دستورات قابل استفاده در گیت

نصب گیت و تنظیمات اولیه در مخزن محلی

```
$ sudo  yum install git
$ git  –version
```

```
$ git config --global user.name "user git"
$ git config --global user.email "email git"
```

ساخت پوشه گیت

```
$ mkdir myproject
$  cd myproject

# initialize Git
$ git init
```

بررسی وضعیت repository

```
$ git status
```

با استفاده از –short تغییرات را فشرده تر به ما می دهد.

اضافه کردن فایل و کامیت

```
$ git add  “file or dir”
```

از پارامتر –all و -A برای اضافه کردن تمام فایل های پوشه گیت استفاده می شود.

اضافه کردن commit .پیشرفت و تغییرات ما را در حین کار پیگیری می کند. Git هر نقطه تغییر commit یا "save point" را در نظر می گیرد. این نقطه ای از پروژه است که اگر باگ پیدا شد یا می خواهید تغییری ایجاد کنید، می توانید به آن بازگردید.

```
$ git commit  -m “message”
```

منظور -m پیامی هست که همیشه باید همراه کامیت باشد کامیت ها با پیام ها از هم متمایز می شوند.
گاهی اوقات تغییرات کوچکی اعمال می شود و و می توان به صورت مستقیم تغییرات را اعمال کنیم برای این کار از -a استفاده میکنیم

```
$ git  commit -a  -m  “update new line file”
```

برای مشخص شدن تمام کامیت ها در یک مخزن می توان از مخزن log گرفت

```
$ git log
```

ساخت شاخه (branch)

فرض کنید یک پروژه بزرگ دارید که دارای بخش های و قسمت های مختلف می باشد و می خواهید هر طرح و قسمت را جداگانه به روز رسانی کنید برای همین گیت با استفاده از برن چ این امکان را به ما می دهد.
مزیت استفاده از branch در گیت

- با یک شاخه جدید به نام new-design، کد را مستقیماً بدون تاثیر بر شاخه اصلی ویرایش کنید
- رفع خطا در شاخه ی دیگر و ادغام آن با شاخه اصلی
- ایجاد طراحی جدید و کار روی ان و ادغام با شاخه اصلی

```
$ git  branch -M namebranche
```

برای استفاده به عنوان main یا پیشفرض از -m استفاده میکنیم.
برای switch کردن در branch های مختلف از دستور زیر استفاده می شود.

```
$ git checkout namebranch
```

برای ایجاد و انتقال به یک branch جدید از -b استفاده می شود

ادغام branch

```
$ git checkout master
$ git merg namebranchmerg
```

بعد از ادغام کردن در در صورت نیاز میتوان branch را حذف کرد

```
$ git  branch -d namebranch
```

زمانی که بین دو branch یک فایل در تضاد باشد در زمان ادغام کردن conflict رخ می دهد پس باید ابتدا فایل مدنظر کامیت کرده و سپس ادغام شود

آشنایی با github و اضافه کردن مخزن local به server

بعد از ساختن Repository و گرفتن ادرس ان باید آدرس را به git در مرحله قبل ساخته ایم اضافه کنیم

```
$ git remote add origin url
```

برای اضافه کردن به صورت زیر عمل میکنیم

```
$ git  push origin master
```

گرفتن پروژه از github و آپدیت نگه داشتن

هنگام کار به عنوان تیم روی یک پروژه، مهم است که همه به روز بمانند.هر زمان که شروع به کار روی یک پروژه می کنید، باید جدیدترین تغییرات را در نسخه محلی خود دریافت کنید.
دستور pull ترکیبی از دو دستور مختلف می باشد fetch , merg که می تواند پروژه را از منبع بگیرد.
دستور fetch تمام تاریخچه تغییرات یک شعبه/ مخزن ردیابی شده را دریافت می کند.بنابراین، در Git محلی خود، به‌روزرسانی‌ها را واکشی کنید تا ببینید چه چیزی در GitHub تغییر کرده است:

```
$ git fetch origin
```

با این دستور می توان آخرین تغییرات را داشته باشید . همچنین می توان تغییرات منابع سرور و محلی را نمایش داد
$ git diff origin/master

دستور merge شاخه فعلی را با یک شاخه مشخص ترکیب می کند.با fetch تأیید کرده‌ایم که به‌روزرسانی‌ها مطابق انتظار هستند، و می‌تواند شاخه فعلی (مستر) خود را با مبدا/مستر ادغام کنیم:

$ git merg origin/master

اما اگر فقط بخواهید مخزن محلی خود را بدون گذراندن تمام آن مراحل به روز کنید از دستور pull استفاده می کنیم
$ git pull origin

مشارکت در پروژه شخصی دیگران

یک fork یک کپی از یک مخزن است. ,زمانی مفید است که می خواهید در پروژه شخص دیگری مشارکت کنید یا پروژه خود را بر اساس پروژه او شروع کنید. fork یک فرمان در Git نیست، بلکه چیزی است که در GitHub و سایر میزبان های مخزن ارائه می شود. l

$ git clone url.io

بعد از گرفتن یک پروژه و ایجاد تغییرات روی یک پروژه به صورت مشارکتی می توان آن را push نمود
$ git push origin