tldr pages book

Simplified and community-driven man pages

Generated on Thu Oct 26 08:37:25 2023

Website: https://tldr.sh

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

Android

am

Android aktivite yöneticisi.

Daha fazla bilgi için: https://developer.android.com/studio/command-line/adb#am.

- Belirtilmiş bir aktiviteyi başlat:

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

- Bir aktivite başlatıp veriyi ona aktar:

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

- Belirtilmiş bir eylem ve kategoriyi karşılayan bir aktivite başlat:

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

- Bir kastı URI'a çevir:

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

bugreport

Bir Android bug raporu göster.

Bu komut yalnızca **`adb shell`** ile kullanılabilir.

Daha fazla bilgi için: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreport.

- Bir Android cihazı için eksizsiz bug raporu göster:

  ```
  bugreport
  ```

bugreportz

Ziplenmiş bir Android bug raporu oluştur.

Bu komut yalnızca **`adb shell`** ile kullanılabilir.

Daha fazla bilgi için: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreportz.

- Bir Android cihazı için ziplenmiş şekilde eksizsiz bir bug raporu oluşturGenerate a complete zipped bug report of an Android device:

  ```
  bugreportz
  ```

- Çalışan `bugreportz` işleminin durumunu göster:

  ```
  bugreportz -p
  ```

- `bugreportz` sürümünü göster:

  ```
  bugreportz -v
  ```

- Yardım görüntüle:

  ```
  bugreportz -h
  ```

cmd

Android servis yöneticisi.

Daha fazla bilgi için: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/cmd/.

- Tüm çalışan servisleri sırala:

  ```
  cmd -l
  ```

- Belirtilen servisi çağır:

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

- Belirtilen argümanlar ile servis çağır:

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

dalvikvm

Android Java sanal makinesi.

Daha fazla bilgi için: https://source.android.com/devices/tech/dalvik.

- Bir Java programı başlar:

```
dalvikvm -classpath {{örnek/dosya.jar}} {{sınıf_ismi}}
```

dumpsys

Android sistem servisleri ile ilgili bilgi sağla.

Bu komut yalnızca **`adb shell`** ile kullanılabilir.

Daha fazla bilgi için: https://developer.android.com/studio/command-line/dumpsys.

- Tüm sistem servisleri için tanısal bir çıktı al:

  ```
  dumpsys
  ```

- Belirtilen sistem servisi için tanısal bir çıktı al:

  ```
  dumpsys {{servis}}
  ```

- `dumpsys` komutunun hakkında bilgi verebileceği tüm servisleri sırala:

  ```
  dumpsys -l
  ```

- Bir servis için servise özel argümanları sırala:

  ```
  dumpsys {{servis}} -h
  ```

- Tanı çıktısından belirtilen servisi çıkart:

  ```
  dumpsys --skip {{servis}}
  ```

- Saniye bazında bir zaman aşımı süresi belirle (varsayılan 10 saniyedir):

  ```
  dumpsys -t {{saniye}}
  ```

getprop

Android sistem özellikleri ile ilgili bilgi görüntüle.

Daha fazla bilgi için: https://manned.org/getprop.

- Android sistem özellikleri ile ilgili bilgi görüntüle:

  ```
  getprop
  ```

- Belirtilen özellik ile ilgili bilgi görüntüle:

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

- SDK API seviyesini görüntüle:

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

- Android sürümünü görüntüle:

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

- Android cihaz modelini görüntüle:

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

- OEM kilit durumunu görüntüle:

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

- Android'in Wi-Fi kartının MAC adreslerini görüntüle:

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

input

Olay kodlarını ve dokunmatik ekran mimiklerini bir Android cihazına yolla.

Bu komut yalnızca **`adb shell`** ile kullanılabilir.

Daha fazla bilgi için: https://developer.android.com/reference/android/view/KeyEvent.html#constants_1.

- Bir Android cihazına tek karakter için etkinlik kodu gönder:

```
input keyevent {{etkinlik_kodu}}
```

- Bir Android cihazına yazı gönder (`%s` boşlukları temsil eder):

```
input text "{{yazı}}"
```

- Bir Android cihazına tek dokunuş gönder:

```
input tap {{x_poz}} {{y_poz}}
```

- Bir Android cihazına kaydırma mimiği gönder:

```
input swipe {{x_başlangıç}} {{y_başlangıç}} {{x_son}}
{{y_son}} {{ms_süre}}
```

- Bir Android cihazına kaydırma mimiği kullanarak uzun dokunuş gönder:

```
input swipe {{x_poz}} {{y_poz}} {{x_poz}} {{y_poz}}
{{ms_süre}}
```

logcat

Sistem mesajlarının kaydını görüntüle.

Daha fazla bilgi için: https://developer.android.com/studio/command-line/logcat.

- Sistem kaydını görüntüle:

```
logcat
```

- Sistem kayıtlarını bir dosyaya yaz:

```
logcat -f {{örnek/dosya}}
```

- Düzenli ifadeye uyan satırları görüntüle:

```
logcat --regex {{düzenli_ifade}}
```

pkg

Termux için paket yönetim aracı.

Daha fazla bilgi için: https://wiki.termux.com/wiki/Package_Management.

- İndirilmiş tüm paketleri yükselt:

  ```
  pkg upgrade
  ```

- Belirtilen paketi indir:

  ```
  pkg install {{paket}}
  ```

- Belirtilen paketi kaldır:

  ```
  pkg uninstall {{paket}}
  ```

- Belirtilen paketi yeniden indir:

  ```
  pkg reinstall {{paket}}
  ```

- Belirtilen paketi ara:

  ```
  pkg search {{paket}}
  ```

pm

Android cihazındaki uygulamalar ile ilgili bilgi göster.

Daha fazla bilgi için: https://developer.android.com/studio/command-line/adb#pm.

- İndirilen tüm uygulamaların sırala:

  ```
  pm list packages
  ```

- İndirilen tüm sistem uygulamalarını sırala:

  ```
  pm list packages -s
  ```

- İndirilen tüm üçüncü el uygulamaları sırala:

  ```
  pm list packages -3
  ```

- Belirtilen anahtar kelimelere uyan uygulamaları sırala:

  ```
  pm list packages {{anahtar_kelimeler}}
  ```

- Belirtilen uygulamanın APK'sine giden yolu görüntüle:

  ```
  pm path {{uygulama}}
  ```

screencap

Bir mobil ekranın ekran görüntüsünü al.

Bu komut sadece **`adb shell`** üzerinden kullanılabilir.

Daha fazla bilgi için: https://developer.android.com/studio/command-line/adb#screencap.

- Bir ekran görüntüsü al:

```
screencap {{dosya/yolu}}
```

settings

Android işletim sistemi ile ilgili bilgi al.

Daha fazla bilgi için: https://adbinstaller.com/commands/adb-shell-settings-5b670d5ee7958178a2955536.

- `global` isim alanındaki ayarların sırasını görüntüle:

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

- Belirtilen ayarın değerini al:

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

- Bir ayarın değerini belirle:

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

- Belirtilen ayarı sil:

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

wm

Bir Android cihazının ekranı ile ilgili bilgi göster.

Bu komut yalnızca **`adb shell`** ile kullanılabilir.

Daha fazla bilgi için: https://adbinstaller.com/commands/adb-shell-wm-5b672b17e7958178a2955538.

- Bir Android cihazının ekranının fiziksel boyutunu görüntüle:

  ```
  wm {{size}}
  ```

- Bir Android cihazının ekranının fiziksel derinliğini görüntüle:

  ```
  wm {{density}}
  ```

Common

7z

Yüksek sıkıştırma oranına sahip dosya sıkıştırıcısı.

Daha fazla bilgi için: https://manned.org/7z.

- Dosya veya dizin arşivle:

```
7z a {{sikistirilmis_dosya.7z}} {{yoldan/dosya_veya_dizine}}
```

- Varolan bir arşivi çözümle (headerlar dahil):

```
7z a {{sifrelenmiş_dosya.7z}} -p{{parola}} -mhe=on
{{sikistirilmis_dosya.7z}}
```

- Varolan 7z dosyasını orijinal dizin yapısıyla dışa aktar:

```
7z x {{sikistirilmis_dosya.7z}}
```

- Arşivi kullanıcı tarafından belirtilmiş çıkış noktasına aktar:

```
7z x {{sikistirilmis_dosya.7z}} -o{{yoldan/çıktıya}}
```

- Arşivi `stdout`'a aktar:

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

- Spesifik bir arşivleme türüyle arşivle:

```
7z a -t{{7z|bzip2|gzip|lzip|tar|zip}} {{sikistirilmis_dosya.
7z}} {{yoldan/dosya_veya_dizine}}
```

- Kullanılabilir arşiv türlerini sırala:

```
7z i
```

- Arşiv dosyasının içeriğini listele:

```
7z l {{sikistirilmis_dosya.7z}}
```

airmon-ng

Kablosuz ağ cihazlarında izleme modunu etkinleştirin.

Daha fazla bilgi için: https://www.aircrack-ng.org/doku.php?id=airmon-ng.

- Kablosuz cihazları ve durumlarını listeleyin:

  ```
  sudo airmon-ng
  ```

- Belirli bir cihaz için izleme modunu açın:

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

- Kablosuz cihazları kullanan rahatsız edici işlemleri sonlandırın:

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

- Belirli bir ağ arabirimi için izleme modunu kapatın:

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

alias

Takma adlar/kısayollar (bir komut dizesi ile değiştirilen sözcükler) oluşturur.

Kısayollar, kabuğun yapılandırma dosyasında (örneğin **`~/.bashrc`**) tanımlanmadığı sürece geçerli kabuk oturumuyla birlikte sona erer.

Daha fazla bilgi için: https://tldp.org/LDP/abs/html/aliases.html.

- Tüm kısayolları listele:

```
alias
```

- Genel bir kısayol oluştur:

```
alias {{kelime}}="{{komut}}"
```

- Bir kısayolun verildigi komutu göster:

```
alias {{kelime}}
```

- Bir kısayolu kaldır:

```
unalias {{kelime}}
```

- `rm`'yi interaktif bir komuta dönüştür:

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

- `la`'yi `ls --all` için kısayol olarak oluştur:

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

anki

Güçlü ve akıllı bir aralıklı tekrar programı.

Daha fazla bilgi için: https://docs.ankiweb.net.

- Çalıştır:

  ```
  anki
  ```

- Belirtilen bir profil ismi ile çalıştır:

  ```
  anki -p {{profil_ismi}}
  ```

- Belirtilen bir dil ile çalıştır:

  ```
  anki -l {{dil}}
  ```

- Belirtilen bir dizinden (konumdan) calıştır:

  ```
  anki -b {{dizin/yolu}}
  ```

arch

Sistem mimarisinin ismini göster.

Ayrıca bakınız: **`uname`**.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/arch.

- Sistemin mimarisini göster:

  ```
  arch
  ```

arp

Sistemin ARP önbelleğini görüntüle ve manipüle et.

Daha fazla bilgi için: https://manned.org/arp.

- Mevcut ARP tablosunu göster:

```
arp -a
```

- Belirli bir girdiyi sil:

```
arp -d {{adres}}
```

- ARP tablosunda bir girdi oluştur:

```
arp -s {{adres}} {{mac_adresi}}
```

awk

Dosyalar üzerinde çalışmak için çok yönlü bir programlama dili.

Daha fazla bilgi için: https://github.com/onetrueawk/awk.

- Boşlukla ayrılmış bir dosyada beşinci sütunu (alan olarak da bilinir) yazdır:

```
awk '{print $5}' {{dosya/yolu/dosya}}
```

- Boşlukla ayrılmış bir dosyada "foo" içeren satırların ikinci sütununu yazdır:

```
awk '/{{foo}}/ {print $2}' {{dosya/yolu/dosya}}
```

- Alan ayırıcı olarak (boşluk yerine) virgül kullanarak dosyadaki her satırın son sütununu yazdır:

```
awk -F ',' '{print $NF}' {{dosya/yolu/dosya}}
```

- Bir dosyanın ilk sütunundaki değerleri topla ve toplamı yazdır:

```
awk '{s+=$1} END {print s}' {{dosya/yolu/dosya}}
```

- İlk satırdan başlayarak her üçüncü satırı yazdır:

```
awk 'NR%3==1' {{dosya/yolu/dosya}}
```

- Koşullara göre farklı değerler yazdır:

```
awk '{if ($1 == "foo") print "Tam eşleşme foo"; else if ($1 ~
"bar") print "Kısmi eşleşme çubuğu"; else print "Baz"}'
{{dosya/yolu/dosya}}
```

- 1. sütun değerinin belirtilen değere eşit olduğu tüm satırları yazdır:

```
awk '($10 == value)'
```

- 1. sütun değeri min ile max arasında olan tüm satırları yazdır:

```
awk '($10 >= min_value && $10 <= max_value)'
```

base32

Bir dosya veya standart veriyi Base32 formatında şifrele veya yalın veri çıktısı olarak deşifre et.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/base32.

- Bir dosyayı şifrele:

```
base32 {{dosyaismi}}
```

- Bir dosyayı deşifre et:

```
base32 --decode {{dosyaismi}}
```

- `stdin`'den şifrele:

```
{{herhangibirkomut}} | base32
```

- `stdin`'den deşifre et:

```
{{herhangibirkomut}} | base32 --decode
```

base64

Bir dosya veya standart veriyi Base64 formatında şifrele veya yalın veri çıktısı olarak deşifre et.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/base64.

- Bir dosyayı şifrele:

```
base64 {{dosyaismi}}
```

- Bir dosyayı deşifre et:

```
base64 --decode {{dosyaismi}}
```

- `stdin`'den şifrele:

```
{{herhangibirkomut}} | base64
```

- `stdin`'den deşifre et:

```
{{herhangibirkomut}} | base64 --decode
```

bat

Dosyaları yazdır ve birleştir.

Sözdizimi vurgulama ve Git entegrasyonuna sahip bir **`cat`** klonu.

Daha fazla bilgi için: https://github.com/sharkdp/bat.

- Bir dosyanın içeriğini standart çıktıya yazdır:

```
bat {{dosya}}
```

- Birkaç dosyayı hedef dosyada birleştir:

```
bat {{dosya1}} {{dosya2}} > {{hedef_dosya}}
```

- Birkaç dosyayı hedef dosyaya ekle:

```
bat {{dosya1}} {{dosya2}} >> {{hedef_dosya}}
```

- Tüm çıktı satırlarını numaralandır:

```
bat -n {{dosya}}
```

- Bir JSON dosyasının sözdizimini vurgula:

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

- Desteklenen tüm dilleri görüntüle:

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

bundler

Bu komut **`bundle`** için bir takma addır.

Daha fazla bilgi için: https://bundler.io/man/bundle.1.html.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr bundle
  ```

cat

Dosyaları yazdır ve birleştir.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/cat.

- Bir dosyanın içeriğini standart çıktıya yazdır:

```
cat {{dosya/yolu}}
```

- Birkaç dosyayı bir çıktı dosyasında birleştir:

```
cat {{dosya/yolu1}} {{dosya/yolu2}} > {{çıktı/dosyası/yolu}}
```

- Birkaç dosyayı bir çıktı dosyasına ekle:

```
cat {{dosya/yolu1}} {{dosya/yolu2}} >> {{çıktı/dosyası/yolu}}
```

- Tüm çıkış satırlarını numaralandır:

```
cat -n {{dosya/yolu}}
```

- Yazdırılamayan ve boşluk karakterleri görüntüle (ASCII değilse `M-` önekiyle):

```
cat -v -t -e {{dosya/yolu}}
```

chroot

Komut veya etkileşimli komut satırını özel kök diziniyle çalıştırır.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/chroot.

- Komutu yeni kök dizini olarak çalıştır:

```
chroot {{yeni/kok/yolu}} {{komut}}
```

- Kullanılacak kullanıcı ve grubu (ID veya isim) belirle:

```
chroot --userspec={{kullanici:grup}}
```

clamav

Bu komut **`clamdscan`** için bir takma addır.

Daha fazla bilgi için: https://www.clamav.net.

- Asıl komutun belgelerini görüntüleyin:

```
tldr clamdscan
```

clang-cpp

Bu komut **`clang++`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr clang++
```

clojure

Bu komut **`clj`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr clj
  ```

cmake

Çok platformlu yapım sistem oluşturucusu.

Hedeflenen sisteme göre Makefile, Visual Studio projeleri ve benzerlerini oluşturur.

Daha fazla bilgi için: https://cmake.org/cmake/help/latest/manual/cmake.1.html.

- Bir Makefile oluştur ve onu aynı dizindeki bir projeyi derlemek için kullan:

  ```
  cmake && make
  ```

- Bir Makefile oluştur ve onu farklı bir "yapim" dizinindeki projeyi derlemek için kullan (kaynak-dışı yapım):

  ```
  cmake -H. -B {{build}} && make -C {{yapim}}
  ```

cola

Bu komut **`git-cola`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr git-cola
  ```

command

Command, kabuğu bir programı herhangi bir fonksiyon ve gömülü özelliğe ve alias'a takılmadan çalıştırmaya zorlar.

Daha fazla bilgi için: https://manned.org/command.

- `ls` programını aynı isimde bir alias olsa dahi çalıştır:

  ```
  command {{ls}}
  ```

- Alias'a atanan özel komutu veya çalıştırılabilir dosyanın yolunu göster:

  ```
  command -v {{komut_ismi}}
  ```

cp

Dosyaları ve dizinleri kopyalayın.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/cp.

- Bir dosyayı başka bir konuma kopyalayın:

```
cp {{dizin/yol/kaynak_dosya.ext}} {{dizin/yol/
hedef_dosya.ext}}
```

- Dosya adını koruyarak bir dosyayı başka bir dizine kopyalayın:

```
cp {{dizin/yol/kaynak_dosya.ext}} {{dizin/yol/
hedeflenen_ana_dizin}}
```

- Bir dizinin içeriğini yinelemeli olarak başka bir konuma kopyalayın (hedef varsa, dizin bunun içine kopyalanır):

```
cp -R {{dizin/yol/kaynak_dizin}} {{dizin/yol/hedef_dizin}}
```

- Bir dizini ayrıntılı modda yinelemeli olarak kopyalayın (dosyaları kopyalandıkça gösterir):

```
cp -vR {{dizin/yol/kaynak_dizin}} {{dizin/yol/hedef_dizin}}
```

- Etkileşimli modda metin dosyalarını başka bir konuma kopyalayın (üzerine yazmadan önce kullanıcıyı uyarır):

```
cp -i {{*.txt}} {{dizin/yol/hedef_dizin}}
```

- Kopyalamadan önce sembolik bağlantıları takip edin:

```
cp -L {{link}} {{dizin/yol/hedef_dizin}}
```

- İlk bağımsız değişkeni hedef dizin olarak kullanın ('xargs ... | cp -t ' için kullanışlıdır):

```
cp -t {{dizin/yol/hedef_dizin}} {{dizin/yol/dosya_veya_dizin1
dizin/yol/dosya_veya_dizin2 ...}}
```

cron

Bu komut **`crontab`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr crontab
  ```

date

Sistem tarihini görüntüleyin veya ayarlayın.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/date.

- Varsayılan yerel biçimi kullanarak geçerli tarihi görüntüleyin:

```
date +"%c"
```

- Geçerli tarihi UTC ve ISO 8601 formatında görüntüleyin:

```
date -u +"%Y-%m-%dT%H:%M:%S%Z"
```

- Geçerli tarihi bir Unix zaman damgası olarak görüntüleyin (Unix zamanından bu yana geçen saniyeler):

```
date +%s
```

- Varsayılan biçimi kullanarak belirli bir tarihi (Unix zaman damgası olarak) görüntüleyin:

```
date -d @1473305798
```

- Belirli bir tarihi Unix zaman damgası biçimine dönüştürün:

```
date -d "{{2018-09-01 00:00}}" +%s --utc
```

- RFC-3339 biçimini kullanarak geçerli tarihi görüntüleyin (`YYYY-AA-GG ss:dd:ss ZD`):

```
date --rfc-3339=s
```

diff

Dosyaları ve dizinleri karşılaştırın.

Daha fazla bilgi için: https://man7.org/linux/man-pages/man1/diff.1.html.

- Dosyaları karşılaştır (`eski_dosya`'yı `yeni_dosya`'ya dönüştürmek için yapılan değişiklikleri listeler):

```
diff {{eski_dosya}} {{yeni_dosya}}
```

- Boşlukları yok sayarak dosyaları karşılaştırın:

```
diff --ignore-all-space {{eski_dosya}} {{yeni_dosya}}
```

- Farkları yan yana göstererek dosyaları karşılaştırın:

```
diff --side-by-side {{eski_dosya}} {{yeni_dosya}}
```

- Farkları birleştirilmiş biçimde (`git diff` tarafından kullanıldığı gibi) göstererek dosyaları karşılaştırın:

```
diff --unified {{eski_dosya}} {{yeni_dosya}}
```

- Dizinleri yinelemeli olarak karşılaştırın (farklı dosya/dizin adlarını ve dosyalarda yapılan değişiklikleri gösterir):

```
diff --recursive {{eski_dizin}} {{yeni_dizin}}
```

- Dizinleri karşılaştırın, yalnızca farklı olan dosyaların adlarını gösterin:

```
diff --recursive --brief {{eski_dizin}} {{yeni_dizin}}
```

- Git için iki metin dosyasının farklarından, var olmayan dosyaları ise boş olarak değerlendirerek bir yama dosyası oluşturun:

```
diff --text --unified --new-file {{eski_dosya}}
{{yeni_dosya}} > {{fark.patch}}
```

dig

DNS sunucularına sorgulama yapmak için kullanılan bir komuttur.

Daha fazla bilgi için: https://manned.org/dig.

- İlgili sunucu ile ilgili IP adresi sorgulaması yapılır:

  ```
  dig +short {{example.com}}
  ```

- DNS sorgulaması için alternatif server kullanılır:

  ```
  dig @{{1.1.1.1}} {{example.com}}
  ```

- Tersine DNS sorgulaması yapılır:

  ```
  dig -x {{1.1.1.1}}
  ```

- Belli başlı DNS sunucuları ile ilgili kayıtları sorgular:

  ```
  dig +short {{example.com}} {{A|MX|TXT|CNAME|NS}}
  ```

dirname

Belirtilen dosya veya yolun ana dizinini hesaplar.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/dirname.

- Belirtilen yolun ana dizinini hesapla:

```
dirname {{dosya_veya_dizine/giden/yol}}
```

- Birden çok yolun ana dizinini hesapla:

```
dirname {{dosya_veya_dizine/giden/yol_1}}
{{dosya_veya_dizine/giden/yol_2}}
```

- Komut çıktısını yeni satır yerine NUL karakteri ile sınırlandırma (`xargs` yazılımı ile kullanırken işe yarar):

```
dirname --zero {{dosya_veya_dizine/giden/yol_1}}
{{dosya_veya_dizine/giden/yol_2}}
```

dirs

Dizin yığını görüntüler veya üzerinde oynama yapar.

Dizin yığını, **`pushd`** ve **`popd`** komutlarıyla üzerinde oynama yapılabilen, son ziyaret edilen dizinleri gösteren bir listedir.

Daha fazla bilgi için: https://www.gnu.org/software/bash/manual/bash.html#Directory-Stack-Builtins.

- Dizin yığınını her madde arasında boşluk olacak şekilde görüntüle:

```
dirs
```

- Dizin yığınını her satır başı tek madde olacak şekilde görüntüle:

```
dirs -p
```

- Dizin yığınında 0'dan başlamak üzere yalnızca nth girişini göster:

```
dirs +{{N}}
```

- Dizin yığınını temizle:

```
dirs -c
```

dirsearch

Ağ yolu tarayıcı.

Daha fazla bilgi için: https://github.com/maurosoria/dirsearch.

- Bir ağ sunucusunu yaygın eklentiler içeren yaygın yollar için tarayın:

```
dirsearch --url {{url}} --extensions-list
```

- Ağ sunucularını içeren bir listeyi `.php` eklentili yaygın yollar için tarayın:

```
dirsearch --url-list {{örnek/url-listesi.txt}} --extensions
{{php}}
```

- Bir ağ sunucusunu yaygın eklentiler içeren belirtilen yollar için tarayın:

```
dirsearch --url {{url}} --extensions-list --wordlist {{path/
to/url-yol-listesi.txt}}
```

- Bir ağ sunucusunu çerez kullanarak tarayın:

```
dirsearch --url {{url}} --extensions {{php}} --cookie
{{cookie}}
```

- Bir ağ sunucusunu `HEAD` HTTP metodunu kullanarak tarayın:

```
dirsearch --url {{url}} --extensions {{php}} --http-method
{{HEAD}}
```

- Bir ağ sunucusunu tarayın ve sonuçları bir `.json` dosyasına kaydedin:

```
dirsearch --url {{url}} --extensions {{php}} --json-report
{{örnek/rapor_dosyası.json}}
```

docker build

Bir Dockerfile'dan imge yaratın.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/build/.

- Mevcut dizindeki Dockerfile'dan bir docker imgesi oluşturun:

```
docker build .
```

- Belirtilen URL'deki Dockerfile'dan bir docker imgesi oluşturun:

```
docker build {{ornekadres.com/ornek-dizin/ornek-docker-projesi}}
```

- Bir docker imgesi oluşturun ve etiketleyin:

```
docker build --tag {{isim:etiket}} .
```

- İmge oluştururken çerez kullanımını etkisizleştirin:

```
docker build --no-cache --tag {{isim:etiket}} .
```

- Belirtilen Dockerfile ile bir docker imgesi oluşturun:

```
docker build --file {{Dockerfile}} .
```

- Kişiselleştirilmiş yapım-zaman değerleriyle oluşturun:

```
docker build --build-arg {{HTTP_PROXY=http://10.20.30.2:1234}} --build-arg {{FTP_PROXY=http://40.50.60.5:4567}} .
```

docker compose

Çoklu konteynerli docker uygulamalarını çalıştırın ve yönetin.

Daha fazla bilgi için: https://docs.docker.com/compose/reference/.

- Tüm konteynerleri listele:

```
docker compose ps
```

- Mevcut dizinde bir `docker-compose.yml` dosyası çalıştırarak arkaplandaki tüm konteynerleri çalıştırın ve başlatın:

```
docker compose up --detach
```

- Tüm konteynerleri çalıştırın ve gerekiyorsa yeniden oluşturun:

```
docker compose up --build
```

- Tüm konteynerleri alternatif bir beste dosyasıyla başlatın:

```
docker compose --file {{yoldan/dosyaya}} up
```

- Çalışan tüm konteynerleri durdurun:

```
docker compose stop
```

- Tüm konteynerleri, ağları, imgeleri ve alanları durdurun ve silin:

```
docker compose down --rmi all --volumes
```

- Tüm konteynerler için logları takip edin:

```
docker compose logs --follow
```

- Belirtilmiş bir konteyner için logları takip edin:

```
docker compose logs --follow {{konteyner_ismi}}
```

docker exec

Halihazırda çalışan bir Docker konteyneri üstünde komut çalıştır.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/exec/.

- Halihazırda çalışan bir konteynerin üstünde interaktif bir kabuk oturumunu çalıştır:

```
docker exec --interactive --tty {{konteyner_ismi}} {{/bin/bash}}
```

- Halihazırda çalışan bir konteynerin üstüne arkaplanda çalışmak üzere (ayrılmış) bir komut çalıştır:

```
docker exec --detach {{konteyner_ismi}} {{komut}}
```

- Belirtilen bir komutu üstünde çalıştırmak adına çalışan dizini seç:

```
docker exec --interactive -tty --workdir {{örnek/dizin}} {{konteyner_ismi}} {{komut}}
```

- Varolan konteyner üstünde arkaplanda çalışmak üzere bir komut çalıştır ancak `stdin`'i açık tut:

```
docker exec --interactive --detach {{konteyner_ismi}} {{komut}}
```

- Çalışmakta olan bir bash oturumu içinde bir çevre değişkeni belirle:

```
docker exec --interactive --tty --env {{değişken_ismi}}={{value}} {{konteyner_ismi}} {{/bin/bash}}
```

- Belirtilmiş bir kullanıcı olarak komut çalıştır:

```
docker exec --user {{kullanıcı}} {{konteyner_ismi}} {{komut}}
```

docker images

Docker imgelerini yönet.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/images/.

- Tüm Docker imgelerini listele:

  ```
  docker images
  ```

- Orta düzeyler de dahil olmak üzere tüm Docker imgelerini sırala:

  ```
  docker images --all
  ```

- Çıktıyı sessiz modda (yalnızca sayısal ID'ler olarak) sırala:

  ```
  docker images --quiet
  ```

- Herhangi bir konteyner tarafından kullanılmayan tüm Docker imgelerini sırala:

  ```
  docker images --filter dangling=true
  ```

- İsminde belirtilen dizeleri taşıyan imgeleri sırala:

  ```
  docker images "{{*isim*}}"
  ```

docker inspect

Docker objelerinde bulunan düşük seviye bilgiyi gösterir.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/inspect/.

- Yardım içeriğini göster:

```
docker inspect
```

- Bir konteyner, imge veya hacim ile ilgili bilgiyi ismini veya ID'sini girerek görüntüle:

```
docker inspect {{konteyner|imge|ID}}
```

- Bir konteynerin IP adresini görüntüle:

```
docker inspect --format='{{range .NetworkSettings.Networks}}
{{.IPAddress}}{{end}}' {{konteyner}}
```

- Konteynerin log dosyasının yolunu görüntüle:

```
docker inspect --format='{{.LogPath}}' {{konteyner}}
```

- Konteynerin imge ismini görüntüle:

```
docker inspect --format='{{.Config.Image}}' {{konteyner}}
```

- Konfigürasyon bilgisini JSON olarak görüntüle:

```
docker inspect --format='{{json .Config}}' {{konteyner}}
```

- Tüm port limanlayıcıları görüntüle:

```
docker inspect --format='{{range $p, $conf
:= .NetworkSettings.Ports}} {{$p}} -> {{(index $conf
0).HostPort}} {{end}}' {{konteyner}}
```

docker logs

Konteyner kaydını yazdırır.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/logs.

- Bir konteyner içindeki kayıtları yazdır:

  ```
  docker logs {{konteyner_ismi}}
  ```

- Kayıtları yazdır ve izle:

  ```
  docker logs -f {{konteyner_ismi}}
  ```

- Son 5 kaydı yazdır:

  ```
  docker logs {{konteyner_ismi}} --tail {{5}}
  ```

- Kayıtları yazdır ve zaman damgaları ile iliştir:

  ```
  docker logs -t {{konteyner_ismi}}
  ```

- Belli bir konteyner çalışma zamanındaki (i.e. 23m, 10s, 2013-01-02T13:23:37) kayıtları yazdır:

  ```
  docker logs {{konteyner_ismi}} --until {{zaman}}
  ```

docker-machine

Docker çalıştıran makineler oluştur ve onları yönet.

Daha fazla bilgi için: https://docs.docker.com/machine/reference/.

- Halihazırda çalışan docker makinelerini sırala:

  ```
  docker-machine ls
  ```

- Belirli bir isim ile docker makinesi oluştur:

  ```
  docker-machine create {{isim}}
  ```

- Bir makinenin durumunu öğren:

  ```
  docker-machine status {{isim}}
  ```

- Bir makineyi başlat:

  ```
  docker-machine start {{isim}}
  ```

- Bir makineyi durdur:

  ```
  docker-machine stop {{isim}}
  ```

- Bir makine hakkındaki bilgileri incele:

  ```
  docker-machine inspect {{isim}}
  ```

docker network

Docker ağları oluştur ve yönet.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/network/.

- docker daemon'daki tüm müsait ve düzenlenmiş ağları sırala:

  ```
  docker network ls
  ```

- Kullanıcı tarafından belirtilmiş bir ağ oluştur:

  ```
  docker network create --driver {{driver_name}} {{ağ_ismi}}
  ```

- Boşluk ile ayrılmış bir ağ listesinin detaylı bilgisini görüntüle:

  ```
  docker network inspect {{ağ_ismi}}
  ```

- Bir konteyneri isim veya ID kullanarak bir ağa bağla:

  ```
  docker network connect {{ağ_ismi}} {{konteyner_ismi|ID}}
  ```

- Bir konteyneri bir ağdan çıkar:

  ```
  docker network disconnect {{ağ_ismi}} {{konteyner_ismi|ID}}
  ```

- Tüm kullanılmayan (hiçbir konteyner tarafından belirtilmeyen) ağları sil:

  ```
  docker network prune
  ```

- Kullanılmayan ağların boşluk ile ayrılmış bir listesini sil:

  ```
  docker network rm {{ağ_ismi}}
  ```

docker ps

Docker konteynerlerini sırala.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/ps/.

- Halihazırda çalışan docker konteynerlerini listele:

```
docker ps
```

- Tüm (durmuş veya çalışan) docker konteynerlerini listele:

```
docker ps --all
```

- En son oluşturulan (durmuş veya çalışan) konteynerleri listele:

```
docker ps --latest
```

- İsimlerinde belirtilen dizeleri içeren konteynerleri filtrele:

```
docker ps --filter="name={{isim}}"
```

- Belirtilen imge ile akrabalık taşıyan konteynerleri filtrele:

```
docker ps --filter "ancestor={{imge}}:{{tag}}"
```

- Konteynerleri çıkış durum koduna göre filtrele:

```
docker ps --all --filter="exited={{kod}}"
```

- Konteynerleri mevcut durumlarına (oluşturulma, çalışma, silinme, durma, çıkma ve ölme) göre sırala:

```
docker ps --filter="status={{mevcut_durum}}"
```

- Belirtilmiş bir hacmi gömen veya belirtilmiş bir yola gömülmüş hacmi içeren konteynerleri filtrele:

```
docker ps --filter="volume={{örnek/dizin}}" --format "table
{{.ID}}\t{{.Image}}\t{{.Names}}\t{{.Mounts}}"
```

docker rmi

Bir veya daha fazla Docker imgesini sil.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/rmi/.

- Yardım göster:

  ```
  docker rmi
  ```

- Bir veya daha fazla imgeyi isimlerini belirterek sil:

  ```
  docker rmi {{imge1 imge2 ...}}
  ```

- Bir imgeyi zorla sil:

  ```
  docker rmi --force {{imge}}
  ```

- Bir imgeyi etiketlenmemiş ana yollarını silmeden sil:

  ```
  docker rmi --no-prune {{imge}}
  ```

docker run

Yeni bir Docker konteynerinde bir komut çalıştır.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/run/.

- Yeni bir konteynerde, etiketlenmiş bir imgeden komut çalıştır:

```
docker run {{imge:etiket}} {{komut}}
```

- Yeni bir konteynerde arkaplanda çalışacak şekilde komut çalıştır ve ID'sini göster:

```
docker run --detach {{imge}} {{komut}}
```

- İnteraktif mod ve pseudo-TTY'deki bir açık-kapalı konteynerde komut çalıştır:

```
docker run --rm --interactive --tty {{imge}} {{komut}}
```

- Yeni bir konteynerde geçebilmiş çevresel değişkenler ile komut çalıştır:

```
docker run --env '{{değişken}}={{değer}}' --env {{değişken}}
{{imge}} {{komut}}
```

- Yeni bir konteynerde bağlama takılı hacimlerle komut çalıştır:

```
docker run --volume {{örnek/host}}:{{örnek/konteyner}}
{{imge}} {{komut}}
```

- Yayınlanmış portları içeren yeni bir konteynerde komut çalıştır:

```
docker run --publish {{host_portu}}:{{konteyner_portu}}
{{imge}} {{komut}}
```

docker save

Bir veya daha fazla docker imgesini arşivlemek için dışa aktar.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/save/.

- Bir imgeyi, `stdout`'u tar arşivine yönlendirerek kaydet:

  ```
  docker save {{imge}}:{{etiket}} > {örnek/dosya.tar}}
  ```

- Bir imgeyi, bir tar arşivine kaydet:

  ```
  docker save --output {{örnek/dosya.tar}} {{imge}}:{{etiket}}
  ```

- Bir imgenin tüm etiketlerini kaydet:

  ```
  docker save --output {{örnek/dosya.tar}} {{imge_ismi}}
  ```

- Bir imgenin belirli etiketlerini kaydetmek için elle seç:

  ```
  docker save --output {{örnek/dosya.tar}} {{imge_ismi:etiket1
  imge_ismi:etiket2 ...}}
  ```

docker secret

Docker swarm sırlarını yönet.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/secret/.

- `stdin`'den yeni bir sır yarat:

  ```
  {{komut}} | docker secret create {{sır_ismi}} -
  ```

- Bir dosyadan yeni sır oluşturun:

  ```
  docker secret create {{sır_ismi}} {{örnek/dosya}}
  ```

- Tüm sırları sırala:

  ```
  docker secret ls
  ```

- Bir veya daha fazla sırra dair detaylı bilgiyi insan dostu bir formatta göster:

  ```
  docker secret inspect --pretty {{sır_ismi1 sır_ismi2 ...}}
  ```

- Bir veya daha fazla sırrı sil:

  ```
  docker secret rm {{sır_ismi1 sır_ismi2 ...}}
  ```

docker service

Bir docker daemon'unun üzerindeki servisleri yönet.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/service/.

- Bir docker daeomon'unun üzerindeki servisleri listele:

```
docker service ls
```

- Yeni bir servis yarat:

```
docker service create --name {{servis_ismi}} {{imge}}:{{etiket}}
```

- Boşluk ile ayrılmış bir servis listesinin detaylı bilgisini görüntüle:

```
docker service inspect {{servis_ismi|ID}}
```

- Boşluk ile ayrılmış bir servis listesinin görevlerini sırala:

```
docker service ps {{servis_ismi|ID}}
```

- Boşluk ile ayrılmış bir servis listesi için belirli bir replika miktarına yüksel:

```
docker service scale {{servis_ismi}}={{replika_miktarı}}
```

- Boşluk ile ayrılmış bir servis listesini sil:

```
docker service rm {{servis_ismi|ID}}
```

docker start

Bir veya daha fazla durmuş konteyneri başlar.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/start/.

- Yardım göster:

  ```
  docker start
  ```

- Bir docker konteynerini başlat:

  ```
  docker start {{konteyner}}
  ```

- Bir konteyneri, ona `stdout` ile `stderr`'i ekleyerek ve sinyaller göndererek başlat:

  ```
  docker start --attach {{konteyner}}
  ```

- Bir veya daha fazla boşlukla ayrılarak belirtilmiş konteynerleri başlar:

  ```
  docker start {{konteyner1 konteyner2 ...}}
  ```

docker stats

Konteynerler için kaynak kullanım istatistiklerinin canlı yayınını görüntüle.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/stats/.

- Çalışan tüm konteynerlerin aynak kullanım istatistiklerinin canlı yayınını görüntüle:

  ```
  docker stats
  ```

- Boşluk ile ayrılmış bir listedeki konteynerlerin canlı yayınını görüntüle:

  ```
  docker stats {{container_ismi}}
  ```

- Konteyner'in CPU kullanım yüzdesini göstermek için sütun formatını değiştir:

  ```
  docker stats --format "{{.Name}}:\t{{.CPUPerc}}"
  ```

- Tüm (çalışan veya durmuş) konteynerler için istatistikleri görüntüle:

  ```
  docker stats --all
  ```

- İstatistikleri canlı yayınlamayı durdur ve yalnızca mevcut durumdaki istatistikleri görüntüle:

  ```
  docker stats --no-stream
  ```

docker swarm

Bir konteyner orkestrasyon aracı.

Daha fazla bilgi için: https://docs.docker.com/engine/swarm/.

- Bir bataklık dizisi oluştur:

```
docker swarm init
```

- Bir yönetici veya işçiye takılmak için token göster:

```
docker swarm join-token {{işçi|yönetici}}
```

- Diziye yeni bir düğüm ekle:

```
docker swarm join --token {{token}} {{manager_node_url:2377}}
```

- Bir işçiyi bataklıktan sil (işçi düğümünün içinde çalıştır):

```
docker swarm leave
```

- Mevcut CA sertifikasını PEM formatında görüntüle:

```
docker swarm ca
```

- Mevcut CA sertifikasını döndür ve yeni sertifikayı görüntüle:

```
docker swarm ca --rotate
```

- Düğüm sertifikaları için geçerli periyodu değiştir:

```
docker swarm update --cert-expiry {{saat}}h{{dakika}}
m{{saniye}}s
```

docker system

Docker verilerini yönet ve sistem bilgisi görüntüle.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/system/.

- Yardım göster:

```
docker system
```

- Docker disk kullanımını göster:

```
docker system df
```

- Disk kullanımı üzerine detaylı bilgi göster:

```
docker system df --verbose
```

- Kullanılmayan veriyi sil:

```
docker system prune
```

- Kullanılmayan ve geçmişte birden çok kez oluşturulan veriyi sil:

```
docker system prune --filter="until={{saat}}h{{dakika}}m"
```

- Docker deamon'dan tam-zamanlı eylemleri görüntüle:

```
docker system events
```

- Geçerli JSON satırları olarak yayınlanan konteynerleden tam-zamanlı eylemleri göster:

```
docker system events --filter 'type=container' --format
'{{json .}}'
```

- Sistem bilgisi göster:

```
docker system info
```

docker

Docker konteyner ve imgelerini yönetir.

`docker run` gibi bazı alt komutların kendi dökümantasyonu bulunmaktadır.

Daha fazla bilgi için: https://docs.docker.com/engine/reference/commandline/cli/.

- Şuan çalışan docker konteynerlerini listele:

  ```
  docker ps
  ```

- Tüm (çalışan veya duran) docker konteynerlerini listele:

  ```
  docker ps -a
  ```

- Bir imgeden özel bir isimle konteyner başlat:

  ```
  docker run --name {{konteyner_ismi}} {{imge}}
  ```

- Varolan bir konteyneri başlat veya durdur:

  ```
  docker {{start|stop}} {{konteyner_ismi}}
  ```

- Bir docker kaydından imge çek:

  ```
  docker pull {{imge}}
  ```

- Halihazırda çalışan bir konteyner içinde komut istemcisi aç:

  ```
  docker exec -it {{konteyner_ismi}} {{sh}}
  ```

- Durmuş bir konteyneri sil:

  ```
  docker rm {{konteyner_ismi}}
  ```

- Bir konteynerin kaydını çek ve takip et:

  ```
  docker logs -f {{konteyner_ismi}}
  ```

fossil-ci

Bu komut **`fossil-commit`** için bir takma addır.

Daha fazla bilgi için: https://fossil-scm.org/home/help/commit.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr fossil-commit
  ```

fossil-delete

Bu komut **`fossil rm`** için bir takma addır.

Daha fazla bilgi için: https://fossil-scm.org/home/help/delete.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr fossil rm
  ```

fossil-forget

Bu komut **`fossil rm`** için bir takma addır.

Daha fazla bilgi için: https://fossil-scm.org/home/help/forget.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr fossil rm
  ```

fossil-new

Bu komut **`fossil-init`** için bir takma addır.

Daha fazla bilgi için: https://fossil-scm.org/home/help/new.

- Asıl komutun belgelerini görüntüleyin:

```
tldr fossil-init
```

fzf

Komut satırı belirsiz bulucu.

Sk'ya benzer.

Daha fazla bilgi için: https://github.com/junegunn/fzf.

- Belirtilen dizindeki tüm dosyalarda FZF'yi başlat:

```
find {{dosya/yolu/dizin}} -type f | fzf
```

- Çalışan süreçler için FZF'yi başlat:

```
ps aux | fzf
```

- `Shift + Tab` ile birden çok dosya seç ve bir dosyaya yaz:

```
find {{dosya/yolu/dizin}} -type f | fzf --multi > {{dosya/
yolu/dosya}}
```

- fzf'yi belirli bir sorgu ile başlat:

```
fzf --query "{{sorgu}}"
```

- Core ile başlayan ve Go, RB veya PY ile biten girişlerde fzf'yi başlat:

```
fzf --query "^core go$ | rb$ | py$"
```

- PYC ile eşleşmeyen ve Travis'e tam olarak eşleşen girişlerde fzf'yi başlat:

```
fzf --query "!pyc 'travis"
```

gh-cs

Bu komut **`gh-codespace`** için bir takma addır.

Daha fazla bilgi için: https://cli.github.com/manual/gh_codespace.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr gh-codespace
  ```

git add

Değiştirilmiş dosyaları indekse ekle.

Daha fazla bilgi için: https://git-scm.com/docs/git-add.

- İndekse bir dosya ekle:

```
git add {{örnek/dosya}}
```

- Tüm (izlenen veya izlenmeyen) dosyaları ekle:

```
git add -A
```

- Yalnızca izlenen dosyaları ekle:

```
git add -u
```

- Yoksayılan dosyaları dahi ekle:

```
git add -f
```

- Dosyaların parçalarını etkileşimli olarak sahnele:

```
git add -p
```

- Belirtilen dosyaların parçalarını etkileşimli olarak sahnele:

```
git add -p {{örnek/dosya}}
```

- Bir dosyayı etkileşimli olarak sahnele:

```
git add -i
```

git am

Yama dosyalarını uygula. E-posta ile commit alırken faydalıdır.

Ayrıca yama dosyalarının üretilmesine yarayan **`git format-patch`** sayfasına bakılması önerilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-am.

- Bir yama dosyasını uygula:

  ```
  git am {{örnek/yama.patch}}
  ```

- Yama dosyası uygulama işlemini durdur:

  ```
  git am --abort
  ```

- Mümkün olacak kadar yama dosyasını uygula ve bu dosyaların uygulanamayan parçalarını bahsi geçen dosyaları reddetmek için kaydet:

  ```
  git am --reject {{örnek/yama.patch}}
  ```

git annex

Git ile dosyaları, dosyaların içeriğine bakmadan yönet.

Daha fazla bilgi için: https://git-annex.branchable.com.

- Git annex ile bir depo başlat:

  ```
  git annex init
  ```

- Bir dosya ekle:

  ```
  git annex add {{örnek/dosya_veya_dizin}}
  ```

- Bir dosya veya dizinin şu anki durumunu göster:

  ```
  git annex status {{örnek/dosya_veya_dizin}}
  ```

- Yerel bir depoyu, uzaktaki bir depo ile senkronize et:

  ```
  git annex {{uzak_bağlantı}}
  ```

- Bir dosya veya dizin al:

  ```
  git annex get {{örnek/dosya_veya_dizin}}
  ```

- Yardım görüntüle:

  ```
  git annex help
  ```

git annotate

Her satırdaki dosyanın yanında en son commit değeri ve yazarını göster.

Ayrıca **`git annotate`** yerine tercih edilen **`git blame`** sayfasına bakılması önerilir.

`git annotate`, git dışındaki sürüm kontrol sistemlerine aşina olanlar için sağlanmıştır.

Daha fazla bilgi için: https://git-scm.com/docs/git-annotate.

- Bir dosyayı, her satırında son commit değeri ve yazarı bulunacak şekilde göster:

  ```
  git annotate {{örnek/dosya}}
  ```

- Bir dosyayı, her satırında son commit değeri ve yazarının e-postası bulunacak şekilde göster:

  ```
  git annotate -e {{örnek/dosya}}
  ```

git apply

İndeks veya dosyalara yama uygula.

Daha fazla bilgi için: https://git-scm.com/docs/git-apply.

- Yamalanan dosyalarla ilgili mesajları yazdır:

```
git apply --verbose {{örnek/dosya}}
```

- Yamalanan dosyaları indekse uygula ve ekle:

```
git apply --index {{örnek/dosya}}
```

- Uzak yama dosyası uygula:

```
curl {{https://ornek.com/dosya.patch}} | git apply
```

- Çıktı için fark statistiği çıkar ve yamayı uygula:

```
git apply --stat --apply {{örnek/dosya}}
```

- Yamayı tersten uygula:

```
git apply --reverse {{örnek/dosya}}
```

- Yama sonucunu çalışan ağacı değiştirmeden indekste sakla:

```
git apply --cache {{örnek/dosya}}
```

git archive

İsimlendirilmiş bir ağaçtan dosyaların arşivini oluştur.

Daha fazla bilgi için: https://git-scm.com/docs/git-archive.

- Mevcut HEAD'deki içerik ile bir tar arşivi oluştur ve içeriği standart çıktı biçiminde göster:

```
git archive --verbose HEAD
```

- Mevcut HEAD'deki içerik ile bir zip arşivi oluştur ve içeriği standart çıktı biçiminde göster:

```
git archive --verbose --format=zip HEAD
```

- Yukarıda yazan madde ile aynı şeyi yap, ama zip arşivini belirtilen dosya olarak yaz:

```
git archive --verbose --output={{örnek/arşiv/dosyası.zip}}
HEAD
```

- Belirtilmiş bir daldaki son commitlerin içeriğinden bir tar arşivi oluştur:

```
git archive --output={{örnek/arşiv/dosyası.tar}} {{dal_ismi}}
```

- Belirtilmiş bir dizindeki içeriklerden tar arşivi oluştur:

```
git archive --output={{örnek/arşiv/dosyası.tar}} HEAD:
{{örnek/dizin}}
```

- Bir takım dosyayı belirtilmiş bir dizinin içinde arşivlemek için başlarına yol ekle:

```
git archive --output={{örnek/arşiv/dosyası.tar}} --
prefix={{başlarına/yol/eklenecek/dosyalar}}/ HEAD
```

git bisect

Bug taşıyan commit'i bulmak için ikili arama kullan.

Git otomatik olarak commit çizelgesi içinde oradan oraya atlayarak yaramaz commit'i saptar.

Daha fazla bilgi için: https://git-scm.com/docs/git-bisect.

- Buglı bilinen bir commit'i ve (genelde eski olan) iyi bir commit'i belirterek ikiye bölme işlemini başlat:

  ```
  git bisect start {{kötü_commit}} {{iyi_commit}}
  ```

- `git bisect`'in seçtiği her commit'i, mevcut soruna sebep olup olmadıklarını test ettikten sonra "bad" (kötü) veya "good" (iyi) olarak işaretle:

  ```
  git bisect {{good|bad}}
  ```

- `git bisect` sorunlu commit'i saptadıktan sonra, ikiye bölme işlemini bitir ve depoyu bahsi geçen commit'den önceki dala geçir:

  ```
  git bisect reset
  ```

- İkiye bölme işlemi sırasında bir commit'i atla:

  ```
  git bisect skip
  ```

git blame

Her satırdaki dosyanın yanında en son commit değeri ve yazarını göster.

Daha fazla bilgi için: https://git-scm.com/docs/git-blame.

- Bir dosyayı, her satırında son commit değeri ve yazarı bulunacak şekilde göster:

  ```
  git blame {{örnek/dosya}}
  ```

- Bir dosyayı, her satırında son commit değeri ve yazarının e-postası bulunacak şekilde göster:

  ```
  git blame -e {{örnek/dosya}}
  ```

git branch

Dallar ile çalışmak için kullanılan ana Git komutu.

Daha fazla bilgi için: https://git-scm.com/docs/git-branch.

- Yerel dalları göster. Mevctu dal * ile vurgulanır:

```
git branch
```

- Tüm dalları (yerel ve uzak bağlantıda olan) göster:

```
git branch -a
```

- Mevcut dalın ismini göster:

```
git branch --show-current
```

- Mevcut commit'e dayanarak yeni bir dal oluştur:

```
git branch {{dal_ismi}}
```

- Belirtilen commit'e dayanarak yeni bir dal oluştur:

```
git branch {{dal_ismi}} {{commit_değeri}}
```

- Bir dalı yeniden adlandır:

```
git branch -m {{eski_dal_ismi}} {{yeni_dal_ismi}}
```

- Yerel bir dalı sil:

```
git branch -d {{dal_ismi}}
```

- Uzaktaki bir dalı sil:

```
git push {{uzak_bağlantı}} --delete {{uzak_dal_ismi}}
```

git bugreport

Sşstem ve kullanıcıdan hata ayıklama bilgisi çeker ve olası bir Git hatasının rapor edilmesi için bu bilgiyi oluşturduğu bir metin dosyasına kaydeder.

Daha fazla bilgi için: https://git-scm.com/docs/git-bugreport.

- Mevcut dizinde yeni bir hata rapor dosyası oluştur:

  ```
  git bugreport
  ```

- Belirtilen dizinde yeni bir hata rapor dosyası oluştur:

  ```
  git bugreport --output-directory {{örnek/dizin}}
  ```

- `strftime` formatında belirtilmiş bir dosya adı ekiyle yeni bir rapor dosyası oluştur:

  ```
  git bugreport --suffix {{%m%d%y}}
  ```

git bundle

Cisim ve referansları bir arşive paketle.

Daha fazla bilgi için: https://git-scm.com/docs/git-bundle.

- Belirtilmiş bir dalın tüm cisim ve referanslarını içeren bir paket dosyası oluştur:

```
git bundle create {{örnek/dosyas.bundle}} {{dal_ismi}}
```

- Tüm dallar için bir paket dosyası oluştur:

```
git bundle create {{örnek/dosyas.bundle}} --all
```

- Mevcut daldaki en son 5 commit için bir paket dosyası oluştur:

```
git bundle create {{örnek/dosya.bundle}} -{{5}} {{HEAD}}
```

- Son 7 günü içeren bir paket dosyası oluştur:

```
git bundle create {{örnek/dosya.bundle}} --since={{7.days}}
{{HEAD}}
```

- Bir paket dosyasının geçerli olduğunu ve mevcut depoya uygulanabileceğini doğrula:

```
git bundle verify {{örnek/dosya.bundle}}
```

- Bir pakette bulunan referansları sırala:

```
git bundle unbundle {{örnek/dosya.bundle}}
```

- Belirtilen dalı paket dosyasından çıkarıp mevcut depoya koy:

```
git pull {{örnek/dosya.bundle}} {{dal_ismi}}
```

git cat-file

Git depo cisimlerine dair içerik, tür ve boyut bilgisini sağla.

Daha fazla bilgi için: https://git-scm.com/docs/git-cat-file.

- HEAD commit'inin byte bazında boyutunu öğren:

  ```
  git cat-file -s HEAD
  ```

- Belirtilen Git cisminin türünü (blob, ağaç, commit, etiket) öğren:

  ```
  git cat-file -t {{8c442dc3}}
  ```

- Git objesinin içeriğini, türüne uygun olarak hoş şekilde yansıt:

  ```
  git cat-file -p {{HEAD~2}}
  ```

git check-attr

`gitattributes` içeriği görüntüleme aracı.

Daha fazla bilgi için: https://git-scm.com/docs/git-check-attr.

- Bir dosyadaki tüm atıfları kontrol et:

  ```
  git check-attr --all {{örnek/dosya}}
  ```

- Bir dosyadaki belirtilmiş atıfın değerini kontrol et:

  ```
  git check-attr {{atıf}} {{örnek/dosya}}
  ```

- Birden fazla dosyadaki belirtilmiş atıfın değerini kontrol et:

  ```
  git check-attr --all {{örnek/dosya1}} {{örnek/dosya2}}
  ```

- Bir veya birden fazla dosyadaki belirtilmiş atıfın değerini kontrol et:

  ```
  git check-attr {{atıf}} {{örnek/dosya1}} {{örnek/dosya2}}
  ```

git check-ignore

Git yoksayma / dışlama (".gitignore") dosyalarını analiz et.

Daha fazla bilgi için: https://git-scm.com/docs/git-check-ignore.

- Bir dosya veya dizinin yoksayıldığı veya sayılmadığını kontrol et:

```
git check-ignore {{örnek/dosya_veya_dizin}}
```

- Birden fazla dosya veya dizinin yoksayıldığı veya sayılmadığını kontrol et:

```
git check-ignore {{örnek/dosya}} {{örnek/dizin}}
```

- Her bir satıra tekabül edecek şekilde `stdin`'den yolisimleri kullan:

```
git check-ignore --stdin < {{örnek/dosya_sırası}}
```

- İndeksi kontrol etme:

```
git check-ignore --no-index {{örnek/dosya_veya_dizin}}
```

- Her yol için eşleşen desene dair detayları dahil et:

```
git check-ignore --verbose {{örnek/dosya_veya_dizin}}
```

git check-mailmap

Bağlantıların kanonik isimleri ve e-posta adreslerini göster.

Daha fazla bilgi için: https://git-scm.com/docs/git-check-mailmap.

- Bir e-posta adresi ile eşleşen kanonik ismi bul:

```
git check-mailmap "<{{örnek@e-posta.com}}>"
```

git check-ref-format

Girilen referans isminin kabul edilebilir olup olmadığını kontrol eder, ve eğer kabul edilemezse sıfır olmayan bir çıktı verir.

Daha fazla bilgi için: https://git-scm.com/docs/git-check-ref-format.

- Belirtilen referans ismini biçimini kontrol et:

```
git check-ref-format {{refs/head/referans_ismi}}
```

- Son kontrol edilen dalın ismini göster:

```
git check-ref-format --branch @{-1}
```

- Bir referans ismi dosyasını normalleştir:

```
git check-ref-format --normalize {{refs/head/referans_ismi}}
```

git checkout-index

Dosyaları indeksten çalışma ağacına kopyala.

Daha fazla bilgi için: https://git-scm.com/docs/git-checkout-index.

- Son commit'den beri silinen dosyaları geri döndür:

```
git checkout-index --all
```

- Son commit'den beri silinen veya değiştirilen dosyaları geri döndür:

```
git checkout-index --all --force
```

- Son commit'den beri değiştirilen dosyaları geri döndür ancak silinenleri yoksay:

```
git checkout-index --all --force --no-create
```

- Tüm ağacın bir kopyasını belirtilen dizinde dışa aktar (sondaki eğik çizgi önemli):

```
git checkout-index --all --force --prefix={{dışa/aktarılacak/dizin/}}
```

git checkout

Bulunulan dalı değiştir veya çalışma ağaçlarını onar.

Daha fazla bilgi için: https://git-scm.com/docs/git-checkout.

- Yeni bir dal oluştur ve bu dala geç:

```
git checkout -b {{dal_ismi}}
```

- Belirtilen bir referansa (dal, uzak/dal, etiket gibi) dayanacak şekilde yeni bir dal oluştur ve bu dala geç:

```
git checkout -b {{dal_ismi}} {{referans}}
```

- Varolan yerel bir dala geç:

```
git checkout {{dal_ismi}}
```

- En son kontrol edilmiş olan dala geç:

```
git checkout -
```

- Uzak bağlantıdaki varolan bir dala geç:

```
git checkout --track {{uzak_bağlantı_adresi}}/{{dal_ismi}}
```

- Mevcut dizindeki sahnelenmemiş tüm değişiklikleri ayır (geri alma benzeri bir komut için `git reset` komutu önerilir):

```
git checkout .
```

- Sahnelenmemiş değişiklikleri belirtilen dosyaya ayır:

```
git checkout {{dosya_ismi}}
```

- Mevcut dizindeki bir dosyayı, belirtilen dalda commit edilmiş sürümü ile değiştirin:

```
git checkout {{dal_ismi}} -- {{dosya_ismi}}
```

git cherry-pick

Varolan commit'ler ile getirilen yenilikleri mevcut dala uygula.

Değişiklikleri başka bir dala aktarmak için, önce **`git checkout`** komutu kullanılmalıdır.

Daha fazla bilgi için: https://git-scm.com/docs/git-cherry-pick.

- Mevcut dala bir commit uygula:

  ```
  git cherry-pick {{commit_ismi}}
  ```

- Mevcut dala belirtilmiş aralıktaki kadar commit uygula (ayrıca `git rebase --onto` komutunun araştırılması önerilir):

  ```
  git cherry-pick {{ilk_commit}}~..{{son_commit}}
  ```

- Mevcut dala birçok (ardışık olmayan) commit uygula:

  ```
  git cherry-pick {{commit_1}} {{commit_2}}
  ```

- Bir commit'in değişikliklerini, herhangi bir yeni commit oluşturmadan çalışan dizine ekle:

  ```
  git cherry-pick --no-commit {{commit}}
  ```

git cherry

Ana depoya aktarılması gereken commit'leri bul.

Daha fazla bilgi için: https://git-scm.com/docs/git-cherry.

- Commit'leri (ve mesajlarını) ana akımda kendilerine tekabül eden commit'ler ile göster:

  ```
  git cherry -v
  ```

- Farklı bir ana akım ve konu dalı belirt:

  ```
  git cherry {{origin}} {{topic}}
  ```

- Commit'leri verilen sınırlamalar ile sınırla:

  ```
  git cherry {{origin}} {{topic}} {{base}}
  ```

git clean

Takip edilmeyen dosyaları çalışma ağacından sil.

Daha fazla bilgi için: https://git-scm.com/docs/git-clean.

- Git tarafından takip edilmeyen dosyaları sil:

  ```
  git clean
  ```

- Git tarafından takip edilmeyen dosyaları etkileşimli bir nizamda sil:

  ```
  git clean -i
  ```

- Hangi dosyaların silinmeye aday olduğunu onları silmeden göster:

  ```
  git clean --dry-run
  ```

- Git tarafından takip edilmeyen dosyaları zorla zil:

  ```
  git clean -f
  ```

- Git tarafından takip edilmeyen dizinleri zorla zil:

  ```
  git clean -fd
  ```

- `.gitignore` ve `.git/info/exclude`'deki yoksayılan dosyalar dahiş olmak üzere takip edilmeyen dosyaları sil:

  ```
  git clean -x
  ```

git clone

Varolan bir dizini klonla.

Daha fazla bilgi için: https://git-scm.com/docs/git-clone.

- Varolan bir depoyu klonla:

```
git clone {{uzak_bağlantıdaki_depo}}
```

- Varolan bir depoyu velirtilen dizine klonla:

```
git clone {{uzak_bağlantıdaki_depo}} {{örnek/dizin}}
```

- Varolan bir depo ve onun alt modüllerini klonla:

```
git clone --recursive {{uzak_bağlantıdaki_depo}}
```

- Yerel bir depoyu klonla:

```
git clone -l {{örnek/yerel/depo}}
```

- Sessizce klonla:

```
git clone -q {{uzak_bağlantıdaki_depo}}
```

- Yalnızca en yeni 10 commit'i çekerek varolan bir depoyu klonla (zaman tasarrufu açısından yararlıdır):

```
git clone --depth {{10}} {{uzak_bağlantıdaki_depo}}
```

- Yalnızca belirtilen bir dalı çekerek varolan bir depoyu klonla:

```
git clone --branch {{isim}} --single-branch
{{uzak_bağlantıdaki_depo}}
```

git column

Kolonlarda veri görüntüle.

Daha fazla bilgi için: https://git-scm.com/docs/git-column.

- Standart çıktıyı çoklu kolonlar olarak biçimlendir:

  ```
  ls | git column --mode={{column}}
  ```

- Standart çıktıyı maksimum `100` birim sütun genişliğinde biçimlendir:

  ```
  ls | git column --mode=column --width={{100}}
  ```

- Standart çıktıyı maksimum `30` birimlik boşluğa sahip situnlar olacak şekilde biçimlendir:

  ```
  ls | git column --mode=column --padding={{30}}
  ```

git commit-graph

Git commit-graph dosyalarını yaz ve doğrula.

Daha fazla bilgi için: https://git-scm.com/docs/git-commit-graph.

- Dizinin yerel `.git` dizinindeki paketlenmiş commit'ler için bir commit-grafik dosyası yaz:

  ```
  git commit-graph write
  ```

- Erişilebilen tüm commitleri içeren bir commit-grafik dosyası yaz:

  ```
  git show-ref --hash | git commit-graph write --stdin-commits
  ```

- `HEAD`'den erişilebilenlerin yanında mevcut commit-grafik dosyasındaki tüm commit'leri içeren bir commit-grafik dosyası oluştur:

  ```
  git rev-parse {{HEAD}} | git commit-graph write --stdin-
  commits --append
  ```

git commit-tree

Commit cisimleri oluşturmaya yarayan düşük seviyeli araç.

Ayrıca **`git commit`** sayfasına bakılması önerilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-commit-tree.

- Belirtilen mesaj ile bir commit cismi oluştur:

```
git commit-tree {{ağaç}} -m "{{mesaj}}"
```

- Bir dosyadan mesaj okuyan bir commit cismi oluştur (`stdin` için `-` ekini kullan):

```
git commit-tree {{ağaç}} -F {{örnek/dosya}}
```

- GPG anahtarıyla imzalanmış bir commit cismi oluştur:

```
git commit-tree {{ağaç}} -m "{{mesaj}}" --gpg-sign
```

- Belirtilen ana commit cismi ile bir commit cismi oluştur:

```
git commit-tree {{ağaç}} -m "{{mesaj}}" -p {{ana_commit_sha}}
```

git commit

Depoya dosya commit'le.

Daha fazla bilgi için: https://git-scm.com/docs/git-commit.

- Sahnelenmiş dosyaları belirtilen mesaj ile commit'le:

  ```
  git commit -m {{mesaj}}
  ```

- Değişiklikleri otomatik olarak sahnele ve mesaj ile commit'le:

  ```
  git commit -a -m {{mesaj}}
  ```

- Değerini değiştirecek şekilde son commit'i yeni sahnelenmiş değişiklikleri ekleyerek güncelle:

  ```
  git commit --amend
  ```

- Yalnızca belirtilmiş (halihazırda sahnelenmiş) dosyaları commit'le:

  ```
  git commit {{örnek/dosya1}} {{örnek/dosya2}}
  ```

git config

Git depoları için yazılan kişisel konfigürasyon seçeneklerini yönet.

Bu konfigürasyonlar lokal (mevcut depo için) veya evrensel (mevcut kullanıcı için) olabilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-config.

- Yalnızca (mevcut depodaki `.git/config`'de saklanan) yerel konfigürasyon kayıtlarını sırala:

  ```
  git config --list --local
  ```

- Yalnızca (bilgisayardaki `~/.gitconfig`'de saklanan) evrensel konfigürasyon kayıtlarını sırala:

  ```
  git config --list --global
  ```

- Yerel veya evrensel olarak tanımlanan tüm konfigürasyon kayıtlarını sırala:

  ```
  git config --list
  ```

- Belirtilen bir konfigürasyon kaydının değerini öğren:

  ```
  git config alias.unstage
  ```

- Belirtilen bir konfigürasyon kaydının evrensel değerini belirle:

  ```
  git config --global alias.unstage "reset HEAD --"
  ```

- Evrensel bir konfigürasyon kaydını varsayılan değerine geri al:

  ```
  git config --global --unset alias.unstage
  ```

- Mevcut depodaki Git konfigürasyonunu varsayılan metin düzenleyici ile düzenle:

  ```
  git config --edit
  ```

- Evrensel Git konfigürasyonunu varsayılan metin düzenleyici ile düzenle:

  ```
  git config --global --edit
  ```

git count-objects

Paketlenmemiş cisimlerin miktarını ve disk tüketimlerini hesapla.

Daha fazla bilgi için: https://git-scm.com/docs/git-count-objects.

- Tüm cisimleri say ve toplam disk tüketimlerini görüntüle:

  ```
  git count-objects
  ```

- Tüm cisimleri say ve toplam disk tüketimlerini insanların okuyabileceği biçimde görüntüle:

  ```
  git count-objects --human-readable
  ```

- Daha fazla ayrıntı görüntüle:

  ```
  git count-objects --verbose
  ```

- Daha fazla ayrıntıyı insanların okuyabileceği biçimde görüntüle:

  ```
  git count-objects --human-readable --verbose
  ```

git credential

Kullanıcı kimlik bilgilerini kurtar ve sakla.

Daha fazla bilgi için: https://git-scm.com/docs/git-credential.

- Kimlik bilgilerini, kullanıcı ismi ve parolayı konfigürasyon dosyası aracılığıyla kurtararak göster:

```
echo "{{url=http://örnek.com}}" | git credential fill
```

- Kimlik bilgilerini sonra kullanma amacıyla saklamak için bütün yapılandırılmış kimlik yardımcılarına gönder:

```
echo "{{url=http://örnek.com}}" | git credential approve
```

- Belirtilen kimlik bilgisini bütün yapılandırılmış kimlik yardımcılarından temizle:

```
echo "{{url=http://örnek.com}}" | git credential reject
```

git describe

Bir nesneye varolan referans üzerinden insanlar tarafından okunabilecek biçimde olan bir isim ver.

Daha fazla bilgi için: https://git-scm.com/docs/git-describe.

- Mevcut commit için (en son eklenmiş etiket, ilave commit'lerin sayısı ve kısaltılmış commit değerini içeren) özel bir isim oluştur:

  ```
  git describe
  ```

- Kısaltılmış commit değeri için 4 haneli bir isim oluştur:

  ```
  git describe --abbrev={{4}}
  ```

- Etiket referans yolu ile bir isim oluştur:

  ```
  git describe --all
  ```

- Bir Git etiketini açıkla:

  ```
  git describe {{v1.0.0}}
  ```

- Belirtilen daldaki son commit için bir isim oluştur:

  ```
  git describe {{dal_ismi}}
  ```

git diff

İzlenen dosyalara değişiklikleri göster.

Daha fazla bilgi için: https://git-scm.com/docs/git-diff.

- Sahnelenmemiş, commit'lenmemiş değişiklikleri göster:

```
git diff
```

- Sahnelenmiş olanlar da dahil olmak üzere tüm commit'lenmemiş değişiklikleri göster:

```
git diff HEAD
```

- Yalnızca sahnelenmiş (eklenmiş ancak commit'lenmemiş) değişiklikleri göster:

```
git diff --staged
```

- Belirtilen bir tarihten itibaren yapılmış tüm commit'lerdeki değişiklikleri göster:

```
git diff 'HEAD@{3 months|weeks|days|hours|seconds ago}'
```

- Belirtilen bir commit'ten itibaren yalnızca üzerinde değişiklik yapılmış dosyaların ismini göster:

```
git diff --name-only {{commit}}
```

- Belirtilen bir commit'ten itibaren yapılmış dosya oluşturma, yeniden adlandırma ve mod değişim işlemlerini göster:

```
git diff --summary {{commit}}
```

- Tek bir dosyayı iki dal veya commit arasında karşılaştır:

```
git diff {{dal_1}}..{{dal_2}} [--] {{örnek/dosya}}
```

- Mevcut daldaki farklı dosyaları başka bir daldakilerle karşılaştır:

```
git diff {{dal}}:{{örnek/dosya2}} {{örnek/dosya}}
```

git difftool

Harici diff araçları kullanarak dosya değişimlerini göster. **`git diff`** ile aynı ayar ve argümanları destekler.

Daha fazla bilgi için: https://git-scm.com/docs/git-difftool.

- Müsait diff araçlarını göster:

  ```
  git difftool --tool-help
  ```

- Varsayılan diff aracını birleştirmeye ayarla:

  ```
  git config --global diff.tool "{{meld}}"
  ```

- Varsayılan diff aracını sahnelenmiş değişiklikleri göstermek için kullan:

  ```
  git difftool --staged
  ```

- Verilen commit'den itibaren yapılmış değişiklikleri göstermek için (opendiff) kullan:

  ```
  git difftool --tool={{opendiff}} {{commit}}
  ```

git fetch

Uzak bir depodaki cisim ve referansları indir.

Daha fazla bilgi için: https://git-scm.com/docs/git-fetch.

- (Eğer belirtildiyse) Uzaktaki varsayılan ana akım depodan son değişiklikleri çek:

  ```
  git fetch
  ```

- Belirtilen uzak ana akım depodan yeni dalları çek:

  ```
  git fetch {{uzak_bağlantı}}
  ```

- Uzaktaki tüm ana akım depolardaki son değişiklikleri çek:

  ```
  git fetch --all
  ```

- Uzaktaki ana akım depodan etiketleri dahi çek:

  ```
  git fetch --tags
  ```

- Ana akım depodan silinmiş uzak dallara giden yerel referansları sil:

  ```
  git fetch --prune
  ```

git flow

Üst seviye depo işlemleri için Git uzantı koleksiyonu.

Daha fazla bilgi için: https://github.com/nvie/gitflow.

- Varolan bir git deposu içinde başlat:

  ```
  git flow init
  ```

- `develop` tabanlı bir özellik dalı üzerinde geliştirmeye başla:

  ```
  git flow feature start {{özellik}}
  ```

- Özellik dalı üzerinde geliştirmeyi bitir, `develop` dalı ile birleştir ve dalı sil:

  ```
  git flow feature finish {{özellik}}
  ```

- Özelliği uzak sunucuya yayınla:

  ```
  git flow feature publish {{özellik}}
  ```

- Başka bir kullanıcı tarafından yayınlanan özelliği al:

  ```
  git flow feature pull origin {{özellik}}
  ```

git format-patch

`.patch` dosyaları oluştur. Commit'leri e-posta olarak gönderirken işe yarar.

Ayrıca benzer bir komut olan **`git am`** sayfasına bakılması önerilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-format-patch.

- Gönderilmemiş tüm commit'ler için otomatik olarak adlandırılan bir `.patch` dosyası oluştur:

  ```
  git format-patch {{origin}}
  ```

- `stdout`'daki belirtilen 2 revizyon arasındaki tüm commit'ler için bir `.patch` dosyası oluştur:

  ```
  git format-patch {{revizyon_1}}..{{revizyon_2}}
  ```

- Son 3 commit için bit `.patch` dosyası oluştur:

  ```
  git format-patch -{{3}}
  ```

git fsck

Git depo indeksindeki düğümlerin geçerliliğini ve bağlantılarını doğrula.

Düzenleme yapılması tavsiye edilmez. Geçersiz düğümleri çözmek için **`git gc`** komutu önerilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-fsck.

- Mevcut depoyu kontrol et:

  ```
  git fsck
  ```

- Bulunan tüm etiketleri sırala:

  ```
  git fsck --tags
  ```

- Bulunan tüm kök düğümleri sırala:

  ```
  git fsck --root
  ```

git gc

Gereksiz dosyaları silerek yerel depoyu optimize et.

Daha fazla bilgi için: https://git-scm.com/docs/git-gc.

- Depoyu optimize et:

```
git gc
```

- Agresifçe optimiize et (daha uzun sürer):

```
git gc --aggressive
```

- Gevşek objeleri kesme (varsayılan olarak keser):

```
git gc --no-prune
```

- Tüm çıktıları sessize al:

```
git gc --quiet
```

- Tam kullanım için yardım göster:

```
git gc --help
```

git-grep

Belirtilen söz dizisini bir deponun geçmişi dahil tüm dosyalarında ara.

Sıradan **`grep`** komutundaki birçok ek bu komut için de aynen geçerlidir.

Daha fazla bilgi için: https://git-scm.com/docs/git-grep.

- İzlenen dosyalarda belirtilen söz dizisini ara:

```
git grep {{söz_dizisi}}
```

- İzlenen dosyalarda belirtilen desene uygun, belirtilen söz dizisini ara:

```
git grep {{söz_dizisi}} -- {{file_glob_pattern}}
```

- Alt modüller de dahil olmak üzere izlenen dosyalarda belirtilen söz dizisini ara:

```
git grep --recurse-submodules {{söz_dizisi}}
```

- Belirtilen depo geçmişinde belirtilen söz dizisini ara:

```
git grep {{söz_dizisi}} {{HEAD~2}}
```

- Belirtilen söz dizisini tüm dallarda ara:

```
git grep {{söz_dizisi}} $(git rev-list --all)
```

git help

Git hakkında yardım bilgisi görüntüleme aracı.

Daha fazla bilgi için: https://git-scm.com/docs/git-help.

- Belirtilmiş Git alt komutu hakkında yardım bilgisi göster:

  ```
  git help {{komut_ismi}}
  ```

- Belirtilmiş Git alt komutu hakkında yardım bilgisini bir ağ tarayıcısında göster:

  ```
  git help --web {{komut_ismi}}
  ```

- Tüm mevcut Git alt komutlarını sırala:

  ```
  git help --all
  ```

- Mevcut rehberleri sırala:

  ```
  git help --guide
  ```

- Mümkün olan tüm konfigürasyon değişkenlerini sırala:

  ```
  git help --config
  ```

git ignore

Önceden belirlenmiş şablonlarla .gitignore dosyaları oluştur.

Daha fazla bilgi için: https://github.com/tj/git-extras/blob/master/Commands.md#git-ignore.

- Mevzut şablonları sırala:

```
git ignore list
```

- Bir .gitignore şablonu oluştur:

```
git ignore {{nesne_a,nesne_b,nesne_n}}
```

git-imerge

İki git dalı arasında aşamalı olarak birleştirme veya taban değiştirme işlemlerini uygula.

Dallar arasındaki uyuşmazlıklar özel commitler ile bölüşülerek uyuşmazlıkları çözmek kolaylaştırılır.

Daha fazla bilgi için: https://github.com/mhagger/git-imerge.

- imerge bazlı taban değiştirme işlemini başlat (işlemden önce tabanı değiştirilmek istenen dalı kontrol et):

  ```
  git imerge rebase {{yerine_geçilecek_dal}}
  ```

- imerge bazlı birleştirme işlemini başlat (işlemden önce birleştirilmek istenen dalı kontrol et):

  ```
  git imerge merge {{birleştirilecek_dal}}
  ```

- Devam eden birleştirme ve taban değiştirme işlemlerinin ASCII diagramını göster:

  ```
  git imerge diagram
  ```

- Uyuşmazlıkları çözdükten sonra imerge işlemine devam et (önce `git add` komutu ile uyuşmayan dosyaları ekle):

  ```
  git imerge continue --no-edit
  ```

- Tüm uyuşmazlıklar çözüldükten sonra imerge işlemini sonlandır:

  ```
  git imerge finish
  ```

- imerge işlemini sonlandır ve belirtilen eski bir dala geri dön:

  ```
  git-imerge remove && git checkout {{eski_dal}}
  ```

git init

Yeni bir yerel Git deposu başlat.

Daha fazla bilgi için: https://git-scm.com/docs/git-init.

- Yeni bir yerel depo başlat:

  ```
  git init
  ```

- Bir depoyu nesne verileri için SHA256 formatı ile başlat (Git versiyonu 2.29 veya üstü olmalıdır):

  ```
  git init --object-format={{sha256}}
  ```

- Yalın bir depo başlat:

  ```
  git init --bare
  ```

git instaweb

gitweb sunucusu başlatmak için yardımcı araç.

Daha fazla bilgi için: https://git-scm.com/docs/git-instaweb.

- Mevcut Git deposu için bir gitweb sunucusu başlat:

```
git instaweb --start
```

- Yalnızca yerel ağda başlat:

```
git instaweb --start --local
```

- Belirtilmiş bir port'da başlat:

```
git instaweb --start --port {{1234}}
```

- Belirtilmiş bir http daemon'u kullan:

```
git instaweb --start --httpd {{lighttpd|apache2|mongoose|
plackup|webrick}}
```

- Ayrıca bir ağ tarayıcısını otomatik olarak başlat:

```
git instaweb --start --browser
```

- Çalışan mevcut gitweb sunucusunu durdur:

```
git instaweb --stop
```

- Çalışan mevcut gitweb sunucusunu yeniden başlat:

```
git instaweb --restart
```

git lfs

Git depolarındaki büyük dosyalarla çalış.

Daha fazla bilgi için: https://git-lfs.github.com.

- Git LFS'i başlat:

```
git lfs install
```

- Belirtilen topağa uygun dosyaları izle:

```
git lfs track '{{*.bin}}'
```

- Git LFS uç nokta URL'sini değiştir (LFS sunucusunun Git sunucusundan ayrı olması durumunda işlevseldir):

```
git config -f .lfsconfig lfs.url {{lfs_uç_nokta_url'si}}
```

- İzlenen kalıpları sırala:

```
git lfs track
```

- Commit'lenmiş izlenen dosyaları sırala:

```
git lfs ls-files
```

- Tğm Git LFS nesnelerini uzak sunucuya gönder (hatayla karşılaşma durumunda faydalıdır):

```
git lfs push --all {{uzak_depo_adresi}} {{dal_ismi}}
```

- Tüm Git LFS nesnelerini çek:

```
git lfs fetch
```

- Tüm Git LFS nesnelerini kontrol et:

```
git lfs checkout
```

git log

Commit geçmişini göster.

Daha fazla bilgi için: https://git-scm.com/docs/git-log.

- Mevcut olandan başlayarak mevcut çalışma ortamındaki git deposunun commit silsilesini ters kronolojik düzende göster:

```
git log
```

- Belirtilen dosya veya dizinin tarihini farklılıklarla beraber göster:

```
git log -p {{dosya/veya/dizin/konumu}}
```

- Her bir commit'de hangi dosya(lar)ın değiştiğinin önizlemesini göster:

```
git log --stat
```

- Mevcut daldaki commit'lerin mesajlarının ilk satırını içeren bir çizelge göster:

```
git log --oneline --graph
```

- Bir depodaki commit, etiket ve dalların tamamını içeren bir çizelge göster:

```
git log --oneline --decorate --all --graph
```

- Mesajları yalnızca belirtilen ifadeleri içeren commit'leri göster (büyük-küçük harfe duyarsız):

```
git log -i --grep {{aranan_ifade}}
```

- Belirtilmiş yazardan gelen, belirtilen sayıda commit göster:

```
git log -n {{sayı}} --author={{yazar}}
```

- İki tarih arasında yapılmış commit'leri göster:

```
git log --before={{tarih}} --after={{tarih}}
```

git ls-files

İndex ve mevcut ağaçtaki dosyalar hakkında bilgi göster.

Daha fazla bilgi için: https://git-scm.com/docs/git-ls-files.

- Silinen dosyaları göster:

```
git ls-files --deleted
```

- Düzenlenen ve silinen dosyaları göster:

```
git ls-files --modified
```

- Yoksayılmış ve izlenmeyen dosyaları göster:

```
git ls-files --others
```

git ls-remote

Çevrimiçi depolardaki isim ve URL bazlı referansları sıralamaya yarayan Git komutu.

İsim veya URL girilmemişse, varsayılan dal veya çevrimiçi dalın kökeni kullanılır.

Daha fazla bilgi için: https://git-scm.com/docs/git-ls-remote.

- Varsayılan çevrimiçi depodaki tüm referansları göster:

```
git ls-remote
```

- Varsayılan çevrimiçi depodaki yalnızca baş referanslarını göster:

```
git ls-remote --heads
```

- Varsayılan çevrimiçi depodaki yalnızca etiket referanslarını göster:

```
git ls-remote --tags
```

- Girilen isim veya URL'de bulunan çevrimiçi depodaki tüm referansları göster:

```
git ls-remote {{depo_adresi}}
```

- Bir çevrimiçi depodaki referansları belirtilen desene göre göster:

```
git ls-remote {{depo_ismi}} "{{desen}}"
```

git ls-tree

Bir ağaç nesnesinin içeriklerini sırala.

Daha fazla bilgi için: https://git-scm.com/docs/git-ls-tree.

- Bir daldaki ağacın içeriklerini sırala:

  ```
  git ls-tree {{dal_name}}
  ```

- Bir commit üstündeki ağacın içeriklerini alt ağaçlara ayırarak sırala:

  ```
  git ls-tree -r {{commit_değeri}}
  ```

- Bir commit üstündeki ağacın yalnızca dosya isimlerini göster:

  ```
  git ls-tree --name-only {{commit_değeri}}
  ```

git merge

Dalları birleştir.

Daha fazla bilgi için: https://git-scm.com/docs/git-merge.

- Mevcut dal ile belirtilen dalı birleştir:

  ```
  git merge {{dal_ismi}}
  ```

- Birleştirme mesajını düzenle:

  ```
  git merge -e {{dal_ismi}}
  ```

- Bir dalı birleştir ve birleştirme commit'i oluştur:

  ```
  git merge --no-ff {{dal_ismi}}
  ```

- Karışıklık durumlarına karşı birleştirme işlemini durdur:

  ```
  git merge --abort
  ```

git mergetool

Birleştirme sırasında yaşanan karışıklıkları çözmek için karışıklık çözücü araçları çalıştırır.

Daha fazla bilgi için: https://git-scm.com/docs/git-mergetool.

- Karışıklıkları çözmek için varsayılan birleştirme aracını başlat:

```
git mergetool
```

- Kullanılabilir birleştirme araçlarını sırala:

```
git mergetool --tool-help
```

- Belirtilen birleştirme aracını başlat:

```
git mergetool --tool {{araç_ismi}}
```

- Her birleştirme aracı çağrılışında harekete geçme:

```
git mergetool --no-prompt
```

- Özellikle grafiksel (GUI) birleştirme aracını kullan (merge.guitool değişkenine göz at):

```
git mergetool --gui
```

- Özellikle normal birleştirme aracını kullan (merge.guitool değişkenine göz at):

```
git mergetool --no-gui
```

git mv

Dosyaları taşı veya yeniden adlandır ve Git indeksini güncelle.

Daha fazla bilgi için: https://git-scm.com/docs/git-mv.

- Depo içindeki dosyayı taşı ve bu hareketi sonraki commit'e ekle:

```
git mv {{dosya/konumu}} {{yeni/dosya/konumu}}
```

- Dosyayı yeniden adlandır ve yeniden adlandırma hareketini sonraki commit'e ekle:

```
git mv {{dosya_ismi}} {{yeni_dosya_ismi}}
```

- Eğer varsa belirtilen hedefteki dosyanın üstüne yaz:

```
git mv --force {{dosya}} {{hedef}}
```

git notes

Nesne notları ekle veya incele.

Daha fazla bilgi için: https://git-scm.com/docs/git-notes.

- Tüm notları ve bağlı oldukları nesneleri sırala:

```
git notes list
```

- Belirtilen nesneye bağlanan tüm notları sırala (varsayılan HEAD'dedir):

```
git notes list [{{nesne}}]
```

- Belirtilen nesneye bağlanan tüm notları göster (varsayılan HEAD'dedir):

```
git notes show [{{nesne}}]
```

- Belirtilen nesneye bir not ekle (varsayılan metin editörü açılır):

```
git notes append {{nesne}}
```

- Mesajı belirterek belirtilen nesneye bir not ekle:

```
git notes append --message="{{messaj_yazısı}}"
```

- Varolan bir notu düzenle (varsayılan HEAD'dedir):

```
git notes edit [{{nesne}}]
```

- Bir notu bir nesneden öbürüne kopyala:

```
git notes copy {{kaynak_nesne}} {{hedef_nesne}}
```

- Belirtilen nesneye eklenen tüm notları sil:

```
git notes remove {{nesne}}
```

git pr

Github çekme isteklerini (pr) yerelde kontrol et.

Daha fazla bilgi için: https://github.com/tj/git-extras/blob/master/Commands.md#git-pr.

- Belirtilen çekme isteğini kontrol et:

```
git pr {{pr_numarası}}
```

- Belirtilen dış bağlantıdan gelen bir çekme isteğini kontrol et:

```
git pr {{pr_numarası}} {{dış_bağlantı}}
```

- Belirtilen URL'den gelen çekme isteğini kontrol et:

```
git pr {{url}}
```

- Eski çekme isteği dallarını temizle:

```
git pr clean
```

git prune

Nesne veritabanından erişilemeyen tüm nesneleri budamaya yarayan git komutu.

Bu komut genelde doğrudan kullanılmasa da Git gc tarafından bir iç komut olarak kullanılmaktadır.

Daha fazla bilgi için: https://git-scm.com/docs/git-prune.

- Git prune tarafından silinebilecek nesneleri onları silmeden raporla:

  ```
  git prune --dry-run
  ```

- Erişilemeyen nesneleri buda ve `stdout`'a budanan şeyleri görüntüle:

  ```
  git prune --verbose
  ```

- Erişilemeyen nesneleri budarken ilerlemeyi göster:

  ```
  git prune --progress
  ```

git pull

Uzak bir depodan dal getir ve yerel depo ile birleştir.

Daha fazla bilgi için: https://git-scm.com/docs/git-pull.

- Varsayılan uzak depodan değişiklikleri indir ve birleştir:

  ```
  git pull
  ```

- Varsayılan uzak depodan değişiklikleri indir ve ileri sarmayı kullan:

  ```
  git pull --rebase
  ```

- Belirtilen uzak depodan ve daldan değişiklikleri indir, ve sonra onları HEAD ile birleştir:

  ```
  git pull {{uzak_bağlantı}} {{dal}}
  ```

git push

Commit'leri uzak depoya yolla.

Daha fazla bilgi için: https://git-scm.com/docs/git-push.

- Mevcut daldaki yerel değişiklikleri onun uzak eşine gönder:

```
git push
```

- Belirtilen daldaki yerel değişiklikleri onun uzak eşine gönder:

```
git push {{uzak_bağlantı}} {{yerel_dal}}
```

- Mevcut dalı bir uzak dal ismi ayarlayarak uzak depoda yayınla:

```
git push {{uzak_bağlantı}} -u {{uzak_dal}}
```

- Yerel dallardaki tüm değişiklikleri onların belirtilen uzak depodaki uzak eşlerine gönder:

```
git push --all {{uzak_bağlantı}}
```

- Uzak depodaki bir dalı sil:

```
git push {{uzak_bağlantı}} --delete {{uzak_dal}}
```

- Yerel eşi olmayan uzak dalları sil:

```
git push --prune {{uzak_bağlantı}}
```

- Daha yzak depoda olmayan etiketleri yayınla:

```
git push --tags
```

git rebase

Bir daldan başka bir dalın üstüne commit'leri tekrar temeller.

Sıklıkla bir dalı commit'leriyle beraber başka bir tabana "taşımak" için kullanılır.

Daha fazla bilgi için: https://git-scm.com/docs/git-rebase.

- Mevcut dalı belirtilen öbür dal üzerine temelle:

```
git rebase {{yeni_taban_dal}}
```

- Commit'lerin sıralanması, çıkartılması, birleştirilmesi veya modifiye edilmesine izin vermek için tekrar temellemeyi etkileşimli olacak şekilde başlat:

```
git rebase -i {{hedef_taban_dalı_veya_commit_değeri}}
```

- Bir birleştirme hatası tarafından durdurulan tekrar temelleme işlemini çekişen dosyaları düzenledikten sonra devam ettir:

```
git rebase --continue
```

- Birleştirme çatışmasından ötürü durdurulan tekrar temelleme işlemini çekişen commit'leri atlayarak devam ettir:

```
git rebase --skip
```

- Devam eden tekrar temelleme işlemini iptal et (örneğin birleştirmede çatışma yaşandığında):

```
git rebase --abort
```

- Mevcut dalın bir parçasını belirtilen eski tabandan yeni tabana taşı:

```
git rebase --onto {{yeni_taban}} {{eski_taban}}
```

- Son 3 commit'i etkileşimli olmayacak şekilde yeniden uygula:

```
git rebase -i {{HEAD~5}}
```

- Herhangi bir çatışmayı çalışan dal sürümünü kurtarmak üzere otomatik olarak çöz (`theirs` argümanı burada ters anlama sahip):

```
git rebase -X theirs {{dal_ismi}}
```

git reflog

HEAD, dal ve etiketler gibi yerel referansların geçirdiği değişimlerin kaydını göster.

Daha fazla bilgi için: https://git-scm.com/docs/git-reflog.

- HEAD için referans kaydını göster:

```
git reflog
```

- Belirtilen dal için referans kaydını göster:

```
git reflog {{dal_ismi}}
```

- Referans kaydında sadece son 5 değişimi göster:

```
git reflog -n {{5}}
```

git remote

İzlenen depolar dizisini (uzak bağlantıları) yönet.

Daha fazla bilgi için: https://git-scm.com/docs/git-remote.

- Varolan uzak bağlantıların isim ve URL'leriyle bir listesini göster:

```
git remote -v
```

- Uzak bağlantı ile ilgili bilgi göster:

```
git remote show {{uzak_bağlantı_ismi}}
```

- Uzak bağlantı ekle:

```
git remote add {{uzak_bağlantı_ismi}}
{{uzak_bağlantı_url'si}}
```

- Uzak bağlantının URL'sini değiştir:

```
git remote set-url {{uzak_bağlantı_ismi}} {{yeni_url}}
```

- Uzak bağlantıyı sil:

```
git remote remove {{uzak_bağlantı_ismi}}
```

- Uzak bağlantıyı yeniden adlandır:

```
git remote rename {{eski_isim}} {{yeni_isim}}
```

git repack

Bir Git deposundaki paketlenmemiş nesneleri paketle.

Daha fazla bilgi için: https://git-scm.com/docs/git-repack.

- Mevcut dizindeki paketlenmemiş nesneleri paketle:

  ```
  git repack
  ```

- Paketlemeden sonra gereksiz nesneleri sil:

  ```
  git repack -d
  ```

git replace

Nesnelerin yerini değiştirmek için referans oluştur, sırala ve sil.

Daha fazla bilgi için: https://git-scm.com/docs/git-replace.

- Öbür commit'lere dokunmadan bir commit'in başka bir commit ile yerini değiştir:

  ```
  git replace {{nesne}} {{yer_değiştirme}}
  ```

- Belirtilen nesnede varolan yer değiştirme referanslarını sil:

  ```
  git replace --delete {{nesne}}
  ```

- Bir nesnenin içeriğini etkileşimli olarak düzenle:

  ```
  git replace --edit {{nesne}}
  ```

git request-pull

Ana projeye yerelde yapılan değişiklikleri kendi ağacına çekmesini sormak için izin hazırla.

Daha fazla bilgi için: https://git-scm.com/docs/git-request-pull.

- v1.1 sürümü ve belirtilen dal arasındaki değişiklikleri özetleyen bir izin üret:

  ```
  git request-pull {{v1.1}} {{https://ornek.com/proje}}
  {{dal_ismi}}
  ```

- `foo` dalındaki v0.1 sürümü ile yereldeki `bar` dalları arasındaki değişiklikleri özetleyen bir izin üret:

  ```
  git request-pull {{v0.1}} {{https://ornek.com/proje}}
  {{foo:bar}}
  ```

git reset

Mevcut Git HEAD'ini belirtilen duruma sıfırlayarak commit'leri veya değişiklikleri geri al.

Eğer bir konum verildiye o konumdaki değişiklikler "geri alınır"; eğer bir commit değeri veya dal verildiyse o commit/dal "geri alınır".

Daha fazla bilgi için: https://git-scm.com/docs/git-reset.

- Her şeyi geri al:

```
git reset
```

- Belirtilen dosya(lar)ı geri al:

```
git reset {{dosya(ların)/konumu}}
```

- Bir dosyanın kısımlarını geri al:

```
git reset -p {{dosya/konumu}}
```

- Son commit'i, dosya sisteminde yapılan değişiklikleri geri almadan geri al:

```
git reset HEAD~
```

- Son iki commit'i onların indeks'e yaptığı değişiklikleri ekleyerek geri al:

```
git reset --soft HEAD~2
```

- Commit'lenmemiş değişiklikleri sahnelenip sahnelenmediklerine bakmaksızın iptal et (sadece sahnelenmemiş değişiklikleri iptal etmek için `git checkout` kullanılır):

```
git reset --hard
```

- Depoyu belirtilen commit'e o zamana kadar yapılan değişiklikleri iptal ederek sıfırla:

```
git reset --hard {{commit}}
```

git restore

Çalışan ağaç dosyalarını onar. Git sürümü 2.23+ olmalıdır.

`git checkout` ve **`git reset`** komutlarına da ayrıca bakılması tavsiye edilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-restore.

- Sahnelenmemiş bir dosyayı mevcut commit'in sürümüne kavuştur:

  ```
  git restore {{dosya/konumu}}
  ```

- Sahnelenmemiş bir dosyayı belirtilen commit'in sürümüne kavuştur:

  ```
  git restore --source {{commit}} {{dosya/konumu}}
  ```

- İzlenen dosyalardaki sahnelenmemiş tüm değişiklikleri iptal et:

  ```
  git restore :/
  ```

- Bir dosyayı sahnelenmemiş hale getir:

  ```
  git restore --staged {{dosya/konumu}}
  ```

- Tüm dosyaları sahnelenmemiş hale getir:

  ```
  git restore --staged :/
  ```

- Dosyalara yapılan sahnelenmiş veya sahnelenmemiş tüm değişiklikleri iptal et:

  ```
  git restore --worktree --staged :/
  ```

- Onarılacak dosya parçalarını etkileşimli olarak seç:

  ```
  git restore --patch
  ```

git rev-list

Değişiklikleri (commit'leri) ters kronolojik sırada sırala.

Daha fazla bilgi için: https://git-scm.com/docs/git-rev-list.

- Mevcut daldaki tüm commit'leri sırala:

```
git rev-list {{HEAD}}
```

- Belirtilen daldaki belirtilen tarihten daha yakın olan commit'leri sırala:

```
git rev-list --since={{'2019-12-01 00:00:00'}} {{dal_ismi}}
```

- Belirtilen commit'deki tüm birleştirme commit'lerini sırala:

```
git rev-list --merges {{commit}}
```

- Belirtilen etiketten itibaren olan commit sayılarını çıkar:

```
git rev-list {{tag_name}}..HEAD --count
```

git rev-parse

Belirtilen sürümler için metaveri görüntüle.

Daha fazla bilgi için: https://git-scm.com/docs/git-rev-parse.

- Bir dalın commit verisini göster:

  ```
  git rev-parse {{dal_ismi}}
  ```

- Mevcut dal ismini göster:

  ```
  git rev-parse --abbrev-ref {{HEAD}}
  ```

- Kök dizinin mutlak konumunu göster:

  ```
  git rev-parse --show-toplevel
  ```

git revert

Öncekilerin etkilerini geri alan yeni bir commit oluştur.

Daha fazla bilgi için: https://git-scm.com/docs/git-revert.

- En son commit'leri geri al:

  ```
  git revert {{@}}
  ```

- En son 5. commit'i geri al:

  ```
  git revert HEAD~{{4}}
  ```

- Birden fazla commit'i geri al:

  ```
  git revert {{dal_ismi~5..dal_ismi~2}}
  ```

- Yeni commit'ler oluşturma, yalnızca çalışan ağacı değiştir:

  ```
  git revert -n {{0c01a9..9a1743}}
  ```

git rm

Dosyaları dizin indeksinden ve yerel dosya sisteminden sil.

Daha fazla bilgi için: https://git-scm.com/docs/git-rm.

- Dosyayı dizin indeksinden ve dosya sisteminden sil:

```
git rm {{dosya}}
```

- Dizini sil:

```
git rm -r {{dizin}}
```

- Dizin indeksinden dosyayı sil lakin yerelde dosyaya dokunma:

```
git rm --cached {{dosya}}
```

git send-email

Bir yama koleksiyonunu e-posta olarak gönder.

Yamalar dosya, dizin veya sürüm listesi olarak tanımlanabilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-send-email.

- Mevcut dizindeki son commit'i gönder:

```
git send-email -1
```

- Belirtilen commit'i gönder:

```
git send-email -1 {{commit}}
```

- Mevcut dizindeki belirtilen sayı kadar (örneğin 10) commit'i gönder:

```
git send-email {{-10}}
```

- Gönderilecek yama serisi için bir giriş e-posta mesajı gönder:

```
git send-email -{{commits_sayı}} --compose
```

- Gönderilecek her bir yama için e-posta mesajını görüntüle ve düzenle:

```
git send-email -{{commits_sayı}} --annotate
```

git shortlog

'git log' çıktısını özetle.

Daha fazla bilgi için: https://git-scm.com/docs/git-shortlog.

- Yapılan tüm commit'lerin yazar ismiyle alfabetik olarak guruplanmış özetini göster:

  ```
  git shortlog
  ```

- Yapılan tüm commit'lerin en çok commit yapan yazar ismi en üstte olacak şekilde özetini göster:

  ```
  git shortlog -n
  ```

- Yapılan tüm commit'lerin yazar bilgilerini (isim ve e-posta) gösterecek şekilde özetini göster:

  ```
  git shortlog -c
  ```

- En son yapılan 5 commit'in özetini göster (sürüm aralığı belirt):

  ```
  git shortlog HEAD~{{5}}..HEAD
  ```

- Mevcut daldaki tüm kullanıcıları, e-postalarını ve yaptıkları commit sayısını göster:

  ```
  git shortlog -sne
  ```

- Tüm dallardaki tüm kullanıcıları, e-postalarını ve yaptıkları commit sayısını göster:

  ```
  git shortlog -sne --all
  ```

git show-branch

Dalları ve içerdikleri commit'leri göster.

Daha fazla bilgi için: https://git-scm.com/docs/git-show-branch.

- Bir daldaki son commit'lerin bir özetini göster:

```
git show-branch {{dal_ismi|referans|commit}}
```

- Çeşitli commit veya daldaki commit'lerin geçmişini karşılaştır:

```
git show-branch {{dal_ismi|referans|commit}}
```

- Tüm uzak takip dallarını karşılaştır:

```
git show-branch --remotes
```

- Hem yerel, hem de uzak takip dallarını karşılaştır:

```
git show-branch --all
```

- Tüm dallardaki son commit'leri sırala:

```
git show-branch --all --list
```

- Belirtilen dalı mevcut dal ile karşılaştır:

```
git show-branch --current {{commit|dal_ismi|referans}}
```

- Bağlı isim yerine commit ismini görüntüle:

```
git show-branch --sha1-name --current {{current|dal_ismi|
referans}}
```

- Commit'lerin ortak atasından sonraki commit'leri belirtilen sayı kadar görüntüle:

```
git show-branch --more {{5}} {{commit|dal_ismi|referans}}
{{commit|dal_ismi|referans}} {{...}}
```

git show-ref

Referans sıralamak için git komutu.

Daha fazla bilgi için: https://git-scm.com/docs/git-show-ref.

- Depodaki tüm referansları göster:

  ```
  git show-ref
  ```

- Yalnızca kafa referanslarını göster:

  ```
  git show-ref --heads
  ```

- Yalnızca etiket referanslarını göster:

  ```
  git show-ref --tags
  ```

- Belirtilen referansın varolduğunu doğrula:

  ```
  git show-ref --verify {{referans/konumu}}
  ```

git show

Çeşitli Git nesnelerini (commit'ler, etiketler vs.) görüntüle.

Daha fazla bilgi için: https://git-scm.com/docs/git-show.

- Son commit'e dair bilgi (değer, mesaj, değişimler ve öbür metaveriler) göster:

```
git show
```

- Belirtilen commit'e dair bilgi göster:

```
git show {{commit}}
```

- Belirtilen etiket ile özleşen commit'e dair bilgi göster:

```
git show {{tag}}
```

- Dalın HEAD'indeki 3. commit'e dair bilgi göster:

```
git show {{dal}}~{{3}}
```

- Commit'in mesajını diff çıktısını önleyerek tek satırda göster:

```
git show --oneline -s {{commit}}
```

- Yalnızca değiştirilen dosyalarla ilgili istatistik (eklenen/silinen karakterler) göster:

```
git show --stat {{commit}}
```

- Yalnızca eklenen, yeniden adlandırılan veya silinen dosyaların listesini göster:

```
git show --summary {{commit}}
```

- Bir dosyanın belirtilen sürümdeki (örneğin dal, etiket veya commit) içeriğini göster:

```
git show {{sürüm}}:{{dosya/konumu}}
```

git sizer

Git depo boyut metriklerini hesaplar ve problem veya rahatsızlığa sebep olabilecek boyutlarda uyarı verir.

Daha fazla bilgi için: https://github.com/github/git-sizer.

- 0'dan büyük önem içeren istatistikleri raporla:

  ```
  git sizer
  ```

- Tüm istatistikleri raporla:

  ```
  git sizer -v
  ```

- İlave seçenekleri gör:

  ```
  git sizer -h
  ```

git stage

Değiştirilmiş dosyaları indekse ekle.

Bu komut **`git add`**'in eş anlamlısıdır.

Daha fazla bilgi için: https://git-scm.com/docs/git-stage.

- İndekse bir dosya ekle:

  ```
  git stage {{örnek/dosya}}
  ```

- Tüm (izlenen veya izlenmeyen) dosyaları ekle:

  ```
  git stage -A
  ```

- Yalnızca izlenen dosyaları ekle:

  ```
  git stage -u
  ```

- Yoksayılan dosyaları dahi ekle:

  ```
  git stage -f
  ```

- Dosyaların parçalarını etkileşimli olarak sahnele:

  ```
  git stage -p
  ```

- Belirtilen dosyaların parçalarını etkileşimli olarak sahnele:

  ```
  git stage -p {{örnek/dosya}}
  ```

- Bir dosyayı etkileşimli olarak sahnele:

  ```
  git stage -i
  ```

git stash

Yerel Git düzenlemelerini geçici bir alanda sakla.

Daha fazla bilgi için: https://git-scm.com/docs/git-stash.

- Yeni (izlenmeyen) dosyalar hariç mevcut değişiklikleri sakla:

```
git stash [push -m {{keyfi_saklama_mesajı}}]
```

- Yeni (izlenmeyen) dosyalar dahil mevcut değişiklikleri sakla:

```
git stash -u
```

- Değiştirilen dosyaların parçalarını etkileşimli şekilde seçip sakla:

```
git stash -p
```

- Tüm saklananları göster (saklanan ismi, bağlı olduğu dal ve mesaj gösterilir):

```
git stash list
```

- Bir saklananı uygula (varsayılan son saklanandır ve stash@{0} olarak belirtilir):

```
git stash apply {{keyfi_saklanan_veya_commit_ismi}}
```

- Bir saklananı uygula (varsayılan stash@{0}), ve eğer uygulanması sıkıntı çıkarmıyorsa onu saklanan listesinden kaldır:

```
git stash pop {{keyfi_saklanan_ismi}}
```

- Bir saklananı bırak (varsayılan stash@{0}):

```
git stash drop {{keyfi_saklanan_ismi}}
```

- Tüm saklananları bırak:

```
git stash clear
```

git status

Bir git deposundaki dosyalara yapılan değişiklikleri göster.

Mevcut commit'e kıyasla değiştirilen, eklenen ve silinen dosyaları sıralar.

Daha fazla bilgi için: https://git-scm.com/docs/git-status.

- Daha commit'e eklenmemiş değiştirilen dosyaları göster:

  ```
  git status
  ```

- Çıktıyı özetlenmiş şekilde göster:

  ```
  git status -s
  ```

- Çıktıda izlenmeyen dosyaları gösterme:

  ```
  git status --untracked-files=no
  ```

- Çıktıyı özetlenmiş şekilde dal bilgisiyle beraber göster:

  ```
  git status -sb
  ```

git stripspace

Gereksiz boşlukları sil.

Daha fazla bilgi için: https://git-scm.com/docs/git-stripspace.

- Gereksiz boşlukları dosyadan kırp:

  ```
  cat {{örnek/dosya}} | git stripspace
  ```

- Gereksiz boşlukları ve Git yorumlarını dosyadan kırp:

  ```
  cat {{örnek/dosya}} | git stripspace --strip-comments
  ```

- Bir dosyadaki tüm satırları Git yorumlarına çevir:

  ```
  git stripspace --comment-lines < {{örnek/dosya}}
  ```

git submodule

Alt modülleri incele, güncelle ve yönet.

Daha fazla bilgi için: https://git-scm.com/docs/git-submodule.

- Deponun belirtilen alt modüllerini indir:

  ```
  git submodule update --init --recursive
  ```

- Bir Git deposunu alt modül olarak ekle:

  ```
  git submodule add {{depo_url'si}}
  ```

- Bir Git deposunu alt modül olarak belirtilen dizinde ekle:

  ```
  git submodule add {{depo_url'si}} {{dizin/konumu}}
  ```

- Tüm alt modülleri son commit'lerine güncelle:

  ```
  git submodule foreach git pull
  ```

git subtree

Proje bağımlılıklarını alt proje olarak yönetmeye yarayan bir araç.

Daha fazla bilgi için: https://manpages.debian.org/latest/git-man/git-subtree.1.en.html.

- Bir Git deposunu alt ağaç olarak ekle:

```
git subtree add --prefix={{dizin/konumu}} --squash
{{depo_url'si}} {{dal_ismi}}
```

- Alt ağaç deposunu son commit'ine güncelle:

```
git subtree pull --prefix={{dizin/konumu}} {{depo_url'si}}
{{dal_ismi}}
```

- Son alt ağaca kadar olan değişiklikleri alt ağaca commit'le:

```
git subtree merge --prefix={{dizin/konumu}}} --squash
{{depo_url'si}} {{dal_ismi}}
```

- Commit'leri bir alt ağaç deposuna yolla:

```
git subtree push --prefix={{dizin/konumu}}} {{depo_url'si}}
{{dal_ismi}}
```

- Bir alt ağacın geçmişinden yeni bir proje geçmişi dışa aktar:

```
git subtree split --prefix={{dizin/konumu}}} {{depo_url'si}}
-b {{dal_ismi}}
```

git svn

Bir alt sürüm deposu ve Git arasında çift yönlü operasyon.

Daha fazla bilgi için: https://git-scm.com/docs/git-svn.

- Bit SVN deposunu klonla:

```
git svn clone {{https://ornek.com/altsürüm_deposu}}
{{yerel_dizin}}
```

- Bir SVN deposunu belirtilen düzenleme numarasından başlayarak klonla:

```
git svn clone -r{{1234}}:HEAD {{https://svn.ornek.net/
altsürüm/depo}} {{yerel_dizin}}
```

- Uzak SVN deposundan yerel klonu güncelle:

```
git svn rebase
```

- Git HEAD'i değiştirmeden uzak SVN deposundan güncellemeleri çek:

```
git svn fetch
```

- SVN deposuna geri commit'le:

```
git svn dcommit
```

git switch

Git dalları arasında geçiş yap. Gir sürümü 2.23+ olmalıdır.

Ayrıca benzer işlev gören **`git checkout`** komutuna bakılması önerilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-switch.

- Varolan bir dala geç:

  ```
  git switch {{dal_ismi}}
  ```

- Yeni bir dal yarat ve ona geç:

  ```
  git switch --create {{dal_ismi}}
  ```

- Varolan commit üzerine yeni bir dal yarat ve ona geç:

  ```
  git switch --create {{dal_ismi}} {{commit}}
  ```

- Önceki dala geç:

  ```
  git switch -
  ```

- Bir dala geç ve tüm alt modülleri uyum için güncelle:

  ```
  git switch --recurse-submodules {{dal_ismi}}
  ```

- Bir dala geç ve mevcut dal ile commit'lenmeyen değişiklikleri bu dal ile birleştir:

  ```
  git switch --merge {{dal_ismi}}
  ```

git tag

Etiketleri oluştur, sırala, sil veya doğrula.

Bir etiket, belirtilmiş bir commit'e bağlı statik bir referanstır.

Daha fazla bilgi için: https://git-scm.com/docs/git-tag.

- Tüm etiketleri sırala:

```
git tag
```

- Belirtilen isim ile mevcut commit'e bağlı bir etiket yarat:

```
git tag {{etiket_ismi}}
```

- Belirtilen isim ile belirtilen commit'e bağlı bir etiket yarat:

```
git tag {{etiket_ismi}} {{commit}}
```

- Belirtilen mesaja sahip açıklamalı bir etiket yarat:

```
git tag {{etiket_ismi}} -m {{etiket_mesajı}}
```

- Belirtilen isimdeki etiketi sil:

```
git tag -d {{etiket_ismi}}
```

- Ana projeden güncellenmiş etiketleri al:

```
git fetch --tags
```

- Belirtilen commit'i içeren/içermiş tüm etiketleri sırala:

```
git tag --contains {{commit}}
```

git update-index

İndeksi manipüle etmeye yarayan bir Git komutu.

Daha fazla bilgi için: https://git-scm.com/docs/git-update-index.

- Düzenlenmiş bir dosya değiştirilmemiş gibi davran (`git status` bunu değişmiş gibi göstermeyecek):

  ```
  git update-index --skip-worktree {{örnek/düzenlenen_dosya}}
  ```

git update-ref

Git referanslarını yaratmak, güncellemek ve silmeye yarayan bir Git komutu.

Daha fazla bilgi için: https://git-scm.com/docs/git-update-ref.

- Bir referansı sil (ilk commit'i hafifçe sıfırlamaya yarar):

```
git update-ref -d {{HEAD}}
```

- Referansı bir mesaj ile güncelle:

```
git update-ref -m {{mesaj}} {{HEAD}} {{4e95e05}}
```

git var

Bir Git mantıksal değişkeninin değerini yazdırır.

Ayrıca bu komuttan daha çok tercih edilen **`git config`**'e bakılması önerilir.

Daha fazla bilgi için: https://git-scm.com/docs/git-var.

- Yerel bir Git mantıksal değişkeninin değerini yazdır:

```
git var {{GIT_AUTHOR_IDENT|GIT_COMMITTER_IDENT|GIT_EDITOR|
GIT_PAGER}}
```

- Tüm Git mantıksal değerlerini sırala:

```
git var -l
```

git worktree

Aynı depoya bağlı çoklu çalışan ağaçları yönet.

Daha fazla bilgi için: https://git-scm.com/docs/git-worktree.

- Belirtilen dala sahip yeni bir dizin yarat:

  ```
  git worktree add {{örnek/dizin}} {{dal}}
  ```

- Yeni bir dala sahip yeni bir dizin yarat:

  ```
  git worktree add {{örnek/dizin}} -b {{yeni_dal}}
  ```

- Bu depoya bağlı tüm çalışan dizinleri sırala:

  ```
  git worktree list
  ```

- Bir çalışma ağacını (çalışma ağacı dizinini sildikten sonra) kaldır:

  ```
  git worktree prune
  ```

git

Dağıtım sürüö kontrol sistemi.

Daha fazla bilgi için: https://git-scm.com/.

- Git sürümünü kontrol et:

  ```
  git --version
  ```

- Genel yardım sayfasını görüntüle:

  ```
  git --help
  ```

- Bir Git alt komutu (`commit`, `log` gibi) için yardım sayfasını görüntüle:

  ```
  git help {{alt_komut}}
  ```

- Bit Git alt komutunu çalıştır:

  ```
  git {{alt_komut}}
  ```

- Bit Git alt komutunu belirtilen depoda çalıştır:

  ```
  git -C {{örnek/depo}} {{alt_komut}}
  ```

- Bir Git alt komutunu belirtilen biçimlendirmeye uygun olarak çalıştır:

  ```
  git -c '{{config.key}}={{değer}}' {{alt_komut}}
  ```

github-label-sync

GitHub etiketlerini senkronize etmeye yarayan komut satırı arayüzü.

Daha fazla bilgi için: https://github.com/Financial-Times/github-label-sync.

- Yerel bir `labels.json` dosyası kullanarak etiketleri senkronize et:

  ```
  github-label-sync --access-token {{token}} {{depo_ismi}}
  ```

- Belirli bir etiketlenen JSON dosyası kullanarak etiketleri senkronize et:

  ```
  github-label-sync --access-token {{token}} --labels {{url|
  örnek/json_dosyası}} {{depo_ismi}}
  ```

- Programı etiketleri gerçekten senkronize etmeden çalıştır:

  ```
  github-label-sync --access-token {{token}} --dry-run
  {{depo_ismi}}
  ```

- `labels.json` içinde olmayan etiketleri sakla:

  ```
  github-label-sync --access-token {{token}} --allow-added-
  labels {{depo_ismi}}
  ```

- `GITHUB_ACCESS_TOKEN` ortam değişkenini kullanarak senkronize et:

  ```
  github-label-sync {{depo_ismi}}
  ```

gitk

Görsel Git depo tarayıcısı.

Daha fazla bilgi için: https://git-scm.com/docs/gitk.

- Mevcut Git deposu için depo tarayıcısını göster:

```
gitk
```

- Belirtilmiş dosya veya dizin için depo tarayıcısını göster:

```
gitk {{path/to/file_or_directory}}
```

- 1 hafta önceden beri yapılan commit'leri göster:

```
gitk --since="{{1 week ago}}"
```

- 1/1/2016 tarihinden önceki commit'leri göster:

```
gitk --until="{{1/1/2016}}"
```

- Tüm dallarda en fazla 100 değişiklik göster:

```
gitk --max-count={{100}} --all
```

gitlab-ctl

Çok amaçlı GitLab yönetim CLI aracı.

Daha fazla bilgi için: https://docs.gitlab.com/omnibus/maintenance/.

- Tüm servislerin durumunu görüntüle:

  ```
  sudo gitlab-ctl status
  ```

- Belirtilen servisin durumunu görüntüle:

  ```
  sudo gitlab-ctl status {{nginx}}
  ```

- Tüm servisleri yeniden başlat:

  ```
  sudo gitlab-ctl restart
  ```

- Belirtilen servisi yeniden başlat:

  ```
  sudo gitlab-ctl restart {{nginx}}
  ```

- Tüm servislerin kaydını görüntüle ve `Ctrl + C` basılana kadar okumaya devam et:

  ```
  sudo gitlab-ctl tail
  ```

- Belirtilen servisin kaydını görüntüle:

  ```
  sudo gitlab-ctl tail {{nginx}}
  ```

gitlab-runner

GitLab koşucuları için CLI aracı.

Daha fazla bilgi için: https://docs.gitlab.com/runner/.

- Bir koşucuyu kayıt ettir:

```
sudo gitlab-runner register --url {{https://
gitlab.ornek.com}} --registration-token {{token}} --name
{{isim}}
```

- Bir koşucuyu Docker çalıştırıcısıyla kayı ettir:

```
sudo gitlab-runner register --url {{https://
gitlab.ornek.com}} --registration-token {{token}} --name
{{isim}} --executor {{docker}}
```

- Bir koşucunun kaydını geri al:

```
sudo gitlab-runner unregister --name {{isim}}
```

- Koşucu servisinin durumunu görüntüle:

```
sudo gitlab-runner status
```

- Koşucu servisini yeniden başlat:

```
sudo gitlab-runner restart
```

- Kayıt edilen koşucuların GitLab'e bağlanabilme durumlarını kontrol et:

```
sudo gitlab-runner verify
```

gitlab

GitLab API'si için Ruby sarıcı ve CLI aracı.

`gitlab ctl` gibi bazı alt komutların kendi kullanım kılavuzları vardır.

Daha fazla bilgi için: https://narkoz.github.io/gitlab/.

- Yeni bir proje oluştur:

  ```
  gitlab create_project {{proje_ismi}}
  ```

- Belirtilen commit ile ilgili bilgi al:

  ```
  gitlab commit {{proje_ismi}} {{commit_değeri}}
  ```

- Bit CI pipeline'ındaki işler ile ilgili bilgi al:

  ```
  gitlab pipeline_jobs {{proje_ismi}} {{pipeline_id'si}}
  ```

- Belirtilen CI işini başlat:

  ```
  gitlab job_play {{proje_ismi}} {{iş_id'si}}
  ```

gitmoji

Commit'lerde emoji kullanmak içni interaktif bir komut satırı aracı.

Daha fazla bilgi için: https://github.com/carloscuesta/gitmoji-cli.

- Commit sihirbazını çalıştır:

  ```
  gitmoji --commit
  ```

- Git hook'u başlat (bu sayede `git commit` çalıştırıldığı zaman `gitmoji` otomatik olarak çalıştırılabilir):

  ```
  gitmoji --init
  ```

- Git hook'u sil:

  ```
  gitmoji --remove
  ```

- Tüm kullanılabilir emojileri ve açıklamalarını sırala:

  ```
  gitmoji --list
  ```

- Belirtilen kelime sırası için emoji sırası ara:

  ```
  gitmoji --search {{kelime1}} {{kelime2}}
  ```

- Ana depodan emojileri güncelle:

  ```
  gitmoji --update
  ```

- Genel tercihleri düzenle:

  ```
  gitmoji --config
  ```

gitsome

GitHub için gh komutuyla erişilebilen terminal tabanlı arayüz.

Ayrıca **`git`** komutları için menu tarzı otomatik tamamlanmış öneriler sunar.

Daha fazla bilgi için: https://github.com/donnemartin/gitsome.

- Otomatik tamamlamayı ve Git ile gh komutları için etkileşimli yardımı etkinleştirmek için gitsome kabuğuna gir:

  ```
  gitsome
  ```

- Mevcut hesap ile GitHub entegrasyonunu ayarla:

  ```
  gh configure
  ```

- Mevcut hesap için bildirimleri (https://github.com/notifications adresinde görülebildiği gibi) sırala:

  ```
  gh notifications
  ```

- Mevcut hesabın yıldızlanan depolarını belirtilen filtre ile sırala:

  ```
  gh starred "{{python 3}}"
  ```

- Belirtilen GitHub deposunun güncel etkileşimini görüntüle:

  ```
  gh feed {{tldr-pages/tldr}}
  ```

- Belirtilen GitHub kullanıcısının güncel etkileşimini varsayılan sayfacı ile (örneğin `less`) göster:

  ```
  gh feed {{torvalds}} -p
  ```

gnmic-sub

Bu komut **`gnmic subscribe`** için bir takma addır.

Daha fazla bilgi için: https://gnmic.kmrd.dev/cmd/subscribe.

- Asıl komutun belgelerini görüntüleyin:

```
tldr gnmic subscribe
```

go bug

Bug bildir.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Start_a_bug_report.

- Bug bildirisini başlatmak için bir website aç:

```
go bug
```

go build

Go kaynaklarını derle.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Compile_packages_and_dependencies.

- Bir 'package main' dosyasını derle (çıktı uzantısız bir dosya ismi olacak):

```
go build {{örnek/konum/main.go}}
```

- Çıktı dosya ismini belirterek derle:

```
go build -o {{örnek/konum/binary}} {{örnek/konum/kaynak.go}}
```

- Bir paket yarat:

```
go build -o {{örnek/konum/binary}} {{örnek/konum/paket}}
```

- Bir ana paketi veri yarış tanımlayıcısını etkinleştirerek çalıştırılabilir olarak derle.

```
go build -race -o {{örnek/konum/çalıştırılabilir}} {{örnek/
konum/ana_paket}}
```

go clean

Obje ve önbellek dosyalarını sil.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Remove_object_files_and_cached_files.

- Hiçbir şeyi silmeden silme komutlarını yazdır:

  ```
  go clean -n
  ```

- Yapım önbelleğini sil:Delete the build cache:

  ```
  go clean -cache
  ```

- Tüm önbelleğe alınan test sonuçlarını sil:

  ```
  go clean -testcache
  ```

- Modül önbelleğni sil:

  ```
  go clean -modcache
  ```

go doc

Bir paket veya sembolün dokümentasyonunu göster.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Show_documentation_for_package_or_symbol.

- Mevcut paket için dokümentasyonu göster:

```
go doc
```

- Paket dokümentasyonunu ve dışa aktarılmış sembolleri göster:

```
go doc {{encoding/json}}
```

- Sembollerin de dokümentasyonunu göster:

```
go doc -all {{encoding/json}}
```

- Kaynakları da göster:

```
go doc -all -src {{encoding/json}}
```

- Belirtilen sembolü göster:

```
go doc -all -src {{encoding/json.Number}}
```

go env

Go toolchain'in kullandığı ortam değişkenlerini yönet.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Print_Go_environment_information.

- Tüm ortam değişkenlerini göster:

  ```
  go env
  ```

- Belirtilen ortam değişkenlerini göster:

  ```
  go env {{GOPATH}}
  ```

- Bir değere ortam değişkeni ata:

  ```
  go env -w {{GOBIN}}={{örnek/konum/dizin}}
  ```

- Ortam değişkeninin değerini sıfırla:

  ```
  go env -u {{GOBIN}}
  ```

go fix

Yeni API'ler kullanmak için paketleri güncelle.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Update_packages_to_use_new_APIs.

- Paketleri yeni API'ler kullanmak için güncelle:

```
go fix {{paketler}}
```

go fmt

Go kaynak dosyalarını formatla.

Değiştirilen dosya isimlerini yazdırır.

Daha fazla bilgi için: https://pkg.go.dev/cmd/go#hdr-Gofmt__reformat__package_sources.

- Mevcut dizindeki Go kaynak dosyalarını formatla:

  ```
  go fmt
  ```

- Belirtilen Go paketini içe aktarım yolunda formatla (`$GOPATH/src`):

  ```
  go fmt {{örnek/konum/paket}}
  ```

- Paketi mevcut dizinde ve tüm öbür alt dizinlerde formatla (`...` ifadesine dikkat):

  ```
  go fmt {{./...}}
  ```

- Hiçbir şeyi düzenlemeden format komutlarının ne yapacağını yazdır:

  ```
  go fmt -n
  ```

- Komut çalışırken arkaplanda hangi komutların çalıştığını yazdır:

  ```
  go fmt -x
  ```

go generate

Kaynak dosyaları içinde komut çalıştırarak Go dosyaları oluştur.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Generate_Go_files_by_processing_source.

- Kaynak dosyaları içinde komut çalıştırarak Go dosyaları oluştur:

```
go generate
```

go get

Bir bağımlılık paketi ekle veya eski GOPATH modunda paket indir.

Daha fazla bilgi için: https://pkg.go.dev/cmd/go#hdr-Add_dependencies_to_current_module_and_install_them.

- `go.mod`'a modül modunda (module-mode) belirtilen bir paket ekle veya paketi GOPATH modunda indir:

  ```
  go get {{ornek.com/pkg}}
  ```

- Paketi module-aware modunda belirtilen sürümde düzenle:

  ```
  go get {{ornek.com/pkg}}@{{v1.2.3}}
  ```

- Belirtilen paketi sil:

  ```
  go get {{ornek.com/pkg}}@{{none}}
  ```

go install

İçe aktarım yollarıyla isimlendirilen paketleri derle ve indir.

Daha fazla bilgi için: https://pkg.go.dev/cmd/go#hdr-Compile_and_install_packages_and_dependencies.

- Mevcut paketi derle ve indir:

```
go install
```

- Belirtilen yerel paketi derle ve indir:

```
go install {{örnek/konum/paket}}
```

- Bir programın son sürümünü mevcut dizindeki `go.mod`'u yoksayarak indir:

```
go install {{golang.org/x/tools/gopls}}@{{latest}}
```

- Bir programın mevcut dizindeki `go.mod`'da belirtilen sürümünü indir:

```
go install {{golang.org/x/tools/gopls}}
```

go list

Paket ve modülleri sırala.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-List_packages_or_modules.

- Paketleri sırala:

```
go list ./...
```

- Standart paketleri sırala:

```
go list std
```

- Paketleri JSON formatında sırala:

```
go list -json time net/http
```

- Modül bağımlılıklarını ve erişilebilir güncellemeleri sırala:

```
go list -m -u all
```

go mod

Modül yönetimi.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Module_maintenance.

- Mevcut dizinde yeni modül başlat:

  ```
  go mod init {{modülİsmi}}
  ```

- Modülleri yerel önbelleğe yükle:

  ```
  go mod download
  ```

- Kaybolan modülleri ekle ve kullanılmayanları sil:

  ```
  go mod tidy
  ```

- Bağlılıkların beklenen içeriğe sahip olduklarını doğrula:

  ```
  go mod verify
  ```

- Tüm bağlılıkların kaynaklarını satıcı dizine kopyala:

  ```
  go mod vendor
  ```

go run

Binary (ikili sayı değeri) kaydetmeden Go kodunu derle ve çalıştır.

Daha fazla bilgi için: https://pkg.go.dev/cmd/go#hdr-Compile_and_run_Go_program.

- Bir Go dosyası çalıştır:

```
go run {{örnek/konum/dosya.go}}
```

- Ana bir Go paketi çalıştır:

```
go run {{örnek/konum/paket}}
```

go test

Go paketlerini test et (dosyalar **`_test.go`** ifadesiyle bitmeli).

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Testing_flags.

- Mevcut dizinde bulunan paketleri test et:

  ```
  go test
  ```

- Mevcut dizindeki paketleri ayrıntılı şekilde test et:

  ```
  go test -v
  ```

- Mevcut dizindeki ve tüm alt dizinlerdeki paketleri test et (`...` ifadesine dikkat):

  ```
  go test -v ./...
  ```

- Mevcut dzindeki paketleri test et ve tüm kalite testlerini çalıştır:

  ```
  go test -v -bench .
  ```

- Mevcut dizindeki paketleri test et ve 50 saniye içinde tüm kalite testlerini çalıştır:

  ```
  go test -v -bench . -benchtime {{50s}}
  ```

- Paketleri kapsamlı bir analiz ile test et:

  ```
  go test -cover
  ```

go tool

Belirtilen bir Go aracını veya komutunu çalıştır.

Bir Go komutunu tipik olarak hata ayıklamak için tek başına bir binary olarak çalıştır.

Daha fazla bilgi için: https://pkg.go.dev/cmd/go#hdr-Run_specified_go_tool.

- Erişilebilir araçları sırala:

  ```
  go tool
  ```

- Go bağ aracını çalıştır:

  ```
  go tool link {{örnek/konum/main.o}}
  ```

- Çalıştırılacak komutu çalıştırmadan yazdır (`whereis`'e benzer):

  ```
  go tool -n {{komut}} {{argümanları}}
  ```

- Belirtilen araç için resmi dokümentasyonu göster:

  ```
  go tool {{komut}} --help
  ```

go version

Go sürümünü yazdır.

Daha fazla bilgi için: https://golang.org/cmd/go/#hdr-Print_Go_version.

- Go sürümünü yazdır:

  ```
  go version
  ```

- Belirtilen çalıştırılabilir dosyanın yapımı için kullanılan Go sürümünü yazdır:

  ```
  go version {{örnek/konum/çalıştırılabilir_dosya}}
  ```

go vet

Go kaynak kodunu kontrol et ve şüpheli yapıları bildir (örneğin Go kaynak dosyalarını tiftik et).

Go vet komutu eğer sorun bulunduysa sıfır olmayan bir çıkış kodu yazdırır. Eğer herhangi bir sorun bulunmadıysa sıfır çıkış kodu yazdırılır.

Daha fazla bilgi için: https://pkg.go.dev/cmd/vet.

- Mevcut dizindeki Go paketini kontrol et:

  ```
  go vet
  ```

- Belirtilen yoldaki Go paketini kontrol et:

  ```
  go vet {{örnek/dosya_veya_dizin}}
  ```

- Go vet ile çalıştırılabilecek erişilebilir kontrolleri sırala:

  ```
  go tool vet help
  ```

- Belirtilen bir kontrol için detayları ve bayrakları göster:

  ```
  go tool vet help {{kontrol_ismi}}
  ```

- Kontrolün sorun bulmasına sebep olan satırları artı N sayıda ek içeriği görüntüle:

  ```
  go vet -c={{N}}
  ```

- Analiz ve hataları JSON formatında çıkart:

  ```
  go vet -json
  ```

go

Go kaynak kodunu yönetmeye yarayan bir araç.

`go build` gibi bazı alt komutların kendı kullanım dokümentasyonları mevcut.

Daha fazla bilgi için: https://golang.org.

- İçe aktarım yolunda belirtilen şekilde bir paketi indir ve yükle:

  ```
  go get {{paket_yolu}}
  ```

- Bir kaynak dosyasını derle ve çalıştır (bir `main` paketine sahip olmalı):

  ```
  go run {{dosya}}.go
  ```

- Bir kaynak dosyasını belirtilen çalıştırılabilir dosyaya derle:

  ```
  go build -o {{çalıştırılabilir}} {{dosya}}.go
  ```

- Mevcut dizinde bulunan paketi derle:

  ```
  go build
  ```

- Mevcut paket için tüm test durumlarını çalıştır (bahsi geçen dosyalar `_test.go` ifadesi ile bitmeli):

  ```
  go test
  ```

- Mevcut paketi derle ve indir:

  ```
  go install
  ```

- Mevcut diizinde yeni bir modül başlat:

  ```
  go mod init {{modül_ismi}}
  ```

google-chrome

Bu komut **`chromium`** için bir takma addır.

Daha fazla bilgi için: https://chrome.google.com.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr chromium
  ```

grep

Düzenli ifadeler (Regex) kullanarak dosyalardaki kalıpları bul.

Daha fazla bilgi için: https://www.gnu.org/software/grep/manual/grep.html.

- Bir dosya içinde kalıp ara:

```
grep "{{aranan_kalıp}}" {{dosya/yolu}}
```

- Tam bir dize ara (düzenli ifadeleri devre dışı bırakır):

```
grep --fixed-strings "{{tam_dize}}" {{dosya/yolu}}
```

- Bir dizindeki tüm dosyalarda bir kalıbı tekrarlı olarak ara, eşleşmelerin satır numaralarını göster, binary dosyaları göz ardı et:

```
grep --recursive --line-number --binary-files={{without-
match}} "{{aranan_kalıp}}" {{dosya/yolu}}
```

- Büyük/küçük harfe duyarsız modda genişletilmiş düzenli ifadeleri (`?`, `+`, `{}`, `()` ve `|` destekler) kullan:

```
grep --extended-regexp --ignore-case "{{aranan_kalıp}}"
{{dosya/yolu}}
```

- Her eşleşmenin etrafında, öncesinde veya sonrasında 3 satır içerik yazdır:

```
grep --{{context|before-context|after-context}}={{3}}
"{{aranan_kalıp}}" {{dosya/yolu}}
```

- Renkli çıktı ile her eşleşme için dosya adını ve satır numarasını yazdır:

```
grep --with-filename --line-number --color=always
"{{aranan_kalıp}}" {{dosya/yolu}}
```

- Bir kalıpla eşleşen satırları ara, yalnızca eşleşen metni yazdır:

```
grep --only-matching "{{aranan_kalıp}}" {{dosya/yolu}}
```

- Bir kalıpla eşleşmeyen satırlar için `stdin`'de arama yap:

```
cat {{dosya/yolu}} | grep --invert-match "{{aranan_kalıp}}"
```

hx

Bu komut **`helix`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr helix
  ```

kafkacat

Bu komut **`kcat`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr kcat
  ```

llvm-ar

Bu komut **`ar`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr ar
  ```

llvm-g++

Bu komut **`clang++`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr clang++
```

llvm-gcc

Bu komut **`clang`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr clang
  ```

llvm-nm

Bu komut **`nm`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr nm
  ```

llvm-objdump

Bu komut **`objdump`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr objdump
  ```

llvm-strings

Bu komut **`strings`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr strings
```

lzcat

Bu komut **xz** için bir takma addır.

Daha fazla bilgi için: https://manned.org/lzcat.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr xz
  ```

lzma

Bu komut **xz** için bir takma addır.

Daha fazla bilgi için: https://manned.org/lzma.

- Asıl komutun belgelerini görüntüleyin:

```
tldr xz
```

man

Kılavuz sayfalarını biçimlendir ve göster.

Daha fazla bilgi için: https://www.man7.org/linux/man-pages/man1/man.1.html.

- Bir komut için man sayfasını görüntüle:

```
man {{komut}}
```

- Sayfanın 7. bölümündeki bir komut için man sayfasını görüntüle:

```
man {{7}} {{komut}}
```

- Mansayfaları için aratılan yolu göster:

```
man --path
```

- Mansayfasını göstermek yerine mansayfasının konumunu göster:

```
man -w {{komut}}
```

- Belirtilen ifadeyi içeren mansayfalarını ara:

```
man -k "{{aranan_ifade}}"
```

minetest

Çok oyunculu sınırsız dünyalı bloklu sandbox oyun motoru.

Ayrıca **`minetestserver`** sayfasına bakılması önerilir.

Daha fazla bilgi için: https://wiki.minetest.net/Minetest.

- Minetest'i kullanıcı modunda başlat:

  ```
  minetest
  ```

- Minetest'i belirtilen dünyayı host edecek şekilde sunucu modunda başlat:

  ```
  minetest --server --world {{isim}}
  ```

- Belirtilmiş bir dosyaya geçmişi yaz:

  ```
  minetest --logfile {{örnek/dosya}}
  ```

- Hataları yalnızca konsola yaz:

  ```
  minetest --quiet
  ```

minetestserver

Çok oyunculu sınırsız dünyalı bloklu sanbox sunucusu.

Ayrıca **`minetest`** sayfasına bakılması önerilir.

Daha fazla bilgi için: https://wiki.minetest.net/Setting_up_a_server.

- Sunucuyu başlar:

  ```
  minetestserver
  ```

- Müsait dünyaları sırala:

  ```
  minetestserver --world list
  ```

- Yüklenecek dünya ismini belirt:

  ```
  minetestserver --world {{dunya_ismi}}
  ```

- Müsait oyun ID'lerini sırala:

  ```
  minetestserver --gameid list
  ```

- Kullanılacak oyunu belirt:

  ```
  minetestserver --gameid {{oyun_id'si}}
  ```

- Belirtilmiş bir port'u dinle:

  ```
  minetestserver --port {{34567}}
  ```

- Başka bir veritabanı yazılımına göç et:

  ```
  minetestserver --migrate {{sqlite3|leveldb|redis}}
  ```

- Sunucuyu başlattıktan sonra interaktif bir terminal aç:

  ```
  minetestserver --terminal
  ```

mkdir

Yeni bir dizin oluştur.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/mkdir.

- Mevcut dizinde ya da verilen dizinde yeni bir dizin oluştur:

```
mkdir {{dizin}}
```

- Mevcut dizinde birden çok dizin oluştur:

```
mkdir {{dizin_1 dizin_2 ...}}
```

- Özyinelemeli şekilde dizin oluştur (iç içe klasörler oluşturmak için kullanışlıdır):

```
mkdir -p {{dizin/yolu}}
```

mscore

Bu komut **`musescore`** için bir takma addır.

Daha fazla bilgi için: https://musescore.org/handbook/command-line-options.

- Asıl komutun belgelerini görüntüleyin:

```
tldr musescore
```

ng

Angular uygulamaları oluşturup yönetmek için bir Komut Satırı Arayüzü (CLI).

Daha fazla bilgi için: https://angular.io/cli.

- Bir klasörün içinde yeni bir Angular uygulaması oluştur:

  ```
  ng new {{proje_ismi}}
  ```

- Bir uygulamaya yeni bir komponent ekle:

  ```
  ng generate component {{komponent_ismi}}
  ```

- Bir uygulamaya yeni bir sınıf ekle:

  ```
  ng generate class {{sınıf_ismi}}
  ```

- Bir uygulamaya yeni bir direktif ekle:

  ```
  ng generate directive {{direktif_ismi}}
  ```

- Uygulamayı çalıştır ve bir sunucu üzerinden yayınla:

  ```
  ng serve
  ```

- Uygulamayı derle:

  ```
  ng build
  ```

- Testleri çalıştır:

  ```
  ng test
  ```

- Angular kurulumunun versiyonunu kontrol et:

  ```
  ng version
  ```

nginx

Nginx web sunucusu.

Daha fazla bilgi için: https://nginx.org/en/.

- Varsayılan konfigürasyon dosyasıyla sunucuyu başlat:

  ```
  nginx
  ```

- Özel bir konfigürasyon dosyasıyla sunucuyu başlat:

  ```
  nginx -c {{konfigürasyon_dosyası}}
  ```

- Konfigürasyon dosyasındaki her göreceli dosya yolu için bir ön ek ile sunucuyu başlat:

  ```
  nginx -c {{konfigürasyon_dosyası}} -p {{göreceli/dosya/yolu/ön/eki}}
  ```

- Çalışan sunucuyu etkilemeden konfigürasyon dosyasını test et:

  ```
  nginx -t
  ```

- Aksamasız bir sinyal göndererek konfigürasyonu tekrar yükle:

  ```
  nginx -s reload
  ```

nm-classic

Bu komut **`nm`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr nm
```

ntl

Bu komut **`netlify`** için bir takma addır.

Daha fazla bilgi için: https://cli.netlify.com.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr netlify
  ```

pio-init

Bu komut **`pio project`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr pio project
```

piodebuggdb

Bu komut **`pio debug`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr pio debug
  ```

platformio

Bu komut **`pio`** için bir takma addır.

Daha fazla bilgi için: https://docs.platformio.org/en/latest/core/userguide/.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr pio
  ```

ptpython3

Bu komut **`ptpython`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr ptpython
  ```

python3

Bu komut **`python`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr python
  ```

r2

Bu komut **`radare2`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr radare2
```

rcat

Bu komut **`rc`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr rc
  ```

rg

Ripgrep, yinelemeli satır-odaklı bir CLI arama aracıdır.

Grep'e daha hızlı bir alternatif olmayı hedefler.

Daha fazla bilgi için: https://github.com/BurntSushi/ripgrep.

- Normal bir ifade için geçerli dizini yinelemeli olarak ara:

```
rg {{normal_ifade}}
```

- Geçerli dizinde, gizli dosyalar ve ".gitignore" da listelenen dosyalar dahil olmak üzere normal ifadeleri yinelemeli olarak ara:

```
rg --no-ignore --hidden {{normal_ifade}}
```

- Normal ifadeyi yalnızca bir dizin alt kümesinde ara:

```
rg {{normal_ifade}} {{dizin_alt_kümesi}}
```

- Bir glob ile eşleşen dosyalarda normal bir ifade ara (örn: `README.*`):

```
rg {{normal_ifade}} --glob {{glob}}
```

- Normal bir ifadeyle eşleşen dosya adlarını ara:

```
rg --files | rg {{normal_ifade}}
```

- Yalnızca eşleşen dosyaları listele (diğer komutlara yönlendirirken kullanışlıdır):

```
rg --files-with-matches {{normal_ifade}}
```

- Verilen normal ifadeyle eşleşmeyen satırları göster:

```
rg --invert-match {{normal_ifade}}
```

- Bir değişmez dizi deseni için arama yap:

```
rg --fixed-strings -- {{dizi}}
```

ripgrep

Bu komut **`rg`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr rg
  ```

su

Kabuk ortamında başka bir kullanıcıya geçiş yapın.

Daha fazla bilgi için: https://manned.org/su.

- Süper kullanıcıya geçin (kök şifresi gerektirir):

```
su
```

- Belirli bir kullanıcıya geçin (kullanıcının şifresini gerektirir):

```
su {{kullanıcıadı}}
```

- Belirli bir kullanıcıya geçin ve tam oturum açma kabuğunu simüle edin:

```
su - {{kullanıcıadı}}
```

- Başka bir kullanıcı olarak bir komut çalıştırın:

```
su - {{kullanıcıadı}} -c "{{komut}}"
```

tar

Arşivleme aracı.

Dosyalar genellikle gzip veya bzip2 gibi bir sıkıştırma yöntemiyle birleştirilir.

Daha fazla bilgi için: https://www.gnu.org/software/tar.

- Bir arşiv oluştur ve dosyaya yaz:

```
tar cf {{hedef.tar}} {{dosya1}} {{dosya2}} {{dosya3}}
```

- Bir gzip arşivi oluştur ve dosyaya yaz:

```
tar czf {{hedef.tar.gz}} {{dosya1}} {{dosya2}} {{dosya3}}
```

- Göreceli yolları kullanarak bir gzip arşivi oluştur:

```
tar czf {{hedef.tar.gz}} --directory={{dizin/yolu}} .
```

- Sıkıştırılmış bir arşiv dosyasını geçerli dizine ayrıntılı şekilde çıkar:

```
tar xvf {{kaynak.tar[.gz|.bz2|.xz]}}
```

- Sıkıştırılmış bir arşiv dosyasını hedef dizine çıkar:

```
tar xf {{kaynak.tar[.gz|.bz2|.xz]}} --directory={{dizin}}
```

- Sıkıştırılmış bir arşiv oluştur ve sıkıştırma yöntemini seçmek için arşiv sonekini kullan:

```
tar caf {{hedef.tar.xz}} {{dosya1}} {{dosya2}} {{dosya3}}
```

- Bir tar arşivinin içeriğini ayrıntılı olarak listele:

```
tar tvf {{kaynak.tar}}
```

- Şablonla eşleşen dosyaları arşivden çıkar:

```
tar xf {{kaynak.tar}} --wildcards "{{*.html}}"
```

tldr-lint

`tldr` sayfalarını gözden geçir ve biçimlendir.

Daha fazla bilgi için: https://github.com/tldr-pages/tldr-lint.

- Tüm sayfaları gözden geçir:

  `tldr-lint {{sayfa_dizini}}`

- Belirtilmiş bir sayfayı `stdout`'a biçimlendir:

  `tldr-lint --format {{page.md}}`

- Bir konumdaki tüm sayfaları biçimlendir:

  `tldr-lint --format --in-place {{sayfa_dizini}}`

tldr

Komut satırı araçları için tldr-pages projesinden basit yardım sayfaları görüntüler.

Daha fazla bilgi için: https://tldr.sh.

- Bir komutun tipik kullanımını göster (ipucu: burayı görüntülemek için kullandığınız komutun aynısı!):

  ```
  tldr {{komut}}
  ```

- Linux için tar tldr sayfasını göster:

  ```
  tldr -p {{linux}} {{tar}}
  ```

- Bir Git alt komutu için yardım al:

  ```
  tldr {{git-checkout}}
  ```

- (Eğer alıcı önbellek oluşumunu destekliyorsa) Yerel paketleri güncelle:

  ```
  tldr -u
  ```

tldrl

`tldr-lint` komutunun aynısı.

Daha fazla bilgi için: https://github.com/tldr-pages/tldr-lint.

- Orijinal komut için yardım sayfasını göster:

  ```
  tldr tldr-lint
  ```

tlmgr-arch

Bu komut **`tlmgr platform`** için bir takma addır.

Daha fazla bilgi için: https://www.tug.org/texlive/tlmgr.html.

- Asıl komutun belgelerini görüntüleyin:

```
tldr tlmgr platform
```

todoman

Bu komut **`todo`** için bir takma addır.

Daha fazla bilgi için: https://todoman.readthedocs.io/.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr todo
  ```

transmission

Bu komut **`transmission-daemon`** için bir takma addır.

Daha fazla bilgi için: https://transmissionbt.com/.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr transmission-daemon
  ```

tree

Mevcut dizinin içeriğini ağaç biçiminde göster.

Daha fazla bilgi için: http://mama.indstate.edu/users/ice/tree/.

- Dosya ve dizinleri `num` değeri kadar derinlikte göster (1 olması durumunda mevcut dizin gösterilir):

```
tree -L {{num}}
```

- Yalnızca dizinleri göster:

```
tree -d
```

- Renklendirme açık olacak şekilde gizli dosyaları dahi göster:

```
tree -a -C
```

- Ağacın satırlarını girintiler yerine tüm yolu belirterek göster:

```
tree -i -f
```

- Tüm dosyaların ve dizinlerin eklenerek artan boyutlarını, insanların okuyabileceği bir biçimde göster:

```
tree -s -h --du
```

- Ağaç hiyerarşisi içindeki dosyaları bir wildcard (glob) kalıbı kullanarak ve aranan özellikteki dosyalara sahip olmayan dizinleri yoksayarak göster:

```
tree -P '{{*.txt}}' --prune
```

- Ağaç hiyerarşisi içindeki dizinleri bir wildcard (glob) kalıbı kullanarak ve istenen dizine atalığı olmayan dizinleri yoksayarak göster:

```
tree -P {{dizin_ismi}} --matchdirs --prune
```

- Ağacı belirtilen dizinleri yoksayarak göster:

```
tree -I '{{dizin_ismi1|dizin_ismi2}}'
```

unlzma

Bu komut **xz** için bir takma addır.

Daha fazla bilgi için: https://manned.org/unlzma.

- Asıl komutun belgelerini görüntüleyin:

```
tldr xz
```

unxz

Bu komut **xz** için bir takma addır.

Daha fazla bilgi için: https://manned.org/unxz.

- Asıl komutun belgelerini görüntüleyin:

```
tldr xz
```

vi

Bu komut **`vim`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr vim
  ```

xzcat

Bu komut **xz** için bir takma addır.

Daha fazla bilgi için: https://manned.org/xzcat.

- Asıl komutun belgelerini görüntüleyin:

```
tldr xz
```

yes

Bir şeyi tekrar tekrar yazdır.

Bu komut genelde yükleme işlemleri sırasında onay için yes yazmak için kullanılır (apt-get gibi).

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/yes.

- Tekrar tekrar "mesaj" yazdır:

```
yes {{mesaj}}
```

- Tekrar tekrar "y" yazdır:

```
yes
```

- `apt-get` komutu tarafından sorulan her şeyi kabul et:

```
yes | sudo apt-get install {{program}}
```

Linux

a2disconf

Debian tabanlı işletim sistemlerinde Apache konfigürasyon dosyasını devre dışı bırak.

Daha fazla bilgi için: https://manpages.debian.org/latest/apache2/a2disconf.8.en.html.

- Bir konfigürasyon dosyasını devre dışı bırak:

```
sudo a2disconf {{konfigürasyon_dosyası}}
```

- Bilgilendirici mesajı gösterme:

```
sudo a2disconf --quiet {{konfigürasyon_dosyası}}
```

a2dismod

Debian tabanlı işletim sistemlerinde bir Apache modülünü devre dışı bırak.

Daha fazla bilgi için: https://manpages.debian.org/latest/apache2/a2dismod.8.en.html.

- Bir modülü devre dışı bırak:

```
sudo a2dismod {{modül}}
```

- Bilgilendirici mesajları gösterme:

```
sudo a2dismod --quiet {{module}}
```

a2dissite

Debian tabanlı işletim sistemlerinde bir Apache sanal hostunu devre dışı bırak.

Daha fazla bilgi için: https://manpages.debian.org/latest/apache2/a2dissite.8.en.html.

- Uzak hostu devre dışı bırak:

```
sudo a2dissite {{sanal_host}}
```

- Bilgilendirici mesajları gösterme:

```
sudo a2dissite --quiet {{sanal_host}}
```

a2enconf

Debian tabanlı işletim sistemlerinde Apache konfigürasyon dosyasını etkinleştir.

Daha fazla bilgi için: https://manpages.debian.org/latest/apache2/a2enconf.8.en.html.

- Bir konfigürasyon dosyasını etkinleştir:

```
sudo a2enconf {{konfigürasyon_dosyası}}
```

- Bilgilendirici mesajları gösterme:

```
sudo a2enconf --quiet {{konfigürasyon_dosyası}}
```

a2enmod

Debian tabanlı işletim sistemlerinde Apache modülünü etkinleştir.

Daha fazla bilgi için: https://manpages.debian.org/latest/apache2/a2enmod.8.en.html.

- Bir modülü etkinleştir:

```
sudo a2enmod {{modül}}
```

- Bilgilendirici mesajları gösterme:

```
sudo a2enmod --quiet {{modül}}
```

a2ensite

Debian tabanlı işletim sistemlerinde Apache sanal hostu etkinleştir.

Daha fazla bilgi için: https://manpages.debian.org/latest/apache2/a2ensite.8.en.html.

- Bir sanal hostu etkinleştir:

```
sudo a2ensite {{sanal_host}}
```

- Bilgilendirici mesajları gösterme:

```
sudo a2ensite --quiet {{sanal_host}}
```

a2query

Apache ve Debian tabanlı işletim sistemlerinde çalışma süresi yapılandırmasını kurtar.

Daha fazla bilgi için: https://manpages.debian.org/latest/apache2/a2query.html.

- Etkinleştirilmiş Apache modüllerini sırala:

```
sudo a2query -m
```

- Belirtilen modülün indirilip indirilmediğini kontrol et:

```
sudo a2query -m {{modül_ismi}}
```

- Etkinleştirilmiş sanal hostları sırala:

```
sudo a2query -s
```

- Mevcut etkinleştirilmiş Çoklu İşlem Modülü'nü görüntüle:

```
sudo a2query -M
```

- Apache sürümünü görüntüle:

```
sudo a2query -v
```

alternatives

Bu komut **`update-alternatives`** için bir takma addır.

Daha fazla bilgi için: https://manned.org/alternatives.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr update-alternatives
  ```

apt-get

Debian ve Ubuntu paket yönetim aracı.

Paket aramak için **`apt-cache`** komutunu kullanın.

Daha fazla bilgi için: https://manpages.debian.org/latest/apt/apt-get.8.html.

- Kullanılabilir paket ve versiyon listesini güncelleyin (diğer `apt-get` komutlarını çalıştırmadan önce kullanmanız önerilir):

  ```
  apt-get update
  ```

- Bir paket yükleyin veya son sürüme güncelleyin:

  ```
  apt-get install {{paket}}
  ```

- Bir paketi silin:

  ```
  apt-get remove {{paket}}
  ```

- Bir paketi ve konfigürasyon dosyalarını silin:

  ```
  apt-get purge {{paket}}
  ```

- Yüklü paketlerin hepsini son sürümlerine yükseltin:

  ```
  apt-get upgrade
  ```

- Yerel depoyu temizleyin - kullanılmayan gereksiz paket dosyalarını (.deb) silin:

  ```
  apt-get autoclean
  ```

- Artık gerekmeyen paketleri silin:

  ```
  apt-get autoremove
  ```

- Yüklenmiş paketleri yükseltin (`upgrade` gibi), ancak gereksiz paketleri silin ve yeni bağımlılıkları memnun edecek ek paketler kurun:

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

apt

Debian tabanlı dağıtımlar için paket yönetim aracı.

Ubuntu 16.04 ve sonraki sürümlerde interaktif kullanıldığında **`apt-get`** için önerilen değiştirme.

Daha fazla bilgi için: https://manpages.debian.org/latest/apt/apt.8.html.

- Kullanılabilir paket ve versiyonların listesini yenile (Bu komutu diğer `apt` komutlarından önce kullanmanız önerilir):

  ```
  apt update
  ```

- Belirli bir paketi arayın:

  ```
  apt search {{paket}}
  ```

- Bir paketin bilgilerini gösterin:

  ```
  apt show {{paket}}
  ```

- Bir paket kurun veya mevcut en son sürüme güncelleyin:

  ```
  apt install {{paket}}
  ```

- Bir paketi kaldırın (bunun için "purge" kullanmak, yapılandırma dosyalarını da kaldırır):

  ```
  apt remove {{paket}}
  ```

- Kurulu tüm paketleri mevcut en yeni sürümlerine yükseltin:

  ```
  apt upgrade
  ```

- Tüm paketleri listeleyin:

  ```
  apt list
  ```

- Kurulu paketleri listeleyin:

  ```
  apt list --installed
  ```

aptitude

Debian ve Ubuntu paket yönetim aracı.

Daha fazla bilgi için: https://manpages.debian.org/latest/aptitude/aptitude.8.html.

- Kullanılabilir paket ve sürüm listesini senkronize et. Bu, herhangi bir aptitude komutunu uygulamadan önce çalıştırılmalıdır:

  ```
  aptitude update
  ```

- Yeni bir paket ve onun bağımlılıklarını kur:

  ```
  aptitude install {{paket}}
  ```

- Paket ara:

  ```
  aptitude search {{paket}}
  ```

- İndirilmiş bir paket ara: (`?installed` bir aptitude arama ifadesidir):

  ```
  aptitude search '?installed({{paket}})'
  ```

- Bir paket ve onun bağımlılıklarını kaldır:

  ```
  aptitude remove {{paket}}
  ```

- Yüklü paketleri son kullanılabilir sürümlerine yükselt:

  ```
  aptitude upgrade
  ```

- Yüklü paketleri yükle (`aptitude upgrade` gibi), gereksizleri sil ve yeni bağımlılıkları karşılamak üzere ek paketler kur:

  ```
  aptitude full-upgrade
  ```

- Bir paketin otomatik yükseltilmesini engellemek için onu beklemede tut:

  ```
  aptitude hold '?installed({{paket}})'
  ```

atool

Çeşitli biçimlerdeki arşivleri yönetin.

Daha fazla bilgi için: https://www.nongnu.org/atool/.

- Bir zip arşivindeki dosyaları listele:

```
atool --list {{arşiv.zip/dosyasının/yolu}}
```

- Bir tar.gz arşivini yeni bir alt dizine (veya yalnızca bir dosya içeriyorsa geçerli dizine) çıkart:

```
atool --extract {{arşiv.tar.gz/dosyasının/yolu}}
```

- İki dosyaya sahip yeni bir 7zip arşivi oluştur:

```
atool --add {{arşiv.7z/dosyasının/yolu}} {{dosya1/yolu}} {{dosya2/yolu}}
```

- Geçerli dizindeki tüm zip ve rar arşivlerini çıkart:

```
atool --each --extract {{*.zip}} {{*.rar}}
```

batcat

Bu komut **`bat`** için bir takma addır.

Daha fazla bilgi için: https://github.com/sharkdp/bat.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr bat
  ```

brightnessctl

GNU/Linux işletim sistemlerinde cihaz parlaklığını okumak ve kontrol etmek için yardımcı program.

Daha fazla bilgi için: https://github.com/Hummer12007/brightnessctl.

- Değiştirilebilir parlaklığa sahip cihazları listele:

```
brightnessctl --list
```

- Ekran arka ışığının şu andaki seviyesini yazdır:

```
brightnessctl get
```

- Ekran parlaklığını belli bir aralık dahilinde belirli yüzdeye eşitle:

```
brightnessctl set {{50%}}
```

- Parlaklığı belirli bir seviyede artır:

```
brightnessctl set {{+10%}}
```

- Parlaklığı belirli bir seviyede düşür:

```
brightnessctl set {{10%-}}
```

bspwm

Bu komut **`bspc`** için bir takma addır.

Daha fazla bilgi için: https://github.com/baskerville/bspwm.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr bspc
  ```

cc

Bu komut **`gcc`** için bir takma addır.

Daha fazla bilgi için: https://gcc.gnu.org.

- Asıl komutun belgelerini görüntüleyin:

```
tldr gcc
```

cgroups

Bu komut **`cgclassify`** için bir takma addır.

Daha fazla bilgi için: https://www.kernel.org/doc/Documentation/cgroup-v2.txt.

- Asıl komutun belgelerini görüntüleyin:

```
tldr cgclassify
```

cp

Dosya ve dizinleri kopyala.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/cp.

- Bir dosyayı başka bir konuma kopyala:

```
cp {{örnek/yol/kaynak_dosya.ext}} {{örnek/yol/
hedef_dosya.ext}}
```

- Bir dosyayı ismini değiştirmeden başka bir dizine kopyala:

```
cp {{örnek/yol/kaynak_dosya.ext}} {{örnek/yol/
hedef_ana_dizin}}
```

- Bir dizinin içeriğini başka bir konuma tekrarlı şekilde kopyala (eğer belirtilen konum varsa dizin onun içine kopyalanır):

```
cp -r {{örnek/yol/kaynak_dizin}} {{örnek/yol/hedef_dizin}}
```

- Bir dizini tekrarlı şekilde ayrıntılı modda kopyala (dosyaları kopyalandıkları gibi gösterir):

```
cp -vr {{örnek/yol/kaynak_dizin}} {{örnek/yol/hedef_dizin}}
```

- Metin dosyalarını interaktif modda başka bir konuma kopyala (üstüne yazmadan önce kullanıcıyı bilgilendirir):

```
cp -i {{*.txt}} {{örnek/yol/hedef_dizin}}
```

- Kopyalamadan önce sembolik linkleri izle:

```
cp -L {{link}} {{örnek/yol/hedef_dizin}}
```

- Kopyalarken kaynak dosyalarının tam konumunu belirt:

```
cp --parents {{kaynak/örnek/yol/dosya}} {{örnek/yol/
hedef_dosya}}
```

dnf

RHEL, Fedora ve CentOS için paket yönetim aracı (yum'un yerini alır).

Daha fazla bilgi için: https://dnf.readthedocs.io.

- Kurulu paketleri kullanılabilir en yeni sürümlere yükselt:

  ```
  sudo dnf upgrade
  ```

- Anahtar kelimeler kullanarak paket ara:

  ```
  dnf search {{anahtar_kelimeler}}
  ```

- Bir paketin ayrıntılarını göster:

  ```
  dnf info {{paket}}
  ```

- Yeni bir paket kur:

  ```
  sudo dnf install {{paket}}
  ```

- Yeni bir paket kur ve tüm soruları otomatik evet olarak yanıtla:

  ```
  sudo dnf -y install {{paket}}
  ```

- Bir paketi kaldır:

  ```
  sudo dnf remove {{paket}}
  ```

- Kurulu paketleri listele:

  ```
  dnf list --installed
  ```

- Verilen dosyayı hangi paketlerin sağladığını bul:

  ```
  dnf provides {{dosya}}
  ```

ip address

IP adresi yönetimi alt komutu.

Daha fazla bilgi için: https://manned.org/ip-address.

- Ağ arayüzlerini ve ilişkili IP adreslerini listele:

  ```
  ip address
  ```

- Yalnızca etkin ağ arayüzlerini gösterecek şekilde filtrele:

  ```
  ip address show up
  ```

- Belirli bir ağ arayüzü hakkındaki bilgileri görüntüle:

  ```
  ip address show dev {{eth0}}
  ```

- Bir ağ arayüzüne bir IP adresi ekle:

  ```
  ip address add {{ip_adresi}} dev {{eth0}}
  ```

- Bir ağ arayüzünden bir IP adresini kaldır:

  ```
  ip address delete {{ip_adresi}} dev {{eth0}}
  ```

- Belirli bir kapsamdaki tüm IP adreslerini bir ağ arayüzünden sil:

  ```
  ip address flush dev {{eth0}} scope {{global|host|link}}
  ```

ip link

Ağ arayüzlerini yönet.

Daha fazla bilgi için: https://man7.org/linux/man-pages/man8/ip-link.8.html.

- Tüm ağ arayüzleriyle ilgili bilgileri göster:

```
ip link
```

- Belirli bir ağ arayüzüyle ilgili bilgileri göster:

```
ip link show {{ethN}}
```

- Bir ağ arayüzünü etkinleştir veya devre dışı bırak:

```
ip link set {{ethN}} {{up|down}}
```

- Bir ağ arayüzüne anlamlı bir ad ver:

```
ip link set {{ethN}} alias "{{LAN Arayüzü}}"
```

- Bir ağ arayüzünün MAC adresini değiştir:

```
ip link set {{ethN}} address {{ff:ff:ff:ff:ff:ff}}
```

- Jumbo çerçeveleri kullanması için bir ağ arayüzünün MTU boyutunu değiştir:

```
ip link set {{ethN}} mtu {{9000}}
```

ip neighbour

Komşu/ARP tablosu yönetimi IP alt komutu.

Daha fazla bilgi için: https://manned.org/ip-neighbour.8.

- Komşu/ARP tablosu girdilerini görüntüle:

```
ip neighbour
```

- `eth0` aygıtının komşu tablosundaki girdileri kaldır:

```
sudo ip neighbour flush dev {{eth0}}
```

- Bir komşu araması gerçekleştir ve bir komşu girdisi döndür:

```
ip neighbour get {{aranacak_ip}} dev {{eth0}}
```

- `eth0` arayüzüne komşu IP adresi için bir ARP girdisi ekle veya sil:

```
sudo ip neighbour {{add|del}} {{ip_adresi}} lladdr
{{mac_adresi}} dev {{eth0}} nud reachable
```

- `eth0` arayüzünde komşu IP adresi için bir ARP girdisini değiştir:

```
sudo ip neighbour {{change|replace}} {{ip_adresi}} lladdr
{{yeni_mac_adresi}} dev {{eth0}}
```

ip-route-list

Bu komut **`ip-route-show`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr ip-route-show
  ```

ip route

IP yönlendirme tablosu yönetimi alt komutu.

Daha fazla bilgi için: https://manned.org/ip-route.

- Yönlendirme tablosunu görüntüle:

```
ip route {{show|list}}
```

- Ağ geçidini kullanan öntanımlı bir yönlendirme ekle:

```
sudo ip route add default via {{ağ_geçidi_ip_adresi}}
```

- `eth0` arayüzünü kullanan öntanımlı bir yönlendirme ekle:

```
sudo ip route add default dev {{eth0}}
```

- Statik bir yönlendirme ekle:

```
sudo ip route add {{hedef_ip_adresi}} via
{{ağ_geçidi_ip_adresi}} dev {{eth0}}
```

- Statik bir yönlendirmeyi sil:

```
sudo ip route del {{hedef_ip_adresi}} dev {{eth0}}
```

- Statik bir yönlendirmeyi değiştir:

```
sudo ip route {{change|replace}} {{hedef_ip_adresi}} via
{{ağ_geçidi_ip_adresi}} dev {{eth0}}
```

- Bir IP adresine ulaşmak için çekirdek tarafından hangi yönlendirmenin kullanılacağını göster:

```
ip route get {{hedef_ip_adresi}}
```

ip rule

IP yönlendirme politikası veri tabanı yönetimi.

Daha fazla bilgi için: https://man7.org/linux/man-pages/man8/ip-rule.8.html.

- Yönlendirme politikasını göster:

```
ip rule {{show|list}}
```

- Paket kaynak adreslerine dayalı yeni bir kural ekle:

```
sudo ip rule add from {{192.168.178.2/32}}
```

- Paket hedef adreslerine dayalı yeni bir kural ekle:

```
sudo ip rule add to {{192.168.178.2/32}}
```

- Paket kaynak adreslerine dayalı bir kuralı sil:

```
sudo ip rule delete from {{192.168.178.2/32}}
```

- Paket hedef adreslerine dayalı bir kuralı sil:

```
sudo ip rule delete to {{192.168.178.2/32}}
```

- Silinen tüm kuralları temizle:

```
ip rule flush
```

- Tüm kuralları bir dosyaya kaydet:

```
ip rule save > {{ip_kuralları.dat/dosyasının/yolu}}
```

- Tüm kuralları bir dosyadan geri yükle:

```
ip rule restore < {{ip_kuralları.dat/dosyasının/yolu}}
```

ip

Yönlendirmeyi, aygıtları, kural yönlendirmesini ve tünelleri görüntüle / değiştir.

`ip address` gibi bazı alt komutların kendi kullanım belgeleri vardır.

Daha fazla bilgi için: https://www.man7.org/linux/man-pages/man8/ip.8.html.

- Arayüzlerin bilgilerini ayrıntılı bir şekilde listele:

  ```
  ip address
  ```

- Arayüzlerin ağ katmanı bilgilerini kısa bir şekilde listele:

  ```
  ip -brief address
  ```

- Arayüzlerin bağlantı katmanı bilgilerini kısa bir şekilde listele:

  ```
  ip -brief link
  ```

- Yönlendirme tablosunu görüntüle:

  ```
  ip route
  ```

- Komşuları (ARP tablosunu) göster:

  ```
  ip neighbour
  ```

- Bir arayüzü etkinleştir/devre dışı bırak:

  ```
  ip link set {{arayüz}} up/down
  ```

- Bir arayüze IP adresi ekle/sil:

  ```
  ip addr add/del {{ip}}/{{maske}} dev {{arayüz}}
  ```

- Öntanımlı yönlendirme ekle:

  ```
  ip route add default via {{ip}} dev {{arayüz}}
  ```

megadl

Bu komut **`megatools-dl`** için bir takma addır.

Daha fazla bilgi için: https://megatools.megous.com/man/megatools-dl.html.

- Asıl komutun belgelerini görüntüleyin:

```
tldr megatools-dl
```

ncal

Bu komut **`cal`** için bir takma addır.

Daha fazla bilgi için: https://manned.org/ncal.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr cal
  ```

nmcli connection

NetworkManager ile bağlantı yönetimi.

Daha fazla bilgi için: https://networkmanager.dev/docs/api/latest/nmcli.html.

- Tüm NetworkManager bağlantılarını listele (ad, UUID, tür ve aygıtı gösterir):

```
nmcli connection
```

- UUID belirterek bağlantıyı etkinleştir:

```
nmcli connection up uuid {{uuid}}
```

- Bağlantıyı devre dışı bırak:

```
nmcli connection down uuid {{uuid}}
```

- IPv4 ve IPv6 otomatik olarak yapılandırılan bir bağlantı oluştur:

```
nmcli connection add ifname {{arayüz_adı}} type {{ethernet}}
ipv4.method {{auto}} ipv6.method {{auto}}
```

- Statik bir yalnızca IPv6 bağlantısı oluştur:

```
nmcli connection add ifname {{arayüz_adı}} type {{ethernet}}
ip6 {{2001:db8::2/64}} gw6 {{2001:db8::1}} ipv6.dns
{{2001:db8::1}} ipv4.method {{ignore}}
```

- Statik bir yalnızca IPv4 bağlantısı oluştur:

```
nmcli connection add ifname {{arayüz_adı}} type {{ethernet}}
ip4 {{10.0.0.7/8}} gw4 {{10.0.0.1}} ipv4.dns {{10.0.0.1}}
ipv6.method {{ignore}}
```

- Bir OVPN dosyasından OpenVPN kullanan bir VPN bağlantısı oluştur:

```
nmcli connection import type {{openvpn}} file
{{vpn_yapılandırması.ovpn/dosyasının/yolu}}
```

nmcli device

NetworkManager ile donanım aygıtı yönetimi.

Daha fazla bilgi için: https://networkmanager.dev/docs/api/latest/nmcli.html.

- Tüm ağ arayüzlerinin durumlarını yazdır:

  ```
  nmcli device status
  ```

- Kullanılabilir kablosuz erişim noktalarını yazdır:

  ```
  nmcli device wifi
  ```

- Belirtilen ad ve parola ile kablosuz ağa bağlan:

  ```
  nmcli device wifi connect {{ssid}} password {{parola}}
  ```

- Geçerli kablosuz ağ için parola ve QR kodunu yazdır:

  ```
  nmcli device wifi show-password
  ```

nmcli monitor

NetworkManager bağlantı durumundaki değişiklikleri izleyin.

Daha fazla bilgi için: https://networkmanager.dev/docs/api/latest/nmcli.html.

- NetworkManager değişikliklerini izlemeye başla:

  ```
  nmcli monitor
  ```

nmcli

NetworkManager'ı denetlemek için bir komut satırı aracı.

`nmcli monitor` gibi bazı alt komutların kendi kullanım belgeleri vardır.

Daha fazla bilgi için: https://networkmanager.dev/docs/api/latest/nmcli.html.

- Bir `nmcli` alt komutunu çalıştır:

  ```
  nmcli {{agent|connection|device|general|help|monitor|
  networking|radio}} {{komut_seçenekleri}}
  ```

- Kullanılan NetworkManager sürümünü görüntüle:

  ```
  nmcli --version
  ```

- Yardımı görüntüle:

  ```
  nmcli --help
  ```

- Bir alt komut için yardımı görüntüle:

  ```
  nmcli {{alt_komut}} --help
  ```

nmtui

NetworkManager'ı denetlemek için metin tabanlı kullanıcı arayüzü.

Gezinmek için ok tuşlarını, seçmek için Enter tuşunu kullanın.

Daha fazla bilgi için: https://networkmanager.dev/docs/api/latest/nmtui.html.

- Kullanıcı arayüzünü aç:

  ```
  nmtui
  ```

- Etkinleştirme veya devre dışı bırakma seçeneğiyle birlikte kullanılabilir bağlantıların bir listesini göster:

  ```
  nmtui connect
  ```

- Belirtilen ağa bağlan:

  ```
  nmtui connect {{ad|uuid|aygıt|SSID}}
  ```

- Belirtilen ağı düzenle/ekle/sil:

  ```
  nmtui edit {{ad|kimlik}}
  ```

- Sistem ana makine adını ayarla:

  ```
  nmtui hostname
  ```

pacman-mirrors

Manjaro Linux için pacman aynalistesi oluşturucu.

pacman-mirrors'ın çalıştırıldığı her vakit, E**sudo pacman -Syyu** komutu ile veritabanının senkronize edilmesi ve sistemin güncellenmesi gerekir.

Ayrıca bakınız: **pacman**.

Daha fazla bilgi için: https://wiki.manjaro.org/index.php?title=Pacman-mirrors.

- Varsayılan ayarlar ile bir aynalistesi oluştur:

  ```
  sudo pacman-mirrors --fasttrack
  ```

- Mevcut aynaların durumunu göster:

  ```
  pacman-mirrors --status
  ```

- Mevcut dalı göster:

  ```
  pacman-mirrors --get-branch
  ```

- Farklı bir dala geç:

  ```
  sudo pacman-mirrors --api --set-branch {{stabil|instabil|
  test_ediliyor}}
  ```

- Sadece IP adresinin bulunduğu ülkenin aynalarını kullanarak bir aynalistesi oluştur:

  ```
  sudo pacman-mirrors --geoip
  ```

pacman --query

Arch Linux paket yönetim aracı.

Ayrıca bakınız: **`pacman`**.

Daha fazla bilgi için: https://man.archlinux.org/man/pacman.8.

- Yüklenmiş paket ve sürümleri sırala:

```
pacman --query
```

- Sadece özellikle indirilmiş paket ve sürümleri sırala:

```
pacman --query --explicit
```

- Hangi paketin belirtilen dosyaya sahip olduğunu bul:

```
pacman --query --owns {{dosya_ismi}}
```

- İndirilmiş bir pakete dair bilgiyi görüntüle:

```
pacman --query --info {{paket_ismi}}
```

- Bir paketin içerdiği dosyaları sırala:

```
pacman --query --list {{paket_ismi}}
```

- Yetim (başka bir pakete bağlılık olarak indirilmiş ancak herhangi bir paket tarafından gerektirilmeyen) paketleri sırala:

```
pacman --query --unrequired --deps --quiet
```

- Mevcut depolarda bulunmayan, indirilmiş paketleri sırala:

```
pacman --query --foreign
```

- Miadı dolmuş paketleri sırala:

```
pacman --query --upgrades
```

pacman --remove

Arch Linux paket yönetim aracı.

Ayrıca bakınız: **`pacman`**.

Daha fazla bilgi için: https://man.archlinux.org/man/pacman.8.

- Bir paket ve bağlılıklarını sil:

```
sudo pacman --remove --recursive {{paket_ismi}}
```

- Bir paketi ve onun hem bağlılıklarını, hem de konfigürasyon dosyalarını sil:

```
sudo pacman --remove --recursive --nosave {{paket_ismi}}
```

- Bir paketi telkin olmaksızın sil:

```
sudo pacman --remove --noconfirm {{paket_ismi}}
```

- Yetim (başka bir pakete bağlılık olarak indirilmiş ancak herhangi bir paket tarafından gerektirilmeyen) paketleri sil:

```
sudo pacman --remove --recursive --nosave $(pacman --query --
unrequired --deps --quiet)
```

- Bir paketi ve ona bağlı olan tüm öbür paketleri sil:

```
sudo pacman --remove --cascade {{paket_ismi}}
```

- (Bir paketin silinme durumunda) Etkilenecek paketleri (silmeden) listele:

```
pacman --remove --print {{paket_ismi}}
```

- Bu alt komut için yardım göster:

```
pacman --remove --help
```

pacman --sync

Arch Linux paket yönetim aracı.

Ayrıca bakınız: **`pacman`**.

Daha fazla bilgi için: https://man.archlinux.org/man/pacman.8.

- Yeni bir paket indir:

```
sudo pacman --sync {{paket_ismi}}
```

- Tüm paketleri senkronize et ve güncelle (bahsi geçen paketleri güncellemeden indirmek için `--downloadonly` eki gereklidir):

```
sudo pacman --sync --refresh --sysupgrade
```

- Tüm paketleri güncelle ve telkin olmaksızın yeni bir tane indir:

```
sudo pacman --sync --refresh --sysupgrade --noconfirm
{{paket_ismi}}
```

- Paket veritabanını girilen ifade ile arat:

```
pacman --sync --search "{{arama_şablonu}}"
```

- Bir paket hakkında bilgi görüntüle:

```
pacman --sync --info {{paket_ismi}}
```

- Bir paket güncellemesi sırasında çakışan dosyaların üstüne yaz:

```
sudo pacman --sync --refresh --sysupgrade --overwrite
{{örnek_dosya}}
```

- Tüm paketleri senkronize et ve güncelle, ancak belli bir paketi yoksay:

```
sudo pacman --sync --refresh --sysupgrade --ignore
{{paket_ismi}}
```

- Kullanılmayan paket ve kullanılmamış depoları çerezlerden sil (tüm paketlerin çerezlerini temizlemek için `--clean` eki iki kez kullanılmalıdır):

```
sudo pacman --sync --clean
```

pacman

Arch Linux paket yönetim aracı.

Ayrıca bakınız: **`pacman-database`**, **`pacman-deptest`**, **`pacman-files`**, **`pacman-key`**, **`pacman-mirrors`**, **`pacman-query`**, **`pacman-remove`**, **`pacman-sync`**, **`pacman-upgrade`**.

Daha fazla bilgi için: https://man.archlinux.org/man/pacman.8.

- Tüm paketleri senkronize et ve güncelle:

```
sudo pacman -Syu
```

- Yeni bir paket indir:

```
sudo pacman -S {{paket_ismi}}
```

- Bir paket ve bağlılıklarını sil:

```
sudo pacman -Rs {{paket_ismi}}
```

- Paket veritabanını girilen ifade ile arat:

```
pacman -Ss "{{arama_şablonu}}"
```

- İndirilmiş paket ve sürümleri sırala:

```
pacman -Q
```

- Sadece özellikle belirtilen paket ve sürümleri sırala:

```
pacman -Qe
```

- Hangi paketin belirtilen dosyaya sahip olduğunu bul:

```
pacman -Qo {{dosya_ismi}}
```

- Paket çerezlerini boş alan açmak için temizle:

```
sudo pacman -Scc
```

print

`run-mailcap` komutunun print özelliğinin öbür adı.

Normalde **`run-mailcap`** komutu mime-tarzı/dosya işlemek için kullanılır.

Daha fazla bilgi için: https://manned.org/print.

- Bir dosyayı yazdır:

  ```
  print {{dosya_ismi}}
  ```

- `run-mailcap` ile yazdır:

  ```
  run-mailcap --action=print {{dosya_ismi}}
  ```

pulseaudio

Ses sistem arkaplan uygulaması ve yöneticisi.

Daha fazla bilgi için: https://www.freedesktop.org/wiki/Software/PulseAudio/.

- Pulseaudio'nun çalışıp çalışmadığını kontrol et (sıfır olmayan çıktı, çalışmadığı anlamına gelir):

  ```
  pulseaudio --check
  ```

- Pulseaudio'yu arkaplanda çalıştır:

  ```
  pulseaudio --start
  ```

- Arkaplanda çalışan tüm pulseaudio uygulamalarını öldür:

  ```
  pulseaudio --kill
  ```

- Müsait modülleri sırala:

  ```
  pulseaudio --dump-modules
  ```

- Belirtilen argümanlarla bir modülü mevcut çalışan arkaplan uygulamasına yükle:

  ```
  pulseaudio --load="{{modül_ismi}} {{argümanlar}}"
  ```

pw-cat

Pipewire üzerinden ses dosyalarını çalın ve kaydedin.

Daha fazla bilgi için: https://fedoraproject.org/wiki/QA:Testcase_PipeWire_PipeWire_CLI.

- Varsayılan hedef üzerinden bir WAV dosyası oynat:

```
pw-cat --playback {{örnek/konum/dosya.wav}}
```

- Örnek bir kaydı farklı bir ses seviyesinde kayda al:

```
pw-cat --record --volume={{0.1}} {{örnek/konum/dosya.wav}}
```

- Örnek bir kaydı farklı bir örnek oran kullanarak kayda al:

```
pw-cat --record --rate={{6000}} {{örnek/konum/dosya.wav}}
```

pw-cli

Pipewire komut satır arayüzü.

Daha fazla bilgi için: https://docs.pipewire.org/page_man_pw_cli_1.html.

- Tüm düğümleri (taban ve kaynakları) ID'leri ile birlikte yazdır:

  ```
  pw-cli list-objects Node
  ```

- Belirtilen ID'ye sahip objeye dair bilgileri yazdır:

  ```
  pw-cli info {{4}}
  ```

- Tüm objelerin bilgilerini yazdır:

  ```
  pw-cli info all
  ```

pw-link

PipeWire'daki portlar arası linkleri yönet.

Daha fazla bilgi için: https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Virtual-Devices.

- Tüm ses çıktı ve girdi portlarını sırala:

```
pw-link --output --input'
```

- Çıktı ve girdi portları arasında bir bağlantı yarat:

```
pw-link {{çıktı_port_ismi}} {{girdi_port_ismi}}
```

- Disconnect two ports:

```
pw-link --disconnect {{çıktı_port_ismi}} {{girdi_port_ismi}}
```

- Yardım sayfası göster:

```
pw-link -h
```

pw-loopback

PipeWire'da geri döngü cihazları yaratma aracı.

Daha fazla bilgi için: https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Virtual-Devices.

- Varsayılan geri döngü davranışına sahip bir geri döngü cihazı yarat:

```
pw-loopback
```

- Hoparlörlere otomatik olarak bağlanan bir geri döngü cihazı yarat:

```
pw-loopback -m '{{[FL FR]}}' --capture-
props='{{media.class=Audio/Sink}}'
```

- Mikrofona otomatik olarak bağlanan bir geri döngü cihazı yarat:

```
pw-loopback -m '{{[FL FR]}}' --playback-
props='{{media.class=Audio/Source}}'
```

- Hiçbir şeye otomatik olarak bağlanmayan salak bir geri döngü cihazı yarat:

```
pw-loopback -m '{{[FL FR]}}' --capture-
props='{{media.class=Audio/Sink}}' --playback-
props='{{media.class=Audio/Source}}'
```

- Hoparlörlere otomatik olarak bağlanan ve taban-kaynak arasında sağ-sol kanalların yerini değiştiren bir geri döngü cihazı yarat:

```
pw-loopback --capture-props='{{media.class=Audio/Sink
audio.position=[FL FR]}}' --playback-
props='{{audio.position=[FR FL]}}'
```

- Mikrofona otomatik olarak bağlanan ve taban-kaynak arasında sağ-sol kanalların yerini değiştiren bir geri döngü cihazı yarat:

```
pw-loopback --capture-props='{{audio.position=[FR FL]}}' --
playback-props='{{media.class=Audio/Source audio.position=[FL
FR]}}'
```

pw-play

pw-cat --playback komutu için kısayol aracı.

Daha fazla bilgi için: https://fedoraproject.org/wiki/QA:Testcase_PipeWire_PipeWire_CLI.

- Tüm erişilebilir oynatma hedeflerini sırala:

  ```
  pw-play --list-targets
  ```

- Varsayılan hedef üzerinden bir WAV sesi oynat:

  ```
  pw-play {{örnek/konum/dosya.wav}}
  ```

- WAV sesini farklı bir ses yüksekliğinde oynat:

  ```
  pw-play --volume={{0.1}} {{örnek/konum/dosya.wav}}
  ```

pw-record

Pipewire aracılığıyla ses dosyalarını kaydedin.

pw-cat --record için kısaltma.

Daha fazla bilgi için: https://fedoraproject.org/wiki/QA:Testcase_PipeWire_PipeWire_CLI.

- Varsayılan hedefi kullanarak örnek bir ses kaydı al:

```
pw-record {{örnek/konum/dosya.wav}}
```

- Farklı bir ses seviyesinde örnek ses kaydı al:

```
pw-record --volume={{0.1}} {{örnek/konum/dosya.wav}}
```

- Farklı bir kayıt oranı kullanarak örnek ses kaydı al:

```
pw-record --rate={{6000}} {{örnek/konum/dosya.wav}}
```

pwd

Mevcut/çalışan dizinin ismini yazdır.

Daha fazla bilgi için: https://www.gnu.org/software/coreutils/pwd.

- Mevcut dizini yazdır:

  ```
  pwd
  ```

- Mevcut dizini yazdır ve tüm symlink'leri çöz (yani "fiziksel" yolu göster):

  ```
  pwd --physical
  ```

- Mevcut mantıksal dizini yazdır:

  ```
  pwd --logical
  ```

pwdx

Bir işlemin çalışan dizinini yazdır.

Daha fazla bilgi için: https://manned.org/pwdx.

- Bir işlemin mevcut çalışan dizinini yazdır:

```
pwdx {{işlem_id'si}}
```

trash

Çöp / geri dönüşüm kutusunu yönetmek için bir komut satırı arayüzü.

Daha fazla bilgi için: https://github.com/andreafrancia/trash-cli.

- Bir dosyayı sil (çöpe gönder):

```
trash {{örnek/dosya}}
```

- Çöpteki dosyaları göster:

```
trash-list
```

- Çöpteki dosyaları geri getir:

```
trash-restore
```

- Çöpü boşalt:

```
trash-empty
```

- Çöpü 10 gün öncesinden daha yeni atılan dosyalar hariç boşalt:

```
trash-empty {{10}}
```

- Çöpte 'foo' ismini taşıyan tüm dosyaları sil:

```
trash-rm foo
```

- Belirtilen konumdaki tüm dosyaları sil:

```
trash-rm {{/detaylı/örnek/konum/dosya_veya_dizin}}
```

ubuntu-bug

Bu komut **`apport-bug`** için bir takma addır.

Daha fazla bilgi için: https://manned.org/ubuntu-bug.

- Asıl komutun belgelerini görüntüleyin:

```
tldr apport-bug
```

wl-copy

Wayland için pano değiştirme aracı.

Ayrıca bakın: **`wl-paste`**.

Daha fazla bilgi için: https://github.com/bugaevc/wl-clipboard.

- Metni panoya kopyala:

```
wl-copy {{metin}}
```

- Komutun çıktısını panoya kopyala:

```
{{komut}} | wl-copy
```

- Yalnızca bir yapıştırma için kopyala ve ardından temizle:

```
wl-copy --paste-once
```

- Panoyu temizle:

```
wl-copy --clear
```

wl-paste

Wayland için panoda saklanan verilere erişim aracı.

Ayrıca bakın: **`wl-copy`**.

Daha fazla bilgi için: https://github.com/bugaevc/wl-clipboard.

- Panonun içeriğini yapıştır:

  ```
  wl-paste
  ```

- Panonun içeriğini bir dosyaya yaz:

  ```
  wl-paste > {{dosya/yolu}}
  ```

- Panonun içeriğini bir komuta aktar:

  ```
  wl-paste | {{komut}}
  ```

wpa_cli

Kablosuz LAN arayüzleri ekleyin ve yapılandırın.

Daha fazla bilgi için: https://manned.org/wpa_cli.

- Kullanılabilir ağları tara:

  ```
  wpa_cli scan
  ```

- Tarama sonuçlarını göster:

  ```
  wpa_cli scan_results
  ```

- Ağ ekle:

  ```
  wpa_cli add_network {{numara}}
  ```

- Bir ağın SSID değerini ayarla:

  ```
  wpa_cli set_network {{numara}} ssid "{{SSID}}"
  ```

- Ağı etkinleştir:

  ```
  wpa_cli enable_network {{numara}}
  ```

- Yapılandırmayı kaydet:

  ```
  wpa_cli save_config
  ```

wpa_passphrase

Belirtilen SSID için bir ASCII paroladan bir WPA-PSK anahtarı oluşturun.

Daha fazla bilgi için: https://manned.org/wpa_passphrase.1.

- Standart girişten parolayı okuyarak belirtilen SSID için WPA-PSK anahtarını hesapla ve görüntüle:

  ```
  wpa_passphrase {{SSID}}
  ```

- Parolayı argüman olarak belirterek belirtilen SSID için WPA-PSK anahtarını hesapla ve görüntüle:

  ```
  wpa_passphrase {{SSID}} {{parola}}
  ```

xeyes

Ekranda fare imlecini takip eden bir çift göz göster.

Daha fazla bilgi için: https://manned.org/xeyes.

- Xeyes'ı yerel makinenin varsayılan ekranında başlat:

  ```
  xeyes
  ```

- Xeyes'ı uzak bir makinenin 0. görüntü ve 0. ekran koordinatlarında başlat:

  ```
  xeyes -display {{uzak_makine}}:{{0}}.{{0}}
  ```

xfce4-screenshooter

XFCE4 ekran görüntüsü aracı.

Daha fazla bilgi için: https://docs.xfce.org/apps/xfce4-screenshooter/start.

- Ekran görüntüsü alma grafik arayüzünü başlat:

```
xfce4-screenshooter
```

- Tüm ekranın ekran görüntüsünü al ve nasıl devam edileceğini belirlemek adına grafik arayüzünü başlat:

```
xfce4-screenshooter --fullscreen
```

- Tüm ekranın ekran görüntüsünü al ve görüntüyü belirtilen dizine kaydet:

```
xfce4-screenshooter --fullscreen --save {{örnek/dizin}}
```

- Ekran görüntüsünü çekmeden önce belli bir süre bekle:

```
xfce4-screenshooter --delay {{saniye_miktarı}}
```

- Ekranın (fare ile seçilecek) belli bir bölümünün görüntüsünü al:

```
xfce4-screenshooter --region
```

- Üzerinde bulunulan pencerenin görüntüsünü al ve panoya kopyala:

```
xfce4-screenshooter --window --clipboard
```

- Üzerinde bulunulan pencerenin görüntüsünü qal ve seçilen bir program ile aç:

```
xfce4-screenshooter --window --open {{gimp}}
```

xfce4-terminal

XFCE4 terminal öykünücüsü.

Daha fazla bilgi için: https://docs.xfce.org/apps/xfce4-terminal/start.

- Yeni bir terminal penceresi aç:

```
xfce4-terminal
```

- Başlangıç başlığı belirle:

```
xfce4-terminal --initial-title "{{başlangıç_başlığı}}"
```

- Mevcut terminal penceresinde yeni bir sekme aç:

```
xfce4-terminal --tab
```

- Yeni bir terminal penceresini belirlenen bir komutu çalıştırarak aç:

```
xfce4-terminal --command "{{argümanlı_komut}}"
```

- Çalıştırılan komutun çalışmayı kesme durumunda dahi terminali kapama:

```
xfce4-terminal --command "{{argümanlı_komut}}" --hold
```

- Her birinde farklı komut çalışacak birçok yeni sekme aç:

```
xfce4-terminal --tab --command "{{komut_a}}" --tab --command
"{{komut_b}}"
```

xrandr

Bir ekran için boyut, yön ve/veya çıkış yansımasını ayarla.

Daha fazla bilgi için: https://www.x.org/releases/current/doc/man/man1/xrandr.1.xhtml.

- Sistemin mevcut durumunu göster (bilinen ekranlar, çözünürlükler, ...):

```
xrandr --query
```

- Bağlantısı kesilmiş çıkışları devre dışı bırak ve bağlanmış olanları varsayılan ayarlar ile devreye sok:

```
xrandr --auto
```

- DisplayPort 1'in çözünürlük ve yenileme hızını 1920x1080, 60Hz olarak değiştir:

```
xrandr --output {{DP1}} --mode {{1920x1080}} --rate {{60}}
```

- HDMI2'nin çözünürlüğünü 1280x1024'e değiştirip, DP1'in sağına koy:

```
xrandr --output {{HDMI2}} --mode {{1280x1024}} --right-of {{DP1}}
```

- VGA1 çıkışını devre dışı bırak:

```
xrandr --output {{VGA1}} --off
```

- LVDS1 için parlaklığı 50% yap:

```
xrandr --output {{LVDS1}} --brightness {{0.5}}
```

xterm

X Ekran Sistemi için terminal öykünücüsü.

Daha fazla bilgi için: https://manned.org/xterm.

- `Örnek` başlığına sahip bir terminal aç:

  ```
  xterm -T {{Örnek}}
  ```

- Terminali tam ekran modunda aç:

  ```
  xterm -fullscreen
  ```

- Terminali lacivert arkaplan ve sarı ön plan (font rengi) ile aç:

  ```
  xterm -bg {{darkblue}} -fg {{yellow}}
  ```

- Terminali satır başına 100 karakter ve sütun başına 35 satır sığacak şekilde, x=200px y=20px koordinatlarında aç:

  ```
  xterm -geometry {{100}}x{{35}}+{{200}}+{{20}}
  ```

- Terminali bir Serif fontu ve 20'ye eşit olan bir font büyüklüğü ile aç:

  ```
  xterm -fa "{{Serif}}" -fs {{20}}
  ```

Osx

aa

Bu komut **yaa** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr yaa
```

g[

Bu komut **`-p linux [`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux [
```

gawk

Bu komut **`-p linux awk`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux awk
```

gb2sum

Bu komut **`-p linux b2sum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux b2sum
```

gbase32

Bu komut **`-p linux base32`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux base32
```

gbase64

Bu komut **`-p linux base64`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux base64
```

gbasename

Bu komut **`-p linux basename`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux basename
```

gbasenc

Bu komut **`-p linux basenc`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux basenc
```

gcat

Bu komut **`-p linux cat`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux cat
```

gchcon

Bu komut **`-p linux chcon`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux chcon
```

gchgrp

Bu komut **`-p linux chgrp`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux chgrp
```

gchmod

Bu komut **`-p linux chmod`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux chmod
```

gchown

Bu komut **`-p linux chown`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux chown
```

gchroot

Bu komut **`-p linux chroot`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux chroot
```

gcksum

Bu komut **`-p linux cksum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux cksum
```

gcomm

Bu komut **`-p linux comm`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux comm
```

gcp

Bu komut **`-p linux cp`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux cp
```

gcsplit

Bu komut **`-p linux csplit`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux csplit
```

gcut

Bu komut **`-p linux cut`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux cut
```

gdate

Bu komut **`-p linux date`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux date
```

gdd

Bu komut **`-p linux dd`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux dd
```

gdf

Bu komut **`-p linux df`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux df
```

gdir

Bu komut **`-p linux dir`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux dir
```

gdircolors

Bu komut **`-p linux dircolors`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux dircolors
```

gdirname

Bu komut **`-p linux dirname`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux dirname
```

gdnsdomainname

Bu komut **`-p linux dnsdomainname`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux dnsdomainname
```

gecho

Bu komut **`-p linux echo`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux echo
```

ged

Bu komut **`-p linux ed`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux ed
```

gegrep

Bu komut **`-p linux egrep`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux egrep
```

genv

Bu komut **`-p linux env`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux env
```

gexpand

Bu komut **`-p linux expand`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux expand
```

gexpr

Bu komut **`-p linux expr`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux expr
```

gfactor

Bu komut **`-p linux factor`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux factor
```

gfalse

Bu komut **`-p linux false`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux false
```

gfgrep

Bu komut **`-p linux fgrep`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux fgrep
```

gfind

Bu komut **`-p linux find`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux find
```

gfmt

Bu komut **`-p linux fmt`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux fmt
```

gfold

Bu komut **`-p linux fold`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux fold
```

gftp

Bu komut **`-p linux ftp`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux ftp
```

ggrep

Bu komut **`-p linux grep`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux grep
```

ggroups

Bu komut **`-p linux groups`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux groups
```

ghead

Bu komut **`-p linux head`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux head
```

ghostid

Bu komut **`-p linux hostid`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux hostid
```

ghostname

Bu komut **`-p linux hostname`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux hostname
```

gid

Bu komut **`-p linux id`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux id
```

gifconfig

Bu komut **`-p linux ifconfig`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux ifconfig
```

gindent

Bu komut **`-p linux indent`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux indent
```

ginstall

Bu komut **`-p linux install`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux install
```

gjoin

Bu komut **`-p linux join`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux join
```

gkill

Bu komut **`-p linux kill`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux kill
```

glibtool

Bu komut **`-p linux libtool`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux libtool
```

glibtoolize

Bu komut **`-p linux libtoolize`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux libtoolize
```

glink

Bu komut **`-p linux link`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux link
```

gln

Bu komut **`-p linux ln`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux ln
```

glocate

Bu komut **`-p linux locate`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux locate
```

glogger

Bu komut **`-p linux logger`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux logger
```

glogname

Bu komut **`-p linux logname`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux logname
```

gls

Bu komut **`-p linux ls`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux ls
```

gmake

Bu komut **`-p linux make`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux make
```

gmd5sum

Bu komut **`-p linux md5sum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux md5sum
```

gmkdir

Bu komut **`-p linux mkdir`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux mkdir
```

gmkfifo

Bu komut **`-p linux mkfifo`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux mkfifo
```

gmknod

Bu komut **`-p linux mknod`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux mknod
```

gmktemp

Bu komut **`-p linux mktemp`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux mktemp
```

gmv

Bu komut **`-p linux mv`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux mv
```

gnice

Bu komut **`-p linux nice`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux nice
```

gnl

Bu komut **`-p linux nl`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux nl
```

gnohup

Bu komut **`-p linux nohup`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux nohup
```

gnproc

Bu komut **`-p linux nproc`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr -p linux nproc
  ```

gnumfmt

Bu komut **`-p linux numfmt`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux numfmt
```

god

Bu komut **`-p linux od`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux od
```

gpaste

Bu komut **`-p linux paste`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux paste
```

gpathchk

Bu komut **`-p linux pathchk`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux pathchk
```

gping

Bu komut **`-p linux ping`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux ping
```

gping6

Bu komut **`-p linux ping6`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux ping6
```

gpinky

Bu komut **`-p linux pinky`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux pinky
```

gpr

Bu komut **`-p linux pr`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux pr
```

gprintenv

Bu komut **`-p linux printenv`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux printenv
```

gprintf

Bu komut **`-p linux printf`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux printf
```

gptx

Bu komut **`-p linux ptx`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux ptx
```

gpwd

Bu komut **`-p linux pwd`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux pwd
```

grcp

Bu komut **`-p linux rcp`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux rcp
```

greadlink

Bu komut **`-p linux readlink`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux readlink
```

grealpath

Bu komut **`-p linux realpath`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux realpath
```

grexec

Bu komut **`-p linux rexec`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux rexec
```

grlogin

Bu komut **`-p linux rlogin`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux rlogin
```

grm

Bu komut **`-p linux rm`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux rm
```

grmdir

Bu komut **`-p linux rmdir`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux rmdir
```

grsh

Bu komut **`-p linux rsh`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux rsh
```

gruncon

Bu komut **`-p linux runcon`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux runcon
```

gsed

Bu komut **`-p linux sed`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sed
```

gseq

Bu komut **`-p linux seq`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux seq
```

gsha1sum

Bu komut **`-p linux sha1sum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sha1sum
```

gsha224sum

Bu komut **`-p linux sha224sum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sha224sum
```

gsha256sum

Bu komut **`-p linux sha256sum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sha256sum
```

gsha384sum

Bu komut **`-p linux sha384sum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sha384sum
```

gsha512sum

Bu komut **`-p linux sha512sum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sha512sum
```

gshred

Bu komut **`-p linux shred`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux shred
```

gshuf

Bu komut **`-p linux shuf`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux shuf
```

gsleep

Bu komut **`-p linux sleep`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sleep
```

gsort

Bu komut **`-p linux sort`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sort
```

gsplit

Bu komut **`-p linux split`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux split
```

gstat

Bu komut **`-p linux stat`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux stat
```

gstdbuf

Bu komut **`-p linux stdbuf`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux stdbuf
```

gstty

Bu komut **`-p linux stty`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux stty
```

gsum

Bu komut **`-p linux sum`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sum
```

gsync

Bu komut **`-p linux sync`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux sync
```

gtac

Bu komut **`-p linux tac`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux tac
```

gtail

Bu komut **`-p linux tail`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux tail
```

gtalk

Bu komut **`-p linux talk`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux talk
```

gtar

Bu komut **`-p linux tar`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux tar
```

gtee

Bu komut **`-p linux tee`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux tee
```

gtelnet

Bu komut **`-p linux telnet`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux telnet
```

gtest

Bu komut **`-p linux test`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux test
```

gtftp

Bu komut **`-p linux tftp`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux tftp
```

gtime

Bu komut **`-p linux time`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux time
```

gtimeout

Bu komut **`-p linux timeout`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux timeout
```

gtouch

Bu komut **`-p linux touch`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux touch
```

gtr

Bu komut **`-p linux tr`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux tr
```

gtraceroute

Bu komut **`-p linux traceroute`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux traceroute
```

gtrue

Bu komut **`-p linux true`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux true
```

gtruncate

Bu komut **`-p linux truncate`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux truncate
```

gtsort

Bu komut **`-p linux tsort`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux tsort
```

gtty

Bu komut **`-p linux tty`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux tty
```

guname

Bu komut **`-p linux uname`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux uname
```

gunexpand

Bu komut **`-p linux unexpand`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux unexpand
```

guniq

Bu komut **`-p linux uniq`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux uniq
```

gunits

Bu komut **`-p linux units`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux units
```

gunlink

Bu komut **`-p linux unlink`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux unlink
```

gupdatedb

Bu komut **`-p linux updatedb`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux updatedb
```

guptime

Bu komut **`-p linux uptime`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux uptime
```

gusers

Bu komut **`-p linux users`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux users
```

gvdir

Bu komut **`-p linux vdir`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux vdir
```

gwc

Bu komut **`-p linux wc`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux wc
```

gwhich

Bu komut **`-p linux which`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux which
```

gwho

Bu komut **`-p linux who`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux who
```

gwhoami

Bu komut **`-p linux whoami`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux whoami
```

gwhois

Bu komut **`-p linux whois`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux whois
```

gxargs

Bu komut **`-p linux xargs`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux xargs
```

gyes

Bu komut **`-p linux yes`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr -p linux yes
```

launchd

Bu komut **`launchctl`** için bir takma addır.

Daha fazla bilgi için: https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/Introduction.html.

- Asıl komutun belgelerini görüntüleyin:

```
tldr launchctl
```

Sunos

devfsadm

`/dev` için yönetim komutu. **`/dev`** ad alanına yönetir.

Daha fazla bilgi için: https://www.unix.com/man-page/sunos/1m/devfsadm.

- Yeni disk ara:

  ```
  devfsadm -c disk
  ```

- Sarkab /dev bağlantılarını temizle ve yeni bir cihaz ara:

  ```
  devfsadm -C -v
  ```

- Komut çalıştırılacağı takdirde ne olacağını gör ancak herhangi bir düzenleme yapma:

  ```
  devfsadm -C -v -n
  ```

dmesg

Kernel mesajlarını görüntüle.

Daha fazla bilgi için: https://www.unix.com/man-page/sunos/1m/dmesg.

- Kernel mesajlarını görüntüle:

  ```
  dmesg
  ```

- Sistemde ne kadar fiziksel hafıza kaldığını göster:

  ```
  dmesg | grep -i memory
  ```

- Kernel mesajlarını terminal ekranına sığacak ve her satıra bir tane gelecek şekilde göster:

  ```
  dmesg | less
  ```

prctl

Çalışan işlemlerin, görevlerin ve projelerin kaynak kontrollerini öğren veya belirle.

Daha fazla bilgi için: https://www.unix.com/man-page/sunos/1/prctl.

- Belirtilen işlemin limit ve izinlerini incele:

  ```
  prctl {{PID}}
  ```

- İşlem limit ve izinlerini makineye dayanıklı fortmattaExamine process limits and permissions in machine parsable format:

  ```
  prctl -P {{PID}}
  ```

- Çalışan işlem için belirtilen limiti öğren:

  ```
  prctl -n process.max-file-descriptor {{PID}}
  ```

prstat

Aktif işlem istatistiklerini bildir.

Daha fazla bilgi için: https://www.unix.com/man-page/sunos/1m/prstat.

- CPU kullanımına ayrılan tüm işlem ve raporların istatiğini incele:

  ```
  prstat
  ```

- Hafıza kullanımına ayrılan tüm işlem ve raporların istatistiğini incele:

  ```
  prstat -s rss
  ```

- Her bir kullanıcı için toplam kullanım özetini bildir:

  ```
  prstat -t
  ```

- Mikrodurum işlem hesap açıklama bilgisini bildir:

  ```
  prstat -m
  ```

- Saniye başı en çok CPU kullanan 5 işlemin listesini yazdır:

  ```
  prstat -c -n 5 -s cpu 1
  ```

snoop

Ağ paketi inceleyici.

SunOS'in tcpdump alternatifi.

Daha fazla bilgi için: https://www.unix.com/man-page/sunos/1m/snoop.

- Belirtilen ağ arayüzünde paketleri yakala:

```
snoop -d {{e1000g0}}
```

- Yakalanan paketleri terminalde göstermek yerine bir dosyaya kaydet:

```
snoop -o {{dosyaismi}}
```

- Belirtilen dosyadan paketlerin ayrıntılı protokol katman özetini görüntüle:

```
snoop -V -i {{dosyaismi}}
```

- Host isminden gelen ağ paketlerini yakala ve belirtilen port'a git:

```
snoop to port {{port}} from host {{hostismi}}
```

- İki IP adresi arasında takas edilen ağ paketleriini yakala ve hex değerlerini göster:

```
snoop -x0 -p4 {{ip_addresi_1}} {{ip_addresi_2}}
```

svcadm

Servisleri idare et.

Daha fazla bilgi için: https://www.unix.com/man-page/linux/1m/svcadm.

- Servis veritabanındaki bir servisi etkinleştir:

  ```
  svcadm enable {{servis_ismi}}
  ```

- Servisi devre dışı bırak:

  ```
  svcadm disable {{servis_ismi}}
  ```

- Çalışan bir servisi yeniden başlat:

  ```
  svcadm restart {{servis_ismi}}
  ```

- Servise yapulandırma dosyalarını yeniden okumasını emret:

  ```
  svcadm refresh {{servis_ismi}}
  ```

- Bir servisi bakım durumundan çıkar ve başlamasını emret:

  ```
  svcadm clear {{servis_ismi}}
  ```

svccfg

Servis yapılandırmalarını içe aktar, dışa aktar ve düzenle.

Daha fazla bilgi için: https://www.unix.com/man-page/linux/1m/svccfg.

- Yapılandırma dosyasını değerlendir:

  ```
  svccfg validate {{smf.xml}}
  ```

- Servis yapılandırma dosyalarını belirtilen dosyaya yazılacak şekilde dışa aktar:

  ```
  svccfg export {{servisismi}} > {{smf.xml}}
  ```

- Dosyadan servis yapılandırmalarını içe aktar/güncelle:

  ```
  svccfg import {{smf.xml}}
  ```

SVCS

Çalışan servislere dair bilgileri sırala.

Daha fazla bilgi için: https://www.unix.com/man-page/linux/1/svcs.

- Tüm çalışan servisleri sırala:

```
svcs
```

- Çalışmayan servisleri sırala:

```
svcs -vx
```

- Belirtilen servise dair bilgileri sırala:

```
svcs apache
```

- Servis kayıt dosyasının yerini göster:

```
svcs -L apache
```

- Servis kayıt dosyasının sonunu görüntüle:

```
tail $(svcs -L apache)
```

truss

İzleme sistem çağrıları için sorun giderme aracı.

SunOS'in strace alternatifi.

Daha fazla bilgi için: https://www.unix.com/man-page/linux/1/truss.

- Bir programı tüm alt işlemleriyle beraber çalıştırarak başlat:

```
truss -f {{program}}
```

- Belirtilen işlemi onun PID değerini girerek izlemeye başla:

```
truss -p {{pid}}
```

- Bir programı argümanları ve çevresel değerlerini göstererek başlar:

```
truss -a -e {{program}}
```

- Her bir sistem çağrısı için zaman, çağrı ve hataları say ve program çıkışında bunların özetini bildir:

```
truss -c -p {{pid}}
```

- Bir işlemi onun çıktısını sistem çağrısıyla süzerek izle:

```
truss -p {{pid}} -t {{system_çağrısı_ismi}}
```

Windows

chrome

Bu komut **`chromium`** için bir takma addır.

Daha fazla bilgi için: https://chrome.google.com.

- Asıl komutun belgelerini görüntüleyin:

```
tldr chromium
```

cinst

Bu komut **`choco install`** için bir takma addır.

Daha fazla bilgi için: https://docs.chocolatey.org/en-us/choco/commands/install.

- Asıl komutun belgelerini görüntüleyin:

  ```
  tldr choco install
  ```

clist

Bu komut **`choco list`** için bir takma addır.

Daha fazla bilgi için: https://docs.chocolatey.org/en-us/choco/commands/list.

- Asıl komutun belgelerini görüntüleyin:

```
tldr choco list
```

cpush

Bu komut **`choco-push`** için bir takma addır.

Daha fazla bilgi için: https://docs.chocolatey.org/en-us/create/commands/push.

- Asıl komutun belgelerini görüntüleyin:

```
tldr choco-push
```

cuninst

Bu komut **`choco uninstall`** için bir takma addır.

Daha fazla bilgi için: https://docs.chocolatey.org/en-us/choco/commands/uninstall.

- Asıl komutun belgelerini görüntüleyin:

```
tldr choco uninstall
```

curl

Bu komut **`curl -p common`** için bir takma addır.

Daha fazla bilgi için: https://curl.se.

- Asıl komutun belgelerini görüntüleyin:

```
tldr curl -p common
```

iwr

Bu komut **`invoke-webrequest`** için bir takma addır.

- Asıl komutun belgelerini görüntüleyin:

```
tldr invoke-webrequest
```

pwsh-where

Bu komut **`Where-Object`** için bir takma addır.

Daha fazla bilgi için: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/where-object.

- Asıl komutun belgelerini görüntüleyin:

```
tldr Where-Object
```

rd

Bu komut **`rmdir`** için bir takma addır.

Daha fazla bilgi için: https://learn.microsoft.com/windows-server/administration/windows-commands/rd.

- Asıl komutun belgelerini görüntüleyin:

```
tldr rmdir
```

sls

Bu komut **`where-object`** için bir takma addır.

Daha fazla bilgi için: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/select-string.

- Asıl komutun belgelerini görüntüleyin:

```
tldr where-object
```

wget

Bu komut **`wget -p common`** için bir takma addır.

Daha fazla bilgi için: https://www.gnu.org/software/wget.

- Asıl komutun belgelerini görüntüleyin:

```
tldr wget -p common
```