

JAVA 웹 개발자 양성과정

Git과 GitHub

2강 – Git CLI 기초

By SoonGu Hong



JAVA 웹 개발자 양성과정

Git과 GitHub

1. Git 초기화

* 내 앱을 버전관리해 보자!

1. Git bash를 켜고~
2. 앱의 코드가 저장될 폴더로 이동하자
3. 그리고 그 폴더에 git 초기화 명령을 하면
 - **\$ git init**
4. 해당 폴더에 .git 이라는 숨김폴더가 만들어진다!
5. 이제 해당 폴더를 로컬 저장소라고 부르며 버전관리가 시작된다!
6. 만약 원격 저장소(GitHub)에서 내 컴퓨터로 코드를 불러오면 로컬 저장소가 자동으로 생겨요~

* Git 초기화 주요 명령어

- pwd : 현재 경로 출력
- cd : 디렉토리 변경
- ls : 현재 디렉토리 내부 파일 목록 출력
- git init : 로컬 저장소 생성 및 초기화

* Git 초기화 실습

```
KITT@kitt-23 MINGW64 ~
$ pwd
/c/Users/KITT

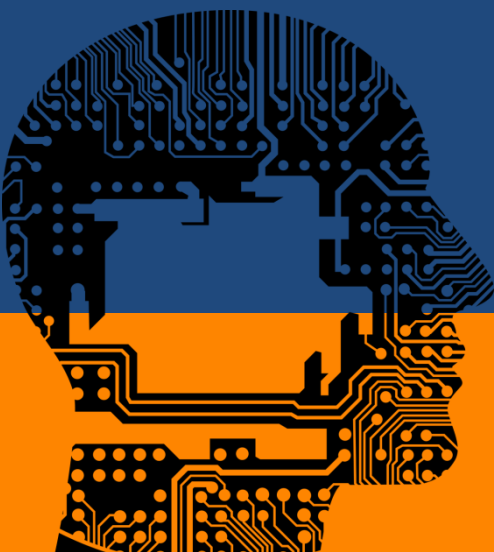
KITT@kitt-23 MINGW64 ~
$ cd d:

KITT@kitt-23 MINGW64 /d
$ ls
'$RECYCLE.BIN'/'      git_practice/'      py_game/'      공유_홍쌤 /
'system volume information'/'  imagedown/'      python_basic_hsg/
aws-keypair/'        msdia80.dll*      sqldeveloper/

KITT@kitt-23 MINGW64 /d
$ cd git_practice/

KITT@kitt-23 MINGW64 /d/git_practice
$ git init
Initialized empty Git repository in D:/git_practice/.git/

KITT@kitt-23 MINGW64 /d/git_practice (master)
$
```



JAVA 웹 개발자 양성과정

Git과 GitHub

2. 버전관리 해보기

* 버전 관리 용어 정리

1. 커밋(commit): 하나의 버전의 시점이 되는 지점

ex)

둘리가 오늘 아침 9시에 우리 앱의 회원가입 기능을 만들고 커밋했다!
(첫번째 버전 생성)

그리고 아침 11시에 도우너가 이어서 로그인 기능을 만들고 커밋했다!
(두번째 버전 생성)

근데 로그인기능이 개판이다 ... ——
(커밋은 버전관리의 세이브지점이 되므로
언제든 첫번째 버전으로 되돌아가는 것이 가능!)

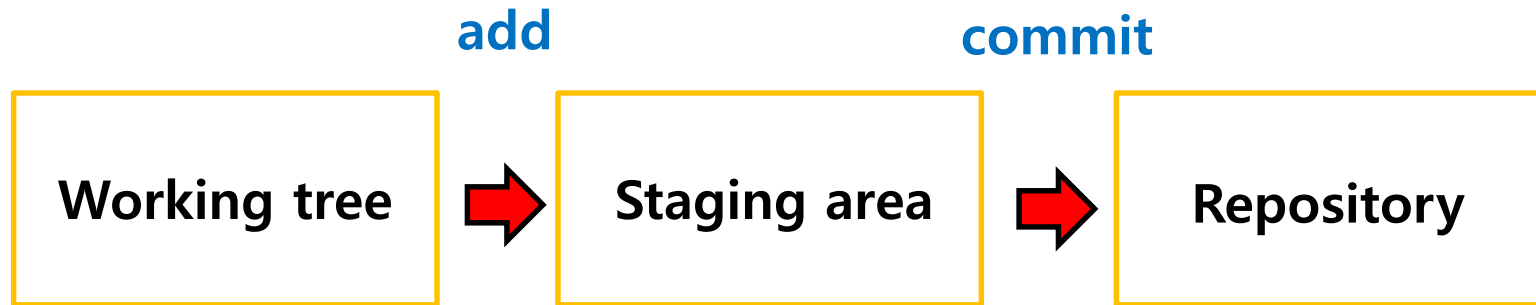


2. 애드(add): 커밋하기를 원하는 파일만 선택

ex) 둘리는 만들고 있는 앱에 유료 API를 사용하고 있다. 해당 API 접속 보안정보가 들어있는 파일을 같이 애드하고 커밋해서 깃허브에 공개된 상태로 올라가버렸네~~
앙 인생 망했어~~~



* 버전 관리 용어 정리



* Git add, commit 실습1

1. 실습 폴더에 README.md, index.html 파일을 만들고 작성해요!
2. README.md파일을 애드! (git add)

```
KITT@kitt-23 MINGW64 /d/git_practice (master)
$ git add README.md
```

3. 그리고 커밋! (git commit -m "메시지")

```
KITT@kitt-23 MINGW64 /d/git_practice (master)
$ git commit -m "README.md를 추가!"
```

4. 커밋이 잘 되었는지 확인! (git log)

```
KITT@kitt-23 MINGW64 /d/git_practice (master)
$ git log
commit f3adab0ad14964c9201705a8fc8330d6c36ea183 (HEAD -> master)
Author: Soon9 <[REDACTED]>
Date: Mon Dec 7 17:10:20 2020 +0900

    README.md를 추가!
```

* Git add, commit 실습2

1. 실습 폴더에 main.js와 common.css 파일을 생성!
2. README.md를 제외한 3개를 동시에 애드하고 싶다!
 - git add . 입력!

```
KITT@kitt-23 MINGW64 /d/git_practice (master)
$ git add .

KITT@kitt-23 MINGW64 /d/git_practice (master)
$ git commit -m "메인 페이지를 추가!"
```

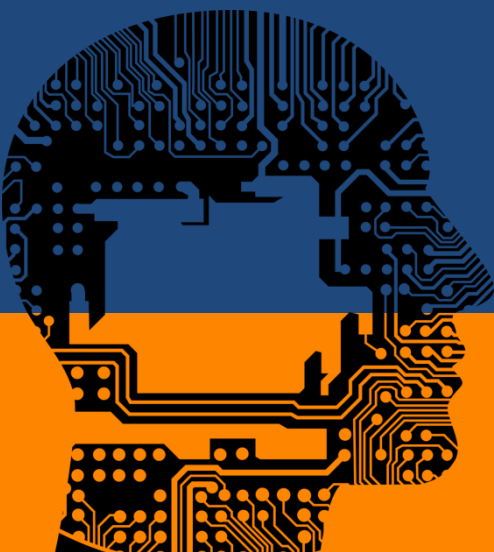
3. git log로 체크!

```
KITT@kitt-23 MINGW64 /d/git_practice (master)
$ git log
commit f88c316a3073b13e8eeba3a2eccbf81c217f1a3d (HEAD -> master)
Author: Soon9 <[REDACTED]>
Date: Mon Dec 7 17:22:02 2020 +0900

    메인 페이지를 추가 !

commit f3adab0ad14964c9201705a8fc8330d6c36ea183
Author: Soon9 <[REDACTED]>
Date: Mon Dec 7 17:10:20 2020 +0900

    README.md를 추가 !
```



JAVA 웹 개발자 양성과정

Git과 GitHub

3. 내 버전 GitHub에 올리기

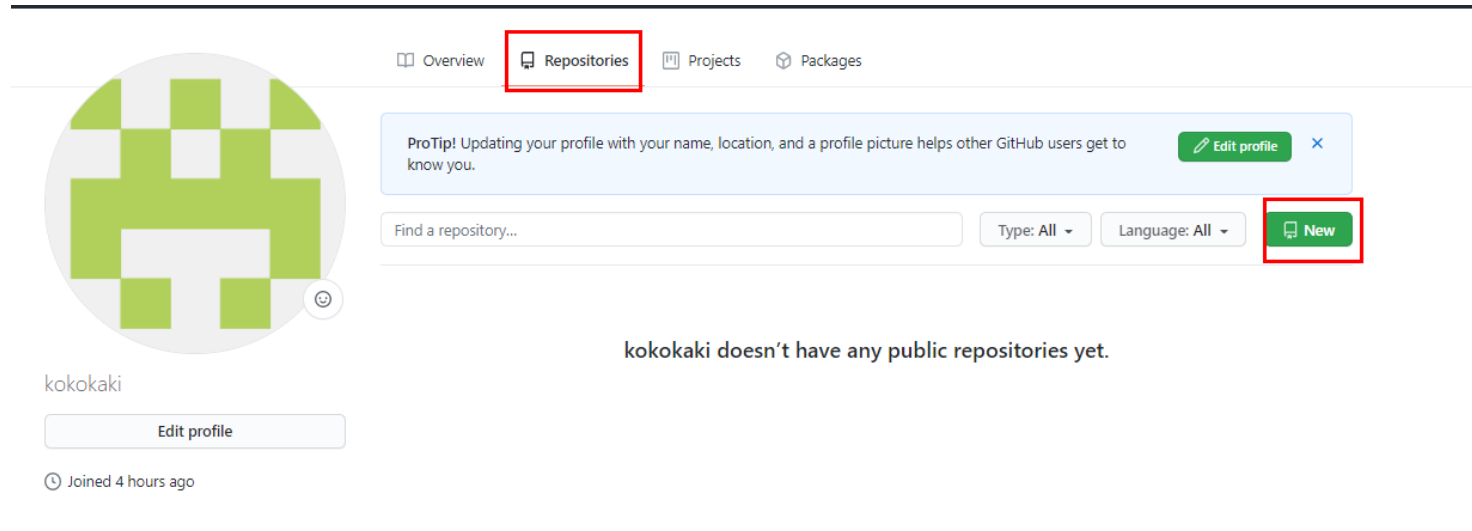
* 로컬 저장소와 원격 저장소

- 지금까지 **로컬저장소**에 버전관리를 하는 것을 배워보았는데요. 로컬저장소에는 안타깝지만 몇가지 단점이 존재합니다.
- 먼저 백업을 안했는데 컴퓨터가 망가져버리면 어떻게 될까요? 아 망했죠??
- 다른 사람들과 같이 버전관리하고 싶으면요? 뭐요? 카톡으로 보내준다고요?
- 이럴 때 필요한게 GitHub와 같은 **원격저장소**를 사용하는 겁니다. 아시겠습니까?



* 원격 저장소 만들기

- <https://github.com/>깃허브닉네임
- Repository -> New



* 원격 저장소 만들기

- Repository name을 작성!

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *



kokokaki ▾

Repository name *

koko-app ✓

Great repository names are short and memorable. Need inspiration? How about [fuzzy-octo-system](#)?

Description (optional)

깃허브 넘모 재밌어~~



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license


A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

* 원격 저장소 만들기

- Repository가 생성되면 원격저장소가 만들어지며 그 주소가 나옵니다.

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

 Set up in Desktop or ☐ HTTPS ☒ SSH

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# koko-app" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/kokokaki/koko-app.git
git push -u origin main
```

* 로컬 저장소 데이터 원격 저장소로 옮기기

1. 내 컴퓨터 로컬저장소에 GitHub 원격저장소 주소 알려주기

\$ git remote add origin 원격저장소 주소

```
KITT@kitt-23 MINGW64 /d/git_practice (master)  
$ git remote add origin https://github.com/kokokaki/koko-app.git
```

2. 만든 커밋 원격저장소로 푸시(push)

\$ git push origin master

```
KITT@kitt-23 MINGW64 /d/git_practice (master)  
$ git push origin master
```


* 로컬 저장소 데이터 원격 저장소로 옮기기

- 깃허브 가서 확인해보기~

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with 'master' branch selected, '1 branch', and '0 tags'. Buttons for 'Go to file', 'Add file', and 'Code' are visible. Below this, a commit by 'soongu' is highlighted, titled '메인페이지를 추가!' (Add main page!), with commit hash 'f88c316' and '1 hour ago'. The commit message is '2 commits'. Below the commit, a list of files is shown: 'README.md', 'common.css', 'index.html', and 'main.js', each with a description of the change and a timestamp of '1 hour ago'. The 'README.md' file is expanded, showing its content: 'Git 공부를 해보아요~' (Let's try learning Git~), 'README파일입니다' (This is the README file), and a bulleted list: '• 마크다운이 뭐지?' (What is markdown?), '• HTML이 뭘까?' (What is HTML?), and '• 커밋~~은 뭐냐고요~' (What is commit~~?).

master 1 branch 0 tags Go to file Add file Code

soongu 메인페이지를 추가! f88c316 1 hour ago 2 commits

File	Commit Message	Time
README.md	README.md를 추가!	1 hour ago
common.css	메인페이지를 추가!	1 hour ago
index.html	메인페이지를 추가!	1 hour ago
main.js	메인페이지를 추가!	1 hour ago

README.md

Git 공부를 해보아요~

README파일입니다

- 마크다운이 뭐지?
- HTML이 뭘까?
- 커밋~~은 뭐냐고요~

감사합니다
THANK YOU