

# Github TIL

## 1. TIL ?

- TIL은 Today I Learned의 줄임말로 개발자 사이에서 매일 자신이 학습한 내용을 commit(기록)하는 것
- github, bitbucket, gitlab과 같은 원격 저장소에서 제공하는 1commit-1grass의 흥미 요소 제공

## 2. TIL 세팅

### (1) Git으로 프로젝트 관리 시작: `git init`

- 자신이 앞으로 학습한 내용을 기록할 `TIL` 폴더를 하나 생성한다. 이때 해당 폴더는 최상단에 생성한다.
- `git bash`에서 `TIL` 폴더로 이동한 이후에 아래의 명령어로 `git` 관리를 시작한다.

```
$ git init
```

### (2) Commit을 위한 Staging: `git add`

- 현재 코드 상태의 스냅샷을 찍기 위한 파일 선택 (== Staging Area에 파일 추가)

```
$ git add [파일 이름] # .은 모든 변경 사항을 staging area로 올림
```

### (3) 버전 관리를 위한 스냅샷 저장: `git commit`

- 현재 상태에 대한 스냅샷을 `commit` 하여, 버전 관리를 진행한다.

```
$ git commit -m "커밋 메시지"
```

### (4) 원격 저장소 정보 추가: `git remote`

- Github 원격(remote) 저장소(repository)를 생성하고 `TIL` 폴더와 연결한다.
- 새로운 원격 저장소가 추가될 때만 입력한다.

```
$ git remote add origin [github 원격 저장소 주소]
```

## (5) 원격 저장소로 코드 `git push`

- 최종적으로 Github 원격 저장소에 push한다.

```
$ git push origin master
```

## (6) 그 외 명령어

- 현재 `git`의 상태를 조회 `git status`

```
$ git status
```

- 버전 관리 이력을 조회

```
$ git log
```

- `git` 설정 (user.name & user.email) : 최초 1회 설정

```
$ git config --global user.name "HyeJoonKim"  
$ git config --global user.email "lynn831@naver.com"
```

## 3. README.md

원격(remote) 저장소(repository)에 대한 정보를 기록하는 마크다운 문서. 일반적으로 해당 프로젝트를 사용하기 위한 방법 등을 기재한다.

### (1) README.md 파일 생성

- README.md 파일을 `TIL` 폴더(최상단)에 생성한다. 이름은 반드시 **README.md**로 설정한다.

```
$ touch README.md
```

### (2) (자신만의) TIL 원칙에 대한 간단한 내용 추가

- 마크다운 작성법 pdf에서 배우고 실습한 내용을 토대로 README.md 파일을 작성한다.
- 형식은 자유롭게 작성하되 마크다운 문법(의미론적)을 지켜서 작성한다.

### (3) 저장 후 버전관리 : `add`, `commit`, `push`

- 작성이 완료되면 아래의 명령어를 통해 commit 이력을 남기고 원격 저장소로 push한다.

```
$ git add README.md
$ git commit -m "add README.md"
$ git push origin master
```

## 4. 추가 학습 내용 관리

---

### (1) 추가 내용 관리

- `TIL` 폴더 내에서 학습을 원하는 내용의 폴더를 생성하고 파일들을 생성한 후 작업을 진행한다.

```
$ mkdir python
```

### (2) 변경 사항을 저장하고, 원격저장소로 옮긴다.

- 업데이트가 완료되면 아래의 명령어를 통해 commit 이력을 남기고 원격 저장소로 push한다.

```
$ git add .
$ git commit -m "학습 내용 추가"
$ git push origin master
```