

LEE,HyunJung

---

# Git & GitHub

GIT은 프로그램소스 같은 것의 변경이력을  
기록하거나 추적하기 위한 분산버전관  
리시스템입니다.

Git이란 무엇이죠?

어렵네요.

분산버전관리시스템이라니...

# 분산버전관리시스템

- ▶ 분산은 여러 사람의 중앙 저장소에 접속하지 않고 작업을 동시에 할 수 있음을 의미해요.
- ▶ 버전관리는 중앙의 저장소에 저장되거나 변경되는 이력을 관리한다는 것을 의미해요.

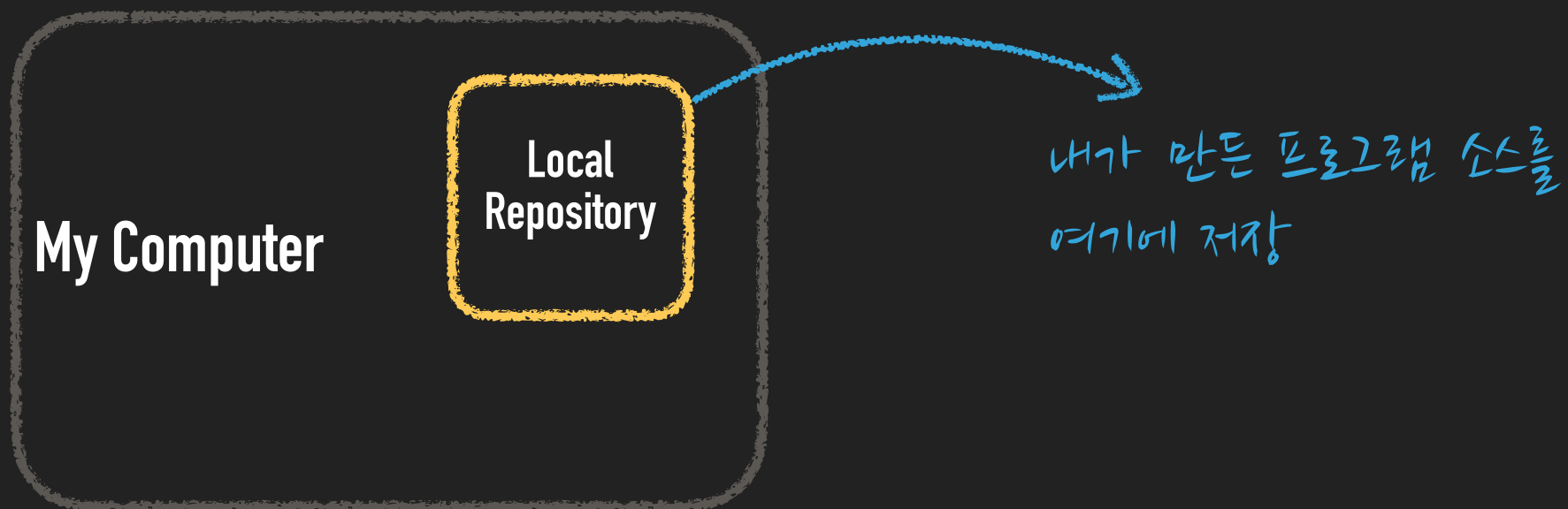


# 더 유식한 말을 해볼까요?

- ▶ 이런 것들을 형상관리도구라고 해요.
- ▶ 형상관리는 소프트웨어 변경사항을 체계적으로 관리하는 것을 의미한답니다.
- ▶ 그 중 git은 최근 많이 사용하고 있는 대표적인 형상관리도구입니다.

# GIT의 구조를 알아볼까요?

git은 저장소가 내컴퓨터(Local repository)에 있어요.



분산이라면서요..

다른사람과 같이 소스를 공유해야 하는데 어찌죠?

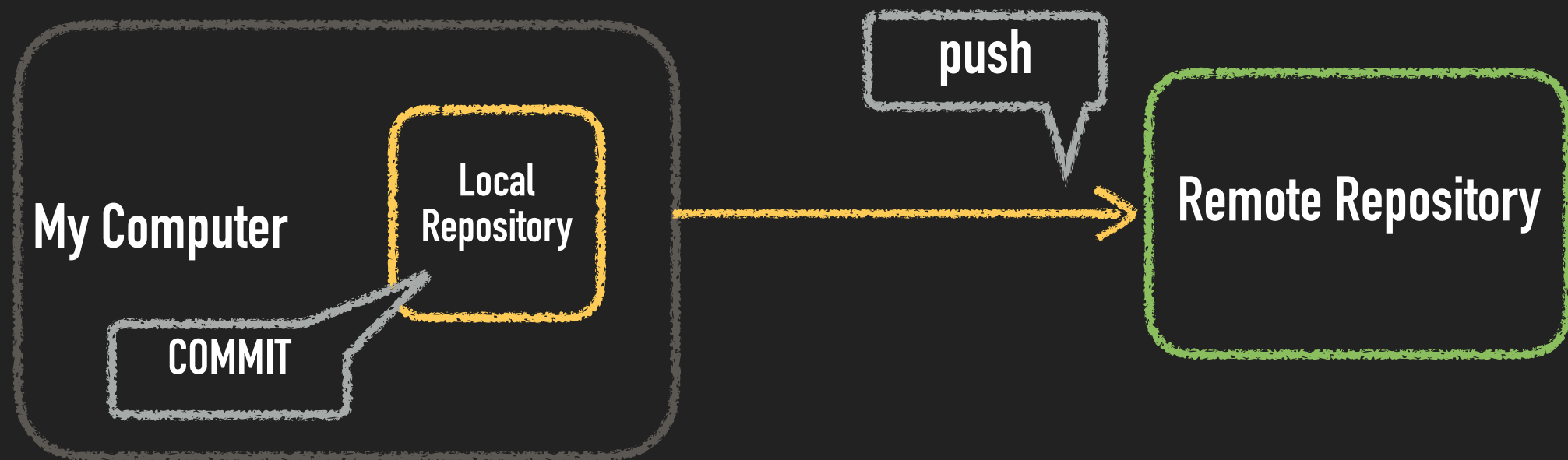
내컴퓨터는 다란 사람이랑 동시에 사용할 수 없어요.

네...고객님...

원격저장소(Remote repository)를 만드시면 됩니다.



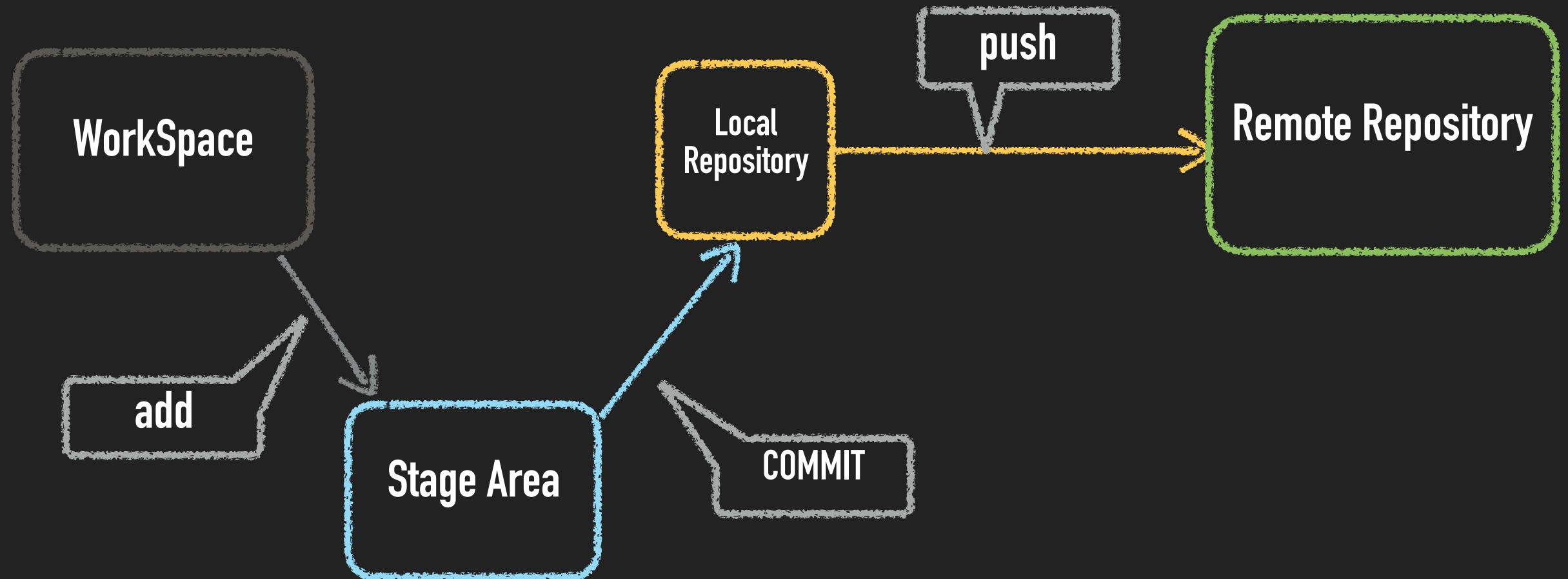
1. commit : 내 컴퓨터의 Local Repository에 작업내용을 저장합니다.
2. push: 자신의 작업내용을 다른사람에게 공유하기 위해 Local Repository에 저장됨.



아직 끝이 아니에요.

Git에는 한단계가 더 존재해요.

Stage 영역이라고도 하고, Index라고도 해요.



이해하셨나요?

Git의 구조는 이거면 충분해요.

작업의 흐름으로 GIT의 명령어 알기.

**git init**새로운 저장소 만들기

폴더를 만들고 그 안에서 명령어를 실행해요.

**git clone**저장소 받아오기

local/remote repository를 복제해요.

```
git clone /local/path
```

```
git clone username@host:/remote/path
```

**git add**작업한 내용을 stage에 올립니다

commit상태 전으로, 곧 commit할 것임을 알려줘요.

```
git clone "file name"
```

**git commit**

변경내용을 확정해요

local repository에 작업한 내용을 반영해요.

```
git commit -m "commit message"
```

**git add**

Remote Repository를 지정해요.

```
git remote add origin "remote server address"
```

**git push**

변경내용을 발송해요

local repository에 있는 변경내용을 remote repository에 반영해요.

```
git push -u origin master
```

## git pull

다른 사람이 한 작업내용을 local repository에 갱신해요  
나의 workspace에 우리 팀이 작업한 내용을 갱신해요.

```
git pull
```

## git merge

내 작업과 다른 사람의 작업한 부분의 충돌 해결하기  
대부분 git에서 자동으로 해결하지만 해결안될 경우, 직접 병합해요.

```
git merge "branch name"
```

## git push

변경내용을 발행해요

local repository에 있는 변경내용을 remote repository에 반영해요.

```
git push -u origin master
```

```
commit 6762dc01e05cdfca004ef65668
Author: Whatwant <whatwant@whatwa
Date: Tue Feb 14 22:13:21 2012
```

```
mv mv_test_action.txt
```

```
commit 67e4212b23ed5f31b2a9dac6d0
Author: Whatwant <whatwant@whatwa
Date: Mon Feb 13 22:31:14 2012
```

```
add mv_test.txt
```

```
commit b904f926b01780b193f6422052
Author: Whatwant <whatwant@whatwa
Date: Sun Feb 12 23:39:26 2012
```

```
rm --cached rm_test3.txt
```

```
commit 838b7d5898ef28051bb14a9a3e
Author: Whatwant <whatwant@whatwa
Date: Sun Feb 12 23:18:48 2012
```

```
rm rm_test2.txt
```

```
commit 3af2b945ed842961220b78602f
Author: Whatwant <whatwant@whatwa
Date: Sun Feb 12 23:09:17 2012
```

## Django girls

# Let's Start



# #TO DO

1. GitHub 가입하기
2. 내 컴퓨터에 Git 설치하기
3. Local repository 를 생성하기
4. 파일을 Local repository에 저장하기
5. Local repository와 Remote repository를 연결하고 저장하기
6. Remote에서 변경된 파일을 Local에 적용하기

# 1. GitHub에 가입하세요.

우리는 Github를 Remote repository로 쓸거예요.

github는 무료입니다. 그러나 계정이 필요해요

가입한 다음, Create repository를 눌러 아래와 같이 새로운 저장소를 만들어봅시다.

<https://github.com/>

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: foxmyth

Repository name: django

Great repository names are short and memorable. Need inspiration? How about effacious-weasel.

Description (optional): test django

Public: Anyone can see this repository. You choose who can commit.

요기에 repository  
이름을 입력하기.

Django girls tutorial  
를 참고하면 좋아  
요.

[tutorial.djangogirls.org/  
ko/deploy/index.html](https://tutorial.djangogirls.org/ko/deploy/index.html)

## 2. Git을 설치하세요.

<http://git-scm.com/> 에서 파일을 다운로드받아 설치한다.



자신의 OS에 맞는 것을  
다운로드 받으세요.

```
django
E:\projects\django>git --version
git version 2.5.1.windows.1
E:\projects\django>
```

내컴퓨터에 Git이 설치되어 있는  
지를 알고 싶을 때는 Command Line  
에서 아래의 명령어를 치세요.

**git --version**

### 3. Local repository를 생성합니다.

django

```
E:\projects>mkdir django
```

저장소를 생성하고 싶은 폴더로 이동하거나 생성해주세요.

```
E:\projects>dir
E 드라이브의 볼륨: OS
볼륨 일련 번호: F8B7-7112
```

Local Repository 생성.

```
E:\projects 디렉터리
```

```
E:\projects\django>git init
Initialized empty Git repository in E:/projects/django/.git/
```

```
2015-12-04 오후 02:28
```

```
2015-12-04 오후 02:28
```

```
2015-11-11 오후 04:02
```

```
2015-12-02 오후 06:05
```

```
2015-12-04 오후 02:28
```

```
2015-10-22 오전 11:29
```

```
2015-12-01 오후 01:00
```

```
2015-10-22 오전 11:26
```

```
2015-10-22 오후 03:47
```

```
E:\projects\django>echo "make test file" > test.txt
```

Test 파일을 하나 생성합니다.

```
E:\projects\django>dir/w
E 드라이브의 볼륨: OS
볼륨 일련 번호: F8B7-7112
```

```
E:\projects\django 디렉터리
```

```
3개 파일
```

```
6개 디렉터리
```

```
[.]          [..]          test.txt
```

```
1개 파일
```

```
19 바이트
```

```
2개 디렉터리 873,197,264,896 바이트 남음
```

```
E:\projects>cd django
```

```
E:\projects\django>
```

```
E:\projects\django>
```

Test 파일은 윈도우 탐색기(또는 Finder)를 통해서 생성해도 되요.

Command Line이 어렵다면 굳이 쓰지 않아도 됩니다. 개발자들도 어려워해요.

## 4. 파일을 Local repository에 저장해봅시다.

```
django
E:\projects\django>git add test.txt
E:\projects\django>git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   test.txt
E:\projects\django>git commit -m "committed first time"
[master (root-commit) f7689b7] committed first time
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
E:\projects\django>git status
On branch master
nothing to commit, working directory clean
E:\projects\django>
```

Test파일을 Stage 영역으로 보냅니다.

이제 Stage에 있는 파일들을 Local Repository에 저장해요.

### Remember it?

local repository에 파일을 저장하는 단계는 2 단계였어요.

stage 와 commit

\* **git status** : 현재 상태를 표시해줘요.

## 5. Remote Repository와 연결해보아요.

Github의 repository 주소를 복사해요.

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/foxmyth/django.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository

echo #  
git in  
git ad  
git co  
git re  
git pu

```
E:\projects#django>git remote add origin https://github.com/foxmyth/django.git
```

```
E:\projects#django>git status  
On branch master  
nothing to commit, working directory clean
```

이제 연결합시다.

```
E:\projects#django>git remote -v  
origin https://github.com/foxmyth/django.git (fetch)  
origin https://github.com/foxmyth/django.git (push)
```

```
E:\projects#django>git push -u origin master  
Username for 'https://github.com': foxhyun@gmail.com  
Password for 'https://foxhyun@gmail.com@github.com':  
Counting objects: 3, done.  
Writing objects: 100% (3/3), 244 bytes | 0 bytes/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
To https://github.com/foxmyth/django.git  
* [new branch]      master -> master  
Branch master set up to track remote branch master from origin by rebasing.
```

Local repository에 있는 파일들을 github로 보내요.

## 6. Github에서 파일을 수정하고, Local Repository에 적용해봐요.

foxmyth / django

Code Issues 0 Pull requests

django / test.txt

Edit file Preview changes

1	"make test file"
2	"edit test file" from remote.

```
E:\projects\django>git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean

E:\projects\django>git pull
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/foxmyth/django
   f7689b7..2e45624  master    -> origin/master
First, rewinding head to replay your work on top of it...
Fast-forwarded master to 2e456247e01ebbf2ab268abe122a22b8c21931f3.

E:\projects\django>echo test.txt
test.txt

E:\projects\django>type test.txt
"make test file"
"edit test file" from remote.

E:\projects\django>
```

다른사람이 수정한 것을 Local repository  
에 반영합니다.

git add 부터 git pull까지  
작업이 끝날때까지 무  
한반복하면 되요.

쉽죠?



```
commit 6762dc01e05cdfca004ef65668
Author: Whatwant <whatwant@whatwa
Date: Tue Feb 14 22:13:21 2012
```

```
mv mv_test_action.txt
```

```
commit 67e4212b23ed5f31b2a9dac6d0
Author: Whatwant <whatwant@whatwa
Date: Mon Feb 13 22:31:14 2012
```

```
add mv_test.txt
```

```
commit b904f926b01780b193f6422052
Author: Whatwant <whatwant@whatwa
Date: Sun Feb 12 23:39:26 2012
```

```
rm --cached rm_test3.txt
```

```
commit 838b7d5898ef28051bb14a9a3e
Author: Whatwant <whatwant@whatwa
Date: Sun Feb 12 23:18:48 2012
```

```
rm rm_test2.txt
```

```
commit 3af2b945ed842961220b78602f
Author: Whatwant <whatwant@whatwa
Date: Sun Feb 12 23:09:17 2012
```

지난 시간에 작업한 Restaurant Project를 Github에 올려보아요.

## Django girls

---

# HomeWork



<https://git-scm.com/downloads/guis>

개발자들도 많이 사용해요.  
사용해보세요.

Command Line(Terminal)이 어렵다면?



Thank you.  
오늘은 여기까지!!

---