

Git Advanced

Working on a new feature

When you want to start a new feature.
You have to create a new branch to work from:

```
(master) git status (CLEAN)

(master) git checkout -b my-feature
```

You've just finished you feature! Now it's time to make a last commit, create a pull request and go back to master branch:

```
(my-feature) git add .

(my-feature) git commit -m 'XXXX'

(my-feature) git push origin my-feature


(my-feature) git status (CLEAN)


(my-feature) gh repo view --web

# GO CREATE A PULL REQUEST ON GITHUB


(my-feature) git checkout master

(master) git pull origin master

(master) git sweep


(master) git checkout -b my-new-feature
```

Getting latest changes from master

When you want to get the latest changes from master on your branch:

```
(my-feature) git add .  
  
(my-feature) git commit -m 'XXXX'  
  
(my-feature) git status (CLEAN)  
  
(my-feature) git checkout master  
  
(master) git pull origin master  
  
(master) git checkout my-feature  
  
(my-feature) git merge master
```

Solving conflicts

When you can't merge your Pull Request due to conflicts.

⚠️ GitHub proposes a web interface to resolve conflicts on Pull Requests. First, try to use their feature. It might ease the process.

Otherwise, follow this process:

```
(my-feature) git status (CLEAN)  
  
(my-feature) git checkout master  
  
(master) git pull origin master  
  
(master) git checkout my-feature  
  
(my-feature) git merge master
```

NOW IT'S TIME TO SOLVE ALL THE CONFLICTS

OPEN SUBLIME TEXT AND FIX THEM: locate them with cmd + shift + f <<<<<<

```
(my-feature) git add .
```

```
(my-feature) git commit -m "Fix conflicts"
```

```
(my-feature) git push origin my-feature
```

```
hub browse
```

GO ON GITHUB, NOW THE PULL REQUEST CAN BE MERGED

```
(my-feature) git checkout master
```

```
(master) git pull origin master
```

```
(master) git sweep
```

```
(master) git checkout -b my-new-feature
```

Fix an accidental commit to master

⚠ This operation can be done ONLY if you HAVEN'T PUSHED to GitHub yet!

When you accidentally committed some changes to `master` and want to move them to a new branch.

You have to create a new branch for your feature. This branch will contain your commit. Finally, you'll have to remove your commit from master.

Here are the steps:

```
(master) git checkout -b my-feature
```

```
(my-feature) git checkout master
```

```
(master) git reset HEAD~ --hard
```

```
(master) git checkout my-feature
```

YOU CAN PUSH TO GITHUB OR CONTINUE TO WORK ON YOUR FEATURE

Sharing/Synching your Repository

Ultimately we want to share our code, normally by synching it to a central repository.

For doing so, we have to add a remote.

```
git remote add origin git@github.com:username/existing_project.git
```

You can list your remotes with:

```
git remote -v
```

The real powerful thing is that you can add multiple different remotes. This is often used in combination with cloud hosting solutions for deploying your code on your server. For instance, you could add a remote named “heroku” which points to your cloud hosting server repository.