata come document root:

```
darkhttpd {{percorso/della/docroot}}
```

- Avvia il server su una specifica porta (8080 di default per utenti non root):

```
darkhttpd {{percorso/della/docroot}} --port {{porta}}
```

- Ascolta solo su uno specifico indirizzo IP (di default, il server ascolta su tutte le interfacce):

```
darkhttpd {{percorso/della/docroot}} --addr {{indirizzo_ip}}
```

# dash

Debian Almquist Shell, una implementazione di **sh** moderna, che conforme a POSIX, (non compatibile con Bash).

Maggiori informazioni: <https://manned.org/dash>.

- Avvia una sessione shell interattiva:

```
dash
```

- Esegui un comando, ed esci subito:

```
dash -c "{{comando}}"
```

- Esegui un script:

```
dash {{percorso/dello/script.sh}}
```

- Esegui comandi da un script, stampando ogni comando prima di eseguirlo:

```
dash -x {{percorso/dello/script.sh}}
```

- Esegui comandi da un script, fermandosi al primo errore:

```
dash -e {{percorso/dello/script.sh}}
```

- Leggi ed esegui comandi dal `stdin`:

```
dash -s
```

# date

Imposta o mostra data e ora di sistema.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/date>.

- Mostra la data corrente utilizzando il formato predefinito della locale corrente:

```
date +%c
```

- Mostra la data corrente in UTC e formato ISO 8601:

```
date -u +%Y-%m-%dT%H:%M:%SZ
```

- Mostra la data corrente come timestamp Unix (secondi dall'epoca Unix):

```
date +%s
```

- Mostra una specifica data (rappresentata come timestamp Unix) utilizzando il formato predefinito:

```
date -d @1473305798
```

- Converti una specifica data in un timestamp Unix:

```
date -d "{{2018-09-01 00:00}}" +%s --utc
```

# dc

Calcolatore a precisione arbitraria. Usa la notazione polacca inversa (RPN).

Maggiori informazioni: [https://www.gnu.org/software/bc/manual/dc-1.05/html\\_mono/dc.html](https://www.gnu.org/software/bc/manual/dc-1.05/html_mono/dc.html).

- Avvia il calcolatore in modalità interattiva:

```
dc
```

- Esegui uno script dc da file:

```
dc -f {{file}}
```

- Calcola 4 per 5 [4 5 \*], sottrai 17 [17 -], e stampa [p] il risultato (utilizzando echo):

```
echo "4 5 * 17 - p" | dc
```

- Setta il numero di posizioni decimali a 7 [7 k], calcola 5 diviso -3 [5 \_3/] e stampa [p] il risultato (utilizzando dc -e):

```
dc -e "7 k 5 _3 / p"
```

- Calcola il rapporto aureo, phi: setta il numero di posizioni decimali a 100 [100 k], e calcola la radice di 5 [5 v] più 1 [1 +], diviso due [2 /] e stampa [p] il risultato:

```
dc -e "100 k 5 v 1 + 2 / p"
```



# dd

Converti e copia un file.

Maggiori informazioni: <https://manned.org/dd.1p>.

- Crea un disco USB avviabile da un file ISO e mostra il progresso:

```
dd if={{percorso/del/file.iso}} of={{/dev/disco_usb}}  
status=progress
```

- Clona un disco su un altro disco a blocchi con grandezza di 4 MiB e scarica le scritture prima che il comando termini:

```
dd bs=4194304 conv=fsync if={{/dev/disco_sorgente}} of={{/  
dev/disco_destinazione}}
```

- Genera un file con un numero specifico di byte randomici utilizzando il driver random del kernel:

```
dd bs={{100}} count={{1}} if=/dev/urandom of={{percorso/del/  
file_random}}
```

- Testa la performance in scrittura di un disco:

```
dd bs={{1024}} count={{1000000}} if=/dev/zero of={{percorso/  
del/file_1GB}}
```

- Crea un backup di sistema, salvalo in un file IMG (può essere ripristinato in seguito scambiando `if` e `of`), e mostra il progresso:

```
dd if={{/dev/disco}} of={{percorso/del/file.img}}  
status=progress
```

# decaffeinate

Converti script CoffeeScript in JavaScript moderno.

Maggiori informazioni: <https://decaffeinate-project.org>.

- Converti un file CoffeeScript in JavaScript:

```
decaffeinate {{percorso/del/file.coffee}}
```

- Converti un file CoffeeScript v2 in JavaScript:

```
decaffeinate --use-cs2 {{percorso/del/file.coffee}}
```

- Converti `require` e `module.exports` in `import` ed `export`:

```
decaffeinate --use-js-modules {{percorso/del/file.coffee}}
```

- Converti un file CoffeeScript, permettendo di esportare nomi:

```
decaffeinate --loose-js-modules {{percorso/del/file.coffee}}
```

# deluge-console

Interfaccia interattiva da linea di comando per il client BitTorrent Deluge.

Maggiori informazioni: <https://deluge-torrent.org>.

- Avvia un'interfaccia interattiva da console:

```
deluge-console
```

- Connetti ad un'istanza del demone di Deluge:

```
connect {{hostname}}:{{porta}}
```

- Aggiungi un torrent al demone:

```
add {{url|magnet|percorso/del/file}}
```

- Mostra informazioni su tutti i torrent:

```
info
```

- Mostra informazioni su di uno specifico torrent:

```
info {{id_torrent}}
```

- Metti in pausa un torrent:

```
pause {{id_torrent}}
```

- Riprendi un torrent:

```
resume {{id_torrent}}
```

- Rimuovi un torrent dal demone:

```
rm {{id_torrent}}
```

# deluge

Client BItTorrent da linea di comando.

Maggiori informazioni: <https://deluge-torrent.org>.

- Scarica un torrent:

```
deluge {{url|magnet|percorso/del/file}}
```

- Scarica un torrent utilizzando uno specifico file di configurazione:

```
deluge -c {{percorso/del/file_configurazione}} {{url|magnet|percorso/del/file}}
```

- Scarica un torrent ed avvia una specifica interfaccia utente:

```
deluge -u {{gtk|web|console}} {{url|magnet|percorso/del/file}}
```

- Scarica un torrent e scrivi il log in un file:

```
deluge -l {{percorso/del/file_log}} {{url|magnet|percorso/del/file}}
```

# deluged

Un processo demone per il client BitTorrent Deluge.

Maggiori informazioni: <https://deluge-torrent.org>.

- Avvia il demone di Deluge:

```
deluged
```

- Avvia il demone di Deluge su di una specifica porta:

```
deluged -p {{porta}}
```

- Avvia il demone di Deluge utilizzando uno specifico file di configurazione:

```
deluged -c {{percorso/del/file_configurazione}}
```

- Avvia il demone di Deluge e scrivi il log in un file:

```
deluged -l {{percorso/del/file_log}}
```

# dep

Strumento di gestione delle dipendenze per progetti Go.

Maggiori informazioni: <https://deployer.org>.

- Inizializza la directory corrente come radice di un progetto Go:

```
dep init
```

- Installa dipendenze mancanti (scannerizza `Gopkg.toml` ed i file `.go`):

```
dep ensure
```

- Mostra lo stato delle dipendenze di un progetto:

```
dep status
```

- Aggiungi una dipendenza al progetto:

```
dep ensure -add {{url_pacchetto}}
```

- Aggiorna le versioni bloccate (in `Gopkg.lock`) di tutte le dipendenze:

```
dep ensure -update
```

# detox

Rinomina file per renderli facili da utilizzare.

Rimuove spazi e altri caratteri fastidiosi come doppi caratteri underline.

Maggiori informazioni: <https://github.com/dharple/detox>.

- Rimuovi spazi ed altri caratteri indesiderabili dal nome di un file:

```
detox {{file}}
```

- Mostra come detox rinominerebbe tutti i file in una directory ricorsivamente:

```
detox --dry-run -r {{directory}}
```

- Rimuovi spazi e altri caratteri indesiderabili da tutti i file in una directory ricorsivamente:

```
detox -r {{directory}}
```

# dexter

Strumento per autenticare utenti kubectl con OpenId Connect.

Maggiori informazioni: <https://github.com/gini/dexter>.

- Crea ed autentica un utente con Google OIDC:

```
dexter auth -i {{id-client}} -s {{segreto-client}}
```

- Sovrascrivi la posizione predefinita della configurazione di kube:

```
dexter auth -i {{id-client}} -s {{segreto-client}} --kube-  
config {{percorso/della/configurazione}}
```



# df

Fornisce una panoramica dello spazio utilizzato dai file system sui dischi.

Maggiori informazioni: <https://manned.org/df.1posix>.

- Mostra tutti i file system ed il loro utilizzo del disco:

```
df
```

- Mostra tutti i file system ed il loro utilizzo del disco in formato leggibile dall'uomo:

```
df -h
```

- Mostra il file system ed contenente il file o directory dato ed il suo utilizzo del disco:

```
df {{percorso/del/file_o_directory}}
```

# dhcpwn

Testa attacchi di esaurimento IP DHCP ed intercetta il traffico DHCP locale.

Maggiori informazioni: <https://github.com/mschwager/dhcpwn>.

- Inonda la rete con richieste di IP:

```
dhcpwn --interface {{interfaccia}} flood --count  
{{numero_di_richieste}}
```

- Intercetta traffico DHCP locale:

```
dhcpwn --interface {{interfaccia}} sniff
```

# diff

Confronta file e directory.

Maggiori informazioni: <https://manned.org/diff>.

- Confronta due file (elenca cambiamenti necessari per trasformare vecchio\_file in nuovo\_file):  
`diff {{vecchio_file}} {{nuovo_file}}`
- Confronta due file ignorando gli spazi:  
`diff -w {{vecchio_file}} {{nuovo_file}}`
- Confronta due file mostrando le differenze fianco a fianco:  
`diff -y {{vecchio_file}} {{nuovo_file}}`
- Confronta due file, mostrando le differenze in formato unificato (come git diff):  
`diff -u {{vecchio_file}} {{nuovo_file}}`
- Confronta due directory ricorsivamente (mostra i nomi dei file/directory diversi e le differenze tra i file):  
`diff -r {{old_directory}} {{new_directory}}`
- Confronta due directory mostrando solamente il nome dei file diversi:  
`diff -rq {{old_directory}} {{new_directory}}`

# diffstat

Crea un istogramma dall'output del comando **diff**.

Maggiori informazioni: <https://manned.org/diffstat>.

- Mostra i cambiamenti in un istogramma:

```
diff {{file1}} {{file2}} | diffstat
```

- Mostra inserimenti, cancellazioni e modifiche come una tabella:

```
diff {{file1}} {{file2}} | diffstat -t
```

# dig

Utilità di lookup DNS.

Maggiori informazioni: <https://manned.org/dig>.

- Mostra gli IP associati ad un hostname (record A):

```
dig +short {{esempio.com}}
```

- Mostra i mail server associati ad uno specifico dominio (record MX):

```
dig +short {{esempio.com}} MX
```

- Specifica un server DNS alternativo a cui fare richiesta:

```
dig @{{8.8.8.8}} {{esempio.com}}
```

- Esegui un lookup DNS inverso su di un indirizzo IP (record PTR):

```
dig -x {{8.8.8.8}}
```

- Trova i nameserver autoritativi per la zona e mostra i record SOA:

```
dig +nssearch {{esempio.com}}
```

- Esegui richieste iterative e mostra l'intero percorso per risolvere il dominio:

```
dig +trace {{esempio.com}}
```

# dircolors

Stampa comandi necessari per settare la variabile d'ambiente LS\_COLOR per stilizzare **ls**, **dir**, etc.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/dircolors>.

- Stampa i comandi per settare LS\_COLOR con i colori predefiniti:

```
dircolors
```

- Stampa i comandi per settare LS\_COLOR con i colori definiti in un file:

```
dircolors {{file}}
```

- Stampa comandi per la Bourne shell:

```
dircolors --bourne-shell
```

- Stampa comandi per la C shell:

```
dircolors --c-shell
```

- Mostra i colori predefiniti per diversi tipi di file ed estensioni:

```
dircolors --print-data
```

# direnv

Estensione della shell per aggiungere o rimuovere variabili d'ambiente in base alla directory corrente.

Maggiori informazioni: <https://github.com/direnv/direnv>.

- Permette il caricamento del `.envrc` presente nella directory corrente:

```
direnv allow {{.}}
```

- Revoca il permesso del `.envrc` presente nella directory corrente:

```
direnv deny {{.}}
```

- Permette la modifica del `.envrc` nell'editor di testo predefinito, in seguito ricarica l'ambiente:

```
direnv edit {{.}}
```

- Ricarica l'ambiente corrente:

```
direnv reload
```

- Mostra delle informazioni di debug:

```
direnv status
```

# dirname

Determina la directory genitore di un determinato file o percorso.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/dirname>.

- Calcola la directory genitore di un dato percorso:

```
dirname {{percorso/del/file_o_directory}}
```

- Calcola la directory genitore di più percorsi:

```
dirname {{percorso/del/file_or_directory1 percorso/del/  
file_or_directory2 ...}}
```

- Delimita l'output utilizzando caratteri NUL invece di una nuova linea (utile in combinazione con xargs):

```
dirname --zero {{percorso/del/file_or_directory1 percorso/  
del/file_or_directory2 ...}}
```



# dirs

Mostra o manipola uno stack di directory.

Uno stack di directory è una lista di directory recentemente visitate che può essere manipolata con i comandi **pushd** e **popd**.

Maggiori informazioni: <https://www.gnu.org/software/bash/manual/bash.html#Directory-Stack-Builtins>.

- Mostra lo stack di directory dividendo i nomi con uno spazio:

```
dirs
```

- Mostra lo stack di directory una per riga:

```
dirs -p
```

- Mostra solo l'ennesima directory dello stack (gli indici partono da 0):

```
dirs +{{N}}
```

- Pulisci lo stack di directory:

```
dirs -c
```

# dive

Un tool per esplorare immagini Docker, contenuti dei livelli, e ridurre la dimensione.

Maggiori informazioni: <https://github.com/wagoodman/dive>.

- Analizza un'immagine Docker:

```
dive {{tag_immagine}}
```

- Costruisci un'immagine ed avvia l'analisi:

```
dive build -t {{tag}}
```

# docker build

Crea un'immagine a partire da un Dockerfile. La creazione di un'immagine Docker è chiamata build.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/buildx/build/>.

- Crea un'immagine Docker usando il Dockerfile nella directory corrente:

```
docker build .
```

- Crea un'immagine Docker usando il Dockerfile disponibile a un dato URL:

```
docker build {{github.com/creack/docker-firefox}}
```

- Crea e tagga un'immagine Docker:

```
docker build --tag {{nome:tag}} .
```

- Non usare la cache per la creazione di un'immagine Docker:

```
docker build --no-cache --tag {{nome:tag}} .
```

- Crea un'immagine Docker usando un dato Dockerfile:

```
docker build --file {{Dockerfile}} .
```

- Crea un'immagine Docker usando variabili fornite in fase di build:

```
docker build --build-arg {{HTTP_PROXY=http://  
10.20.30.2:1234}} --build-arg {{FTP_PROXY=http://  
40.50.60.5:4567}} .
```

# docker compose

Esegui e gestisci applicazioni Docker composte da più container.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/compose/>.

- Elenca i container in esecuzione:

```
docker compose ps
```

- Crea ed avvia tutti i container in background utilizzando il file `docker-compose.yml` nella directory corrente:

```
docker compose up --detach
```

- Avvia tutti i container, buildandoli di nuovo se necessario:

```
docker compose up --build
```

- Avvia tutti i contenitori specificando un nome di progetto e utilizzando un file compose alternativo:

```
docker compose -p {{nome_di_progetto}} --file {{percorso/del/file}} up
```

- Ferma tutti i container in esecuzione:

```
docker compose stop
```

- Ferma e rimuovi tutti i container, reti, immagini e volumi:

```
docker compose down --rmi all --volumes
```

- Segui i log di tutti i container:

```
docker compose logs --follow
```

- Segui i log di un container specifico:

```
docker compose logs --follow {{nome_container}}
```

# docker container

Gestisci container Docker.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/container/>.

- Elenca i container Docker attualmente in esecuzione:

```
docker container ls
```

- Avvia uno o più container fermi:

```
docker container start {{nome_container1}}  
{{nome_container2}}
```

- Termina uno o più container in esecuzione:

```
docker container kill {{nome_container}}
```

- Ferma uno o più container in esecuzione:

```
docker container stop {{nome_container}}
```

- Sospendi tutti i processi dentro uno o più container:

```
docker container pause {{nome_container}}
```

- Mostra informazioni dettagliate su uno o più container:

```
docker container inspect {{nome_container}}
```

- Esporta il filesystem di un container come archivio tar:

```
docker container export {{nome_container}}
```

- Crea una nuova immagine dai cambiamenti di un container:

```
docker container commit {{nome_container}}
```

# docker cp

Copia file o directory tra il filesystem di un container e quello locale (host).

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/container/cp/>.

- Copia un file o una directory dall'host a un container:

```
docker cp {{percorso/del/file_o_directory_su_host}}  
{{nome_container}}:{{percorso/del/  
file_o_directory_su_container}}
```

- Copia un file o una directory da un container all'host:

```
docker cp {{nome_container}}:{{percorso/del/  
file_o_directory_su_container}} {{percorso/del/  
file_o_directory_su_host}}
```

- Copia un file o una directory dall'host a un container, seguendo un link simbolico (non copiare il link simbolico, ma direttamente il file da lui referenziato):

```
docker cp --follow-link {{percorso/del/  
link_simbolico_su_host}} {{nome_container}}:{{percorso/del/  
file_o_directory_su_container}}
```

# docker exec

Esegui un comando su un Docker container in esecuzione.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/container/exec/>.

- Avvia una shell interattiva all'interno di un container in esecuzione:

```
docker exec --interactive --tty {{nome_container}} {{/bin/bash}}
```

- Esegui un comando in background ("detached") su un container in esecuzione:

```
docker exec --detach {{nome_container}} {{comando}}
```

- Seleziona la directory di lavoro in cui eseguire un dato comando:

```
docker exec --interactive --tty --workdir {{percorso/della/directory}} {{nome_container}} {{comando}}
```

- Esegui un comando in background su un container esistente, mantenendo aperto stdin:

```
docker exec --interactive --detach {{nome_container}}  
{{comando}}
```

- Imposta una variabile d'ambiente in una sessione Bash in esecuzione:

```
docker exec --interactive --tty --env {{nome_variabile}}  
={{valore}} {{nome_container}} {{/bin/bash}}
```

- Specifica l'utente da usare per eseguire un comando:

```
docker exec --user {{utente}} {{nome_container}} {{comando}}
```

# docker image

Gestisci immagini Docker.

Vedi anche **docker build**, **docker import** e **docker pull**.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/image/>.

- Elenca tutte le immagini Docker locali:

```
docker image ls
```

- Elimina le immagini Docker locali inutilizzate:

```
docker image prune
```

- Cancella tutte le immagini inutilizzate (non solo quelle sprovviste di tag):

```
docker image prune --all
```

- Mostra la cronologia di un'immagine Docker locale:

```
docker image history {{immagine}}
```



# docker images

Gestisci immagini Docker.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/image/ls/>.

- Elenca tutte le immagini Docker:

```
docker images
```

- Elenca tutte le immagini Docker incluse quelle intermedie:

```
docker images -a
```

- Elenca in modalità silenziosa (solo gli ID numerici):

```
docker images -q
```

- Elenca tutte le immagini Docker che non sono usate da alcun container:

```
docker images --filter dangling=true
```

# docker inspect

Mostra informazioni a basso livello di oggetti Docker.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/inspect/>.

- Mostra aiuto:

```
docker inspect
```

- Mostra informazioni su un container, immagine o volume usando un nome o un identificativo (ID):

```
docker inspect {{nome_container|nome_immagine|ID}}
```

- Mostra l'indirizzo IP di un container:

```
docker inspect --format '{{range.NetworkSettings.Networks}}\n{{.IPAddress}}\n{{end}}' {{nome_container}}
```

- Mostra il percorso dei file di log di un container:

```
docker inspect --format='{{.LogPath}}' {{nome_container}}
```

- Mostra il nome dell'immagine di un container:

```
docker inspect --format='{{.Config.Image}}'\n{{nome_container}}
```

- Mostra le informazioni di configurazione in formato JSON:

```
docker inspect --format='{{json .Config}}'\n{{nome_container}}
```

- Mostra il binding di tutte le porte:

```
docker inspect --format='{{range $p, $conf\n:= .NetworkSettings.Ports}} {{.HostPort}} -> {{(index $conf\n0).HostPort}} {{end}}' {{nome_container}}
```

# docker logs

Mostra i log di un container.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/container/logs/>.

- Mostra i log di un container:

```
docker logs {{nome_container}}
```

- Segui i log di un container:

```
docker logs -f {{nome_container}}
```

- Mostra le ultime 5 righe:

```
docker logs {{nome_container}} --tail {{5}}
```

- Mostra i log mettendo un timestamp in coda:

```
docker logs -t {{nome_container}}
```

- Mostra i log avvenuti prima di un dato momento nell'esecuzione del container (ad esempio, 23m, 10s, 2013-01-02T13:23:37):

```
docker logs {{nome_container}} --until {{momento}}
```

# docker-machine

Crea e gestisci macchine che utilizzano Docker.

Maggiori informazioni: <https://github.com/docker/machine>.

- Elenca macchine Docker in esecuzione:

```
docker-machine ls
```

- Crea una nuova macchina Docker con uno specifico nome:

```
docker-machine create {{nome}}
```

- Mostra lo stato di una macchina:

```
docker-machine status {{nome}}
```

- Avvia una macchina:

```
docker-machine start {{nome}}
```

- Arresta una macchina:

```
docker-machine stop {{nome}}
```

- Ispeziona le informazioni su di una macchina:

```
docker-machine inspect {{nome}}
```

# docker network

Crea e gestisci reti docker.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/network/>.

- Elenca le reti disponibili configurate sul Docker daemon:

```
docker network ls
```

- Crea una rete definita da un utente:

```
docker network create --driver {{nome_del_driver}}  
{{nome_rete}}
```

- Mostra informazioni dettagliate su una lista di reti (separata da spazi):

```
docker network inspect {{nome_rete_1 nome_rete_2}}
```

- Connetti un container ad una rete usando il suo nome o ID:

```
docker network connect {{nome_rete}} {{nome_container|ID}}
```

- Disconnetti un container da una rete:

```
docker network disconnect {{nome_rete}} {{nome_container|ID}}
```

- Elimina le reti inutilizzate (non referenziate da alcun container):

```
docker network prune
```

- Elimina una lista di reti (separata da spazi) inutilizzate:

```
docker network rm {{nome_rete_1 nome_rete_2}}
```

# docker

Gestisci container ed immagini Docker.

Alcuni comandi aggiuntivi, come **run**, hanno la propria documentazione.

Maggiori informazioni: <https://docs.docker.com/reference/cli/docker/>.

- Elenca tutti i container Docker (in esecuzione e arrestati):

```
docker ps --all
```

- Avvia un container da una immagine, con un nome personalizzato:

```
docker run --name {{nome_container}} {{immagine}}
```

- Avvia o arresta un container esistente:

```
docker {{start|stop}} {{nome_container}}
```

- Scarica (pull) un'immagine dal Docker registry:

```
docker pull {{image}}
```

- Avvia una shell all'interno di un container in esecuzione:

```
docker exec -it {{nome_container}} {{sh}}
```

- Rimuovi un container arrestato:

```
docker rm {{nome_container}}
```

- Ottieni e visualizza i log di un container:

```
docker logs -f {{nome_container}}
```

# dokku

Mini-Heroku basato su Docker (PaaS, Platform As A Service).

Distribuisci facilmente molteplici app sul tuo server in diversi linguaggi utilizzando un singolo comando **git-push**.

Maggiori informazioni: <https://github.com/dokku/dokku>.

- Elenca app in esecuzione:

```
dokku apps
```

- Crea un'app:

```
dokku apps:create {{nome_app}}
```

- Rimuovi un'app:

```
dokku apps:destroy {{nome_app}}
```

- Installa un plugin:

```
dokku plugin:install {{url_completo_repo}}
```

- Collega un database ad un'app:

```
dokku {{db}}:link {{nome_db}} {{nome_app}}
```

# dot

Strumento da linea di comando per produrre disegni a livelli di grafi orientati.

Maggiori informazioni: <https://graphviz.org/doc/info/command.html>.

- Renderizza un'immagine determinando il nome del file di output dal nome del file di input ed il formato:

```
dot -Tpng -O {{percorso/del/file.dot}}
```

- Crea una SVG da un file DOT:

```
dot -Tsvg -o {{percorso/del/file_output.svg}} {{percorso/del/file.dot}}
```



# dotnet

Strumenti .NET da linea di comando multiplatforma per .NET Core.

Alcuni comandi aggiuntivi, come **build**, hanno la propria documentazione.

Maggiori informazioni: <https://learn.microsoft.com/dotnet/core/tools>.

- Inizializza un nuovo progetto .NET:

```
dotnet new {{nome_abbreviato_template}}
```

- Ripristina pacchetti nuget:

```
dotnet restore
```

- Costruisci ed esegui il progetto .NET nella directory corrente:

```
dotnet run
```

- Esegui una applicazione dotnet pacchettizzata (solo il runtime è necessario, il resto dei comandi richiedono .NET Core SDK):

```
dotnet {{percorso/della/applicazione.dll}}
```

# doxygen

Un sistema di documentazione per diversi linguaggi di programmazione.

Maggiori informazioni: <https://www.doxygen.nl>.

- Genera il modello di configurazione di default `Doxyfile`:

```
doxygen -g
```

- Genera un modello di file di configurazione:

```
doxygen -g {{percorso/del/file_config}}
```

- Genera la documentazione utilizzando un file di configurazione esistente:

```
doxygen {{percorso/del/file_config}}
```

# drill

Esegui varie query DNS.

Maggiori informazioni: <https://manned.org/drill>.

- Mostra gli IP associati ad un hostname (record A):

```
drill {{esempio.com}}
```

- Lookup the mail server(s) associated with a given domain name (MX record):

```
drill mx {{esempio.com}}
```

- Recupera tutti i tipi di record per un dato dominio:

```
drill any {{esempio.com}}
```

- Specifica un server DNS alternativo da interrogare:

```
drill {{esempio.com}} @{{8.8.8.8}}
```

- Esegui un lookup DNS inverso su di un indirizzo IP (record PTR):

```
drill -x {{8.8.8.8}}
```

- Esegui un tracciamento DNSSEC dai root server fino al dominio:

```
drill -TD {{esempio.com}}
```

- Mostra record DNSKEY per un dominio:

```
drill -s dnskey {{esempio.com}}
```

# drush

Shell da linea di comando ed interfaccia di scripting per Drupal.

Maggiori informazioni: <https://www.drush.org>.

- Scarica il modulo "foo":

```
drush dl {{foo}}
```

- Scarica la versione 7.x-2.1-beta1 del modulo "foo":

```
drush dl {{foo}}-7.x-2.1-beta1
```

- Abilita il modulo "foo":

```
drush en {{foo}}
```

- Disabilita il modulo "foo":

```
drush dis {{foo}}
```

- Pulisci tutte le cache:

```
drush cc all
```

- Pulisci le cache CSS e JavaScript:

```
drush cc css-js
```

# du

Utilizzo del disco: stima e riassumi lo spazio utilizzato da file e directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/du>.

- Elenca le dimensioni di una directory ed ogni sotto-directory, nell'unità specificata (B/KiB/MiB):

```
du -{{b|k|m}} {{percorso/della/directory}}
```

- Elenca le dimensioni di una directory ed ogni sotto-directory, in formato umanamente leggibile (seleziona automaticamente l'unità appropriata per ogni dimensione):

```
du -h {{percorso/della/directory}}
```

- Mostra la dimensione di una singola directory, in unità umanamente leggibili:

```
du -sh {{percorso/della/directory}}
```

- Mostra in formato umanamente leggibile le dimensioni di una directory e tutti i file e directory in essa contenuti:

```
du -ah {{percorso/della/directory}}
```

- Elenca le dimensioni umanamente leggibili di una directory e di ogni sotto-directory, fino ad N livelli di profondità:

```
du -h --max-depth=N {{percorso/della/directory}}
```

- Mostra le dimensioni umanamente leggibili di tutti i file `.jpg` nelle sottodirectory della directory corrente, e mostra il totale cumulativo alla fine:

```
du -ch {{*/*.jpg}}
```

# duplicity

Crea archivi incrementali, compressi, cifrati con controllo di versione.

Può caricare i backup su una varietà di servizi backend.

Maggiori informazioni: <https://duplicity.gitlab.io>.

- Esegui il backup di una directory via FTPS su una macchina remota, cifrandolo con una password:

```
FTP_PASSWORD={{password_login_ftp}}  
PASSPHRASE={{password_cifratura}} duplicity {{percorso/della/  
directory_sorgente}} {{ftps://utente@hostname/percorso/della/  
directory_target/}}
```

- Esegui il backup di una directory in un server Amazon S3, facendo un backup completo ogni mese:

```
duplicity --full-if-older-than {{1M}} s3://{{nome_bucket[/  
prefisso]}}
```

- Elimina le versioni più vecchie di un anno da un backup salvato in un server WebDAV:

```
FTP_PASSWORD={{password_login_webdav}} duplicity remove-  
older-than {{1Y}} --force {{webdav[s]://  
utente@hostname[:porta]/directory}}
```

- Elenca i backup disponibili:

```
duplicity collection-status "file://{{percorso/assoluto/  
della/directory/di/backup}}"
```

- Elenca i file in un backup salvato su una macchina remota, via SSH:

```
duplicity list-current-files --time {{YYYY-MM-DD}} scp://  
{{utente@hostname}}/{{percorso/della/directory/backup}}
```

- Ripristina una sotto-directory da un backup locale cifrato con GnuPG in una posizione precisa:

```
PASSPHRASE={{password_chiave_gpg}} duplicity restore --  
encrypt-key {{id_chiave_gpg}} --path-to-restore {{percorso/  
relativo/sotto_directory}} file://{{percorso/assoluto/della/  
directory/di/backup}} {{percorso/della/directory/dove/  
ripristinare}}
```

# ebook-convert

Converti e-book in differenti formati, come PDF, EPUB, Mobi.

Parte dello strumento Calibre per librerie e-book.

Maggiori informazioni: <https://manual.calibre-ebook.com/generated/en/ebook-convert.html>.

- Converti un e-book in un altro formato:

```
ebook-convert {{percorso/del/file_input}} {{file_output}}
```

# echo

Stampa a schermo gli argomenti forniti.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/echo>.

- Stampa un messaggio di testo. Nota: le virgolette sono opzionali:

```
echo "{{Hello World}}"
```

- Stampa un messaggio con il contenuto di variabili di ambiente:

```
echo "{{La mia path è $PATH}}"
```

- Stampa un messaggio senza il carattere di nuova linea finale:

```
echo -n "{{Hello World}}"
```

- Aggiungi un messaggio in coda ad un file:

```
echo "{{Hello World}}" >> {{file.txt}}
```

- Abilita l'interpretazione delle sequenze di escape con il backslash (caratteri speciali):

```
echo -e "{{Colonna 1\tColonna 2}}"
```



# ect

Efficiente Tool di Compressione (o ECT) è un ottimizzatore di file scritto in C++. Supporta file PNG, JPEG, gzip e Zip.

Maggiori informazioni: <https://github.com/fhanau/Efficient-Compression-Tool>.

- Comprimi un file:

```
ect {{file.png}}
```

- Comprimi un file con il massimo livello di compressione utilizzando più thread:

```
ect -9 --mt-deflate {{file.png}}
```

- Comprimi tutti i file in una directory ricorsivamente, mantenendo la data di modifica originale:

```
ect -keep -recurse {{percorso/della/directory}}
```

# ed

L'originale editor di testo Unix.

Maggiori informazioni: [https://www.gnu.org/software/ed/manual/ed\\_manual.html](https://www.gnu.org/software/ed/manual/ed_manual.html).

- Avvia ed per editare un documento vuoto (che può essere salvato come nuovo file nella directory corrente):

```
ed
```

- Avvia ed per editare un documento vuoto, con : come indicatore del prompt di comandi:

```
ed -p :
```

- Avvia ed per editare un file esistente (mostra il numero di byte del file caricato):

```
ed -p : {{percorso/del/file}}
```

- Attiva o disattiva la stampa di spiegazioni per gli errori (di default, le spiegazioni non sono stampate ed appare solo un ?):

```
H
```

- Aggiungi del testo al documento corrente. Indica il completamento inserendo un punto da solo su una nuova linea:

```
a<Enter>{{text_to_insert}}<Enter>.
```

- Stampa l'intero documento (, è una scorciatoia per il range 1,\$ che copre dall'inizio alla fine del documento):

```
,p
```

- Scrivi il documento corrente su un nuovo file (il nome del file può essere omissso se ed è stato avviato con un file esistente):

```
w {{nome_file}}
```

- Termina ed:

```
q
```

# electrum

Ergonomico wallet (portafogli) Bitcoin e gestore di chiavi private.

Maggiori informazioni: <https://electrum.org>.

- Crea un nuovo wallet:

```
electrum -w {{nuovo_wallet.dat}} create
```

- Ripristina un wallet esistente da un seed offline:

```
electrum -w {{wallet_ripristinato.dat}} restore -o
```

- Crea una transazione firmata offline:

```
electrum mktx {{destinatario}} {{ammontare}} -f 0.0000001 -F  
{{mittente}} -o
```

- Mostra tutti gli indirizzi del wallet per la ricezione:

```
electrum listaddresses -a
```

- Firma un messaggio:

```
electrum signmessage {{indirizzo}} {{messaggio}}
```

- Verifica un messaggio:

```
electrum verifymessage {{indirizzo}} {{firma}} {{messaggio}}
```

- Connettiti solo ad una specifica istanza electrum-server:

```
electrum -p socks5:{{127.0.0.1}}:9050 -s  
{{56ckl5obj37gpcu.onion}}:50001:t -1
```

# elinks

Un browser testuale simile a lynx.

Maggiori informazioni: <http://elinks.or.cz>.

- Avvia elinks:

```
elinks
```

- Termina elinks:

```
<Ctrl> + C
```

- Stampa l'output di una pagina web nella console, colorando il testo con codici di controllo ANSI:

```
elinks -dump -dump-color-mode {{1}} {{url}}
```

# elm

Compila ed esegui file sorgente Elm.

Maggiori informazioni: <https://elm-lang.org>.

- Inizializza un progetto Elm, generando un file `elm.json`:

```
elm init
```

- Avvia una shell Elm interattiva:

```
elm repl
```

- Compila un file Elm, scrivendo il risultato in un file `index.html`:

```
elm make {{sorgente}}
```

- Compila un file Elm, scrivendo il risultato in un file JavaScript:

```
elm make {{sorgente}} --output={{destinazione}}.js
```

- Avvia un web server locale che compila file Elm al caricamento delle pagine:

```
elm reactor
```

- Installa un pacchetto Elm da <https://package.elm-lang.org>:

```
elm install {{author}}/{{package}}
```

# emacs

Editor di testo in tempo reale, estendibile, personalizzabile e auto documentato.

Maggiori informazioni: <https://www.gnu.org/software/emacs>.

- Avvia emacs in modalità console (senza finestra X):

```
emacs -nw
```

- Apri un file in emacs:

```
emacs {{nome_file}}
```

- Esci da emacs (salva i buffer e termina):

```
<Ctrl> + X, <Ctrl> + C
```

# emacsclient

Apri file in un server emacs esistente.

Maggiori informazioni: [https://www.gnu.org/software/emacs/manual/html\\_node/emacs/emacsclient-Options.html](https://www.gnu.org/software/emacs/manual/html_node/emacs/emacsclient-Options.html).

- Apri un file in un server Emacs esistente (utilizzando la GUI se disponibile):

```
emacsclient {{nome_file}}
```

- Apri un file in modalità console (senza finestra X):

```
emacsclient -nw {{nome_file}}
```

- Apri un file in un frame Emacs esistente e ritorna immediatamente:

```
emacsclient -n {{nome_file}}
```

# ember

Utilità da linea di comando per Ember.js.

Utilizzato per creare e mantenere applicazioni Ember.js.

Maggiori informazioni: <https://cli.emberjs.com>.

- Crea una nuova applicazione Ember:

```
ember new {{nome_app}}
```

- Crea un nuovo addon Ember:

```
ember addon {{nome_addon}}
```

- Builda un progetto:

```
ember build
```

- Builda un progetto in modalità produzione:

```
ember build -prod
```

- Avvia un server di sviluppo:

```
ember serve
```

- Esegui una suite di test:

```
ember test
```

- Esegui un blueprint per generare una route o un componente:

```
ember generate {{tipo}} {{nome}}
```

- Installa un addon ember-cli:

```
ember install {{nome_addon}}
```



# enca

Rileva e converti l'encoding di file di testo.

Maggiori informazioni: <https://github.com/nijel/enca>.

- Rileva l'encoding di uno o più file in base alla locale di sistema:

```
enca {{file1 file2 ...}}
```

- Rileva l'encoding specificando un linguaggio nel formato di locale POSIX/C (e.g. zh\_CN, en\_US):

```
enca -L {{linguaggio}} {{file1 file2 ...}}
```

- Converti file ad uno specifico encoding:

```
enca -L {{linguaggio}} -x {{encoding}} {{file1 file2 ...}}
```

- Crea una copia di un file esistente utilizzando un encoding diverso:

```
enca -L {{linguaggio}} -x {{encoding_finale}} <  
{{file_originale}} > {{nuovo_file}}
```

# enscript

Converti file di testo in PostScript, HTML, RTF, ANSI ed overstrike.

Maggiori informazioni: <https://www.gnu.org/software/enscript>.

- Genera un file PostScript da un file di testo:

```
enscript {{percorso/del/file_input}} --output={{percorso/del/
file_output}}
```

- Genera un file in un linguaggio differente da PostScript:

```
enscript {{percorso/del/file_input}} --
language={{linguaggio}} --output={{percorso/del/file_output}}
```

- Genera un file PostScript con layout orizzontale, dividendo la pagina in colonne (massimo 9):

```
enscript {{percorso/del/file_input}} --
columns={{numero_colonne}} --landscape --output={{percorso/
del/file_output}}
```

- Mostra linguaggi e formati file disponibili per evidenziare la sintassi:

```
enscript --help-highlight
```

- Genera un file PostScript con evidenziazione della sintassi e colori per uno specifico linguaggio:

```
enscript {{percorso/del/file_input}} --color=1 --
highlight={{linguaggio}} --output={{percorso/del/
file_output}}
```

# entr

Esegui comandi arbitrari al cambiamento di file.

Maggiori informazioni: <https://eradman.com/entrproject/>.

- Ricompila con `make` se qualsiasi file in qualsiasi sottodirectory cambia:

```
{{ag -l}} | entr {{make}}
```

- Ricompila e testa con `make` se qualsiasi file sorgente `.c` nella directory corrente cambia:

```
{{ls *.c}} | entr {{'make && make test'}}
```

- Invia il segnale `SIGTERM` ad un sottoprocesso ruby precedentemente avviato prima di eseguire `ruby main.rb`:

```
{{ls *.rb}} | entr -r {{ruby main.rb}}
```

- Esegui un comando con il file cambiato (`/_`) come argomento:

```
{{ls *.sql}} | entr {{psql -f}} /_
```

# env

Mostra le variabili d'ambiente o esegui un programma in un ambiente modificato.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/env>.

- Mostra le variabili d'ambiente:

```
env
```

- Esegui un programma. Utilizzato spesso in script dopo lo shebang (!) per cercare il percorso del programma:

```
env {{programma}}
```

- Resetta l'ambiente ed esegui un programma:

```
env -i {{programma}}
```

- Rimuovi una variabile dall'ambiente ed esegui un programma:

```
env -u {{variabile}} {{programma}}
```

- Setta una variabile ed esegui un programma:

```
env {{variabile}}={{valore}} {{programma}}
```

# envsubst

Sostituisci variabili di ambiente con il loro valore in stringhe di formato della shell.

Le variabili da sostituire devono essere nella forma `${var}` oppure `$var`.

Maggiori informazioni: [https://www.gnu.org/software/gettext/manual/html\\_node/envsubst-Invocation.html](https://www.gnu.org/software/gettext/manual/html_node/envsubst-Invocation.html).

- Sostituisci variabili di ambiente in `stdin` e stampa l'output su `stdout`:

```
echo '{{$HOME}}' | envsubst
```

- Sostituisci variabili di ambiente in un file input e stampa l'output su `stdout`:

```
envsubst < {{percorso/del/file_input}}
```

- Sostituisci variabili di ambiente in un file input e scrivi l'output su un file:

```
envsubst < {{percorso/del/file_input}} > {{percorso/del/file_output}}
```

- Sostituisci in un file input le variabili di ambiente specificate in una lista separata da spazi:

```
envsubst '{{$USER $HOME $SHELL}}' < {{percorso/del/file_input}}
```

# erl

Esegui e gestisci programmi nel linguaggio di programmazione Erlang.

Maggiori informazioni: <https://www.erlang.org>.

- Compila ed esegui un programma Erlang sequenziale come un comune script e poi esci:

```
erlc {{file}} && erl -noshell  
'{{modulo:funzione(argomenti)}}, init:stop().'
```

- Connetti ad un nodo Erlang in esecuzione:

```
erl -remsh {{nome_nodo}}@{{hostname}} -sname {{soprannome}} -  
hidden -setcookie {{cookie_nodo_remoto}}
```

- Fai caricare alla shell Erlang dei moduli da una directory:

```
erl -pa {{directory_con_file_beam}}
```

# eslint

Utilità di linting per JavaScript e JSX.

Maggiori informazioni: <https://eslint.org>.

- Crea una configurazione eslint:

```
eslint --init
```

- Esegui il linting di un dato set di file:

```
eslint {{percorso/del/file1.js percorso/del/file2.js ...}}
```

- Risolvi gli errori di linting:

```
eslint --fix
```

- Esegui il linting utilizzando un determinato file di configurazione:

```
eslint -c {{percorso/del/file_config}} {{percorso/del/  
file1.js percorso/del/file2.js ...}}
```

# espeak

Usa la sintesi vocale per parlare tramite il dispositivo audio di output predefinito.

Maggiori informazioni: <https://espeak.sourceforge.net>.

- Pronuncia una frase ad alta voce:

```
espeak "Mi piace andare in bici."
```

- Pronuncia il contenuto di un file ad alta voce:

```
espeak -f {{nome_file}}
```

- Salva l'output su un file audio WAV, invece che parlare direttamente:

```
espeak -w {{nome_file.wav}} "È GNU più Linux."
```

- Usa una voce differente:

```
espeak -v {{voce}}
```



# exa

Un moderno sostituto per **ls** (elenca i contenuti di una directory).

Maggiori informazioni: <https://the.exa.website>.

- Elenca i file nella directory corrente, uno per riga:

```
exa --oneline
```

- Elenca tutti i file, inclusi quelli nascosti:

```
exa --all
```

- Elenca tutti i file e mostra informazioni (permessi, dimensione e data di ultima modifica):

```
exa --long --all
```

- Elenca i file, ordinandoli per dimensione decrescente:

```
exa --reverse --sort={{size}}
```

- Mostra un albero dei file con 3 livelli di profondità:

```
exa --long --tree --level={{3}}
```

- Elenca i file e mostra informazioni, ordinandoli per ultima modifica (più vecchi prima):

```
exa --long --sort={{modified}}
```

# exec

Sostituisci il processo corrente con un altro.

Maggiori informazioni: <https://manned.org/exec.1posix>.

- Sostituisci con il comando specificato utilizzando le variabili di ambiente correnti:

```
exec {{comando -con -flag}}
```

- Sostituisci con il comando specificato utilizzando un ambiente vuoto:

```
exec -c {{comando -con -flag}}
```

- Sostituisci con il comando specificato ed esegui il login con la shell predefinita:

```
exec -l {{comando -con -flag}}
```

- Sostituisci con il comando specificato e cambia il nome del processo:

```
exec -a {{nuovo_nome_processo}} {{comando -con -flag}}
```

# exiftool

Leggi e scrivi metadati nei file.

Maggiori informazioni: <https://exiftool.org>.

- Rimuovi tutti i metadati EXIF dai file specificati:

```
exiftool -All= {{file1 file2 ...}}
```

- Muovi avanti di 1 ora la data in cui sono state scattate tutte le foto contenute in una directory:

```
exiftool "-AllDates+=0:0:0 1:0:0" {{percorso/della/  
directory}}
```

- Muovi indietro di 1 giorno e 2 ore la data in cui sono state scattate tutte le immagini JPEG:

```
exiftool "-AllDates-=0:0:1 2:0:0" -ext jpg
```

- Cambia solo il campo `DateTimeOriginal` sottraendo 1.5 ore e non tenere file di backup:

```
exiftool -DateTimeOriginal-=1.5 -overwrite_original
```

- Rinomina ricorsivamente tutti i file JPEG in una directory in base al campo `DateTimeOriginal`:

```
exiftool '-filename<DateTimeOriginal' -d %Y-%m-%d_%H-%M-%S%  
%lc.%e {{percorso/della/directory}} -r -ext jpg
```

# expand

Converti caratteri tab in spazi.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/expand>.

- Converti tab in un file in spazi, scrivendo su standard output:

```
expand {{file}}
```

- Converti i tab in spazi, leggendo da standard input:

```
expand
```

- Non convertire i tab dopo caratteri di spaziatura:

```
expand -i {{file}}
```

- Sostituisci i tab con un determinato numero di spazi, non 8 (default):

```
expand -t {{numero_spazi}} {{file}}
```

- Utilizza una lista separata da virgole di posizioni esplicite di tab:

```
expand -t {{1,4,6}}
```

# fastboot

Comunica con il dispositivo Android connessione quando in modalità bootloader (la situazione in cui **adb** non funziona).

Maggiori informazioni: [https://cs.android.com/android/platform/superproject/+main:system/core/fastboot](https://cs.android.com/android/platform/superproject/+/main:system/core/fastboot).

- Sblocca il bootloader:

```
fastboot oem unlock
```

- Ri-blocca il bootloader:

```
fastboot oem lock
```

- Riavvia il dispositivo da modalità fastboot, nuovamente in modalità fastboot:

```
fastboot reboot bootloader
```

- Esegue in Flash di una data immagine:

```
fastboot flash {{file.img}}
```

- Esegue il Flash di una recovery image personalizzata:

```
fastboot flash recovery {{file.img}}
```

- Mostra i dispositivi connessi:

```
fastboot devices
```

- Mostra tutte le informazioni su un dispositivo:

```
fastboot getvar all
```

# ffmpeg

Tool per convertire audio e video.

Maggiori informazioni: <https://ffmpeg.org>.

- Estrai l'audio da un video e salvalo come MP3:

```
ffmpeg -i {{percorso/del/video.mp4}} -vn {{percorso/del/
audio}}.mp3
```

- Sequenza immagini numerate (foto\_1.jpg, foto\_2.jpg, ecc) per creare un video o una GIF:

```
ffmpeg -i {{percorso/del/frame_%d.jpg}} -f image2
{{video.mpg|video.gif}}
```

- Estrai un singolo fotogramma da un video al timestamp mm:ss e salvalo come immagine di dimensioni 128x128:

```
ffmpeg -ss {{mm:ss}} -i {{percorso/del/video.mp4}} -frames 1
-s {{128x128}} -f image2 {{percorso/del/image.png}}
```

- Taglia un video da un momento iniziale mm:ss a un momento finale mm:ss (rimuovi la flag -to per tagliare fino alla fine):

```
ffmpeg -ss {{mm:ss}} -to {{mm2:ss2}} -i {{percorso/del/
video.mp4}} -codec copy {{percorso/del/output.mp4}}
```

- Converti un video AVI a MP4. Audio AAC a 128kbit, video h264 a CRF 23:

```
ffmpeg -i {{percorso/del/input_video}}.avi -codec:a aac -b:a
128k -codec:v libx264 -crf 23 {{percorso/del/
output_video}}.mp4
```

- Effettua un remux di un video MKV a MP4 senza re-encodare gli stream audio o video:

```
ffmpeg -i {{percorso/del/{input_video}}}.mkv -codec copy
{{percorso/del/output_video}}.mp4
```

- Converti un video MP4 a codec VP9. Per ottenere la migliore qualità possibile, usa un valore di CRF (consigliabile tra 15-35) e -b:v DEVE essere 0:

```
ffmpeg -i {{percorso/del/input_video}}.mp4 -codec:v libvpx-
vp9 -crf {{30}} -b:v 0 -codec:a libopus -vbr on -threads
{{number_of_threads}} {{percorso/del/output_video}}.webm
```

# ffprobe

Analizzatore di flussi multimediali.

Maggiori informazioni: <https://ffmpeg.org/ffprobe.html>.

- Visualizza tutte le informazioni disponibili sui flussi di un file multimediale (audio, video, immagini, etc):

```
ffprobe -v error -show_streams {{file.mp4}}
```

- Visualizza la durata del contenuto:

```
ffprobe -v error -show_entries format=duration -of  
default=noprint_wrappers=1:nokey=1 {{file.mp4}}
```

- Visualizza la frequenza dei fotogrammi di un video:

```
ffprobe -v error -select_streams v:0 -show_entries  
stream=avg_frame_rate -of default=noprint_wrappers=1:nokey=1  
{{video.mp4}}
```

- Visualizza la larghezza o l'altezza di un video:

```
ffprobe -v error -select_streams v:0 -show_entries  
stream={{width|height}} -of  
default=noprint_wrappers=1:nokey=1 {{video.mp4}}
```

- Visualizza il bit-rate medio di un video:

```
ffprobe -v error -select_streams v:0 -show_entries  
stream=bit_rate -of default=noprint_wrappers=1:nokey=1  
{{video.mp4}}
```

# file

Determina il tipo di file.

Maggiori informazioni: <https://manned.org/file>.

- Fornisce una descrizione del tipo di file specificato. Funziona anche con file senza estensione:

```
file {{nome_file}}
```

- Controlla dentro un file zip e determina il tipo dei file in esso contenuti:

```
file -z {{foo.zip}}
```

- Permette a `file` di leggere file speciali o di dispositivo:

```
file -s {{nome_file}}
```

- Non si limita al primo tipo di file trovato; continua a leggere il file fino alla fine:

```
file -k {{nome_file}}
```

- Determina il tipo MIME di un file:

```
file -i {{nome_file}}
```



# flac

Codifica, decodifica e controlla file flac.

Maggiori informazioni: <https://xiph.org/flac>.

- Converte un file wav in un file flac (questo creerà un file flac nella medesima posizione del file wav):

```
flac {{percorso/del/file.wav}}
```

- Codifica un file wav in flac, specificando il nome del risultato:

```
flac -o {{percorso/del/file_compresso.flac}} {{percorso/del/  
file_originale.wav}}
```

- Decodifica un file wav in flac, specificando il nome del risultato:

```
flac -d -o {{percorso/del/file_decompresso.wav}} {{percorso/  
del/file_originale.flac}}
```

- Controlla che un file flac sia codificato correttamente:

```
flac -t {{percorso/del/file.flac}}
```

# flake8

Programma per verificare lo stile e la qualità di un codice Python.

Maggiori informazioni: <https://flake8.pycqa.org/>.

- Analizza ricorsivamente un file o una directory:

```
flake8 {{percorso/del/file_o_directory}}
```

- Analizza ricorsivamente un file o una directory e mostra le righe contenenti errori:

```
flake8 --show-source {{percorso/del/file_o_directory}}
```

- Analizza ricorsivamente un file o una directory e ignora la lista delle regole specificate. (La lista con tutte le regole è consultabile su [flake8rules.com](https://flake8rules.com)):

```
flake8 --ignore {{regola1,regola2}} {{percorso/del/  
file_o_directory}}
```

- Analizza ricorsivamente un file o una directory ma esclude i file che corrispondono a una sottostringa o a un glob:

```
flake8 --exclude {{sottostringa1,glob2}} {{percorso/del/  
file_o_directory}}
```

# fmt

Riformatta i paragrafi di un file di testo unendoli e limitando la larghezza delle righe a un dato numero di caratteri (di default 75).

Maggiori informazioni: <https://www.gnu.org/software/coreutils/fmt>.

- Riformatta un file:

```
fmt {{percorso/del/file}}
```

- Riformatta un file producendo linee di (al massimo) n caratteri:

```
fmt -w {{n}} {{percorso/del/file}}
```

- Riformatta un file senza unire assieme le linee più corte della data larghezza:

```
fmt -s {{percorso/del/file}}
```

- Riformatta un file usando una spaziatura uniforme (1 spazio tra due parole e 2 spazi tra due paragrafi):

```
fmt -u {{percorso/del/file}}
```

# fossil ci

Questo comando è un alias per **fossil commit**.

Maggiori informazioni: <https://fossil-scm.org/home/help/commit>.

- Consulta la documentazione del comando originale:

```
tldr fossil-commit
```

# fossil delete

Questo comando è un alias per **fossil rm**.

Maggiori informazioni: <https://fossil-scm.org/home/help/delete>.

- Consulta la documentazione del comando originale:

```
tldr fossil rm
```

# fossil forget

Questo comando è un alias per **fossil rm**.

Maggiori informazioni: <https://fossil-scm.org/home/help/forget>.

- Consulta la documentazione del comando originale:

```
tldr fossil rm
```

# fossil new

Questo comando è un alias per **fossil init**.

Maggiori informazioni: <https://fossil-scm.org/home/help/new>.

- Consulta la documentazione del comando originale:

```
tldr fossil-init
```

# gh cs

Questo comando è un alias per **gh codespace**.

Maggiori informazioni: [https://cli.github.com/manual/gh\\_codespace](https://cli.github.com/manual/gh_codespace).

- Consulta la documentazione del comando originale:

```
tldr gh-codespace
```



# git add

Aggiungi file nuovi o modificati all'area di stage.

Maggiori informazioni: <https://git-scm.com/docs/git-add>.

- Aggiungi un file all'area di stage:

```
git add {{percorso/del/file}}
```

- Aggiungi tutti i file (tracciati e non tracciati):

```
git add -A
```

- Aggiungi solo i file già tracciati:

```
git add -u
```

- Aggiungi anche i file ignorati:

```
git add -f
```

- Aggiungi parti di un file in modo interattivo:

```
git add -p {{percorso/del/file}}
```

# git am

Applica file di patch. Utile quando si ricevono commit via email.

Vedi anche **git format-patch** per generare file di patch.

Maggiori informazioni: <https://git-scm.com/docs/git-am>.

- Applica un file di patch:

```
git am {{percorso/del/file.patch}}
```

- Interrompi l'applicazione di un file di patch:

```
git am --abort
```

- Applica quanto possibile di un file di patch, salvando le parti non applicabili in file .rej:

```
git am --reject {{percorso/del/file.patch}}
```

# git archive

Crea un archivio dei file nell'albero di lavoro.

Maggiori informazioni: <https://git-scm.com/docs/git-archive>.

- Crea un archivio tar del contenuto in HEAD e stampa il risultato su standard output:

```
git archive {{-v|--verbose}} HEAD
```

- Crea un archivio zip del contenuto in HEAD e stampa il risultato su standard output:

```
git archive {{-v|--verbose}} --format zip HEAD
```

- Come sopra, ma scrivi l'archivio zip su file:

```
git archive {{-v|--verbose}} {{-o|--output}} {{percorso/del/  
file.zip}} HEAD
```

- Crea un archivio tar dell'ultimo commit sul ramo specificato:

```
git archive {{-o|--output}} {{percorso/del/file.tar}}  
{{nome_ramo}}
```

- Crea un archivio tar del contenuto di una specifica directory:

```
git archive {{-o|--output}} {{percorso/del/file.tar}} HEAD:  
{{percorso/della/directory}}
```

- Anteponi un percorso ad ogni file così da archivarlo in una directory specifica:

```
git archive {{-o|--output}} {{percorso/del/file.tar}} --  
prefix {{percorso/da/anteporre}}/ HEAD
```

# git bisect

Usa la ricerca binaria per trovare il commit che ha introdotto un bug.

Git salta automaticamente avanti ed indietro nell'albero dei commit per restringere progressivamente il campo fino al commit colpevole.

Maggiori informazioni: <https://git-scm.com/docs/git-bisect>.

- Avvia una ricerca su un intervallo di commit definito dal commit "cattivo" contenente il bug ed un altro commit "buono" privo del bug (solitamente più vecchio):

```
git bisect start {{commit_cattivo}} {{commit_buono}}
```

- Contrassegna ogni commit selezionato da `git bisect` come "bad" (cattivo) o "good" (buono) dopo averlo testato per verificare la presenza del bug:

```
git bisect {{good|bad}}
```

- Una volta che `git bisect` ha individuato il commit che ha introdotto il bug, termina la sessione di ricerca e torna al ramo precedente:

```
git bisect reset
```

- Ignora un commit durante la ricerca (ad esempio uno che fallisce i test per un motivo diverso dal bug ricercato):

```
git bisect skip
```

# git blame

Mostra hash del commit ed ultimo autore per ogni riga di un file.

Maggiori informazioni: <https://git-scm.com/docs/git-blame>.

- Stampa il contenuto di un file annotando ogni riga con l'hash del commit e il nome dell'autore:

```
git blame {{file}}
```

- Stampa il contenuto di un file annotando ogni riga con l'hash del commit e l'indirizzo email dell'autore:

```
git blame {{-e|--show-email}} {{file}}
```

# git branch

Il principale comando Git per lavorare con i rami.

Maggiori informazioni: <https://git-scm.com/docs/git-branch>.

- Elenca i rami locali. Il ramo corrente è evidenziato da un \*:

```
git branch
```

- Elenca tutti i rami (locali e remoti):

```
git branch -a
```

- Crea un nuovo ramo a partire dal commit corrente:

```
git branch {{nome_ramo}}
```

- Crea un nuovo ramo a partire dal commit specificato:

```
git branch {{nome_ramo}} {{hash_commit}}
```

- Rinomina un ramo (non applicabile sul ramo corrente):

```
git branch {{-m|--move}} {{vecchio_nome}} {{nuovo_nome}}
```

- Cancella un ramo locale (non applicabile sul ramo corrente):

```
git branch {{-d|--delete}} {{nome_ramo}}
```

# git bundle

Colloca oggetti e riferimenti in un archivio.

Maggiori informazioni: <https://git-scm.com/docs/git-bundle>.

- Crea un file bundle che contiene tutti gli oggetti e riferimenti di un dato ramo:

```
git bundle create {{percorso/del/file.bundle}} {{nome_ramo}}
```

- Crea un file bundle di tutti i rami:

```
git bundle create {{percorso/del/file.bundle}} --all
```

- Crea un file bundle degli ultimi 5 commit sul ramo corrente:

```
git bundle create {{percorso/del/file.bundle}} -5 {{HEAD}}
```

- Crea un file bundle degli ultimi 7 giorni:

```
git bundle create {{percorso/del/file.bundle}} --since=7.days  
{{HEAD}}
```

- Verifica che un file bundle sia valido e possa essere applicato al repository in uso:

```
git bundle verify {{percorso/del/file.bundle}}
```

- Stampa su standard output la lista di riferimenti contenuti in un bundle:

```
git bundle unbundle {{percorso/del/file.bundle}}
```

- Dato un file bundle, estrai un ramo specifico nel repository in uso:

```
git pull {{percorso/del/file.bundle}} {{nome_ramo}}
```

# git cat-file

Visualizza il contenuto di un oggetto Git nel repository o mostrane dimensione e tipo.

Maggiori informazioni: <https://git-scm.com/docs/git-cat-file>.

- Mostra la dimen[s]ione del commit HEAD in byte:

```
git cat-file -s HEAD
```

- Mostra il [t]ipo (blob, albero, commit, tag) di un oggetto Git:

```
git cat-file -t {{8c442dc3}}
```

- Stam[p]a il contenuto di un oggetto Git, formattato in base al tipo:

```
git cat-file -p {{HEAD~2}}
```



# git check-ignore

Analizza ed esegui il debug di ".gitignore" e dei file esclusi.

Maggiori informazioni: <https://git-scm.com/docs/git-check-ignore>.

- Verifica se un file o una directory sono ignorati:

```
git check-ignore {{percorso/del/file_o_directory}}
```

- Verifica se più file o directory sono ignorati:

```
git check-ignore {{percorso/del/file_o_directory1 percorso/
del/file_o_directory2 ...}}
```

- Leggi i percorsi di file o directory da `stdin` (uno per riga) invece che dalla riga di comando:

```
git check-ignore --stdin < {{percorso/della/
lista_dei_file_o_directory}}
```

- Non controllare nell'indice (usato per determinare il motivo per cui alcuni percorsi non sono ignorati):

```
git check-ignore --no-index {{percorso/del/file_o_directory1
percorso/del/file_o_directory2 ...}}
```

- Includi dettagli sul pattern corrispondente per ogni percorso specificato:

```
git check-ignore --verbose {{percorso/del/file_o_directory1
percorso/del/file_o_directory2 ...}}
```

# git checkout

Cambia rami o ripristina i file dell'albero di lavoro.

Maggiori informazioni: <https://git-scm.com/docs/git-checkout>.

- Crea e passa ad un nuovo ramo:

```
git checkout -b {{nome_ramo}}
```

- Crea e passa ad un nuovo ramo a partire dal riferimento specificato (ramo-locale, remote/ramo-remoto, tag sono alcuni esempi di riferimenti validi):

```
git checkout -b {{nome_ramo}} {{riferimento}}
```

- Passa ad un ramo locale esistente:

```
git checkout {{nome_ramo}}
```

- Passa ad un ramo remoto esistente:

```
git checkout --track {{nome_repository_remoto}}/{{nome_ramo}}
```

- Annulla tutte le modifiche nella directory corrente che non sono state aggiunte all'area di stage (vedi `git reset` per più comandi simili):

```
git checkout .
```

- Annulla tutte le modifiche di un dato file non aggiunte all'area di stage:

```
git checkout {{nome_file}}
```

- Sostituisci un file con il contenuto del suo corrispondente localizzato su un altro ramo:

```
git checkout {{nome_ramo}} -- {{nome_file}}
```

# git cherry-pick

Applica al ramo corrente le modifiche introdotte da commit esistenti.

Per applicare le modifiche ad un altro ramo, usa prima **git checkout** per passare al ramo desiderato.

Maggiori informazioni: <https://git-scm.com/docs/git-cherry-pick>.

- Applica un commit al ramo corrente:

```
git cherry-pick {{commit}}
```

- Applica una sequenza di commit al ramo corrente (vedi anche `git rebase --onto`):

```
git cherry-pick {{commit_iniziale}}~..{{commit_finale}}
```

- Applica un insieme di commit non sequenziali al ramo corrente:

```
git cherry-pick {{commit_1 commit_2 ...}}
```

- Aggiungi le modifiche introdotte da un commit alla directory di lavoro, ma senza creare un nuovo commit:

```
git cherry-pick --no-commit {{commit}}
```

# git clean

Elimina i file non tracciati dall'albero di lavoro.

Maggiori informazioni: <https://git-scm.com/docs/git-clean>.

- Elimina i file che non sono tracciati da Git:

```
git clean
```

- Elimina in modo interattivo i file non tracciati da Git:

```
git clean {{-i|--interactive}}
```

- Mostra quali file non tracciati sarebbero eliminati, senza però eliminarli davvero:

```
git clean --dry-run
```

- Forza l'eliminazione dei file non tracciati da Git:

```
git clean {{-f|--force}}
```

- Forza l'eliminazione delle directory non tracciate da Git:

```
git clean {{-f|--force}} -d
```

- Elimina i file non tracciati, compresi quelli da ignorare elencati in `.gitignore` e `.git/info/exclude`:

```
git clean -x
```

# git clone

Clona un repository esistente.

Maggiori informazioni: <https://git-scm.com/docs/git-clone>.

- Clona un repository remoto esistente:

```
git clone {{url_repository_remoto}}
```

- Clona un repository remoto insieme ai suoi sottomoduli:

```
git clone --recursive {{url_repository_remoto}}
```

- Clona un repository locale:

```
git clone -l {{percorso/del/repository/locale}}
```

- Clona in modalità silenziosa:

```
git clone -q {{url_repository_remoto}}
```

- Clona un repository remoto scaricando solo i 10 commit più recenti del ramo principale (utile per risparmiare tempo):

```
git clone --depth {{10}} {{url_repository_remoto}}
```

# git commit

Salva file nell'area di stage in una nuova istantanea del tuo repository.

Maggiori informazioni: <https://git-scm.com/docs/git-commit>.

- Committa sul repository i file nell'area di stage con un messaggio:

```
git commit -m "{{messaggio}}"
```

- Aggiungi all'area di stage tutti i file modificati e committali con un messaggio:

```
git commit -a -m "{{messaggio}}"
```

- Sostituisci l'ultimo commit con le modifiche attualmente salvate nell'area di stage:

```
git commit --amend
```

- Committa solo i file specificati (tra quelli presenti nell'area di stage):

```
git commit {{percorso/del/file1 percorso/del/file2 ...}}
```

# git config

Configura le impostazioni di uno o più repository Git.

Le configurazioni possono essere sia locali (per il repository corrente) che globali (per l'utente corrente).

Maggiori informazioni: <https://git-scm.com/docs/git-config>.

- Elenca solo le opzioni di configurazione locali (salvate in `.git/config` nel repository corrente):

```
git config --list --local
```

- Elenca solo le opzioni di configurazione globali (salvate in `~/.gitconfig`):

```
git config --list --global
```

- Elenca tutte le opzioni di configurazione impostate, sia locali che globali:

```
git config --list
```

- Mostra il valore di una data opzione di configurazione:

```
git config alias.unstage
```

- Imposta il valore globale di una data opzione di configurazione:

```
git config --global alias.unstage "reset HEAD --"
```

- Ripristina una opzione di configurazione globale al suo valore di default:

```
git config --global --unset alias.unstage
```

# git describe

Rendi il nome di un oggetto Git più leggibile usando i riferimenti disponibili.

Maggiori informazioni: <https://git-scm.com/docs/git-describe>.

- Crea un nome univoco per il commit corrente (il nome contiene i tag più recenti, il numero di commit aggiuntivi, e l'hash breve del commit):

```
git describe
```

- Crea un nome di 4 cifre per l'hash breve del commit:

```
git describe --abbrev={{4}}
```

- Genera un nome che includa anche il percorso di riferimento:

```
git describe --all
```

- Descrivi un tag Git:

```
git describe {{v1.0.0}}
```

- Crea un nome per l'ultimo commit di un dato ramo:

```
git describe {{nome_ramo}}
```



# git diff

Mostra le modifiche ai file tracciati.

Maggiori informazioni: <https://git-scm.com/docs/git-diff>.

- Mostra le modifiche non ancora nell'area di stage:

```
git diff
```

- Mostra tutte le modifiche non ancora salvate in un commit (incluse quelle nell'area di stage):

```
git diff HEAD
```

- Mostra solo le modifiche nell'area di stage (aggiunte, ma non ancora aggiunte ad un commit):

```
git diff --staged
```

- Mostra le modifiche di tutti i commit a partire da una certa data/ora (un'espressione temporale come ad esempio "1 week 2 days" o una data ISO):

```
git diff 'HEAD@{3 months|weeks|days|hours|seconds ago}'
```

- Mostra le differenze tramite delle statistiche come quali file modificati o l'istogramma e il totale delle righe inserite/cancellate:

```
git diff --stat {{commit}}
```

- Mostra un riepilogo dei file creati, rinominati o la cui modalità è cambiata a partire da un dato commit:

```
git diff --summary {{commit}}
```

- Confronta le versioni di un dato file tra due rami o commit:

```
git diff {{ramo_1}}..{{ramo_2}} [--] {{percorso/del/file}}
```

- Confronta le versioni di più file tra il ramo corrente e un altro ramo:

```
git diff {{ramo}}:{{percorso/del/file2}} {{percorso/del/file}}
```

# git difftool

Mostra le modifiche ai file tracciati usando uno strumento Diff esterno. Accetta le stesse opzioni e argomenti di Git diff.

Maggiori informazioni: <https://git-scm.com/docs/git-difftool>.

- Elenca gli strumenti Diff disponibili:

```
git difftool --tool-help
```

- Imposta meld come strumento Diff predefinito:

```
git config --global diff.tool "{{meld}}"
```

- Usa lo strumento Diff predefinito per mostrare le modifiche nell'area di stage:

```
git difftool --staged
```

- Usa uno specifico strumento Diff (opendiff) per mostrare le modifiche a partire da un dato commit:

```
git difftool --tool={{opendiff}} {{commit}}
```

# git fetch

Scarica oggetti e riferimenti da un repository remoto.

Maggiori informazioni: <https://git-scm.com/docs/git-fetch>.

- Scarica le ultime modifiche dal repository remoto di origine (upstream) di default, se definito:

```
git fetch
```

- Scarica i nuovi rami da un dato repository remoto di origine:

```
git fetch {{nome_repository_remoto}}
```

- Scarica le ultime modifiche da tutti i repository remoti di origine:

```
git fetch --all
```

- Scarica anche i tag dal repository remoto di origine:

```
git fetch --tags
```

- Elimina i riferimenti locali ai rami remoti che sono stati eliminati dal repository di origine:

```
git fetch --prune
```

# git format-patch

Prepara file .patch. Utile per l'invio di commit via email.

Vedi anche **git am**, che permette di applicare file .patch.

Maggiori informazioni: <https://git-scm.com/docs/git-format-patch>.

- Crea un file .patch (il nome è assegnato automaticamente) con i commit non ancora inviati al repository remoto:

```
git format-patch {{origin}}
```

- Scrivi su stdout un file .patch per l'intervallo di commit definito dai due commit dati:

```
git format-patch {{commit_1}}..{{commit_2}}
```

- Scrivi un file .patch per gli ultimi 3 commit:

```
git format-patch -{{3}}
```

# git gc

Ottimizza il repository locale ripulendolo dai file non necessari.

Maggiori informazioni: <https://git-scm.com/docs/git-gc>.

- Ottimizza il repository:

```
git gc
```

- Ottimizza il repository in modo più aggressivo, impiegando più tempo:

```
git gc --aggressive
```

- Non eliminare gli oggetti non tracciati (sono eliminati di default):

```
git gc --no-prune
```

- Non mostrare alcun output:

```
git gc --quiet
```

- Mostra utilizzo completo:

```
git gc --help
```

# git-grep

Cerca stringhe nello storico dei file tracciati nel repository.

Supporta molti degli stessi parametri accettati dal comando **grep** tradizionale.

Maggiori informazioni: <https://git-scm.com/docs/git-grep>.

- Cerca una stringa nei file tracciati:

```
git grep {{stringa_ricercata}}
```

- Cerca una stringa nei file tracciati che soddisfano un dato pattern:

```
git grep {{stringa_ricercata}} -- {{file_glob_pattern}}
```

- Cerca una stringa nei file tracciati, sottomoduli inclusi:

```
git grep --recurse-submodules {{stringa_ricercata}}
```

- Cerca una stringa in uno dato momento della cronologia del repository:

```
git grep {{stringa_ricercata}} {{HEAD~2}}
```

- Cerca una stringa in tutti i rami:

```
git grep {{stringa_ricercata}} $(git rev-list --all)
```

# git help

Mostra informazioni di aiuto su Git.

Maggiori informazioni: <https://git-scm.com/docs/git-help>.

- Mostra aiuto per uno specifico sottocomando Git:

```
git help {{sottocomando}}
```

- Visualizza aiuto per un sottocomando Git in un browser web:

```
git help --web {{sottocomando}}
```

- Mostra un elenco dei sottocomandi Git disponibili:

```
git help --all
```

- Mostra un elenco delle guide disponibili:

```
git help --guide
```

- Mostra tutte le possibili variabili di configurazione:

```
git help --config
```

# git ignore-io

Genera file .gitignore usando template predefiniti.

Maggiori informazioni: <https://github.com/tj/git-extras/blob/master/Commands.md#git-ignore-io>.

- Elenca i template disponibili:

```
git ignore-io list
```

- Genera un template .gitignore:

```
git ignore-io {{elemento_a,elemento_b,elemento_n}}
```



# git-imerge

Esegui un'unione (merge) o rebase tra due rami Git in modo incrementale.

Eventuali conflitti tra i due rami sono tracciati in coppie di commit distinti, per semplificarne la risoluzione.

Maggiori informazioni: <https://github.com/mhagger/git-imerge>.

- Avvia un rebase usando imerge (dopo aver fatto checkout sul ramo da spostare):

```
git imerge rebase {{ramo_su_cui_eseguire_il_rebase}}
```

- Avvia un'unione usando imerge (dopo aver fatto checkout sul ramo di destinazione):

```
git imerge merge {{ramo_da_unire}}
```

- Mostra con un diagramma ASCII lo stato di esecuzione dell'unione o rebase:

```
git imerge diagram
```

- Continua con l'operazione di imerge dopo aver risolto i conflitti (dopo aver aggiunto i file in conflitto con `git add`):

```
git imerge continue --no-edit
```

- Concludi l'operazione di imerge dopo aver risolto tutti i conflitti:

```
git imerge finish
```

- Interrompi l'operazione di imerge e ritorna al ramo precedente:

```
git-imerge remove && git checkout {{ramo_precedente}}
```

# git init

Inizializza un nuovo repository Git locale.

Maggiori informazioni: <https://git-scm.com/docs/git-init>.

- Inizializza un nuovo repository locale:

```
git init
```

- Inizializza un repository con il nome specificato per il ramo iniziale:

```
git init --initial-branch={{nome_ramo}}
```

- Inizializza un repository usando SHA256 per gli hash degli oggetti (richiede Git versione 2.29+):

```
git init --object-format={{sha256}}
```

- Inizializza un repository di soli dati, adatto per essere usato come server remoto accessibile via SSH:

```
git init --bare
```

# git instaweb

Helper per avviare un server gitweb.

Maggiori informazioni: <https://git-scm.com/docs/git-instaweb>.

- Avvia un server gitweb dal repository corrente:

```
git instaweb --start
```

- Resta in ascolto solo su localhost:

```
git instaweb --start --local
```

- Resta in ascolto su una porta specifica:

```
git instaweb --start --port {{1234}}
```

- Usa un HTTP daemon specifico:

```
git instaweb --start --httpd {{lighttpd|apache2|mongoose|  
plackup|webrick}}
```

- Avvia automaticamente anche un web browser:

```
git instaweb --start --browser
```

- Interrompi il server gitweb in esecuzione:

```
git instaweb --stop
```

- Riavvia il server gitweb in esecuzione:

```
git instaweb --restart
```

# git lfs

Lavora con file di grandi dimensioni in repository Git.

Maggiori informazioni: <https://git-lfs.com>.

- Inizializza Git LFS:

```
git lfs install
```

- Tieni traccia dei file che soddisfano un criterio glob:

```
git lfs track '{{*.bin}}'
```

- Cambia l'URL endpoint di Git LFS (utile quando server LFS e server Git sono separati):

```
git config {{-f|--file}} .lfsconfig lfs.url  
{{lfs_url_endpoint}}
```

- Elenca i criteri tracciati:

```
git lfs track
```

- Elenca i file tracciati che sono già stati salvati in un commit:

```
git lfs ls-files
```

- Invia tutti gli oggetti Git LFS al server remoto (utile in caso di errori):

```
git lfs push --all {{nome_repository_remoto}} {{nome_ramo}}
```

- Scarica tutti gli oggetti Git LFS:

```
git lfs fetch
```

- Ripristina gli oggetti Git LFS:

```
git lfs checkout
```

# git log

Mostra la cronologia dei commit.

Maggiori informazioni: <https://git-scm.com/docs/git-log>.

- Mostra la sequenza dei commit del ramo del repository in uso, a partire dal commit corrente e andando in ordine cronologico inverso:

```
git log
```

- Mostra la cronologia di un dato file o directory, mostrando anche le modifiche:

```
git log {{-p|-u|--patch}} {{percorso/del/file_o_directory}}
```

- Offri una panoramica dei file che sono cambiati ad ogni commit:

```
git log --stat
```

- Mostra il grafo dei commit nel ramo corrente, includendo solo la prima riga di ogni messaggio di commit:

```
git log --oneline --graph
```

- Mostra il grafo di tutti i commit, tag e rami dell'intero repository:

```
git log --oneline --decorate --all --graph
```

- Mostra solo i commit il cui messaggio contiene una data stringa (ignorando maiuscole/minuscole):

```
git log {{-i|--regexp-ignore-case}} --grep  
{{stringa_da_cercare}}
```

- Mostra gli ultimi N commit di un certo autore:

```
git log {{-n|--max-count}} {{numero}} --author "{{autore}}"
```

- Mostra i commit effettuati tra due date (yyyy-mm-dd):

```
git log --before "{{2017-01-29}}" --after "{{2017-01-17}}"
```

# git ls-remote

Elenca i riferimenti in un repository remoto dato un nome o un URL.

Qualora né nome né URL siano specificati, il ramo predefinito è upstream - se configurato - oppure origin.

Maggiori informazioni: <https://git-scm.com/docs/git-ls-remote>.

- Mostra tutti i riferimenti nel repository remoto predefinito:

```
git ls-remote
```

- Mostra solo i riferimenti HEAD nel repository remoto predefinito:

```
git ls-remote --heads
```

- Mostra solo i riferimenti a tag nel repository remoto predefinito:

```
git ls-remote --tags
```

- Mostra tutti i riferimenti da un repository remoto dato un nome o URL:

```
git ls-remote {{url_repository}}
```

- Filtra i riferimenti da un repository remoto rispetto a un dato criterio:

```
git ls-remote {{nome_repository}} "{{criterio}}"
```

# git ls-tree

Elenca il contenuto di un oggetto albero.

Maggiori informazioni: <https://git-scm.com/docs/git-ls-tree>.

- Mostra il contenuto dell'albero su un ramo:

```
git ls-tree {{nome_ramo}}
```

- Mostra il contenuto dell'albero su un commit, procedendo ricorsivamente nei sotto-alberi:

```
git ls-tree -r {{hash_commit}}
```

- Mostra solo il nome dei file dell'albero su un commit:

```
git ls-tree --name-only {{hash_commit}}
```

# git merge

Esegui un'unione (merge) tra due rami Git.

Maggiori informazioni: <https://git-scm.com/docs/git-merge>.

- Avvia un'unione con il tuo ramo corrente:

```
git merge {{ramo_da_unire}}
```

- Avvia un'unione e cambia il messaggio predefinito:

```
git merge --edit {{ramo_da_unire}}
```

- Avvia un'unione e committa l'operazione:

```
git merge --no-ff {{ramo_da_unire}}
```

- Interrompi un'unione in caso di conflitti:

```
git merge --abort
```



# git mv

Sposta o rinomina file e aggiorna l'indice Git.

Maggiori informazioni: <https://git-scm.com/docs/git-mv>.

- Sposta i file nella repository e aggiungi l'operazione al commit successivo:

```
git mv {{percorso/del/file}} {{nuovo/percorso/del/file}}
```

- Rinomina i file e aggiungi l'operazione al commit successivo:

```
git mv {{file}} {{file_rinominato}}
```

- Sposta sovrascrivendo eventuali file esistenti nel percorso di destinazione:

```
git mv --force {{percorso/del/file}} {{nuovo/percorso/del/  
file}}
```

# git prune

Elimina dal database degli oggetti quelli non più raggiungibili.

Questo comando è usato più spesso internamente da Git gc piuttosto che in modo diretto.

Maggiori informazioni: <https://git-scm.com/docs/git-prune>.

- Elenca quali oggetti saranno eliminati da Git prune senza eliminarli definitivamente:

```
git prune --dry-run
```

- Elimina gli oggetti non raggiungibili e stampare un elenco su `stdout`:

```
git prune --verbose
```

- Elimina gli oggetti non raggiungibili, mostrando lo stato di avanzamento:

```
git prune --progress
```

# git pull

Scarica oggetti e riferimenti (fetch) da un repository remoto e avvia un'unione (merge) con il ramo corrente.

Maggiori informazioni: <https://git-scm.com/docs/git-pull>.

- Scarica le ultime modifiche dal repository remoto e avvia un'unione:

```
git pull
```

- Scarica le ultime modifiche dal repository remoto e avvia un rebase:

```
git pull --rebase
```

- Scarica le ultime modifiche da uno specifico ramo remoto e avvia un'unione con il ramo corrente:

```
git pull {{nome_repository_remoto}} {{nome_ramo}}
```

# git push

Invia i commit ad un repository remoto.

Maggiori informazioni: <https://git-scm.com/docs/git-push>.

- Invia le modifiche fatte nel ramo corrente locale al corrispondente ramo remoto:

```
git push
```

- Invia le modifiche fatte in uno specifico ramo locale al corrispondente ramo remoto:

```
git push {{nome_repository_remoto}} {{nome_ramo}}
```

- Invia le modifiche fatte in uno specifico ramo locale al corrispondente ramo remoto ed imposta il ramo remoto come destinazione di default per i push/pull del ramo locale:

```
git push -u {{nome_repository_remoto}} {{nome_ramo}}
```

- Invia le modifiche fatte in uno specifico ramo locale ad uno specifico ramo remoto:

```
git push {{nome_repository_remoto}} {{nome_ramo}}:  
{{nome_ramo_remoto}}
```

- Invia le modifiche fatte in ogni ramo locale ai corrispondenti rami remoti in uno specifico repository remoto:

```
git push --all {{nome_repository_remoto}}
```

- Cancella un ramo di un repository remoto:

```
git push {{nome_repository_remoto}} --delete  
{{nome_ramo_remoto}}
```

- Cancella i rami remoti che non hanno un ramo locale corrispondente:

```
git push --prune {{nome_repository_remoto}}
```

- Pubblica i tag che non sono già presenti nel repository remoto:

```
git push --tags
```

# git rebase

Applica i commit di un ramo su un ramo differente.

Tipicamente usato per riallineare (rebase) due rami, creando copie dei commit nella nuova posizione.

Maggiori informazioni: <https://git-scm.com/docs/git-rebase>.

- Riallinea il ramo corrente con il ramo specificato:

```
git rebase {{ramo_della_nuova_base}}
```

- Avvia un rebase interattivo, che consente di riordinare, omettere, unire o modificare i commit:

```
git rebase {{-i|--interactive}} {{nome_ramo_o_commit_hash}}
```

- Prosegui con un rebase che era stato sospeso da un errore di unione, dopo aver risolto i conflitti:

```
git rebase --continue
```

- Prosegui con un rebase che era stato sospeso da conflitti di unione, ignorando i commit in conflitto:

```
git rebase --skip
```

- Interrompi un rebase in corso (ad esempio perché interrotto da un conflitto di unione):

```
git rebase --abort
```

- Sposta parti del ramo corrente su una base differente, specificando la vecchia base di partenza:

```
git rebase --onto {{nuova_base}} {{vecchia_base}}
```

- Applica gli ultimi 5 commit locali, consentendo di riordinarli, ometterli, unirli o modificarli:

```
git rebase {{-i|--interactive}} {{HEAD~5}}
```

- Risolvi automaticamente i conflitti a favore del ramo di versione corrente (la parola chiave `theirs` ha qui un significato opposto):

```
git rebase {{-X|--strategy-option}} theirs {{nome_ramo}}
```

# git reflog

Mostra la cronologia delle modifiche a riferimenti locali come HEAD, rami o tag.

Maggiori informazioni: <https://git-scm.com/docs/git-reflog>.

- Mostra il reflog di HEAD:

```
git reflog
```

- Mostra il reflog di uno specifico ramo:

```
git reflog {{nome_ramo}}
```

- Mostra le ultime 5 voci del reflog:

```
git reflog {{-n|--max-count}} 5
```

# git remote

Gestisci i collegamenti remoti ("remote") di un repository locale.

Maggiori informazioni: <https://git-scm.com/docs/git-remote>.

- Mostra l'elenco dei collegamenti remoti, con il loro nome e URL:

```
git remote {{-v|--verbose}}
```

- Mostra informazioni su un remote:

```
git remote show {{nome_remote}}
```

- Aggiungi un remote:

```
git remote add {{nome_remote}} {{url_remote}}
```

- Modifica l'URL di un remote (usa --add per preservare gli URL esistenti):

```
git remote set-url {{nome_remoto}} {{nuovo_url}}
```

- Elimina un remote:

```
git remote remove {{nome_remote}}
```

- Rinomina un remote:

```
git remote rename {{vecchio_nome}} {{nuovo_nome}}
```

# git repack

Comprimi gli oggetti decompressi in un repository Git.

Maggiori informazioni: <https://git-scm.com/docs/git-repack>.

- Comprimi gli oggetti decompressi nella directory corrente:

```
git repack
```

- Rimuovi eventuali archivi ridondanti dopo la compressione:

```
git repack -d
```



# git replace

Crea, elenca, ed elimina riferimenti ad oggetti sostituiti.

Maggiori informazioni: <https://git-scm.com/docs/git-replace>.

- Sostituisci un commit con un altro, senza modificare gli altri commit:

```
git replace {{oggetto}} {{oggetto_sostitutivo}}
```

- Cancella riferimenti esistenti ad un oggetto sostituito:

```
git replace --delete {{oggetto}}
```

- Modifica il contenuto di un oggetto in modo interattivo:

```
git replace --edit {{oggetto}}
```

# git reset

Annulla commit o rimuovi modifiche dall'area di stage, reimpostando l'HEAD corrente su uno specifico stato.

Se viene fornito un percorso, il comando reset si interpreta come "rimuovi dall'area di stage"; se viene fornito l'hash di un commit o un ramo, si interpreta come "annulla commit".

Maggiori informazioni: <https://git-scm.com/docs/git-reset>.

- Rimuovi tutto dall'area di stage:

```
git reset
```

- Rimuovi dall'area di stage uno o più file:

```
git reset {{percorso/del/file1 percorso/del/file2 ...}}
```

- Rimuovi dall'area di stage solo alcune porzioni di un file in modo interattivo:

```
git reset --patch {{percorso/del/file}}
```

- Annulla l'ultimo commit, preservando tutte le modifiche nel filesystem:

```
git reset HEAD~
```

- Annulla gli ultimi due commit, aggiungendo all'area di stage le modifiche relative:

```
git reset --soft HEAD~2
```

- Annulla le modifiche non committate, indipendentemente se siano presenti nell'area di stage o meno (usa `git checkout` per queste ultime):

```
git reset --hard
```

- Reimposta il repository su un dato commit, annullando qualsiasi tipo di modifica precedente:

```
git reset --hard {{commit}}
```

# git restore

Ripristina i file dell'albero di lavoro. Richiede versioni di Git successive alla 2.23.

Vedi anche **git checkout**.

Maggiori informazioni: <https://git-scm.com/docs/git-restore>.

- Ripristina un file cancellato dal contenuto del commit corrente (HEAD):

```
git restore {{percorso/del/file}}
```

- Ripristina un file alla versione di un commit differente:

```
git restore --source {{commit}} {{percorso/del/file}}
```

- Annulla le modifiche ai file nell'area di stage, ripristinandoli all'HEAD:

```
git restore .
```

# git rev-list

Elenca le revisioni (commit) in ordine cronologico inverso.

Maggiori informazioni: <https://git-scm.com/docs/git-rev-list>.

- Mostra tutti i commit del ramo corrente:

```
git rev-list {{HEAD}}
```

- Mostra i commit più recenti di una certa data, su uno specifico ramo:

```
git rev-list --since={{'2019-12-01 00:00:00'}} {{nome_ramo}}
```

- Mostra tutti i commit di unione (merge commit) associati a uno specifico commit:

```
git rev-list --merges {{commit}}
```

# git rev-parse

Mostra i metadati relativi a specifiche revisioni.

Maggiori informazioni: <https://git-scm.com/docs/git-rev-parse>.

- Mostra l'hash del commit di un ramo:

```
git rev-parse {{nome_ramo}}
```

- Mostra il nome del ramo corrente:

```
git rev-parse --abbrev-ref {{HEAD}}
```

- Mostra il percorso assoluto della directory di root:

```
git rev-parse --show-toplevel
```

# git revert

Crea nuovi commit che invertano i risultati dei commit precedenti.

Maggiori informazioni: <https://git-scm.com/docs/git-revert>.

- Inverti il commit più recente:

```
git revert {{HEAD}}
```

- Inverti il quintultimo commit:

```
git revert HEAD~{{4}}
```

- Inverti più commit:

```
git revert {{nome_ramo~5..nome_ramo~2}}
```

- Inverti senza creare nuovi commit, ma modificando l'albero di lavoro:

```
git revert -n {{0c01a9..9a1743}}
```

# git rm

Cancella file dall'indice del repository e dal filesystem locale.

Maggiori informazioni: <https://git-scm.com/docs/git-rm>.

- Cancella un file dall'indice del repository e dal filesystem:

```
git rm {{file}}
```

- Cancella una directory:

```
git rm -r {{directory}}
```

- Cancella un file dall'indice del repository ma non dal filesystem locale:

```
git rm --cached {{file}}
```

# git send-email

Invia una raccolta di patch via email.

Le patch possono essere specificate come file, directory, o liste di revisione.

Maggiori informazioni: <https://git-scm.com/docs/git-send-email>.

- Invia l'ultimo commit nel ramo corrente:

```
git send-email -1
```

- Invia un commit specifico:

```
git send-email -1 {{commit}}
```

- Invia 10 commit nel ramo corrente:

```
git send-email {{-10}}
```

- Invia un'email con un messaggio introduttivo alla serie di patch:

```
git send-email -{{numero_di_commit}} --compose
```

- Revisiona e modifica il messaggio email per ogni patch da inviare:

```
git send-email -{{numero_di_commit}} --annotate
```



# git shortlog

Riassume l'output di **git log**.

Maggiori informazioni: <https://git-scm.com/docs/git-shortlog>.

- Mostra un riassunto dei commit fatti, raggruppati alfabeticamente per nome dell'autore:

```
git shortlog
```

- Mostra un riassunto dei commit fatti, ordinati per numero di commit:

```
git shortlog {{-n|--numbered}}
```

- Mostra un riassunto dei commit fatti, raggruppati per identità dell'utente che ha eseguito il commit (nome e email):

```
git shortlog {{-c|--committer}}
```

- Mostra un riassunto degli ultimi 5 commit (richiesti sottoforma di intervallo di revisione):

```
git shortlog HEAD~5..HEAD
```

- Mostra tutti gli utenti, email e numero di commit nel ramo corrente:

```
git shortlog {{-s|--summary}} {{-n|--numbered}} {{-e|--email}}
```

- Mostra tutti gli utenti, email e numero di commit in tutti i rami:

```
git shortlog {{-s|--summary}} {{-n|--numbered}} {{-e|--email}} --all
```

# git show-branch

Mostra rami e relativi commit.

Maggiori informazioni: <https://git-scm.com/docs/git-show-branch>.

- Mostra un sommario degli ultimi commit in un ramo:

```
git show-branch {{nome_ramo|riferimento|commit}}
```

- Confronta commit nella cronologia di più commit o rami:

```
git show-branch {{nome_ramo|riferimento|commit}}
```

- Confronta tutti i rami remoti tracciati:

```
git show-branch --remotes
```

- Confronta i rami locali e remoti:

```
git show-branch --all
```

- Mostra gli ultimi commit di tutti i rami:

```
git show-branch --all --list
```

- Confronta un dato ramo con quello corrente:

```
git show-branch --current {{commit|nome_ramo|riferimento}}
```

- Mostra il nome del commit e non il nome relativo:

```
git show-branch --sha1-name --current {{current|nome_ramo|  
riferimento}}
```

- Mostra un numero aggiuntivo di commit oltre il predecessore comune:

```
git show-branch --more {{5}} {{commit|nome_ramo|riferimento}}  
{{commit|nome_ramo|riferimento}} {{...}}
```

# git show-ref

Elenca i riferimenti.

Maggiori informazioni: <https://git-scm.com/docs/git-show-ref>.

- Mostra tutti i riferimenti nel repository:

```
git show-ref
```

- Mostra solo i riferimenti agli HEAD:

```
git show-ref --heads
```

- Mostra solo i riferimenti ai tag:

```
git show-ref --tags
```

- Verifica che un certo riferimento esista:

```
git show-ref --verify {{percorso/del/riferimento}}
```

# git show

Mostra vari tipi di oggetti Git (commit, tag, etc.).

Maggiori informazioni: <https://git-scm.com/docs/git-show>.

- Mostra informazioni sull'ultimo commit (hash, messaggio, modifiche, ed altri metadati):

```
git show
```

- Mostra informazioni su un dato commit:

```
git show {{commit}}
```

- Mostra informazioni sul commit associato ad un tag specifico:

```
git show {{tag}}
```

- Mostra informazioni sul terzo commit dalla cima del ramo:

```
git show {{ramo}}~{{3}}
```

- Mostra il messaggio di commit su linea singola, senza mostrare il diff:

```
git show --oneline -s {{commit}}
```

- Mostra solo la lista dei file modificati in un commit:

```
git show --stat {{commit}}
```

- Mostra il contenuto di un file ad una data revisione (ad esempio, in un ramo, tag o commit):

```
git show {{revisione}}:{{percorso/del/file}}
```

# git sizer

Calcola varie metriche dimensionali su repository Git e avvisa quando i loro valori rilevano problematicità.

Maggiori informazioni: <https://github.com/github/git-sizer>.

- Mostra solo le statistiche che hanno un livello di rischio maggiore di 0:

```
git sizer
```

- Mostra tutte le statistiche:

```
git sizer -v
```

- Mostra opzioni aggiuntive:

```
git sizer -h
```

# git stage

Questo comando è un alias per **git add**.

Maggiori informazioni: <https://git-scm.com/docs/git-stage>.

- Consulta la documentazione del comando originale:

```
tldr git add
```

# git stash

Salva in un'area temporanea (stash) modifiche Git locali.

Maggiori informazioni: <https://git-scm.com/docs/git-stash>.

- Salva in un'area temporanea modifiche locali, tranne i file nuovi e non tracciati:

```
git stash push -m {{messaggio_di_stash_opzionale}}
```

- Includi nello stash anche i file nuovi e non tracciati:

```
git stash -u
```

- Seleziona per lo stash parti di file modificati in modo interattivo:

```
git stash -p
```

- Elenca tutti gli stash, mostrandone il nome, ramo relativo e messaggio:

```
git stash list
```

- Applica uno stash (quello predefinito è l'ultimo, chiamato stash@{0}):

```
git stash apply {{nome_o_commit_stash_opzionale}}
```

- Applica uno stash (il predefinito è stash@{0}) e rimuovilo dalla lista degli stash se non ha causato conflitti:

```
git stash pop {{nome_stash_opzionale}}
```

- Rimuovi tutti gli stash:

```
git stash clear
```

# git status

Mostra le modifiche ai file in un repository Git.

Elenca i file modificati, aggiunti e cancellati rispetto al commit corrente.

Maggiori informazioni: <https://git-scm.com/docs/git-status>.

- Mostra i file modificati che non sono stati ancora committati:

```
git status
```

- Mostra l'output in formato ridotto:

```
git status -s
```

- Nascondi i file non tracciati dall'output:

```
git status --untracked-files=no
```

- Mostra informazioni sul ramo ed in formato ridotto:

```
git status -sb
```



# git submodule

Ispeziona, aggiorna e gestisce moduli secondari (submodule).

Maggiori informazioni: <https://git-scm.com/docs/git-submodule>.

- Installa specifici moduli secondari di un repository:

```
git submodule update --init --recursive
```

- Aggiungi un repository Git come modulo secondario:

```
git submodule add {{url_repository}}
```

- Aggiungi un repository Git come modulo secondario alla directory specificata:

```
git submodule add {{url_repository}} {{percorso/della/  
directory}}
```

- Aggiorna tutti i moduli secondari al loro commit più recente:

```
git submodule foreach git pull
```

# git subtree

Strumento per gestire le dipendenze di un progetto come progetti secondari.

Maggiori informazioni: <https://manpages.debian.org/latest/git-man/git-subtree.1.html>.

- Aggiungi un repository Git come albero secondario:

```
git subtree add --prefix={{percorso/della/directory/}} --squash {{url_repository}} {{master}}
```

- Aggiorna l'albero secondario di un repository al suo commit più recente:

```
git subtree pull --prefix={{percorso/della/directory/}} {{url_repository}} {{master}}
```

- Unisci un albero secondario al ramo principale (master):

```
git subtree merge --prefix={{percorso/della/directory/}} --squash {{url_repository}} {{master}}
```

- Invia commit all'albero secondario di un repository:

```
git subtree push --prefix={{percorso/della/directory/}} {{url_repository}} {{master}}
```

- Estrai la cronologia di un nuovo progetto dalla cronologia di un albero secondario:

```
git subtree split --prefix={{percorso/della/directory/}} {{url_repository}} -b {{nome_ramo}}
```

# git svn

Operazioni bidirezionali tra repository Subversion e Git.

Maggiori informazioni: <https://git-scm.com/docs/git-svn>.

- Clona un repository SVN:

```
git svn clone {{https://esempio.com/repo_subversion}}  
{{directory_locale}}
```

- Clona un repository SVN a partire da uno specifico numero di revisione:

```
git svn clone {{-r|--revision}} {{1234}}:HEAD {{https://  
svn.esempio.net/subversion/repo}} {{directory_locale}}
```

- Aggiorna una copia locale da un repository SVN remoto:

```
git svn rebase
```

- Scarica aggiornamenti da un repository SVN remoto senza spostare l'HEAD  
Git:

```
git svn fetch
```

- Invia un commit a un repository SVN:

```
git svn commit
```

# git switch

Passa ad altri rami. Richiede versioni di Git successive alla 2.23.

Vedi anche **git checkout**.

Maggiori informazioni: <https://git-scm.com/docs/git-switch>.

- Passa ad un altro ramo esistente:

```
git switch {{nome_ramo}}
```

- Crea un nuovo ramo e passa a quel ramo:

```
git switch --create {{nome_ramo}}
```

- Crea un nuovo ramo a partire da un commit esistente e passa a quel ramo:

```
git switch --create {{nome_ramo}} {{commit}}
```

- Torna al ramo precedente:

```
git switch -
```

- Passa ad un ramo ed aggiorna tutti i moduli secondari associati:

```
git switch --recurse-submodules {{nome_ramo}}
```

- Passa ad un ramo e uniscilo automaticamente al ramo corrente, include le modifiche non committate:

```
git switch --merge {{nome_ramo}}
```

# git tag

Crea, elenca, cancella o verifica tag.

Un tag è un riferimento statico a uno specifico commit.

Maggiori informazioni: <https://git-scm.com/docs/git-tag>.

- Mostra tutti i tag:

```
git tag
```

- Crea un tag con un nome, puntandolo al commit corrente:

```
git tag {{nome_tag}}
```

- Crea un tag con un nome, puntandolo ad un dato commit:

```
git tag {{nome_tag}} {{commit}}
```

- Crea un tag annotandolo con un messaggio:

```
git tag {{nome_tag}} -m {{messaggio_tag}}
```

- Cancella un tag, dato il suo nome:

```
git tag -d {{nome_tag}}
```

- Scarica tag aggiornati da upstream:

```
git fetch --tags
```

- Mostra tutti i tag i cui predecessori includono uno specifico commit:

```
git tag --contains {{commit}}
```

# git update-ref

Crea, aggiorna e cancella riferimenti Git.

Maggiori informazioni: <https://git-scm.com/docs/git-update-ref>.

- Cancella un riferimento, utile per resettare il primo commit in modo soft:

```
git update-ref -d {{HEAD}}
```

- Aggiorna un riferimento con un messaggio:

```
git update-ref -m {{messaggio}} {{HEAD}} {{4e95e05}}
```

# git worktree

Gestisci gli alberi di lavoro collegati allo stesso repository.

Maggiori informazioni: <https://git-scm.com/docs/git-worktree>.

- Crea una nuova directory a partire da uno specifico ramo:

```
git worktree add {{percorso/della/directory}} {{ramo}}
```

- Crea una nuova directory a partire da un nuovo ramo:

```
git worktree add {{percorso/della/directory}} -b  
{{nuovo_ramo}}
```

- Mostra tutte le directory di lavoro collegate al repository corrente:

```
git worktree list
```

- Cancella un albero di lavoro (dopo averne cancellato la directory):

```
git worktree prune
```

# git

Sistema di controllo versione distribuito.

Alcuni comandi aggiuntivi, come **commit**, hanno la propria documentazione.

Maggiori informazioni: <https://git-scm.com/>.

- Controlla la versione di Git:

```
git --version
```

- Mostra informazioni di aiuto generali:

```
git --help
```

- Mostra aiuto per uno specifico comando:

```
git help {{command}}
```

- Esegui un comando Git:

```
git {{command}}
```



# gnmic sub

Questo comando è un alias per **gnmic subscribe**.

Maggiori informazioni: <https://gnmic.kmrd.dev/cmd/subscribe>.

- Consulta la documentazione del comando originale:

```
tldr gnmic subscribe
```

# gocryptfs

Filesystem crittografato scritto in Go.

Maggiori informazioni: <https://github.com/rfjakob/gocryptfs>.

- Inizializzare un filesystem crittografato:

```
gocryptfs -init {{percorso/della/directory_cifrata}}
```

- Montare un filesystem crittografato:

```
gocryptfs {{percorso/della/directory_cifrata}} {{percorso/  
del/punto_di_mount}}
```

- Montare un filesystem usando la master key invece della password:

```
gocryptfs --masterkey {{percorso/della/directory_cifrata}}  
{{percorso/del/punto_di_mount}}
```

- Cambiare la password:

```
gocryptfs --passwd {{percorso/della/directory_cifrata}}
```

- Generare uno snapshot cifrato di una directory:

```
gocryptfs --reverse {{percorso/della/directory}} {{percorso/  
della/directory_cifrata}}
```

# google-chrome

Questo comando è un alias per **chromium**.

Maggiori informazioni: <https://chrome.google.com>.

- Consulta la documentazione del comando originale:

```
tldr chromium
```

# head

Stampa a schermo le prime linee di un file.

Maggiori informazioni: <https://manned.org/head.1p>.

- Stampa a schermo le prime linee di un file:

```
head -n {{numero_di_linee}} {{file}}
```

- Stampa a schermo i primi byte di un file:

```
head -c {{numero_di_byte}} {{file}}
```

- Stampa a schermo tutto il file meno le ultime linee:

```
head -n -{{numero_di_linee}} {{file}}
```

- Stampa a schermo tutto il file meno gli ultimi byte:

```
head -c -{{numero_di_byte}} {{file}}
```

# hello

Stampa a schermo "Hello, world!", "hello, world" oppure del testo personalizzabile.

Maggiori informazioni: <https://www.gnu.org/software/hello/>.

- Stampa a schermo "Hello, world!":

```
hello
```

- Stampa a schermo "hello, world", nel modo tradizionale:

```
hello --traditional
```

- Stampa a schermo un messaggio:

```
hello --greeting="{{messaggio}}"
```

# helm

Helm è un gestore di pacchetti per Kubernetes.

Alcuni comandi aggiuntivi, come **install**, hanno la propria documentazione.

Maggiori informazioni: <https://helm.sh/>.

- Crea una helm chart:

```
helm create {{nome_chart}}
```

- Aggiungi un nuovo repository helm:

```
helm repo add {{nome_repository}}
```

- Elenca i repositories helm:

```
helm repo list
```

- Aggiorna i repositories helm:

```
helm repo update
```

- Cancella un repository helm:

```
helm repo remove {{nome_repository}}
```

- Installa una helm chart:

```
helm install {{nome_chart}} {{nome_repository}}/  
{{nome_chart}}
```

- Scarica una helm chart sottoforma di archivio tar:

```
helm get {{nome_chart_rilasciata}}
```

- Aggiorna le dipendenze helm:

```
helm dependency update
```

# hx

Questo comando è un alias per **helix**.

- Consulta la documentazione del comando originale:

`tldr helix`

# hyperfine

Strumento di benchmarking con interfaccia CLI.

Maggiori informazioni: <https://github.com/sharkdp/hyperfine/>.

- Esegui un benchmark di base, eseguendo almeno 10 esecuzioni:

```
hyperfine '{{make}}'
```

- Esegui un benchmark comparativo:

```
hyperfine '{{make target1}}' '{{make target2}}'
```

- Modifica il numero minimo di esecuzioni di benchmark:

```
hyperfine --min-runs {{7}} '{{make}}'
```

- Esegui benchmark con periodo di riscaldamento:

```
hyperfine --warmup {{5}} '{{make}}'
```

- Esegui un comando prima di ogni esecuzione di benchmark (per cancellare le cache, etc.):

```
hyperfine --prepare '{{make clean}}' '{{make}}'
```

- Esegui un benchmark in cui un singolo parametro cambia per ogni esecuzione:

```
hyperfine --prepare '{{make clean}}' --parameter-scan  
{{num_threads}} {{1}} {{10}} '{{make -j {{num_threads}}}'
```



# id3tag

Strumento per leggere, scrivere, e manipolare i tag (etichette) ID3v1 e ID3v2 di file MP3.

Maggiori informazioni: <https://manned.org/id3tag>.

- Imposta l'etichetta dell'artista e del titolo in un file MP3:

```
id3tag --artist {{artista}} --song {{titolo}} {{percorso/del/  
file.mp3}}
```

- Imposta il titolo dell'album di tutti i file MP3 nella directory corrente:

```
id3tag --album {{album}} {{*.mp3}}
```

- Fornisce altro aiuto:

```
id3tag --help
```

# jrnl

Una semplice applicazione da linea di comando per tenere un diario.

Maggiori informazioni: <https://jrnl.sh>.

- Inserisci una nuova nota con il tuo editor:

```
jrnl
```

- Inserimento veloce di una nota:

```
jrnl {{today at 3am}}: {{titolo}}. {{contenuto}}
```

- Mostra le ultime dieci note inserite:

```
jrnl -n {{10}}
```

- Mostra tutto quello che è successo dall'inizio dello scorso anno fino all'inizio di marzo:

```
jrnl -from "{{last year}}" -until {{march}}
```

- Modifica tutte le note taggate con "texas" e "history":

```
jrnl {{@texas}} -and {{@history}} --edit
```

# kafkacat

Questo comando è un alias per **kcat**.

- Consulta la documentazione del comando originale:

`tldr kcat`

# kubectl

Interfaccia interattiva da linea di comando per eseguire comandi sui clusters Kubernetes.

Alcuni comandi aggiuntivi, come **run**, hanno la propria documentazione..

Maggiori informazioni: <https://kubernetes.io/docs/reference/kubectl/>.

- Elenca le informazioni su una risorsa in maniera dettagliata:

```
kubectl get {{pod|service|deployment|ingress|...}} -o wide
```

- Aggiorna il pod specificato con l'etichetta 'unhealthy' e il valore 'true':

```
kubectl label pods {{nome}} unhealthy=true
```

- Elenca tutte le risorse:

```
kubectl get all
```

- Mostra l'utilizzo delle risorse (CPU/Memory/Storage) di nodi o pods:

```
kubectl top {{pod|nodo}}
```

- Mostra l'indirizzo del master e i servizi del cluster:

```
kubectl cluster-info
```

- Mostra la spiegazione di un campo specifico:

```
kubectl explain {{pods.spec.containers}}
```

- Mostra i logs di un container in un pod o in una risorsa specificata:

```
kubectl logs {{nome_pod}}
```

- Esegue un comando in un pod esistente:

```
kubectl exec {{nome_pod}} -- {{ls /}}
```

# less

Apri un file per lettura interattiva, permettendo di scorrere e cercare testo al suo interno.

Maggiori informazioni: <https://greenwoodsoftware.com/less/>.

- Apri un file:

```
less {{file}}
```

- Muoviti una pagina in giù / su:

```
<Spazio> (giù), b (su)
```

- Vai alla fine o all'inizio del file:

```
G (fine), g (inizio)
```

- Cerca una stringa in avanti (usa n/N per muoverti al risultato successivo/precedente):

```
/{{stringa}}
```

- Cerca una stringa indietro (usa n/N per muoverti al risultato successivo/precedente):

```
?{{stringa}}
```

- Aggiorna il file dinamicamente:

```
F
```

- Apri il file in un editor di testo:

```
v
```

- Esci:

```
q
```

# llvm-ar

Questo comando è un alias per **ar**.

- Consulta la documentazione del comando originale:

`tldr ar`

# llvm-g++

Questo comando è un alias per **clang++**.

- Consulta la documentazione del comando originale:

```
tldr clang++
```

# llvm-gcc

Questo comando è un alias per **clang**.

- Consulta la documentazione del comando originale:

`tldr clang`



# llvm-nm

Questo comando è un alias per **nm**.

- Consulta la documentazione del comando originale:

`tldr nm`

# llvm-objdump

Questo comando è un alias per **objdump**.

- Consulta la documentazione del comando originale:

```
tldr objdump
```

# llvm-strings

Questo comando è un alias per **strings**.

- Consulta la documentazione del comando originale:

```
tldr strings
```

# ln

Crea un collegamento a un file o a una directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/ln>.

- Crea un collegamento simbolico a un file (o directory):

```
ln -s {{percorso/del/file}} {{percorso/del/collegamento}}
```

- Sovrascrivi un collegamento esistente in modo che punti a un nuovo file:

```
ln -sf {{percorso/del/nuovo/file}} {{percorso/del/collegamento}}
```

- Crea un collegamento fisico a un file:

```
ln {{percorso/del/file}} {{percorso/del/collegamento}}
```

# ls

Elenca i contenuti di una directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/ls>.

- Elenca i file nella directory corrente, uno per riga:

```
ls -l
```

- Elenca tutti i file, inclusi quelli nascosti:

```
ls -a
```

- Elenca tutti i file, aggiungendo / in coda ai nomi delle directory:

```
ls -F
```

- Elenca tutti i file e mostra informazioni (permessi, proprietà, dimensione e data di ultima modifica):

```
ls -la
```

- Elenca tutti i file e mostra informazioni con la dimensione esposta usando un formato facilmente leggibile (KiB, MiB, GiB):

```
ls -lh
```

- Elenca tutti i file e mostra informazioni, ordinandoli per dimensione decrescente:

```
ls -lsR
```

- Elenca tutti i file e mostra informazioni, ordinandoli per data di ultima modifica (i più vecchi prima):

```
ls -ltr
```

- Elenca solo le directory:

```
ls -d */
```

# lzcat

Questo comando è un alias per **xz**.

Maggiori informazioni: <https://manned.org/lzcat>.

- Consulta la documentazione del comando originale:

```
tldr xz
```

# lzma

Questo comando è un alias per **xz**.

Maggiori informazioni: <https://manned.org/lzma>.

- Consulta la documentazione del comando originale:

```
tldr xz
```

# magick convert

Strumento della suite immagineMagick per la conversione di immagini.

Maggiori informazioni: <https://imagemagick.org/script/convert.php>.

- Converti un'immagine da JPEG a PNG:

```
magick convert {{immagine.jpg}} {{immagine.png}}
```

- Scala un'immagine al 50% delle sue dimensioni originali:

```
magick convert {{immagine.png}} -resize 50% {{immagine2.png}}
```

- Scala un'immagine ad una dimensione massima di 640x480 mantenendo le proporzioni originali:

```
magick convert {{immagine.png}} -resize 640x480  
{{immagine2.png}}
```

- Concatena più immagini orizzontalmente:

```
magick convert {{immagine1.png}} {{immagine2.png}}  
{{immagine3.png}} +append {{immagine123.png}}
```

- Crea una GIF da una serie di immagini con un intervallo di 100ms tra ogni immagine:

```
magick convert {{immagine1.png}} {{immagine2.png}}  
{{immagine3.png}} -delay {{100}} {{animazione.gif}}
```

- Crea un'immagine a tinta unita di un determinato colore:

```
magick convert -size {{800x600}} "xc:{{#ff0000}}"  
{{immagine.png}}
```



# man

Formatta e mostra pagine manuale.

Maggiori informazioni: <https://manned.org/man>.

- Mostra la pagina di manuale di un comando:

```
man {{comando}}
```

- Mostra la pagina di manuale per un comando dalla sezione 7:

```
man {{7}} {{comando}}
```

- Mostra il percorso in cui vengono cercate le pagine:

```
man --path
```

- Mostra la posizione di una pagina invece che la pagina stessa:

```
man -w {{comando}}
```

- Cerca pagine di manuale che contengano una certa stringa:

```
man -k {{ricerca}}
```

# md5sum

Calcola i checksum crittografici di tipo MD5.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/md5sum>.

- Calcolare il checksum MD5 di un file:

```
md5sum {{percorso/del/file}}
```

- Calcola i checksum MD5 per più di un file:

```
md5sum {{percorso/del/file1}} {{percorso/del/file2}}
```

- Verifica che tutti i file abbiano checksum corrispondenti al file di MD5SUM:

```
md5sum -c {{percorso/del/file.md5}}
```

- Calcola il checksum MD5 dal standard input:

```
echo "{{testo}}" | md5sum
```

# minifab

Strumento per semplificare il settaggio e il deployment di una blockchain Hyperledger Fabric.

Maggiori informazioni: <https://github.com/hyperledger-labs/minifabric>.

- Crea la blockchain Hyperledger Fabric:

```
minifab up -i {{versione_minifab}}
```

- Rimuovi la blockchain Hyperledger Fabric:

```
minifab down
```

- Installa smart contract su un canale:

```
minifab install -n {{nome_smart_contract}}
```

- Installa smart contract su un canale specificando la versione:

```
minifab install -n {{nome_smart_contract}} -v  
{{versione_smart_contract}}
```

- Inizializza smart contract dopo installazione/aggiornamento dello stesso:

```
minifab approve,commit,initialize,discover
```

- Interroga smart contract con argomenti:

```
minifab invoke -n {{nome_smart_contract}} -p  
'{{nome_metodo}}', '{{arg0}}', '{{arg1}}', ...'
```

- Interroga la blockchain:

```
minifab blockquery {{numero_blocco}}
```

- Esegui direttamente l'applicazione:

```
minifab apprun -l {{linguaggio_di_programmazione}}
```

# mkdir

Crea directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/mkdir>.

- Crea una directory nella directory corrente o in un dato percorso:

```
mkdir {{directory}}
```

- Crea directory ricorsivamente (utile per creare directory annidate):

```
mkdir {{-p|--parents}} {{percorso/della/directory}}
```

# more

Aprire un file per la lettura interattiva, con scorrimento e ricerca.

Maggiori informazioni: <https://manned.org/more.1p>.

- Apri un file:

```
more {{percorso/del/file}}
```

- Apri un file visualizzato da una riga specifica:

```
more +{{linea_numero}} {{percorso/del/file}}
```

- Mostra aiuto:

```
more --help
```

- Prossima pagina:

```
<Spazio>
```

- Cerca una stringa (tocca n per passare alla corrispondenza successiva):

```
/{{qualcose}}
```

- Smettere:

```
q
```

- Mostra aiuto per comando interattivo:

```
h
```

# mount

Fornisce accesso a un intero filesystem in una directory specifica.

Maggiori informazioni: <https://manned.org/mount.8>.

- Mostra tutti i filesystem montati:

```
mount
```

- Monta un dispositivo in una directory:

```
mount -t {{tipo_di_filesystem}} {{percorso/del/dispositivo}}  
{{percorso/della/directory_desiderata}}
```

- Monta un CD-ROM (con il filetype ISO9660) a /cdrom (sola lettura):

```
mount -t {{iso9660}} -o ro {{/dev/cdrom}} {{/cdrom}}
```

- Monta tutti i filesystem definiti in /etc/fstab:

```
mount -a
```

- Monta un filesystem specifico descritto in /etc/fstab (ad esempio /dev/sda1 /my\_drive ext2 defaults 0 2):

```
mount {{/my_drive}}
```

- Monta una directory in un'altra directory:

```
mount --bind {{percorso/della/vecchia_directory}} {{percorso/  
della/nuova_directory}}
```

# mpv

Un player audio/video basato su MPlayer.

Maggiori informazioni: <https://mpv.io>.

- Riproduci un file video o audio:

```
mpv {{file}}
```

- Salta avanti/indietro di 5 secondi:

```
SINISTRA <oppure> DESTRA
```

- Salta indietro/avanti di 1 minuto:

```
GIÙ <oppure> SU
```

- Riduci o aumenta la velocità di riproduzione del 10%:

```
[ <oppure> ]
```

- Riproduci un file a una velocità specifica (da 0.01 a 100, normalmente 1):

```
mpv --speed {{velocità}} {{file}}
```

- Riproduci un file usando un profilo definito nel file `mpv.conf`:

```
mpv --profile {{nome_profilo}} {{file}}
```

# mscore

Questo comando è un alias per **musescore**.

Maggiori informazioni: <https://musescore.org/handbook/command-line-options>.

- Consulta la documentazione del comando originale:

```
tldr musescore
```



# mv

Sposta o rinomina file e directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/mv>.

- Rinomina un file o una directory quando la destinazione non è una directory esistente:

```
mv {{percorso/del/file}} {{percorso/di/destinazione}}
```

- Sposta un file o una directory in una directory esistente:

```
mv {{percorso/di/origine}} {{percorso/della/  
directory_esistente}}
```

- Sposta più file in una directory esistente, mantenendo i nomi dei file invariati:

```
mv {{percorso/di/origine1 percorso/di/origine2 ...}}  
{{percorso/della/directory_esistente}}
```

- Non richiedere conferma prima di sovrascrivere i file esistenti:

```
mv -f {{percorso/di/origine}} {{percorso/di/destinazione}}
```

- Richiedi conferma prima di sovrascrivere i file esistenti, indipendentemente dalle autorizzazioni dei file:

```
mv -i {{percorso/di/origine}} {{percorso/di/destinazione}}
```

- Non sovrascrivere i file esistenti nella destinazione:

```
mv -n {{percorso/di/origine}} {{percorso/di/destinazione}}
```

- Sposta i file in modalità dettagliata, mostrando i file dopo che sono stati spostati:

```
mv -v {{percorso/di/origine}} {{percorso/di/destinazione}}
```

# neofetch

Strumento CLI per visualizzare informazioni al OS, software e hardware.

Maggiori informazioni: <https://github.com/dylanaraps/neofetch>.

- Stampa secondo la configurazione predefinita e genera una configurazione, se è la prima volta:

```
neofetch
```

- Attiva o disattiva la visualizzazione di una riga di informazioni nell'output, dove 'infoname' è il nome della funzione nel config, e.g. 'memory':

```
neofetch --{{enable|disable}} {{infoname}}
```

- Nascondi/mostri l'architettura del OS:

```
neofetch --os_arch {{on|off}}
```

- Attiva/disattiva marca CPU nell' output:

```
neofetch --cpu_brand {{on|off}}
```

# nm-classic

Questo comando è un alias per **nm**.

- Consulta la documentazione del comando originale:

`tldr nm`

# nmap

Nmap è un tool per port scanning ed esplorazione di rete.

Alcune funzionalità diventano attive solamente con privilegi d'amministratore.

Maggiori informazioni: <https://nmap.org/book/man.html>.

- Controlla se un indirizzo IP è attivo, e indovina il suo sistema operativo:

```
nmap -O {{ip_o_nome_host}}
```

- Cerca di determinare se gli host specificati sono attivi e quali sono i loro nomi:

```
sudo nmap -sn {{ip_o_nome_host}}  
{{opzionale_altro_indirizzo}}
```

- Attiva scripts, segnalazione di servizi, OS fingerprinting e traceroute:

```
nmap -A {{indirizzo_o_indirizzi}}
```

- Scansiona una specifica lista di porte (usa -p- per tutte le porte 1-65535):

```
nmap -p {{porta1,porta2,...,portaN}} {{indirizzo_o_indirizzi}}
```

- Determina vulnerabilità e informazioni di un host eseguendo una scansione di tutte le porte, servizi e versioni con tutti gli script di default NSE attivi:

```
nmap -sC -sV {{indirizzo_o_indirizzi}}
```

# node

Piattaforma JavaScript Server-side (Node.js).

Maggiori informazioni: <https://nodejs.org>.

- Esegue un file JavaScript:

```
node {{percorso/al/file}}
```

- Avvia una REPL (shell interattiva):

```
node
```

- Esegue il codice JavaScript che viene specificato come argomento:

```
node -e "{{codice}}"
```

- Valuta un'espressione e ne stampa il risultato, questo comando specifico è utile per vedere le versioni delle dipendenze di node:

```
node -p "process.versions"
```

- Attiva il debugger mettendo in pausa l'esecuzione finché il codice sorgente non viene caricato:

```
node --no-lazy --inspect-brk {{percorso/al/file}}
```

# npx

Esegue file binari dai pacchetti **npm**.

Maggiori informazioni: <https://github.com/npm/npx>.

- Esegue un file binario di uno specifico modulo:

```
npx {{nome_modulo}} {{argomenti_comando}}
```

- Nel caso in cui ci siano più binari per lo stesso modulo, specifica il pacchetto da eseguire:

```
npx --package {{nome_pacchetto}} {{nome_modulo}}
```

- Lancia un comando unicamente se questo esiste nel path corrente o in `node_modules/.bin`:

```
npx --no-install {{comando}} {{argomenti_comando}}
```

- Esegue il file binario del modulo specificato evitando di mostrare l'output prodotto dallo stesso npx:

```
npx --quiet {{nome_modulo}} {{argomenti_comando}}
```

- Mostra il menù d'aiuto:

```
npx --help
```

# ntl

Questo comando è un alias per **netlify**.

Maggiori informazioni: <https://cli.netlify.com>.

- Consulta la documentazione del comando originale:

```
tldr netlify
```

# nvim

Neovim, un editor di testo basato su Vim che offre molti diversi modi di manipolare e navigare il testo.

Premere **i** per entrare in modalità inserimento (insert mode), **<Esc>** per uscire e tornare alla modalità normale (normal mode).

Maggiori informazioni: <https://neovim.io>.

- Aprire un file:

```
nvim {{file}}
```

- Entrare nella modalità per scrivere testo (insert mode):

```
<Esc>i
```

- Copiare ("yank") o cancellare ("delete") la linea corrente (può poi essere copiata con **p** o **P**):

```
<Esc>{{yy|dd}}
```

- Annullare l'ultima operazione fatta:

```
<Esc>u
```

- Cercare uno specifico pattern nel file (premere **n/N** per navigare tra le occorrenze successive/precedenti):

```
<Esc>/{{patter_da_cercare}}<Enter>
```

- Eseguire una sostituzione tramite espressione regolare nell'intero file:

```
<Esc>:%s/{{espressione_regolare}}/{{sostituzione}}/g<Enter>
```

- Salvare (scrivere) il file per poi uscire:

```
{{<Esc>ZZ|<Esc>:x<Enter>|<Esc>:wq<Enter>}}
```

- Uscire senza salvare:

```
<Esc>:q!<Enter>
```



# opusenc

Converte audio WAV o FLAC in Opus.

Maggiori informazioni: <https://opus-codec.org/docs/opus-tools/opusenc.html>.

- Converte un file WAV in un file Opus usando le opzioni predefinite:

```
opusenc {{percorso/del/file_originale.wav}} {{percorso/del/
file_convertito.opus}}
```

- Converte un audio stereo alla massima qualità possibile:

```
opusenc --bitrate {{512}} {{percorso/del/file_originale.wav}}
{{percorso/del/file_convertito.opus}}
```

- Converte un audio con canali surround 5.1 alla massima qualità possibile:

```
opusenc --bitrate {{1536}} {{percorso/del/
file_originale.flac}} {{percorso/del/file_convertito.opus}}
```

- Converte l'audio di una voce alla minima qualità possibile:

```
opusenc {{percorso/del/file_originale.wav}} --downmix-mono --
bitrate {{6}} {{percorso/del/file_convertito.opus}}
```

# pio init

Questo comando è un alias per **pio project**.

- Consulta la documentazione del comando originale:

```
tldr pio project
```

# piodebuggdb

Questo comando è un alias per **pio debug**.

- Consulta la documentazione del comando originale:

```
tldr pio debug
```

# platformio

Questo comando è un alias per **pio**.

Maggiori informazioni: <https://docs.platformio.org/en/latest/core/userguide/>.

- Consulta la documentazione del comando originale:

```
tldr pio
```

# poetry

Gestore di pacchetti e dipendenze per Python.

Maggiori informazioni: <https://python-poetry.org/docs/cli/>.

- Crea un nuovo progetto Poetry nella directory specificata:

```
poetry new {{nome_progetto}}
```

- Installa una dipendenza e le relative sottodipendenze:

```
poetry add {{dipendenza}}
```

- Inizializza interattivamente la nuova directory come un nuovo progetto Poetry:

```
poetry init
```

- Recupera l'ultima versione di ciascuna dipendenza e aggiorna il file `poetry.lock`:

```
poetry update
```

- Esegue un comando all'interno dell'ambiente virtuale del progetto:

```
poetry run {{comando}}
```

# powershell

Shell della riga di comando e linguaggio di scripting progettato appositamente per l'amministrazione dei sistemi.

Guarda anche: **pwsh**.

Maggiori informazioni: [https://learn.microsoft.com/powershell/module/microsoft.powershell.core/about/about\\_pwsh](https://learn.microsoft.com/powershell/module/microsoft.powershell.core/about/about_pwsh).

- Avvia una sessione di shell interattiva:

```
powershell
```

- Avvia una sessione shell interattiva senza caricare le configurazioni di avvio:

```
powershell -NoProfile
```

- Esegui specifici comandi:

```
powershell -Command "{{echo 'powershell è eseguito'}}"
```

- Esegui uno specifico script:

```
powershell -File {{percorso/dello/script.ps1}}
```

- Avvia una sessione con una versione specifica di PowerShell:

```
powershell -Version {{versione}}
```

- Impedisce l'uscita dalla shell dopo aver eseguito i comandi di avvio:

```
powershell -NoExit
```

- Specifica il formato dei dati inviati a PowerShell:

```
powershell -InputFormat {{Text|XML}}
```

- Specifica come formattare l'output da PowerShell:

```
powershell -OutputFormat {{Text|XML}}
```

# ptpython3

Questo comando è un alias per **ptpython**.

- Consulta la documentazione del comando originale:

```
tldr ptpython
```

# pyenv virtualenv

Crea ambienti virtuali basati sulle distribuzioni Python che si hanno installate.

Maggiori informazioni: <https://github.com/pyenv/pyenv-virtualenv>.

- Crea un nuovo ambiente virtuale basato su Python 3.6.6:

```
pyenv virtualenv {{3.6.6}} {{virtualenv_name}}
```

- Elenca tutti gli ambienti virtuali esistenti:

```
pyenv virtualenvs
```

- Attiva un ambiente virtuale:

```
pyenv activate {{virtualenv_name}}
```

- Disattiva l'ambiente virtuale corrente:

```
pyenv deactivate
```



# pyenv

Passa da una distribuzione all'altra di Python in modo semplice.

Maggiori informazioni: <https://github.com/pyenv/pyenv>.

- Elenca i comandi disponibili:

```
pyenv commands
```

- Elenca tutte le distribuzioni di Python presenti nella directory `${PYENV_ROOT}/versions`:

```
pyenv versions
```

- Elenca tutte le versioni di Python che possono essere installate da upstream:

```
pyenv install --list
```

- Installa una distribuzione di Python nella directory `${PYENV_ROOT}/versions`:

```
pyenv install {{2.7.10}}
```

- Disinstalla una distribuzione di Python dalla directory `${PYENV_ROOT}/versions`:

```
pyenv uninstall {{2.7.10}}
```

- Imposta la distribuzione di Python da utilizzare globalmente sulla macchina:

```
pyenv global {{2.7.10}}
```

- Imposta la distribuzione di Python da utilizzare nella directory corrente e in tutte le relative sottodirectory:

```
pyenv local {{2.7.10}}
```

# python

Interprete del linguaggio Python.

Maggiori informazioni: <https://www.python.org>.

- Avvia una REPL (shell interattiva):

```
python
```

- Esegue lo script contenuto in un file Python:

```
python {{script.py}}
```

- Esegue uno script all'interno della shell interattiva:

```
python -i {{script.py}}
```

- Esegue un'espressione Python:

```
python -c "{{expression}}"
```

- Esegue lo script di un modulo presente nella libreria:

```
python -m {{modulo}} {{argomenti}}
```

- Installa un pacchetto usando pip:

```
python -m pip install {{nome_pacchetto}}
```

- Esegue il debug interattivo di uno script Python:

```
python -m pdb {{script.py}}
```

# python3

Questo comando è un alias di **python**.

- Consulta la documentazione del comando originale:

`tldr python`

# r2

Questo comando è un alias per **radare2**.

- Consulta la documentazione del comando originale:

```
tldr radare2
```

# rcat

Questo comando è un alias per **rc**.

- Consulta la documentazione del comando originale:

`tldr rc`

# readlink

Segue un collegamento simbolico e ne recupera le informazioni.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/readlink>.

- Restituisce il percorso originale a cui il collegamento simbolico fa riferimento:

```
readlink {{nome_file}}
```

- Ottiene il percorso assoluto di un file:

```
readlink -f {{nome_file}}
```

# rm

Rimuovi file o directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/rm>.

- Rimuovi file:

```
rm {{percorso/del/file1 percorso/del/file2 ...}}
```

- Rimuovi file, senza chiedere conferma o mostrare messaggi di errore:

```
rm -f {{percorso/della/directory}}
```

- Rimuovi file interattivamente, chiedendo conferma prima di rimuovere ogni file:

```
rm -i {{percorso/del/file1 percorso/del/file2 ...}}
```

- Rimuovi file in modalità verbosa, scrivendo un messaggio a schermo per ogni file rimosso:

```
rm -v {{percorso/del/file}}
```

- Rimuovi ricorsivamente una directory e tutti i suoi contenuti:

```
rm -r {{percorso/della/directory}}
```

# sudo

Esegue un singolo comando come superuser o come un altro utente.

Maggiori informazioni: <https://www.sudo.ws/sudo.html>.

- Esegui un comando come superuser:

```
sudo {{less /var/log/syslog}}
```

- Modifica un file come superuser con il tuo editor di default:

```
sudo -e {{/etc/fstab}}
```

- Esegui un comando come un altro utente e/o gruppo:

```
sudo -u {{utente}} -g {{gruppo}} {{id -a}}
```

- Ripeti l'ultimo comando prefissandolo con "sudo" (funziona solo in Bash, Zsh, ecc):

```
sudo !!
```

- Fai partire la shell di default con i privilegi da superuser:

```
sudo -i
```



# tac

Visualizza e concatena file con righe in ordine inverso.

Guarda anche: **cat**.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/tac>.

- Concatena file specifici in ordine inverso:

```
tac {{percorso/del/file1 percorso/del/file2 ...}}
```

- Visualizza `stdin` in ordine inverso:

```
{{cat percorso/del/file}} | tac
```

- Usa un `[s]`eparatore specifico:

```
tac -s {{separatore}} {{percorso/del/file1 percorso/del/file2 ...}}
```

- Usa un `[r]`egex specifico come `[s]`eparatore:

```
tac -r -s {{separatore}} {{percorso/del/file1 percorso/del/file2 ...}}
```

- Utilizzare un separatore `[b]`prima di ciascun file:

```
tac -b {{percorso/del/file1 percorso/del/file2 ...}}
```

# task

Gestore della lista dei TODO.

Maggiori informazioni: <https://taskwarrior.org/docs/>.

- Aggiungere un nuovo task:

```
task add {{thing_to_do}}
```

- Lista dei task:

```
task list
```

- Contrassegnare un task come completato:

```
task {{task_id}} done
```

- Modificare un task:

```
task {{task_id}} modify {{new_thing_to_do}}
```

- Eliminare un task:

```
task {{task_id}} delete
```

# tldr

Questo comando è un alias per **tldr-lint**.

Maggiori informazioni: <https://github.com/tldr-pages/tldr-lint>.

- Consulta la documentazione del comando originale:

```
tldr tldr-lint
```

# tlmgr arch

Questo comando è un alias per **tlmgr platform**.

Maggiori informazioni: <https://www.tug.org/texlive/tlmgr.html>.

- Consulta la documentazione del comando originale:

```
tldr tlmgr platform
```

# todo

Un semplice gestore per i todo da linea di comando.

Maggiori informazioni: <https://todoman.readthedocs.io>.

- Elenco dei task che possono essere inizializzati:

```
todo list --startable
```

- Aggiungere un nuovo task alla lista delle cose da fare per lavoro:

```
todo new {{cose_da_fare}} --list {{lavoro}}
```

- Aggiungere una località ad un task con un dato ID:

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

- Mostrare i dettagli di un task:

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

- Contrassegnare un task con un ID specifico come completato:

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

- Eliminare un task:

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

- Eliminare un task completato e ripristinare gli ID dei task rimanenti:

```
todo flush
```

# tree

Mostra i contenuti della directory corrente come un albero.

Maggiori informazioni: <https://manned.org/tree>.

- Stampa file e directory fino al 'num'-esimo livello di profondità (dove 1 significa la directory corrente):

```
tree -L {{num}}
```

- Stampa solamente le directory:

```
tree -d
```

- Stampa anche i file nascosti con la colorazione attiva:

```
tree -a -C
```

- Stampa l'albero senza linee di indentazione, mostrando invece il percorso completo (usa **-N** per non convertire caratteri non stampabili in sequenze di escape):

```
tree -i -f
```

- Stampa la dimensione di ogni file e la dimensione totale di ogni directory, in formato leggibile dall'utente:

```
tree -s -h --du
```

- Stampa i file all'interno dell'albero gerarchico, utilizzando espressioni di metacaratteri (glob pattern) per escludere le directory che non contengono file corrispondenti alla ricerca:

```
tree -P '{{*.txt}}' --prune
```

- Stampa le directory all'interno dell'albero gerarchico, utilizzando espressioni di metacaratteri (glob pattern) per escludere le directory che non sono progenitori di quelle desiderate:

```
tree -P {{nomi_di_directory}} --matchdirs --prune
```

- Stampa l'albero ignorando le directory date:

```
tree -I '{{nome_directory1|nome_directory2}}'
```

# unlzma

Questo comando è un alias per **xz**.

Maggiori informazioni: <https://manned.org/unlzma>.

- Consulta la documentazione del comando originale:

```
tldr xz
```

# unxz

Questo comando è un alias per **xz**.

Maggiori informazioni: <https://manned.org/unxz>.

- Consulta la documentazione del comando originale:

```
tldr xz
```



# vi

Questo comando è un alias per **vim**.

- Consulta la documentazione del comando originale:

```
tldr vim
```

# vim

Vi IMproved, un editor di testo per programmatori che fornisce diverse modalità per modificare testo.

Premi **i** per entrare in insert mode e **<Esc>** per tornare in normal mode dove non puoi inserire testo normalmente.

Maggiori informazioni: <https://www.vim.org>.

- Apri un file in vim:

```
vim {{file}}
```

- Annulla l'ultima operazione:

```
<Esc>u
```

- Cerca un pattern nel file (usa **n/N** per spostarti al risultato successivo/precedente):

```
/{{espressione_regolare}}<Invio>
```

- Effettua una sostituzione tramite espressione regolare nell'intero file:

```
:%s/{{espressione_regolare}}/{{sostituzione}}/g<Invio>
```

# vlc

Lettore multimediale multi-piattaforma.

Maggiori informazioni: [https://wiki.videolan.org/Documentation:Command\\_line/](https://wiki.videolan.org/Documentation:Command_line/).

- Riproduce un file:

```
vlc {{percorso/del/file}}
```

- Riproduce a schermo intero:

```
vlc --fullscreen {{percorso/del/file}}
```

- Riproduce con audio disattivato:

```
vlc --no-audio {{percorso/del/file}}
```

- Riproduce in ciclo continuo:

```
vlc --loop {{percorso/del/file}}
```

- Riproduce il video in un certo URL:

```
vlc {{https://www.youtube.com/watch?v=oHg5SJYRHA0}}
```

# wget

Scarica file dal Web.

Supporta HTTP, HTTPS, FTP.

Maggiori informazioni: <https://www.gnu.org/software/wget>.

- Scarica il contenuto dell'URL in un file (dal nome "abcd" in questo caso):

```
wget {{https://esempio.com/abcd}}
```

- Scarica il contenuto dell'URL in un file (dal nome "efgh" in questo caso):

```
wget --output-document {{efgh}} {{https://esempio.com/abcd}}
```

- Scarica una singola pagina web e tutte le sue risorse (script, immagini, stili, ecc..) aspettando 3 secondi dopo ogni richiesta:

```
wget --page-requisites --convert-links --wait=3 {{https://esempio.com/pagina_web.html}}
```

- Scarica tutti i file elencati nella directory e nelle sue sotto-directory (non scarica gli elementi incorporati nella pagina):

```
wget --mirror --no-parent {{https://esempio.com/unqualchepercorso/}}
```

- Limita la velocità di download e il numero di tentativi di connessione:

```
wget --limit-rate={{300k}} --tries={{100}} {{https://esempio.com/unqualchepercorso/}}
```

- Scarica un file da un server HTTP trasmettendo le credenziali tramite Basis Auth (funzione anche con FTP):

```
wget --user={{nome_utente}} --password={{password}} {{https://esempio.com}}
```

- Riprende un download incompleto:

```
wget --continue {{https://esempio.com}}
```

- Scarica tutti gli URL contenuti in un file di testo in una directory specificata:

```
wget --directory-prefix {{percorso/della/directory}} --input-file {{lista_di_URL.txt}}
```

# xkill

Termina una finestra interattivamente attraverso la sessione grafica.

Vedi anche **kill** e **killall**.

Maggiori informazioni: <https://www.x.org/releases/current/doc/man/man1/xkill.1.xhtml>.

- Mostra un cursore per terminare una finestra cliccandoci sopra con il tasto sinistro del mouse (premendo qualsiasi altro tasto si cancella l'azione):

```
xkill
```

# xxd

Mostra la rappresentazione esadecimale (hexdump) di un file binario e viceversa.

Maggiori informazioni: <https://manned.org/xxd>.

- Creare l'hexdump di un file binario e mostrare l'output:

```
xxd {{file_di_input}}
```

- Creare l'hexdump di un file binario e salvare il risultato in un file:

```
xxd {{file_di_input}} {{file_di_output}}
```

- Mostrare un output in una versione un po' più compatta, dove gli zero consecutivi vengono sostituiti da un asterisco:

```
xxd -a {{file_di_input}}
```

- Mostrare l'output in 10 colonne di un ottetto (byte) ciascuna:

```
xxd -c {{10}} {{file_di_input}}
```

- Mostrare l'output solo fino ad una lunghezza massima di 32 bytes:

```
xxd -l {{32}} {{file_di_input}}
```

- Mostrare l'output in modalità normale, senza spazi tra le colonne:

```
xxd -p {{file_di_input}}
```

- Eseguire l'operazione inversa, ovvero da un hexdump creare il binario e salvarlo in un file:

```
xxd -r -p {{file_di_input}} {{file_di_output}}
```

# xzcat

Questo comando è un alias per **xz**.

Maggiori informazioni: <https://manned.org/xzcat>.

- Consulta la documentazione del comando originale:

```
tldr xz
```

# youtube-dl

Scarica video da YouTube ed altri siti web.

Maggiori informazioni: <https://rg3.github.io/youtube-dl/>.

- Scarica un video od una playlist:

```
youtube-dl '{{https://www.youtube.com/watch?v=oHg5SJYRHA0}}'
```

- Elenca tutti i formati in cui un video od una playlist sono disponibili:

```
youtube-dl --list-formats '{{https://www.youtube.com/watch?v=Mwa0_nE9H7A}}'
```

- Scarica un video od una playlist con la qualità specificata:

```
youtube-dl --format '{{best[height<=480]}}' '{{https://www.youtube.com/watch?v=oHg5SJYRHA0}}'
```

- Scarica l'audio di un video e lo converte in file MP3:

```
youtube-dl -x --audio-format {{mp3}} '{{url}}'
```

- Scarica l'audio di migliore qualità e il video di migliore qualità e li unisce:

```
youtube-dl -f bestvideo+bestaudio '{{url}}'
```

- Scarica una playlist di video e li salva come file MP4 dai nomi personalizzati:

```
youtube-dl --format {{mp4}} -o "{{%(title)s di %(uploader)s del %(upload_date)s in %(playlist)s.%(ext)s}}" '{{url}}'
```

- Scarica, assieme al video, i sottotitoli in una lingua specificata:

```
youtube-dl --sub-lang {{it}} --write-sub '{{https://www.youtube.com/watch?v=Mwa0_nE9H7A}}'
```

- Scarica una playlist, ne estrae l'audio e lo salva in formato mp3:

```
youtube-dl -f "bestaudio" --continue --no-overwrites --ignore-errors --extract-audio --audio-format mp3 -o "%(title)s.%(ext)s" '{{url_della_playlist}}'
```



# zsh

Z SHell.

Inteprete da linea di comando compatibile con **bash** e **sh**.

Maggiori informazioni: <https://www.zsh.org>.

- Fai partire l'interprete interattivo da linea di comando:

```
zsh
```

- Esegui un comando passandolo come parametro:

```
zsh -c {{comando}}
```

- Esegui comandi da un file (script):

```
zsh {{file}}
```

- Esegui comandi da un file e stampali a schermo mentre vengono eseguiti:

```
zsh -x {{file}}
```

Linux

# a2disconf

Disattiva un file di configurazione Apache su Sistemi Operativi basati su Debian.

Maggiori informazioni: <https://manned.org/a2disconf.8>.

- Disattiva un file di configurazione:

```
sudo a2disconf {{file_di_configurazione}}
```

- Non mostrare messaggi informativi:

```
sudo a2disconf --quiet {{file_di_configurazione}}
```

# a2dismod

Disattiva un modulo Apache su sistemi operativi basati su Debian.

Maggiori informazioni: <https://manned.org/a2dismod.8>.

- Disattiva un modulo:

```
sudo a2dismod {{modulo}}
```

- Non mostrare messaggi informativi:

```
sudo a2dismod --quiet {{modulo}}
```

# a2dissite

Disattiva un virtual host Apache su sistemi operativi basati su Debian.

Maggiori informazioni: <https://manned.org/a2dissite.8>.

- Disattiva un virtual host:

```
sudo a2dissite {{virtual_host}}
```

- Non mostrare messaggi informativi:

```
sudo a2dissite --quiet {{virtual_host}}
```

# a2enconf

Attiva un file di configurazione Apache su sistemi operativi basati su Debian.

Maggiori informazioni: <https://manned.org/a2enconf.8>.

- Attiva un file di configurazione:

```
sudo a2enconf {{file_di_configurazione}}
```

- Non mostrare messaggi informativi:

```
sudo a2enconf --quiet {{file_di_configurazione}}
```

# a2enmod

Attiva un modulo Apache su sistemi operativi basati su Debian.

Maggiori informazioni: <https://manned.org/a2enmod.8>.

- Attiva un modulo:

```
sudo a2enmod {{modulo}}
```

- Non mostrare messaggi informativi:

```
sudo a2enmod --quiet {{modulo}}
```

# a2ensite

Attiva un virtual host Apache su sistemi operativi basati su Debian.

Maggiori informazioni: <https://manned.org/a2ensite.8>.

- Attiva un virtual host:

```
sudo a2ensite {{virtual_host}}
```

- Non mostrare messaggi informativi:

```
sudo a2ensite --quiet {{virtual_host}}
```



# a2query

Recupera la configurazione di runtime da Apache su sistemi operativi basati su Debian.

Maggiori informazioni: <https://manned.org/a2query>.

- Lista i moduli Apache attivi:

```
sudo a2query -m
```

- Controlla se un modulo specifico è installato:

```
sudo a2query -m {{nome_modulo}}
```

- Lista virtual host attivi:

```
sudo a2query -s
```

- Mostra il Modulo Multi-Processo correntemente attivo:

```
sudo a2query -M
```

- Mostra la versione di Apache:

```
sudo a2query -v
```

# acpi

Mostra lo stato e le informazioni termiche della batteria.

Maggiori informazioni: <https://sourceforge.net/projects/acpiclient/files/acpiclient/>.

- Mostra le informazioni sulla batteria:

```
acpi
```

- Mostra le informazioni termiche:

```
acpi -t
```

- Mostra le informazioni sul dispositivo di raffreddamento:

```
acpi -c
```

- Mostra le informazioni termiche in gradi Fahrenheit:

```
acpi -tf
```

- Mostra tutte le informazioni:

```
acpi -V
```

- Estrae le informazioni da `/proc` invece che da `/sys`:

```
acpi -p
```

# add-apt-repository

Gestisce le definizioni di repository APT.

Maggiori informazioni: <https://manned.org/apt-add-repository>.

- Aggiunge un nuovo repository APT:

```
add-apt-repository {{identificativo_del_repository}}
```

- Rimuove un repository APT:

```
add-apt-repository --remove {{identificativo_del_repository}}
```

- Aggiorna la cache dei pacchetti dopo aver aggiunto un repository:

```
add-apt-repository --update {{identificativo_del_repository}}
```

- Attiva i pacchetti sorgente:

```
add-apt-repository --enable-source  
{{identificativo_del_repository}}
```

# adduser

Servizio per aggiungere utenti.

Maggiori informazioni: <https://manned.org/adduser>.

- Crea un nuovo utente con una directory home predefinita e richiede all'utente di impostare una password:

```
adduser {{nome_utente}}
```

- Crea un utente senza una directory home:

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

- Crea un utente con una directory home nel percorso specificato:

```
adduser --home {{percorso/della/home}} {{nome_utente}}
```

- Crea un nuovo utente con l'interprete di comandi(shell) specificato come shell di accesso:

```
adduser --shell {{percorso/della/shell}} {{nome_utente}}
```

- Crea un nuovo utente appartenente al gruppo specificato:

```
adduser --ingroup {{gruppo}} {{nome_utente}}
```

- Aggiunge un utente esistente al gruppo specificato:

```
adduser {{nome_utente}} {{gruppo}}
```

# alternatives

Questo comando è un alias per **update-alternatives**.

Maggiori informazioni: <https://manned.org/alternatives>.

- Consulta la documentazione del comando originale:

```
tldr update-alternatives
```

# apk

Gestore pacchetti di Alpine Linux.

Maggiori informazioni: <https://manned.org/apk>.

- Aggiorna gli indici di tutti i repository remoti:

```
apk update
```

- Installa un nuovo pacchetto:

```
apk add {{pacchetto}}
```

- Disinstalla un pacchetto:

```
apk del {{pacchetto}}
```

- Prova a riparare un pacchetto o ad aggiornare un pacchetto senza modificarne le dipendenze:

```
apk fix {{paquet}}
```

- Ricerca pacchetti data una parola chiave:

```
apk search {{parola_chiave}}
```

- Ottieni informazioni su un pacchetto:

```
apk info {{pacchetto}}
```

# apt-add-repository

Gestisce le definizioni di repository APT.

Maggiori informazioni: <https://manned.org/apt-add-repository.1>.

- Aggiunge un nuovo repository APT:

```
apt-add-repository {{identificativo_del_repository}}
```

- Rimuove un repository APT:

```
apt-add-repository --remove {{identificativo_del_repository}}
```

- Aggiorna la cache dei pacchetti dopo aver aggiunto un repository:

```
apt-add-repository --update {{identificativo_del_repository}}
```

- Attiva i pacchetti sorgente:

```
apt-add-repository --enable-source  
{{identificativo_del_repository}}
```

# apt-cache

Strumenti di Debian e Ubuntu per richiedere informazioni sui pacchetti.

Maggiori informazioni: <https://manned.org/apt-cache.8>.

- Cerca un pacchetto nelle sorgenti attuali:

```
apt-cache search {{query}}
```

- Mostra informazioni su un pacchetto:

```
apt-cache show {{pacchetto}}
```

- Mostra se un pacchetto è installato ed aggiornato:

```
apt-cache policy {{pacchetto}}
```

- Mostra le dipendenze di un pacchetto:

```
apt-cache depends {{pacchetto}}
```

- Mostra i pacchetti che dipendono da un particolare pacchetto:

```
apt-cache rdepends {{pacchetto}}
```



# apt-file

Cerca un file dentro un pacchetto APT, includendo quelli non ancora installati.

Maggiori informazioni: <https://manned.org/apt-file.1>.

- Aggiorna il database dei metadati:

```
sudo apt update
```

- Cerca i pacchetti che contengono un file o un percorso specificato:

```
apt-file search {{parte/del/filename}}
```

- Elenca i contenuti di un pacchetto specifico:

```
apt-file list {{nome_del_pacchetto}}
```

# apt-get

Servizio di gestione dei pacchetti per Debian e Ubuntu.

Cerca i pacchetti usando **apt-cache**.

Maggiori informazioni: <https://manned.org/apt-get.8>.

- Aggiorna la lista dei pacchetti e delle loro versioni disponibili (è consigliato eseguire questo comando prima di altri comandi `apt-get`):

```
apt-get update
```

- Installa un pacchetto, o lo aggiorna all'ultima versione disponibile:

```
apt-get install {{pacchetto}}
```

- Rimuove un pacchetto:

```
apt-get remove {{pacchetto}}
```

- Rimuove un pacchetto ed i suoi file di configurazione:

```
apt-get purge {{pacchetto}}
```

- Aggiorna tutti i pacchetti installati alla versione disponibile più recente:

```
apt-get upgrade
```

- Pulisce gli archivi locali - rimuovendo i file (.deb) da scaricamenti interrotti che non possono più essere scaricati:

```
apt-get autoclean
```

- Rimuove tutti i pacchetti che non sono più necessari:

```
apt-get autoremove
```

- Aggiorna tutti i pacchetti installati (come `upgrade`), rimuovendo i pacchetti obsoleti ed installando ulteriori pacchetti per soddisfare le nuove dipendenze:

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

# apt-key

Servizio di gestione delle chiavi per il gestore di pacchetti APT su Debian ed Ubuntu.

Maggiori informazioni: <https://manned.org/apt-key.8>.

- Elenca le chiavi fidate:

```
apt-key list
```

- Aggiunge una chiave al portachiavi delle chiavi fidate:

```
apt-key add {{file_chiave_pubblica.asc}}
```

- Elimina una chiave dal portachiavi delle chiavi fidate:

```
apt-key del {{id_chiave}}
```

- Aggiunge una chiave remota al portachiavi delle chiavi fidate:

```
wget -q0 - {{https://indirizzo.tld/filename.key}} | apt-key  
add -
```

- Aggiunge una chiave da un server di chiavi con il solo ID della chiave:

```
apt-key adv --keyserver {{pgp.mit.edu}} --recv  
{{ID_DELLA_CHIAVE}}
```

# apt-mark

Servizio per cambiare lo stato di un pacchetto installato.

Maggiori informazioni: <https://manned.org/apt-mark.8>.

- Contrassegna un pacchetto come installato automaticamente:

```
sudo apt-mark auto {{nome_del_pacchetto}}
```

- Mantiene un pacchetto alla sua versione attuale e ne previene l'aggiornamento:

```
sudo apt-mark hold {{nome_del_pacchetto}}
```

- Consente ad un pacchetto di essere nuovamente aggiornato:

```
sudo apt-mark unhold {{nome_del_pacchetto}}
```

- Mostra i pacchetti installati manualmente:

```
apt-mark showmanual
```

- Visualizza i pacchetti mantenuti alla versione attuale che non sono stati aggiornati:

```
apt-mark showhold
```

# apt

Servizio di gestione dei pacchetti per distribuzioni basate su Debian.

Rimpiazzo raccomandato di **apt-get** quando usato interattivamente su Ubuntu 16.04 e versioni successive.

Maggiori informazioni: <https://manned.org/apt.8>.

- Aggiorna la lista dei pacchetti e delle loro versioni disponibili (è consigliato eseguire questo comando prima di altri comandi **apt**):

```
sudo apt update
```

- Cerca per un dato pacchetto:

```
apt search {{pacchetto}}
```

- Mostra le informazioni su un pacchetto:

```
apt show {{pacchetto}}
```

- Installa un pacchetto, o lo aggiorna all'ultima versione disponibile:

```
sudo apt install {{pacchetto}}
```

- Rimuove un pacchetto (usando **purge** rimuove anche i suoi file di configurazione):

```
sudo apt remove {{pacchetto}}
```

- Aggiorna tutti i pacchetti installati alla versione disponibile più recente:

```
sudo apt upgrade
```

- Elenca tutti i pacchetti:

```
apt list
```

- Elenca i pacchetti installati:

```
apt list --installed
```

# batcat

Questo comando è un alias per **bat**.

Maggiori informazioni: <https://github.com/sharkdp/bat>.

- Consulta la documentazione del comando originale:

```
tldr bat
```

# beep

Un'utilità per emettere un segnale acustico all'altoparlante del PC.

Maggiori informazioni: <https://github.com/spkr-beep/beep>.

- Emetti un suono:

```
beep
```

- Emetti un suono che si ripete:

```
beep -r {{ripetizioni}}
```

- Emetti un suono a una specifica frequenza (Hz) e durata (millisecondi):

```
beep -f {{frequenza}} -l {{durata}}
```

- Riproduci ogni nuova frequenza e durata come un segnale acustico distinto:

```
beep -f {{frequenza}} -l {{durata}} -n -f {{frequenza}} -l {{durata}}
```

- Suona la scala di do maggiore:

```
beep -f {{262}} -n -f {{294}} -n -f {{330}} -n -f {{349}} -n  
-f {{392}} -n -f {{440}} -n -f {{494}} -n -f {{523}}
```

# CC

Questo comando è un alias per **gcc**.

Maggiori informazioni: <https://gcc.gnu.org>.

- Consulta la documentazione del comando originale:

```
tldr gcc
```



# chcon

Cambia contesto di sicurezza SELinux di file o directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/chcon>.

- Mostra il contesto di sicurezza di un file:

```
ls -lZ {{percorso/del/file}}
```

- Cambia il contesto di sicurezza di un file usandone un'altro come riferimento:

```
chcon --reference={{file_di_riferimento}} {{file}}
```

- Cambia l'intero contesto di sicurezza SELinux di un file:

```
chcon {{utente}}:{{ruolo}}:{{tipo}}:{{range/livello}}  
{{file}}
```

- Cambia solo l'utente di un contesto di sicurezza SELinux:

```
chcon -u {{utente}} {{file}}
```

- Cambia solo il ruolo di un contesto di sicurezza SELinux:

```
chcon -r {{ruolo}} {{file}}
```

- Cambia solo il tipo di un contesto di sicurezza SELinux:

```
chcon -t {{tipo}} {{file}}
```

- Cambia solo il range/livello di un contesto di sicurezza SELinux:

```
chcon -l {{range/livello}} {{file}}
```

# cp

Copia file e directory.

Maggiori informazioni: <https://www.gnu.org/software/coreutils/cp>.

- Copia un file in un'altra posizione:

```
cp {{percorso/del/file_da_copiare.est}} {{percorso/del/
file_di_destinazione.est}}
```

- Copia un file all'interno di una directory, mantenendone uguale il nome:

```
cp {{percorso/del/file_da_copiare.est}} {{percorso/della/
directory}}
```

- Copia ricorsivamente i contenuti di una directory in un'altra posizione (se la destinazione esiste, la directory è copiata al suo interno):

```
cp -r {{percorso/della/directory_da_copiare}} {{percorso/di/
destinazione}}
```

- Copia una directory ricorsivamente in modalità prolissa (mostra i file mentre vengono copiati):

```
cp -vr {{percorso/della/directory_da_copiare}} {{percorso/di/
destinazione}}
```

- Copia i file di testo in un'altra posizione, in modalità interattiva (richiede conferma all'utente prima di sovrascrivere):

```
cp -i {{*.txt}} {{percorso/di/destinazione}}
```

- Segue i collegamenti simbolici prima di copiare:

```
cp -L {{collegamento}} {{percorso/di/destinazione}}
```

- Utilizza il percorso completo dei file originali, creando ogni directory intermedia mancante mentre durante la copia:

```
cp --parents {{percorso/dei/file/da/copiare}} {{percorso/del/
file/destinazione}}
```

# deluser

Rimuovi un account utente o un utente da un gruppo.

Maggiori informazioni: <https://manned.org/deluser>.

- Rimuovi un utente:

```
deluser {{nome}}
```

- Rimuovi un utente insieme alla sua directory home e raccolta mail:

```
deluser -r {{nome}}
```

- Rimuovi un utente da un gruppo:

```
deluser {{nome}} {{gruppo}}
```

# dpkg-deb

Impacchetta, spacchetta e fornisce informazioni su archivi Debian.

Maggiori informazioni: <https://manned.org/dpkg-deb>.

- Mostra le informazioni riguardo ad un pacchetto:

```
dpkg-deb --info {{percorso/del/file.deb}}
```

- Mostra il nome e la versione del pacchetto in una singola riga:

```
dpkg-deb --show {{percorso/del/file.deb}}
```

- Elenca i contenuti del pacchetto:

```
dpkg-deb --contents {{percorso/del/file.deb}}
```

- Estrae i contenuti del pacchetto in una directory:

```
dpkg-deb --extract {{percorso/del/file.deb}} {{percorso/
della/directory}}
```

- Crea un pacchetto a partire da una directory specificata:

```
dpkg-deb --build {{percorso/della/directory}}
```

# dpkg-query

Uno strumento che mostra informazioni sui pacchetti installati.

Maggiori informazioni: <https://manned.org/dpkg-query.1>.

- Elenca tutti i pacchetti installati:

```
dpkg-query -l
```

- Elenca i pacchetti installati con nomi che combaciano con una data espressione:

```
dpkg-query -l '{{espressione_pattern}}'
```

- Elenca tutti i file installati da una pacchetto:

```
dpkg-query -L {{nome_del_pacchetto}}
```

- Mostra le informazioni riguardanti un pacchetto:

```
dpkg-query -s {{nome_del_pacchetto}}
```

# dpkg

Gestore di pacchetti Debian.

Alcuni comandi aggiuntivi, come **deb**, hanno la propria documentazione.

Maggiori informazioni: <https://manned.org/dpkg>.

- Installa un pacchetto:

```
dpkg -i {{percorso/del/file.deb}}
```

- Rimuove un pacchetto:

```
dpkg -r {{nome_del_pacchetto}}
```

- Elenca i pacchetti installati:

```
dpkg -l {{espressione_per_la_ricerca}}
```

- Elenca i contenuti di un pacchetto:

```
dpkg -L {{nome_del_pacchetto}}
```

- Elenca i contenuti di un file pacchetto locale:

```
dpkg -c {{percorso/del/file.deb}}
```

- Trova a quale pacchetto appartiene un file:

```
dpkg -S {{filename}}
```

# enum4linux

Strumento per ottenere informazioni da Windows e Samba da un sistema remoto.

Maggiori informazioni: <https://labs.portcullis.co.uk/tools/enum4linux/>.

- Ottieni informazioni utilizzando tutti i metodi disponibili:

```
enum4linux -a {{host_remoto}}
```

- Ottieni informazioni utilizzando le credenziali fornite:

```
enum4linux -u {{nome_utente}} -p {{password}} {{host_remoto}}
```

- Ottieni la lista utenti:

```
enum4linux -U {{host_remoto}}
```

- Mostra le risorse condivise:

```
enum4linux -S {{host_remoto}}
```

- Ottieni informazioni riguardo al sistema operativo:

```
enum4linux -o {{host_remoto}}
```

# fsck

Controlla l'integrità di un filesystem o lo ripara. Il filesystem non dev'essere montato al momento in cui il comando viene eseguito.

Maggiori informazioni: <https://manned.org/fsck>.

- Controlla il filesystem `/dev/sdX`, riportando eventuali blocchi danneggiati:

```
sudo fsck {{/dev/sdX}}
```

- Controlla il filesystem `/dev/sdX`, riportando eventuali blocchi danneggiati e per ognuno consente all'utente di scegliere interattivamente se ripararlo:

```
sudo fsck -r {{/dev/sdX}}
```

- Controlla il filesystem `/dev/sdX`, riportando eventuali blocchi danneggiati e riparandoli automaticamente:

```
sudo fsck -a {{/dev/sdX}}
```



# halt

Arresta, spegne o riavvia la macchina.

Maggiori informazioni: <https://manned.org/halt.8>.

- Arresta la macchina:

```
halt
```

- Spegne la macchina:

```
halt --poweroff
```

- Riavvia la macchina:

```
halt --reboot
```

# i3

Window Manager Dinamico.

Tasto Windows=Super=Mod4, Alt=Mod1.

Maggiori informazioni: <https://i3wm.org/docs/userguide.html>.

- Avvia i3 (non può essere eseguito contemporaneamente con un altro gestore di finestre):

`i3`

- Apri un terminale:

`<Super> + <Invio>`

- Crea un nuovo workspace:

`<Super> + <Shift> + {{numero}}`

- Salta a workspace numero n:

`<Super> + {{n}}`

- Per il prossimo nuovo, dividi orizzontalmente:

`<Super> + h`

- Per il prossimo nuovo, dividi verticalmente:

`<Super> + v`

- Avvia dmenu, lanciatore di programmi:

`<Super> + D`

# ip route list

Questo comando è un alias per **ip route show**.

- Consulta la documentazione del comando originale:

`tldr ip-route-show`

# ip

Mostra / manipola routing, dispositivi, criteri di routing e tunnel.

Alcuni sottocomandi, come **ip address**, hanno una propria documentazione d'uso.

Maggiori informazioni: <https://www.manned.org/ip.8>.

- Elenca le interfacce con informazioni dettagliate:

```
ip address
```

- Elenca le interfacce con informazioni brevi sul livello di rete:

```
ip -brief address
```

- Elenca le interfacce con informazioni brevi sul livello di collegamento:

```
ip -brief link
```

- Visualizza la tabella di routing:

```
ip route
```

- Mostra i vicini (tabella ARP):

```
ip neighbour
```

- Attiva/disattiva un'interfaccia:

```
ip link set {{interfaccia}} {{up|down}}
```

- Aggiungi/elimina un indirizzo IP a/da un'interfaccia:

```
ip addr add/del {{ip}}/{{mask}} dev {{interfaccia}}
```

- Aggiungi una route predefinita:

```
ip route add default via {{ip}} dev {{interfaccia}}
```

# iptables

Programma che permette di configurare tabelle, catene e regole fornite dal firewall del kernel Linux.

Maggiori informazioni: <https://manned.org/iptables>.

- Visualizza catene, regole e contatori di pacchetti/byte per la tabella dei filtri:

```
sudo iptables -vnL
```

- Imposta regola ad una catena:

```
sudo iptables -P {{catena}} {{regola}}
```

- Appendi regola ad una catena di policy per IP:

```
sudo iptables -A {{catena}} -s {{ip}} -j {{regola}}
```

- Appendi regola ad una catena di policy per IP considerando protocollo e porta:

```
sudo iptables -A {{catena}} -s {{ip}} -p {{protocollo}} --  
dport {{porta}} -j {{regola}}
```

- Cancella regola da una catena:

```
sudo iptables -D {{catena}} {{numero_di_linea_della_regola}}
```

- Salva la configurazione di ip tables di una specifica tabella in un file:

```
sudo iptables-save -t {{nome_tabella}} > {{percorso/del/  
file_iptables}}
```

- Ripristina la configurazione di iptables da un file:

```
sudo iptables-restore < {{percorso/del/file_iptables}}
```

# lsmod

Mostra lo stato dei moduli del kernel Linux.

Vedi anche **modprobe**, che carica i moduli del kernel.

Maggiori informazioni: <https://manned.org/lsmod>.

- Elenca tutti i moduli del kernel attualmente caricati:

```
lsmod
```

# lsusb

Visualizza le informazioni su i bus USB e i dispositivi a loro connessi.

Maggiori informazioni: <https://manned.org/lsusb>.

- Elenca tutti i dispositivi USB disponibili:

```
lsusb
```

- Visualizza la gerarchia USB come un albero:

```
lsusb -t
```

- Elenca informazioni prolisse riguardo ai dispositivi USB:

```
lsusb --verbose
```

- Elenca solamente i dispositivi con un certo id fornitore e ID prodotto:

```
lsusb -d {{fornitore}}:{{prodotto}}
```

# megadl

Questo comando è un alias per **megatools-dl**.

Maggiori informazioni: <https://megatools.megous.com/man/megatools-dl.html>.

- Consulta la documentazione del comando originale:

```
tldr megatools-dl
```



# mkfs

Costruisce un filesystem Linux su una partizione del disco rigido.

Questo comando è deprecato in favore degli strumenti specifici per filesystem: `mkfs.tipo`.

Maggiori informazioni: <https://manned.org/mkfs>.

- Costruisce un filesystem Linux ext2 su una partizione:

```
mkfs {{percorso/della/partizione}}
```

- Costruisce un filesystem del tipo specificato:

```
mkfs -t {{ext4}} {{percorso/della/partizione}}
```

- Costruisce un filesystem del specificato e controlla la presenza di settori danneggiati:

```
mkfs -c -t {{ntfs}} {{percorso/della/partizione}}
```

# modinfo

Estrae le informazioni riguardarti un modulo del kernel Linux.

Maggiori informazioni: <https://manned.org/modinfo>.

- Elenca tutti gli attributi di un modulo del kernel:

```
modinfo {{modulo_del_kernel}}
```

- Elenca solamente gli attributi specificati:

```
modinfo -F {{author|description|license|parm|filename}}  
{{modulo_del_kernel}}
```

# modprobe

Aggiunge o rimuove moduli del kernel Linux.

Maggiori informazioni: <https://manned.org/modprobe>.

- Fa finta di carica un modulo nel kernel, ma non lo fa veramente:

```
sudo modprobe --dry-run {{nome_del_modulo}}
```

- Carica un modulo nel kernel:

```
sudo modprobe {{nome_del_modulo}}
```

- Rimuove un modulo dal kernel:

```
sudo modprobe --remove {{nome_del_modulo}}
```

- Rimuove dal kernel un modulo e quelli che dipendono da quest'ultimo:

```
sudo modprobe --remove-dependencies {{nome_del_modulo}}
```

- Mostra le dipendenza di un modulo del kernel:

```
sudo modprobe --show-depends {{nome_del_modulo}}
```

# mpicc

Involucro Open MPI per il compilatore di C.

Shell che esegue sul compilatore, aggiungono i rilevanti argomenti e linkers necessari a compilare/collegare programmi Open MPI, invocando il sottostante compilatore di C per effettuare le effettive operazioni.

Maggiori informazioni: <https://www.mpich.org/static/docs/latest/www1/mpicc.html>.

- Compila un file sorgente in un file oggetto:

```
mpicc -c {{percorso/del/file.c}}
```

- Linka un file oggetto file e genera un eseguibile:

```
mpicc -o {{executable}} {{percorso/del/file.o}}
```

- Linka e compila i file sorgente in un solo comando:

```
mpicc -o {{executable}} {{percorso/del/file.c}}
```

# ncal

Questo comando è un alias per **cal**.

Maggiori informazioni: <https://manned.org/ncal>.

- Consulta la documentazione del comando originale:

```
tldr cal
```

# nmtui

Interfaccia utente solo testo per NetworkManager.

Usa le frecce e invio per navigare.

Maggiori informazioni: <https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmtui.html>.

- Apri interfaccia utente:

```
nmtui
```

- Mostra le reti disponibili, con opzioni per attivare o disattivare:

```
nmtui connect
```

- Per la connessione a una rete:

```
nmtui connect {{nome|uuid|device|SSID}}
```

- Cambia/Agiunge/Elimina una rete:

```
nmtui edit {{nome|id}}
```

- Imposta un hostname nuovo:

```
nmtui hostname
```

# pacman

Gestore pacchetti di Arch Linux.

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

Per comandi equivalenti in altri gestori pacchetti, vedi <https://wiki.archlinux.org/title/Pacman/Rosetta>.

Maggiori informazioni: <https://manned.org/pacman.8>.

- Sincronizza e aggiorna tutti i pacchetti:

```
sudo pacman -Syu
```

- Installa un nuovo pacchetto:

```
sudo pacman -S {{nome_pacchetto}}
```

- Rimuovi un pacchetto e le sue dipendenze:

```
sudo pacman -Rs {{nome_pacchetto}}
```

- Cerca nel database un pacchetto contenente uno specifico file:

```
pacman -F "{{nome_file}}"
```

- Mostra i pacchetti installati e le versioni:

```
pacman -Q
```

- Mostra solo i pacchetti esplicitamente installati e le versioni:

```
pacman -Qe
```

- Mostra i pacchetti orfani (installati come dipendenze ma attualmente non richiesti da nessun altro pacchetto):

```
pacman -Qtdq
```

- Svuota l'intera cache di pacman:

```
sudo pacman -Scc
```

# pidof

Ottiene l'ID di un processo a partire dal suo nome.

Maggiori informazioni: <https://manned.org/pidof>.

- Elenca gli ID di tutti i processi con un dato nome:

```
pidof {{bash}}
```

- Elenca un solo ID di processo con il nome specificato:

```
pidof -s {{bash}}
```

- Elenca gli ID dei processi con un dato includendo anche gli script:

```
pidof -x {{script.py}}
```

- Uccide tutti i processi con un dato nome:

```
kill $(pidof {{nome}})
```



# poweroff

Chiude il sistema.

Maggiori informazioni: <https://www.manned.org/poweroff>.

- Spegne il sistema:

```
sudo poweroff
```

# pulseaudio

Programma che permette di gestire il daemon audio di sistema.

Maggiori informazioni: <https://www.freedesktop.org/wiki/Software/PulseAudio/>.

- Controlla se PulseAudio è in esecuzione. Se il programma non è attivo viene restituito un exit code diverso da 0:

```
pulseaudio --check
```

- Esegue il daemon di PulseAudio in background:

```
pulseaudio --start
```

- Interrompe l'esecuzione del daemon di PulseAudio:

```
pulseaudio --kill
```

- Mostra i moduli disponibili:

```
pulseaudio --dump-modules
```

- Carica un modulo all'interno del daemon in esecuzione con gli argomenti specificati:

```
pulseaudio --load="{{nome_modulo}} {{argomenti}}"
```

# radeontop

Mostra Utilizzo di AMD GPUs.

Maggiori informazioni: <https://github.com/clbr/radeontop>.

- Mostra utilizzo del AMD GPU principale:

```
radeontop
```

- Inizia output con colore:

```
radeontop --color
```

- Scegli un GPU specifico (il numero del bus è il primo numero nell'output di `lspci`):

```
radeontop --bus {{bus_numero}}
```

- Specifica la frequenza di aggiornamento del display (più alto aggiunge più sovraccarico al GPU):

```
radeontop --ticks {{aggiornamenti_per_secondo}}
```

# reset

Reinizializza il terminale corrente. Cancella l'intera schermata del terminale.

Maggiori informazioni: <https://manned.org/reset>.

- Reinizializza il terminale corrente:

```
reset
```

- Visualizza invece il tipo di terminale:

```
reset -q
```

# shutdown

Spegni e riavvia il sistema.

Maggiori informazioni: <https://manned.org/shutdown.8>.

- Spegni il sistema immediatamente:

```
shutdown -h now
```

- Riavvia il sistema immediatamente:

```
shutdown -r now
```

- Riavvia il sistema in 5 minuti:

```
shutdown -r +{{5}} &
```

- Spegni il sistema alle 13:

```
shutdown -h 13:00
```

- Annulla un'operazione programmata di riavvio o spegnimento:

```
shutdown -c
```

# smbclient

Programma client simile ad FTP per server SMB/CIFS.

Maggiori informazioni: <https://manned.org/smbclient>.

- Connettiti ad una share (all'utente verrà richiesta la password; `exit` per uscire dalla sessione):

```
smbclient {{//server/share}}
```

- Connettiti con un altro nome utente:

```
smbclient {{//server/share}} --user {{username}}
```

- Connettiti con un altro gruppo di lavoro:

```
smbclient {{//server/share}} --workgroup {{dominio}} --user {{username}}
```

- Connettiti con un nome utente ed una password:

```
smbclient {{//server/share}} --user {{username%password}}
```

- Scarica un file dal server:

```
smbclient {{//server/share}} --directory {{percorso/della/directory}} --command "get {{file.txt}}"
```

- Carica un file sul server:

```
smbclient {{//server/share}} --directory {{percorso/della/directory}} --command "put {{file.txt}}"
```

- Elenca le share di un server in modo anonimo:

```
smbclient --list={{server}} --no-pass
```

# ubuntu-bug

Questo comando è un alias per **apport-bug**.

Maggiori informazioni: <https://manned.org/ubuntu-bug>.

- Consulta la documentazione del comando originale:

```
tldr apport-bug
```

# ufw

Ufw (Uncomplicated Firewall) - Firewall Semplice.

Frontend per **iptables** per semplificare la configurazione di un firewall.

Maggiori informazioni: <https://wiki.ubuntu.com/UncomplicatedFirewall>.

- Accendi ufw:

```
ufw enable
```

- Spegni ufw:

```
ufw disable
```

- Mostra le regole di ufw, con i numeri corrispondenti:

```
ufw status numbered
```

- Aperto al traffico in entrata sulla porta 5432, con un commento che identifica il servizio:

```
ufw allow {{5432}} comment "{{servizio}}"
```

- Aperto solo al traffico TCP da 192.168.0.4 a qualsiasi indirizzo su questo host, sulla porta 22:

```
ufw allow proto {{tcp}} from {{192.168.0.4}} to {{any}} port {{22}}
```

- Blocchi traffico su porta 80 su questo host:

```
ufw deny {{80}}
```

- Nega tutto il traffico UDP alla porta 22:

```
ufw deny proto {{udp}} from {{any}} to {{any}} port {{22}}
```

- Elimina una regola particolare. Il numero della regola può essere trovato con "ufw status numbered":

```
ufw delete {{numero_della_regola}}
```



# uname

Uname stampa informazioni a proposito della macchina e del sistema operativo in esecuzione su di essa.

Maggiori informazioni: [https://www.gnu.org/software/coreutils/manual/html\\_node/uname-invocation.html](https://www.gnu.org/software/coreutils/manual/html_node/uname-invocation.html).

- Stampa tutte le informazioni:

```
uname --all
```

- Stampa il nome del kernel corrente:

```
uname --kernel-name
```

- Stampa il nome host corrente del nodo di rete:

```
uname --nodename
```

- Stampa la release del kernel corrente:

```
uname --kernel-release
```

- Stampa la versione del kernel corrente:

```
uname --kernel-version
```

- Stampa il nome corrente della macchina hardware:

```
uname --machine
```

- Stampa il tipo di processore corrente:

```
uname --processor
```

- Stampa il nome del sistema operativo corrente:

```
uname --operating-system
```

# zathura

Un visualizzatore di documenti ispirato a vim, con una linea di comando integrata.

Assicurati che sia installata un backend (poppler, PostScript, o DjVu).

Maggiori informazioni: <https://pwmt.org/projects/zathura/>.

- Aprire un file:

```
zathura {{percorso/del/file}}
```

- Navigare sinistra/sopra/sotto/destra:

```
{{H|J|K|L|arrow keys}}
```

- Ruotare:

```
r
```

- Invertire i colori:

```
<Ctrl> + R
```

- Cercare all'interno del testo una stringa:

```
/{{string}}
```

- Creare/eliminare i segnalibri:

```
:{{bmark|bdelete}} {{bookmark_name}}
```

- Lista dei segnalibri:

```
:blist
```

# zypper

Sistema di Gestione dei Pacchetti di SUSE e openSUSE.

Maggiori informazioni: [https://en.opensuse.org/SDB:Zypper\\_manual](https://en.opensuse.org/SDB:Zypper_manual).

- Sincronizza il database dei pacchetti e delle versioni disponibili:

```
zypper refresh
```

- Installa un nuovo pacchetto:

```
zypper install {{pacchetto}}
```

- Rimuovi un pacchetto:

```
zypper remove {{pacchetto}}
```

- Aggiorna i pacchetti installati alle ultime versioni disponibili:

```
zypper update
```

- Cerca usando nome o parola chiave:

```
zypper search {{nome|parole_chiave}}
```

Osx

# aa

Questo comando è un alias per **yaa**.

- Consulta la documentazione del comando originale:

```
tldr yaa
```

# afinfo

Analizzatore dei metadata dei file audio per macOS.

Comando integrato di macOS.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/afinfo.1.html>.

- Mostra le informazioni relative ad un file audio specificato:

```
afinfo {{percorso/del/file}}
```

- Mostra una breve descrizione del file audio:

```
afinfo --brief {{percorso/del/file}}
```

- Mostra i metadati ed i contenuti dell'InfoDictionary del file audio:

```
afinfo --info {{percorso/del/file}}
```

- Mostra l'output in formato xml:

```
afinfo --xml {{percorso/del/file}}
```

- Mostra i problemi del file audio (se ce ne sono):

```
afinfo --warnings {{percorso/del/file}}
```

- Mostra la pagina di aiuto:

```
afinfo --help
```

# afplay

Riproduttore audio a riga di comando.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/afplay.1.html>.

- Riproduci un file audio (fino a quando non finisce la riproduzione):

```
afplay {{percorso/del/file}}
```

- Riproduci un file audio al doppio della velocità:

```
afplay --rate {{2}} {{percorso/del/file}}
```

- Riproduci un file audio alla metà della velocità:

```
afplay --rate {{0.5}} {{percorso/del/file}}
```

- Riproduci i primi N secondi di un file audio:

```
afplay --time {{N}} {{percorso/del/file}}
```

# airport

Strumento di configurazione delle reti senza fili.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/airport.1.html>.

- Mostra le informazioni relative allo stato attuale delle connessioni senza fili:

```
airport --getinfo
```

- Intercetta il traffico delle connessioni senza fili sul primo canale:

```
airport sniff {{1}}
```

- Ricerca le reti senza fili disponibili:

```
airport --scan
```

- Disassocia dalla rete airport corrente:

```
sudo airport --disassociate
```



# apachectl

Interfaccia di controllo del server HTTP Apache per macOS.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/apachectl.8.html>.

- Avvia il demone `org.apache.httpd`:

```
apachectl start
```

- Ferma il demone:

```
apachectl stop
```

- Riavvia il demone:

```
apachectl restart
```

# archey

Semplice strumento per mostrare con stile le informazioni di sistema.

Maggiori informazioni: <https://github.com/joshfinnie/archey-osx>.

- Mostra le informazioni di sistema:

```
archey
```

- Mostra le informazioni di sistema (senza colorazione del testo):

```
archey --nocolor
```

- Mostra le informazioni di sistema usando MacPorts invece di Homebrew:

```
archey --macports
```

- Mostra le informazioni di sistema senza controllare l'indirizzo IP:

```
archey --offline
```

# as

Assembler GNU portabile.

Progettato principalmente per assemblare l'output di **gcc** ed utilizzarlo con **ld**.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/as.1.html>.

- Assembla un file, scrivendo l'output su a.out:

```
as {{percorso/del/file.s}}
```

- Assembla l'output nel file dato:

```
as {{percorso/del/file.s}} -o {{percorso/del/out.o}}
```

- Genera l'output più velocemente saltando gli spazi e senza preprocessare i commenti. (Questo comando dovrebbe essere utilizzato solo con compilatori fidati):

```
as -f {{percorso/del/file.s}}
```

- Includi un percorso dato alla lista delle directory in cui cercare i file specificati nelle direttive `.include`:

```
as -I {{percorso/della/directory}} {{percorso/del/file.s}}
```

# asr

Ripristina (copia) un'immagine disco dentro a un volume.

Il nome del comando sta per Apple Software Restore (software di ripristino Apple).

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/asr.8.html>.

- Ripristina un'immagine disco su un volume specifico:

```
sudo asr restore --source {{nome_immagine.dmg}} --target  
{{percorso/del/volume}}
```

- Distruggi il volume specifico prima di ripristinare:

```
sudo asr restore --source {{nome_immagine.dmg}} --target  
{{percorso/del/volume}} --erase
```

- Salta la verifica dopo il ripristino:

```
sudo asr restore --source {{nome_immagine.dmg}} --target  
{{percorso/del/volume}} --noverify
```

- Clona i volumi senza utilizzare un'immagine disco intermedia:

```
sudo asr restore --source {{percorso/del/volume}} --target  
{{percorso/del/volume/clonato}}
```

# base64

Codifica e decodifica utilizzando la rappresentazione in base64.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/base64.1.html>.

- Codifica un file:

```
base64 --input={{file_da_codificare}}
```

- Decodifica un file:

```
base64 --decode --input={{file_da_decodificare}}
```

- Codifica da stdin:

```
echo -n "{{testo_da_codificare}}" | base64
```

- Decodifica da stdin:

```
echo -n {{testo_da_decodificare}} | base64 --decode
```

# caffeinate

Impedisci al mac di sospendersi.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/caffeinate.8.html>.

- Impedisci la sospensione per un'ora (3600 secondi):

```
caffeinate -u -t {{3600}}
```

- Impedisci la sospensione fino al completamento dell'esecuzione di un comando:

```
caffeinate -s {{comando}}
```

- Impedisci la sospensione fino alla pressione della combinazione di tasti Ctrl-C:

```
caffeinate -i
```

# diskutil

Strumento per gestire i dischi locali e i volumi.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/diskutil.8.html>.

- Mostra tutti i dischi correnti, le partizioni e i volumi montati:

```
diskutil list
```

- Ripara le strutture dati del filesystem di un volume:

```
diskutil repairVolume {{/dev/diskX}}
```

- Smonta un volume:

```
diskutil unmountDisk {{/dev/diskX}}
```

- Estrai un CD/DVD (smontando prima dell'estrazione):

```
diskutil eject {{/dev/disk1}}
```

# g[

Questo comando è un alias per **-p linux** [.

- Consulta la documentazione del comando originale:

```
tldr -p linux [
```



# gawk

Questo comando è un alias per **-p linux awk**.

- Consulta la documentazione del comando originale:

```
tldr -p linux awk
```

# gb2sum

Questo comando è un alias per **-p linux b2sum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux b2sum
```

# gbase32

Questo comando è un alias per **-p linux base32**.

- Consulta la documentazione del comando originale:

```
tldr -p linux base32
```

# gbase64

Questo comando è un alias per **-p linux base64**.

- Consulta la documentazione del comando originale:

```
tldr -p linux base64
```

# gbasename

Questo comando è un alias per **-p linux basename**.

- Consulta la documentazione del comando originale:

```
tldr -p linux basename
```

# gbasenc

Questo comando è un alias per **-p linux basenc**.

- Consulta la documentazione del comando originale:

```
tldr -p linux basenc
```

# gcat

Questo comando è un alias per **-p linux cat**.

- Consulta la documentazione del comando originale:

```
tldr -p linux cat
```

# gchcon

Questo comando è un alias per **-p linux chcon**.

- Consulta la documentazione del comando originale:

```
tldr -p linux chcon
```



# gchgrp

Questo comando è un alias per **-p linux chgrp**.

- Consulta la documentazione del comando originale:

```
tldr -p linux chgrp
```

# gchmod

Questo comando è un alias per **-p linux chmod**.

- Consulta la documentazione del comando originale:

```
tldr -p linux chmod
```

# gchown

Questo comando è un alias per **-p linux chown**.

- Consulta la documentazione del comando originale:

```
tldr -p linux chown
```

# gchroot

Questo comando è un alias per **-p linux chroot**.

- Consulta la documentazione del comando originale:

```
tldr -p linux chroot
```

# gcksum

Questo comando è un alias per **-p linux cksum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux cksum
```

# gcomm

Questo comando è un alias per **-p linux comm**.

- Consulta la documentazione del comando originale:

```
tldr -p linux comm
```

# gcp

Questo comando è un alias per **-p linux cp**.

- Consulta la documentazione del comando originale:

```
tldr -p linux cp
```

# gcsplit

Questo comando è un alias per **-p linux csplit**.

- Consulta la documentazione del comando originale:

```
tldr -p linux csplit
```



# gcut

Questo comando è un alias per **-p linux cut**.

- Consulta la documentazione del comando originale:

```
tldr -p linux cut
```

# gdate

Questo comando è un alias per **-p linux date**.

- Consulta la documentazione del comando originale:

```
tldr -p linux date
```

# gdd

Questo comando è un alias per **-p linux dd**.

- Consulta la documentazione del comando originale:

```
tldr -p linux dd
```

# gdf

Questo comando è un alias per **-p linux df**.

- Consulta la documentazione del comando originale:

```
tldr -p linux df
```

# gdir

Questo comando è un alias per **-p linux dir**.

- Consulta la documentazione del comando originale:

```
tldr -p linux dir
```

# gdircolors

Questo comando è un alias per **-p linux dircolors**.

- Consulta la documentazione del comando originale:

```
tldr -p linux dircolors
```

# gdirname

Questo comando è un alias per **-p linux dirname**.

- Consulta la documentazione del comando originale:

```
tldr -p linux dirname
```

# gdnsdomainname

Questo comando è un alias per **-p linux dnsdomainname**.

- Consulta la documentazione del comando originale:

```
tldr -p linux dnsdomainname
```



# gecho

Questo comando è un alias per **-p linux echo**.

- Consulta la documentazione del comando originale:

```
tldr -p linux echo
```

# ged

Questo comando è un alias per **-p linux ed**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ed
```

# gegrep

Questo comando è un alias per **-p linux egrep**.

- Consulta la documentazione del comando originale:

```
tldr -p linux egrep
```

# genv

Questo comando è un alias per **-p linux env**.

- Consulta la documentazione del comando originale:

```
tldr -p linux env
```

# gexpand

Questo comando è un alias per **-p linux expand**.

- Consulta la documentazione del comando originale:

```
tldr -p linux expand
```

# gexpr

Questo comando è un alias per **-p linux expr**.

- Consulta la documentazione del comando originale:

```
tldr -p linux expr
```

# gfactor

Questo comando è un alias per **-p linux factor**.

- Consulta la documentazione del comando originale:

```
tldr -p linux factor
```

# gfalse

Questo comando è un alias per **-p linux false**.

- Consulta la documentazione del comando originale:

```
tldr -p linux false
```



# gfgrep

Questo comando è un alias per **-p linux fgrep**.

- Consulta la documentazione del comando originale:

```
tldr -p linux fgrep
```

# gfind

Questo comando è un alias per **-p linux find**.

- Consulta la documentazione del comando originale:

```
tldr -p linux find
```

# gfmt

Questo comando è un alias per **-p linux fmt**.

- Consulta la documentazione del comando originale:

```
tldr -p linux fmt
```

# gfold

Questo comando è un alias per **-p linux fold**.

- Consulta la documentazione del comando originale:

```
tldr -p linux fold
```

# gftp

Questo comando è un alias per **-p linux ftp**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ftp
```

# ggrep

Questo comando è un alias per **-p linux grep**.

- Consulta la documentazione del comando originale:

```
tldr -p linux grep
```

# ggroups

Questo comando è un alias per **-p linux groups**.

- Consulta la documentazione del comando originale:

```
tldr -p linux groups
```

# ghead

Questo comando è un alias per **-p linux head**.

- Consulta la documentazione del comando originale:

```
tldr -p linux head
```



# ghostid

Questo comando è un alias per **-p linux ghostid**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ghostid
```

# ghostname

Questo comando è un alias per **-p linux hostname**.

- Consulta la documentazione del comando originale:

```
tldr -p linux hostname
```

# gid

Questo comando è un alias per **-p linux id**.

- Consulta la documentazione del comando originale:

```
tldr -p linux id
```

# gifconfig

Questo comando è un alias per **-p linux ifconfig**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ifconfig
```

# gindent

Questo comando è un alias per **-p linux indent**.

- Consulta la documentazione del comando originale:

```
tldr -p linux indent
```

# ginstall

Questo comando è un alias per **-p linux install**.

- Consulta la documentazione del comando originale:

```
tldr -p linux install
```

# gjoin

Questo comando è un alias per **-p linux join**.

- Consulta la documentazione del comando originale:

```
tldr -p linux join
```

# gkill

Questo comando è un alias per **-p linux kill**.

- Consulta la documentazione del comando originale:

```
tldr -p linux kill
```



# glibtool

Questo comando è un alias per **-p linux libtool**.

- Consulta la documentazione del comando originale:

```
tldr -p linux libtool
```

# glibtoolize

Questo comando è un alias per **-p linux libtoolize**.

- Consulta la documentazione del comando originale:

```
tldr -p linux libtoolize
```

# glink

Questo comando è un alias per **-p linux link**.

- Consulta la documentazione del comando originale:

```
tldr -p linux link
```

# gln

Questo comando è un alias per **-p linux ln**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ln
```

# glocate

Questo comando è un alias per **-p linux locate**.

- Consulta la documentazione del comando originale:

```
tldr -p linux locate
```

# glogger

Questo comando è un alias per **-p linux logger**.

- Consulta la documentazione del comando originale:

```
tldr -p linux logger
```

# glogname

Questo comando è un alias per **-p linux logname**.

- Consulta la documentazione del comando originale:

```
tldr -p linux logname
```

# gls

Questo comando è un alias per **-p linux ls**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ls
```



# gmake

Questo comando è un alias per **-p linux make**.

- Consulta la documentazione del comando originale:

```
tldr -p linux make
```

# gmd5sum

Questo comando è un alias per **-p linux md5sum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux md5sum
```

# gmkdir

Questo comando è un alias per **-p linux mkdir**.

- Consulta la documentazione del comando originale:

```
tldr -p linux mkdir
```

# gmkfifo

Questo comando è un alias per **-p linux mkfifo**.

- Consulta la documentazione del comando originale:

```
tldr -p linux mkfifo
```

# gmknod

Questo comando è un alias per **-p linux mknod**.

- Consulta la documentazione del comando originale:

```
tldr -p linux mknod
```

# gmkttemp

Questo comando è un alias per **-p linux mktemp**.

- Consulta la documentazione del comando originale:

```
tldr -p linux mktemp
```

# gmv

Questo comando è un alias per **-p linux mv**.

- Consulta la documentazione del comando originale:

```
tldr -p linux mv
```

# gnice

Questo comando è un alias per **-p linux nice**.

- Consulta la documentazione del comando originale:

```
tldr -p linux nice
```



# gnl

Questo comando è un alias per **-p linux nl**.

- Consulta la documentazione del comando originale:

```
tldr -p linux nl
```

# gnohup

Questo comando è un alias per **-p linux nohup**.

- Consulta la documentazione del comando originale:

```
tldr -p linux nohup
```

# gnproc

Questo comando è un alias per **-p linux nproc**.

- Consulta la documentazione del comando originale:

```
tldr -p linux nproc
```

# gnumfmt

Questo comando è un alias per **-p linux numfmt**.

- Consulta la documentazione del comando originale:

```
tldr -p linux numfmt
```

# god

Questo comando è un alias per **-p linux od**.

- Consulta la documentazione del comando originale:

```
tldr -p linux od
```

# gpaste

Questo comando è un alias per **-p linux paste**.

- Consulta la documentazione del comando originale:

```
tldr -p linux paste
```

# gpathchk

Questo comando è un alias per **-p linux pathchk**.

- Consulta la documentazione del comando originale:

```
tldr -p linux pathchk
```

# gping

Questo comando è un alias per **-p linux ping**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ping
```



# gping6

Questo comando è un alias per **-p linux ping6**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ping6
```

# gpinky

Questo comando è un alias per **-p linux pinky**.

- Consulta la documentazione del comando originale:

```
tldr -p linux pinky
```

# gpr

Questo comando è un alias per **-p linux pr**.

- Consulta la documentazione del comando originale:

```
tldr -p linux pr
```

# gprintenv

Questo comando è un alias per **-p linux printenv**.

- Consulta la documentazione del comando originale:

```
tldr -p linux printenv
```

# gprintf

Questo comando è un alias per **-p linux printf**.

- Consulta la documentazione del comando originale:

```
tldr -p linux printf
```

# gptx

Questo comando è un alias per **-p linux ptx**.

- Consulta la documentazione del comando originale:

```
tldr -p linux ptx
```

# gpwd

Questo comando è un alias per **-p linux pwd**.

- Consulta la documentazione del comando originale:

```
tldr -p linux pwd
```

# grcp

Questo comando è un alias per **-p linux rcp**.

- Consulta la documentazione del comando originale:

```
tldr -p linux rcp
```



# greadlink

Questo comando è un alias per **-p linux readlink**.

- Consulta la documentazione del comando originale:

```
tldr -p linux readlink
```

# grealpath

Questo comando è un alias per **-p linux realpath**.

- Consulta la documentazione del comando originale:

```
tldr -p linux realpath
```

# grexec

Questo comando è un alias per **-p linux rexec**.

- Consulta la documentazione del comando originale:

```
tldr -p linux rexec
```

# grlogin

Questo comando è un alias per **-p linux rlogin**.

- Consulta la documentazione del comando originale:

```
tldr -p linux rlogin
```

# grm

Questo comando è un alias per **-p linux rm**.

- Consulta la documentazione del comando originale:

```
tldr -p linux rm
```

# grmdir

Questo comando è un alias per **-p linux rmdir**.

- Consulta la documentazione del comando originale:

```
tldr -p linux rmdir
```

# grsh

Questo comando è un alias per **-p linux rsh**.

- Consulta la documentazione del comando originale:

```
tldr -p linux rsh
```

# gruncon

Questo comando è un alias per **-p linux runcon**.

- Consulta la documentazione del comando originale:

```
tldr -p linux runcon
```



# gsed

Questo comando è un alias per **-p linux sed**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sed
```

# gseq

Questo comando è un alias per **-p linux seq**.

- Consulta la documentazione del comando originale:

```
tldr -p linux seq
```

# gsha1sum

Questo comando è un alias per **-p linux sha1sum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sha1sum
```

# gsha224sum

Questo comando è un alias per **-p linux sha224sum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sha224sum
```

# gsha256sum

Questo comando è un alias per **-p linux sha256sum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sha256sum
```

# gsha384sum

Questo comando è un alias per **-p linux sha384sum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sha384sum
```

# gsha512sum

Questo comando è un alias per **-p linux sha512sum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sha512sum
```

# gshred

Questo comando è un alias per **-p linux shred**.

- Consulta la documentazione del comando originale:

```
tldr -p linux shred
```



# gshuf

Questo comando è un alias per **-p linux shuf**.

- Consulta la documentazione del comando originale:

```
tldr -p linux shuf
```

# gsleep

Questo comando è un alias per **-p linux sleep**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sleep
```

# gsort

Questo comando è un alias per **-p linux sort**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sort
```

# gsplit

Questo comando è un alias per **-p linux split**.

- Consulta la documentazione del comando originale:

```
tldr -p linux split
```

# gstat

Questo comando è un alias per **-p linux stat**.

- Consulta la documentazione del comando originale:

```
tldr -p linux stat
```

# gstdbuf

Questo comando è un alias per **-p linux stdbuf**.

- Consulta la documentazione del comando originale:

```
tldr -p linux stdbuf
```

# gstty

Questo comando è un alias per **-p linux stty**.

- Consulta la documentazione del comando originale:

```
tldr -p linux stty
```

# gsum

Questo comando è un alias per **-p linux sum**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sum
```



# gsync

Questo comando è un alias per **-p linux sync**.

- Consulta la documentazione del comando originale:

```
tldr -p linux sync
```

# gtac

Questo comando è un alias per **-p linux tac**.

- Consulta la documentazione del comando originale:

```
tldr -p linux tac
```

# gtail

Questo comando è un alias per **-p linux tail**.

- Consulta la documentazione del comando originale:

```
tldr -p linux tail
```

# gtalk

Questo comando è un alias per **-p linux talk**.

- Consulta la documentazione del comando originale:

```
tldr -p linux talk
```

# gtar

Questo comando è un alias per **-p linux tar**.

- Consulta la documentazione del comando originale:

```
tldr -p linux tar
```

# gtee

Questo comando è un alias per **-p linux tee**.

- Consulta la documentazione del comando originale:

```
tldr -p linux tee
```

# gtelnet

Questo comando è un alias per **-p linux telnet**.

- Consulta la documentazione del comando originale:

```
tldr -p linux telnet
```

# gtest

Questo comando è un alias per **-p linux test**.

- Consulta la documentazione del comando originale:

```
tldr -p linux test
```



# gtftp

Questo comando è un alias per **-p linux tftp**.

- Consulta la documentazione del comando originale:

```
tldr -p linux tftp
```

# gtime

Questo comando è un alias per **-p linux time**.

- Consulta la documentazione del comando originale:

```
tldr -p linux time
```

# gtimeout

Questo comando è un alias per **-p linux timeout**.

- Consulta la documentazione del comando originale:

```
tldr -p linux timeout
```

# gtouch

Questo comando è un alias per **-p linux touch**.

- Consulta la documentazione del comando originale:

```
tldr -p linux touch
```

# gtr

Questo comando è un alias per **-p linux tr**.

- Consulta la documentazione del comando originale:

```
tldr -p linux tr
```

# gtracroute

Questo comando è un alias per **-p linux traceroute**.

- Consulta la documentazione del comando originale:

```
tldr -p linux traceroute
```

# gtrue

Questo comando è un alias per **-p linux true**.

- Consulta la documentazione del comando originale:

```
tldr -p linux true
```

# gtruncate

Questo comando è un alias per **-p linux truncate**.

- Consulta la documentazione del comando originale:

```
tldr -p linux truncate
```



# gtsort

Questo comando è un alias per **-p linux tsort**.

- Consulta la documentazione del comando originale:

```
tldr -p linux tsort
```

# gtty

Questo comando è un alias per **-p linux tty**.

- Consulta la documentazione del comando originale:

```
tldr -p linux tty
```

# guname

Questo comando è un alias per **-p linux uname**.

- Consulta la documentazione del comando originale:

```
tldr -p linux uname
```

# gunexpand

Questo comando è un alias per **-p linux unexpand**.

- Consulta la documentazione del comando originale:

```
tldr -p linux unexpand
```

# guniq

Questo comando è un alias per **-p linux uniq**.

- Consulta la documentazione del comando originale:

```
tldr -p linux uniq
```

# gunits

Questo comando è un alias per **-p linux units**.

- Consulta la documentazione del comando originale:

```
tldr -p linux units
```

# gunlink

Questo comando è un alias per **-p linux unlink**.

- Consulta la documentazione del comando originale:

```
tldr -p linux unlink
```

# gupdatedb

Questo comando è un alias per **-p linux updatedb**.

- Consulta la documentazione del comando originale:

```
tldr -p linux updatedb
```



# guptime

Questo comando è un alias per **-p linux uptime**.

- Consulta la documentazione del comando originale:

```
tldr -p linux uptime
```

# gusers

Questo comando è un alias per **-p linux users**.

- Consulta la documentazione del comando originale:

```
tldr -p linux users
```

# gvdir

Questo comando è un alias per **-p linux vdir**.

- Consulta la documentazione del comando originale:

```
tldr -p linux vdir
```

# gwc

Questo comando è un alias per **-p linux wc**.

- Consulta la documentazione del comando originale:

```
tldr -p linux wc
```

# gwhich

Questo comando è un alias per **-p linux which**.

- Consulta la documentazione del comando originale:

```
tldr -p linux which
```

# gwho

Questo comando è un alias per **-p linux who**.

- Consulta la documentazione del comando originale:

```
tldr -p linux who
```

# gwhoami

Questo comando è un alias per **-p linux whoami**.

- Consulta la documentazione del comando originale:

```
tldr -p linux whoami
```

# gwhois

Questo comando è un alias per **-p linux whois**.

- Consulta la documentazione del comando originale:

```
tldr -p linux whois
```



# gxargs

Questo comando è un alias per **-p linux xargs**.

- Consulta la documentazione del comando originale:

```
tldr -p linux xargs
```

# gyes

Questo comando è un alias per **-p linux yes**.

- Consulta la documentazione del comando originale:

```
tldr -p linux yes
```

# log

Visualizza, esporta e configura i sistemi di log.

Maggiori informazioni: <https://keith.github.io/xcode-man-pages/log.1.html>.

- Mostra i log di sistema in diretta:

```
log stream
```

- Mostra i log mandati a `syslog` da processi con un PID specifico:

```
log stream --process {{ID_processo}}
```

- Mostra i log mandati a `syslog` da processi con un nome specifico:

```
log show --predicate "process == '{{process_name}}'"
```

- Esporta sul disco tutti i log dell'ultima ora:

```
sudo log collect --last {{1h}} --output {{percorso/del/  
file.logarchive}}
```

Windows

# attrib

Mostra o cambia gli attributi di file e cartelle.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/attrib>.

- Mostra tutti gli attributi dei file in una cartella:

```
attrib
```

- Mostra tutti gli attributi dei file in una cartella specifica:

```
attrib {{percorso\della\cartella}}
```

- Mostra tutti gli attributi dei file e delle cartelle nella cartella corrente:

```
attrib /d
```

- Mostra tutti gli attributi dei file in una cartella e le sue sotto-cartelle:

```
attrib /s
```

- Aggiungi l'attributo [r]ead-only o [a]rchive o [s]ystem o [h]idden o not content [i]ndexed a file o cartelle:

```
attrib +{{r|a|s|h|i}} {{percorso\del\file_o_cartella1  
percorso\del\file_o_cartella2 ...}}
```

- Rimuovi un attributo specifico da un file o una cartella:

```
attrib -{{r|a|s|h|i}} {{percorso\del\file_o_cartella1  
percorso\del\file_o_cartella2 ...}}
```

# cd

Mostra informazioni riguardo alla directory corrente e permette di muoversi in un'altra directory.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/cd>.

- Mostra il percorso (path) della directory corrente:

```
cd
```

- Spostati nella directory superiore:

```
cd ..
```

- Spostati ad una specifica directory sullo stesso disco:

```
cd {{percorso\della\directory}}
```

- Spostati ad una specifica directory in un altro disco:

```
cd /d {{C}}:{{percorso\della\directory}}
```

# chkdsk

Controlla il file system e i metadata dei dischi per cercare errori.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/chkdsk>.

- Specifica la lettera del disco (seguita da due punti ':'), monta una partizione o un disco da controllare:

```
chkdsk {{disco}}
```

- Ripara gli errori di un disco specifico:

```
chkdsk {{disco}} /f
```

- Smonta un disco specifico prima di eseguire il controllo:

```
chkdsk {{volume}} /x
```

- Cambia la dimensione dei file di log in una dimensione specifica (solo per NTFS):

```
chkdsk /l{{dimensione}}
```

# choco apikey

Gestisci le chiavi API per le fonti di Chocolatey.

Maggiori informazioni: <https://chocolatey.org/docs/commands-apikey>.

- Mostra una lista di fonti e le loro chiavi API:

```
choco apikey
```

- Mostra una specifica fonte e la sua chiave API:

```
choco apikey --source "{{url_della_fonte}}"
```

- Imposta una chiave API per una fonte:

```
choco apikey --source "{{url_della_fonte}}" --key  
"{{chiave_api}}"
```

- Rimuovi una chiave API per una fonte:

```
choco apikey --source "{{url_della_fonte}}" --remove
```



# choco install

Installa uno o più pacchetti con Chocolatey.

Maggiori informazioni: <https://chocolatey.org/docs/commands-install>.

- Installa uno o più pacchetti separati da spazio:

```
choco install {{pacchetto1 pacchetto2 ...}}
```

- Installa pacchetti da un file di configurazione personalizzato:

```
choco install {{percorso\del\file_di_pacchetti.config}}
```

- Installa un file nuspec o nupkg specifico:

```
choco install {{percorso\del\file}}
```

- Installa una nuova versione specifica di un pacchetto:

```
choco install {{pacchetto}} --version {{versione}}
```

- Consenti l'installazione di più versioni di un pacchetto:

```
choco install {{pacchetto}} --allow-multiple
```

- Conferma automaticamente tutte le richieste:

```
choco install {{pacchetto}} --yes
```

- Specifica una fonte personalizzata per ricevere pacchetti:

```
choco install {{pacchetto}} --source {{url_fonte|alias}}
```

- Fornisci un nome utente e una password per l'autenticazione:

```
choco install {{pacchetto}} --user {{nome_utente}} --password {{password}}
```

# choco

Interfaccia a linea di comando per il gestore pacchetti Chocolatey.

Alcuni sottocomandi come **choco install** hanno la loro documentazione.

Maggiori informazioni: <https://chocolatey.org>.

- Esegui comandi Chocolatey:

```
choco {{comando}}
```

- Guarda il messaggio d'aiuto:

```
choco -?
```

- Guarda il messaggio d'aiuto di un comando specifico:

```
choco {{command}} -?
```

- Controlla la versione di Chocolatey:

```
choco --version
```

# chrome

Questo comando è un alias per **chromium**.

Maggiori informazioni: <https://chrome.google.com>.

- Consulta la documentazione del comando originale:

```
tldr chromium
```

# cinst

Questo comando è un alias per **choco install**.

Maggiori informazioni: <https://docs.chocolatey.org/en-us/choco/commands/install>.

- Consulta la documentazione del comando originale:

```
tldr choco install
```

# clip

Copia il contenuto di input negli Appunti di Windows.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/clip>.

- Manda l'output della riga di comando negli Appunti di Windows:

```
{{dir}} | clip
```

- Copia i contenuti di un file negli appunti di Windows:

```
clip < {{percorso\per\file.ext}}
```

- Copia il testo con una nuova riga finale negli appunti di Windows:

```
echo {{testo_generico}} | clip
```

- Copia il testo senza una nuova riga finale negli appunti di Windows:

```
echo | set /p="testo_generico" | clip
```

# clist

Questo comando è un alias per **choco list**.

Maggiori informazioni: <https://docs.chocolatey.org/en-us/choco/commands/list>.

- Consulta la documentazione del comando originale:

```
tldr choco list
```

# cls

Pulisce lo schermo del terminale.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/cls>.

- Pulisce lo schermo:

```
cls
```

# cmd

L'interprete dei comandi di Windows.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/cmd>.

- Lancia una nuova istanza dell'interprete dei comandi:

```
cmd
```

- Esegue il comando specificato e poi esce:

```
cmd /c {{echo Ciao mondo}}
```

- Esegue il comando specificato e poi apre una shell interattiva:

```
cmd /k {{echo Ciao mondo}}
```

- Disabilita l'uso di `echo` nell'output di un comando:

```
cmd /q
```

- Abilita o disabilita le estensioni ai comandi:

```
cmd /e:{{on|off}}
```

- Abilita o disabilita l'espansione delle variabili d'ambiente:

```
cmd /v:{{on|off}}
```

- Forza l'encoding delle stringhe in Unicode per l'output:

```
cmd /u
```



# color

Imposta i colori di primo piano e di sfondo della console.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/color>.

- Imposta i colori della console sui valori predefiniti:

```
color
```

- Elenca i valori di colore disponibili e le informazioni dettagliate:

```
color /?
```

- Imposta il primo piano e lo sfondo della console su un colore specifico utilizzando numeri esadecimali (1-9, a-f):

```
color {{codice_primo_piano}}{{codice_sfondo}}
```

# cpush

Questo comando è un alias per **choco-push**.

Maggiori informazioni: <https://docs.chocolatey.org/en-us/create/commands/push>.

- Consulta la documentazione del comando originale:

```
tldr choco-push
```

# cuninst

Questo comando è un alias per **choco uninstall**.

Maggiori informazioni: <https://docs.chocolatey.org/en-us/choco/commands/uninstall>.

- Consulta la documentazione del comando originale:

```
tldr choco uninstall
```

# date

Mostra o imposta la data del sistema.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/date>.

- Mostra la data corrente e chiedi di impostarne una nuova (lascia vuoto per lasciare quella corrente):

```
date
```

- Mostra la data corrente senza chiedere di impostarne una nuova:

```
date /t
```

- Cambia la data corrente a una data specifica:

```
date {{mese}}-{{giorno}}-{{anno}}
```

# del

Cancella uno o più file.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/del>.

- Cancella uno o più (separati da uno spazio) file o pattern:

```
del {{pattern_di_file}}
```

- Chiedi conferma prima di cancellare ogni file:

```
del {{pattern_di_file}} /p
```

- Forza l'eliminazione di un file a sola lettura:

```
del {{pattern_di_file}} /f
```

- Elimina in modo ricorsivo i file da tutte le sottodirectory:

```
del {{pattern_di_file}} /s
```

- Non chiedere conferma quando si eliminano i file in base a un carattere globale:

```
del {{pattern_di_file}} /q
```

- Mostra il messaggio d'aiuto e fa vedere la lista di attributi disponibili:

```
del /?
```

- Cancella dei file in base a degli attributi specifici:

```
del {{pattern_di_file}} /a {{attributo}}
```

# dir

Elenca i contenuti di una directory.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/dir>.

- Mostra i contenuti della directory corrente:

```
dir
```

- Mostra i contenuti di una directory specifica:

```
dir {{percorso\della\directory}}
```

- Mostra i contenuti della directory corrente includendo quelli nascosti:

```
dir /A
```

- Mostra i contenuti di una directory specifica includendo quelli nascosti:

```
dir {{percorso\della\directory}} /A
```

# diskpart

Gestore di dischi, volumi e partizioni.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/diskpart>.

- Esegui diskpart da solo in un prompt dei comandi da amministratore per inserire la sua riga di comando:

```
diskpart
```

- Elenca tutti i dischi:

```
list disk
```

- Seleziona un volume:

```
select volume {{volume}}
```

- Assegna una lettera di unità al volume selezionato:

```
assign letter {{lettera}}
```

- Crea una nuova partizione:

```
create partition primary
```

- Attiva il volume selezionato:

```
active
```

- Esci da diskpart:

```
exit
```

# exit

Esci dall'istanza CMD corrente o dal file batch corrente.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/exit>.

- Esci dall'istanza CMD corrente:

```
exit
```

- Esci dallo script [b]atch corrente:

```
exit /b
```

- Esci usando uno specifico codice d'uscita:

```
exit {{2}}
```



# explorer

Il file explorer di Windows.

Maggiori informazioni: <https://ss64.com/nt/explorer.html>.

- Apri Windows Explorer:

```
explorer
```

- Apri Windows Explorer nella directory corrente:

```
explorer .
```

- Apri Windows Explorer in una directory specifica:

```
explorer {{percorso\della\directory}}
```

# ftp

Trasferisci file in modo interattivo tra un server FTP locale e remoto.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/ftp>.

- Connettiti a un server FTP remoto:

```
ftp {{host}}
```

- Accedi come utente anonimo:

```
ftp -A {{host}}
```

- Disabilita l'accesso automatico alla connessione iniziale:

```
ftp -n {{host}}
```

- Esegui un file contenente una lista di comandi FTP:

```
ftp -s:{{percorso\to\file}} {{host}}
```

- Carica file (espressione glob):

```
mget {{*.png}}
```

- Carica file (espressione glob):

```
mput {{*.zip}}
```

- Cancella file sul server remoto:

```
mdelete {{*.txt}}
```

- Mostra la pagina di aiuto dettagliato:

```
ftp --help
```

# ipconfig

Mostra e gestisce le configurazioni di rete di Windows.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/ipconfig>.

- Mostra la lista delle schede di rete:

```
ipconfig
```

- Mostra la lista dettagliata delle schede di rete:

```
ipconfig /all
```

- Rinnova l'indirizzo IP di una scheda di rete:

```
ipconfig /renew {{scheda_di_rete}}
```

- Libera gli indirizzi IP per una scheda di rete:

```
ipconfig /release {{scheda_di_rete}}
```

- Rimuovi tutti i dati dalla cache DNS:

```
ipconfig /flushdns
```

# iwr

Questo comando è un alias per **invoke-webrequest**.

Maggiori informazioni: <https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest>.

- Consulta la documentazione del comando originale:

```
tldr invoke-webrequest
```

# logoff

Termina una sessione di accesso.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/logoff>.

- Termina la sessione in uso:

```
logoff
```

- Termina una sessione con il suo nome o ID:

```
logoff {{nome_sessione|id_sessione}}
```

- Termina la sessione su un server specifico connesso tramite RDP:

```
logoff {{name_sessione|id_sessione}} /server:{{nome_server}}
```

# mount

Montare le condivisioni di rete del Network File System (NFS).

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/mount>.

- Monta una share con la lettera di unità "Z":

```
mount \\{{nome_del_computer}}\{{nome_della_share}} {{Z:}}
```

- Monta una share con la successiva lettera di unità disponibile:

```
mount \\{{nome_del_computer}}\{{nome_della_share}} *
```

- Monta una share con un timeout di lettura in secondi (predefinito a 0,8, può essere 0,9 o da 1 a 60):

```
mount -o timeout={{secondi}} \\{{nome_del_computer}}\{{nome_della_share}} {{Z:}}
```

- Monta una share e riprova fino a 10 volte se fallisce:

```
mount -o retry=10 \\{{nome_del_computer}}\{{nome_della_share}} {{Z:}}
```

- Monta una share forzando la distinzione tra maiuscole e minuscole:

```
mount -o casesensitive \\{{nome_del_computer}}\{{nome_della_share}} {{Z:}}
```

- Monta una share come utente anonimo:

```
mount -o anon \\{{nome_del_computer}}\{{nome_della_share}} {{Z:}}
```

- Monta una share utilizzando un tipo di montaggio specifico:

```
mount -o mtype={{soft|hard}} \\{{nome_del_computer}}\{{nome_della_share}} {{Z:}}
```

# pwsh where

Questo comando è un alias per **Where-Object**.

Maggiori informazioni: <https://learn.microsoft.com/powershell/module/microsoft.powershell.core/where-object>.

- Consulta la documentazione del comando originale:

```
tldr Where-Object
```

# rd

Questo comando è un alias per **rmdir**.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/rd>.

- Consulta la documentazione del comando originale:

```
tldr rmdir
```



# robocopy

Copia robusta di file e directory.

Di default, i file saranno copiati solo se la sorgente e la destinazione hanno timestamp o dimensioni diverse.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/robocopy>.

- Copia tutti i file .jpg e .bmp da una directory ad un'altra:

```
robocopy {{percorso\della\directory_sorgente}}  
{{percorso\della\directory_destinazione}} {{*.jpg}} {{*.bmp}}
```

- Copia tutti i file e le sottodirectory, includendo anche quelle vuote:

```
robocopy {{percorso\della\directory_sorgente}}  
{{percorso\della\directory_destinazione}} /E
```

- Mirror/Sincronizza una directory, eliminando tutto ciò che non è nella sorgente e includendo tutti gli attributi e le autorizzazioni:

```
robocopy {{percorso\della\directory_sorgente}}  
{{percorso\della\directory_destinazione}} /MIR /COPYALL
```

- Copia tutti i file e le sottodirectory, escludendo i file di origine più vecchi rispetto ai file di destinazione:

```
robocopy {{percorso\della\directory_sorgente}}  
{{percorso\della\directory_destinazione}} /E /X0
```

- Elencare tutti i file di dimensioni maggiori o uguali a 50 MB anziché copiarli:

```
robocopy {{percorso\della\directory_sorgente}}  
{{percorso\della\directory_destinazione}} /MIN:{{52428800}} /  
L
```

- Consenti la ripresa se la connessione di rete viene interrotta, limita i tentativi di ripresa a 5 ed il tempo di attesa a 15 secondi:

```
robocopy {{percorso\della\directory_sorgente}}  
{{percorso\della\directory_destinazione}} /Z /R:5 /W:15
```

- Mostra l'aiuto dettagliato:

```
robocopy /?
```

# sls

Questo comando è un alias per **Select-String**.

Maggiori informazioni: <https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/select-string>.

- Consulta la documentazione del comando originale:

```
tldr select-string
```

# tasklist

Mostra una lista dei processi in esecuzione su una macchina locale o remota.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/tasklist>.

- Mostra i processi in esecuzione:

```
tasklist
```

- Mostra i processi in esecuzione in un formato specifico:

```
tasklist /fo {{table|list|csv}}
```

- Mostra i processi in esecuzione utilizzando il nome del file .exe o .dll specificato:

```
tasklist /m {{pattern_di_moduli}}
```

- Mostra i processi in esecuzione su una macchina remota:

```
tasklist /s {{nome_macchina_remota}} /u {{nome_utente}} /p {{password}}
```

- Visualizzare i servizi utilizzando ogni processo:

```
tasklist /svc
```

# whoami

Mostra informazioni riguardo all'utente utilizzato.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/whoami>.

- Mostra il nome dell'utente corrente:

```
whoami
```

- Mostra i gruppi di cui l'utente corrente fa parte:

```
whoami /groups
```

- Mostra i privilegi dell'utente corrente:

```
whoami /priv
```

- Mostra l'UPN (User Principal Name) dell'utente corrente:

```
whoami /upn
```

- Mostra l'ID di logon dell'utente corrente:

```
whoami /logonid
```

- Mostra tutte le informazioni riguardo all'utente corrente:

```
whoami /all
```

# winget

Gestore pacchetti a linea di comando di Windows.

Maggiori informazioni: <https://learn.microsoft.com/windows/package-manager/winget>.

- Installa un pacchetto:

```
winget install {{pacchetto}}
```

- Mostra informazioni su un pacchetto:

```
winget show {{pacchetto}}
```

- Cerca un pacchetto:

```
winget search {{pacchetto}}
```

- Aggiorna tutti i pacchetti all'ultima versione:

```
winget upgrade --all
```

- Elenca tutti i pacchetti installati che possono essere gestiti con winget:

```
winget list --source winget
```

# xcopy

Copia di file e directory.

Maggiori informazioni: <https://learn.microsoft.com/windows-server/administration/windows-commands/xcopy>.

- Copia il/i file nella destinazione specificata:

```
xcopy {{percorso\del\file_o_directory}}  
{{percorso\della\destinazione}}
```

- Elenca i file da copiare prima di copiarli verso la destinazione:

```
xcopy {{percorso\del\file_o_directory}}  
{{percorso\della\destinazione}} /p
```

- Copia solo la struttura della directory senza i file:

```
xcopy {{percorso\del\file_o_directory}}  
{{percorso\della\destinazione}} /t
```

- Copia le directory vuote:

```
xcopy {{percorso\del\file_o_directory}}  
{{percorso\della\destinazione}} /e
```

- Mantieni le politiche di accesso della sorgente (ACL) nella directory di destinazione:

```
xcopy {{percorso\del\file_o_directory}}  
{{percorso\della\destinazione}} /o
```

- Continua l'operazione dopo l'interruzione della connessione di rete:

```
xcopy {{percorso\del\file_o_directory}}  
{{percorso\della\destinazione}} /z
```

- Sovrascrivi automaticamente i file di destinazione già esistenti:

```
xcopy {{percorso\del\file_o_directory}}  
{{percorso\della\destinazione}} /y
```

- Mostra l'aiuto dettagliato:

```
xcopy /?
```

