

## Github로 팀 협업하기

1. 팀장이 팀협업을 위한 git repository 생성하고 master 브랜치에 팀 프로젝트 코드 올림.  
브랜치를 생성하지 않으면 master로 자동 올라감

master는 Project 최종 브랜치

```
git init
git config --global --add safe.directory D:/ai_x/lecNote/11_1stProject
git add .
git commit -m "first commit"
git remote add origin git 주소
git push -u origin master
```

2. 팀원들과 git공유를 위해, settings 메뉴를 통해 **Add people** 로 팀원 초대

팀원은 초대 수락(메일확인) **View invitation** **Accept invitation**

3. 팀원들이 Github에서 소스코드 다운로드

```
> git clone git주소 폴더이름
```

주소는 깃허브의 **Code**버튼에서 복사

폴더이름 생략가능. 폴더이름이 있을 경우, 그 폴더가 생성되고 그 안에 코드들이 다운로드되고, 폴더이름이 생략될 경우, 깃허브 레파지토리 이름으로 폴더가 자동생성되고, 그 안에 코드들이 다운로드된다.

4. Github에서 내 브랜치(branch) 만들기

```
> git checkout -b 브랜치이름
```

Switched to a new branch '브랜치이름'이라는 메시지가 뜸


5. 팀원 : 내 브랜치에 소스 코드 업데이트 하기

```
git add .
```

```
git commit -m "버전 이름"
```

```
git push origin 위에서만들었던브랜치이름
```

git에 접속하여 **compare & pull request** 버튼 이용 : "master로 통합해 줘"하는 의미


 hong had recent pushes less than a minute ago


**Compare & pull request**

위의 메시지가 뜨지 않으면 [Pull requests] 메뉴 이용(master에 갈 작업 있음)












## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn](#)

 base: master < > compare: comlec > ✓ **Able to merge.** These branches can be automatically merged.

**Add a title**

**Add a description**  

Write Preview H B I    |    |  |  @   

xx월 xx일 member login 작업  
뭐시기 뭐시기

Create pull request >

6. 병합하기(브랜치에 있는 내용과 master의 내용 병합)

Pull requests에서 **merge pull request** 이용. 아래와 같이 cmd명령어를 사용할 수 있음.

(cf) **git checkout master**

**git merge 브랜치이름**

7. 마스터 브랜치에서 소스 가져오기(병합이 끝나면 모든 팀원들 pull)

**git pull origin master**

**git checkout master**

**git pull origin master**

**git checkout 브랜치이름**

pull과 push를 반복하면서 서로 협업한다

8. 브랜치끼리 이동하기

**git checkout 브랜치이름** (git switch 브랜치이름)

내가 내 브랜치에서 마스터 브랜치로 이동을 하고 싶거나 다른 브랜치로 이동하고 싶을 때 사용

9. 브랜치 확인

**git branch**

10. 브랜치 삭제

```
git checkout master
```

```
git branch -D 삭제할브랜치이름 (로컬 정보에 삭제)
```

```
git push origin --delete 삭제할브랜치이름 (github 사이트에서도 삭제)
```

11. 협의한 meeting 내용은 Issues에 올리기도 합니다

12. Git 협업시 가장 주의할 점은 "충돌을 피하는 것". 같은 파일을 2명 이상의 조원이 동시에 수정하고 git add 하면 충돌됨

13. master 브랜치에 있는 내용을 develop branch에 복사

```
git checkout -b develop
```

```
git push -u origin develop
```