

# Git with GitHub

Jade

IndiFlex 시니어코딩

# Git 시작하기

Git 첫번째 시간

IndiFlex 시니어코딩

# Settings

Download Git Client

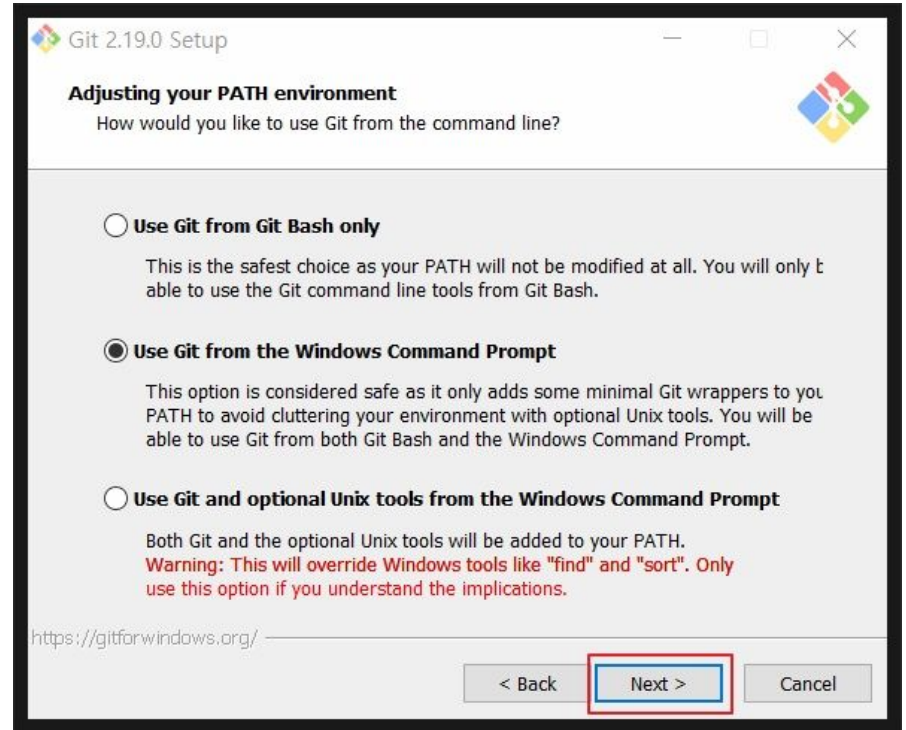
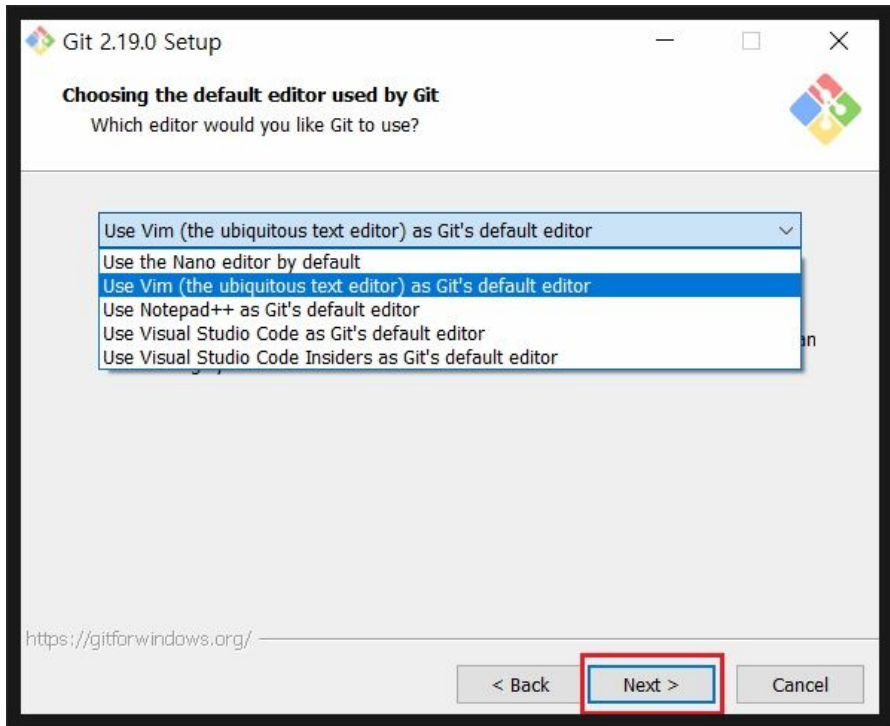
- <https://git-scm.com/downloads>

SCM Service Site

- [GitHub](#) : 무료는 public만 가능, 안정적
- [BitBucket](#) : public/private 모두 무료, 안정적인 편
- 기타: [GitLab, Codebase, CloudForge, etc](#)

(참고) windows에서 GitHub에 한글 깨질 때

- cmd"명령프롬프트"> **chcp 65001**

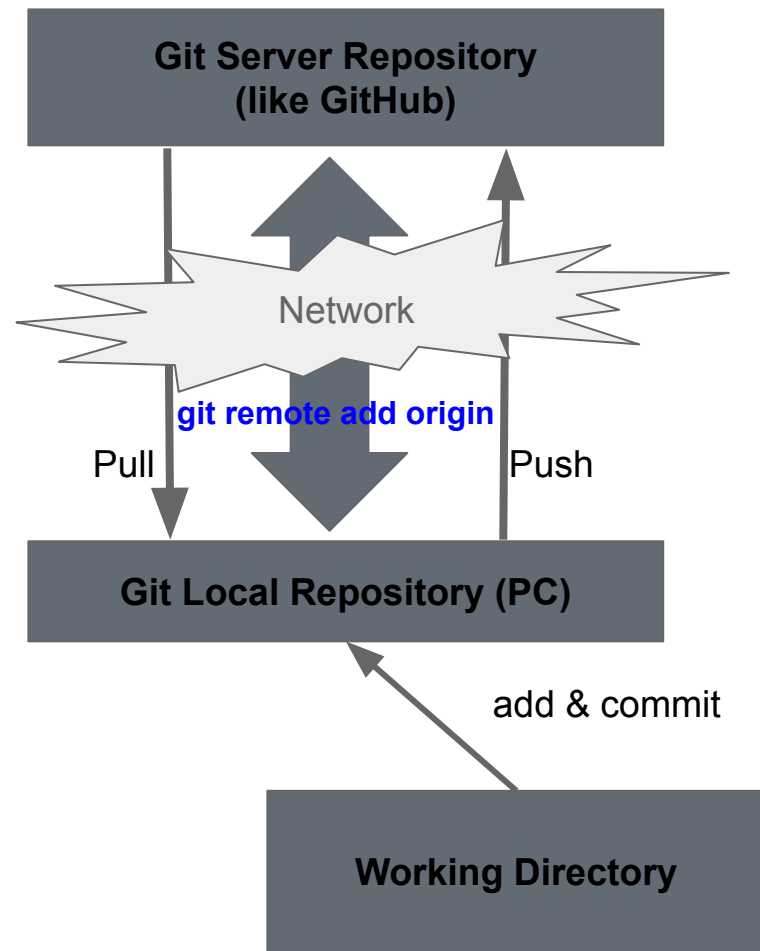


# Git Config

## (환경 설정)

1. # 바탕화면 `git bash` 실행
2. `git config --list`
3. `git config user.name`  
`git config --global user.name <github-name>`
4. `git config user.email`  
`git config --global user.email <email>`
5. `git config` 한눈에 보기  
`cat ~/.gitconfig`

# git structure (구조)



# 로컬 저장소 만들기 (initialize local repository)

1. `cd <work-dir>`
2. `git init`
3. `.gitignore` 파일 작성
4. `git add --all`
5. `git add <file명>` 또는 `git add .`
6. `git commit -am "first commit message"`
7. `git remote add origin <git-remote-url>`
8. `git push -u origin master`  
`git push -fu origin master`
9. `git log`
10. `git status` # staging 상태 확인

(참고) GitHub RemoteURL 패턴

<https://github.com/<사용자이름>/<저장소명>.git>

# git clone & pull



**git clone** <github-remote-url>

- GitHub에 만든 repo와 클론하기

#git pull : 서버의 최신 소스를 다운받기

1. **git pull**
2. touch aaa.txt
3. **git add --all`q** hk1`
4. **git commit -am "message"**
5. **git push**

(참고: 서버와 연결된 repo 끊기)

**git remote rm origin**



# Git with GitHub

## Reset & Merge



IndiFlex 시니어코딩

# Source History Reset & Revert (Git Log & Reset)



## GitBash Command

- `git log --graph --oneline`
- `git reset --[hard | soft | mixed]`
- # (주의) **hard**는 시계(모든것)를 되돌림!!
- cf. `git revert` # history쌓임, 소스 그대로

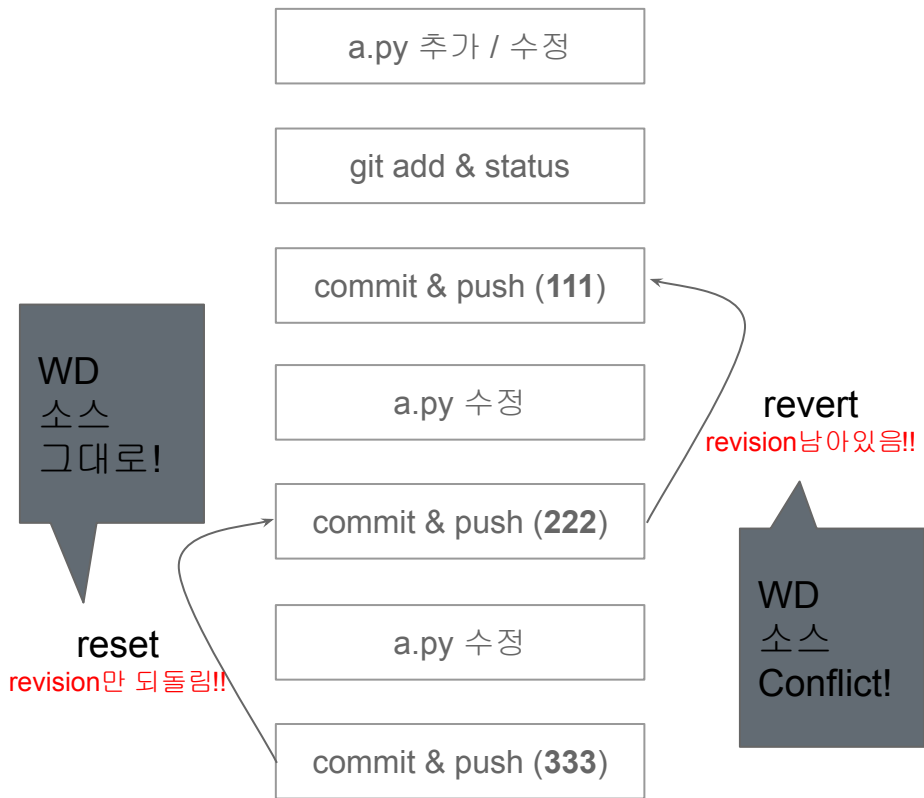
## GitHub (Click Revision or Comment msg)

- Repository Home
- Source Code View
- Split > Let's Attach Comment

## GitLens (Visual Studio Code)

- Install VSCode GitLens Extension
- History (Revisions)
- Branches
- **OLD** | **New**

# (실습) Reset vs Revert



git push origin +master

cf. git reset --hard <revision>은 소스도 되돌림!!

# VSCode GitLens after revert

```
sfdsakfjkals;dsfdsakfjkalssfdas
<<<<<<< HEAD
a
sdf
dsfdsakfjkalssfdasdfasd
dsfdsakfjkalsdsa
f
dsfdsakfjkalssfdasdfasdf
as
dsfdsakfjkals
dsfdsakfjkalsdsa
f
ads
d
sasdfa
=====
DATE=`date +%Y-%m-%d` "%H:%M`
MSG="$DATE lesson"
if [ $# -gt 0 ]; then
    MSG="$DATE - $1"
fi
git add --all
git commit -am "${MSG}"
git push
>>>>>> parent of cacf34a... LAST
```

2

현재 변경 사항 수락 | 수신 변경 사항 수락 | 두 변경 사항 모두 수락 | 변경 사항 비교

3

<<<<<< HEAD (현재 변경 사항)

4

a

5

sdf

6

dsfdsakfjkalssfdasdfasd

7

dsfdsakfjkalsdsa

8

f

9

dsfdsakfjkalssfdasdfasdf

10

as

11

dsfdsakfjkals

12

dsfdsakfjkalsdsa

13

f

14

ads

15

d

16

sasdfa

17

=====

18

DATE=`date +%Y-%m-%d` "%H:%M`

19

MSG="\$DATE lesson"

20

if [ \$# -gt 0 ]; then

21

MSG="\$DATE - \$1"

22

fi

23

git add --all

24

git commit -am "\${MSG}"

25

git push

26

>>>>>> parent of cacf34a... LAST (수신 변경 사항)

27

28

29

# conflict & merge

- **conflict**  
<<<< HEAD (server와 충돌난 부분)  
>>>> local
- **editor에서 수정**
- **저장 후** 아래 명령 수행  
git add --all  
git commit -am "aaaaa"
- git log --graph --decorate --oneline
- git status
- **at VSCode GitLens**

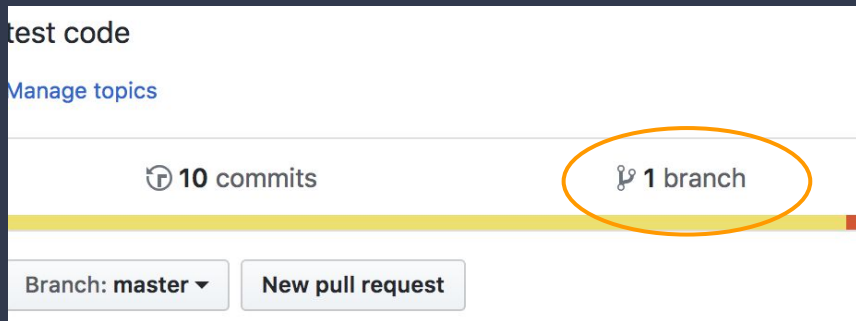
# Git with GitHub

## Branch



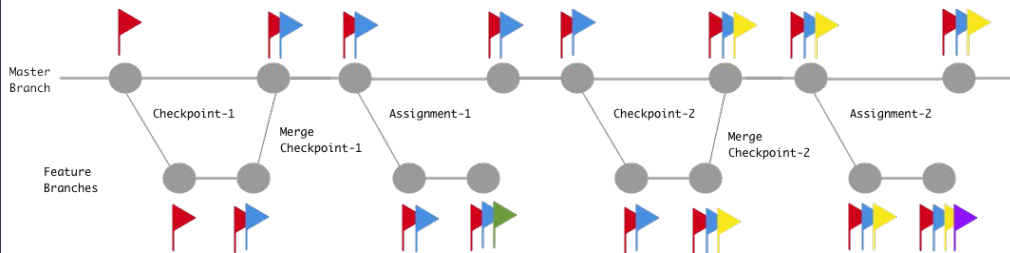
IndiFlex 시니어코딩

# branch



1. `git branch <branch-name>` # branch 만들기
2. `git branch` # branch 전체 보기
3. `git checkout <branch-name>` # 전환
4. 소스 수정 & add & commit
5. `git push origin <branch-name>`
6. `git log`
7. 다른동료폴더> `git clone <git-remote-url>`  
OR  
다른동료폴더> `git pull`

# 여기서 branch 내용까지 받아지면 성공  
# branch를 작업하는 쪽 외에는 소스 안보임



# branch (cont'd)

## <conflict scenario>

Me (testbranch)

Others (master)

A.java 수정

A.java 수정

merge to master

commit & push

commit & pull & push to master

**conflict !!**

소스 수정 & commit & push

commit & push

pull & 끝!!

# master로 merge하기 (작업폴더에서!!)

git checkout master #master로 변경

**git merge <branch-name>**

git branch

git log

# conflict (non fast-forward)

git status

# branch 삭제

local> git branch -d <branch-name>

local> git branch

remote> git push --delete origin <branch-name>

From github.com:indiflex/swptest

b968750..16c84a3 master -> origin/master

Auto-merging src/main/java/com/jade/swp/controller/UserController.java

CONFLICT (content): Merge conflict in src/main/java/com/jade/swp/controller/UserController.java

Automatic merge failed; fix conflicts and then commit the result.

```
public String googlecallback(Model model, @RequestParam String id) {
    System.out.println("GGGGGGG>> 여기는 googleCallback: " + id);
    <<<<<<<< HEAD
    System.out.println("Model: " + model.toString());
    =====
    System.out.println(model.toString());
    System.out.println("This is master written!");
    >>>>>>> 16c84a3905636eddc6a1a179a35150e838219c5e
```



# git branch 보기

# 로컬(PC) 브랜치 보기  
`git branch`

# 서버(GitHub) 브랜치 보기  
`git branch -r`

# 양쪽 모두 보기  
`git branch -a`

#서버의 특정 브랜치 가져오기  
`git checkout -t origin/<branch-name>`

#

# git branch (cont'd)

```
git diff <branch-name>
```

```
git diff <branch-name> <file-name>
```

```
git checkout -p <branch-name> <file-name>
```

```
git checkout -p <branch-name>
```

# 브랜치의 특정 파일만 master로 커밋해서 올리기

```
git checkout master
```

```
git add <file-name>
```

```
git commit -am "message"
```

```
git push origin master
```

# More..

- # GitHub Collaborator & folk
    - collaborators: Repository의 공동 작업자
    - folk : 다른 사람의 Repository 연결하기
- (주의: folk된 repo. 경로는 원본과 다름!!)

#서버의 특정 브랜치 가져오기

```
git checkout -t origin/<branch-name>
```

# 특정 브랜치 pull 하기

```
git pull origin <branch-name>
```

cf. git push origin <branch-name>

# history 완전 삭제

# BGF 설치

- <https://rtyley.github.io/bfg-repo-cleaner/>

# 삭제 작업

1. 파일 삭제
2. add & commint & push
3. `git clone --mirror <url>`
4. `bgf --delete-files`
5. push

```
$> git clone --mirror git@github.com:indiflex/hello.git
$> cd hello.git
$> java -jar ../bfg.jar --delete-files bigquery.json .
$> git reflog expire --expire=now --all && git gc
--prune=now --aggressive
$> git push
```

# 소스 저장소로 이동후!!

```
$> git pull --allow-unrelated-histories
```

# Git with GitHub

총정리



IndiFlex 시니어코딩

## (Tip) Fork 받은 repository 최신 소스로 merge하기

1. Go to Forked Repository
2. Compare to Original Repository
3. Change the **base fork** to your Repository
  - or Select **To** and **From**
4. Change the head fork to the Original Repository
5. Create pull request
6. Confirm Merge