

GitHub를 이용한 포트폴리오 제작

GitHub는 분산 관리 시스템인 Git을 사용하여 프로젝트를 지원하는 웹 호스팅 서비스입니다.

Git은 소스 코드 관리를 위한 분산 관리 시스템입니다. 분산 관리 시스템을 쉽게 말하면, 여러 명의 개발자가 특정 프로젝트를 자신의 컴퓨터로 협업하여 개발하면서 버전을 관리할 수 있는 시스템을 뜻합니다. 즉, 여러 명의 개발자가 작업하여 업로드 한 내용을 버전으로 관리할 수 있습니다.

최근에 개발자라면 너도나도 다 Git 아이디를 가지고 있으며, 대부분의 회사에서도 채용할 때 Git 아이디를 요구하는 회사들이 늘어났습니다.

이는 어떤 프로젝트를 해봤는지 코딩 능력을 보려고 하는 의도도 있지만, 협업을 해야 되는 회사에서 협업 경험과 Git에 대한 숙련도를 보기 위함이 큼니다.

GitHub는 Git을 지원하는 서비스 중 가장 유명한 호스팅 서비스이기 때문에, 유명한 오픈 소스 라이브러리들이 많이 업로드 되어 있습니다.

GitHub의 간단한 사용 방법을 알아봅시다. Git의 명령어는 많지만, 이번 강좌에서는 Git Repository를 만드는 것과 포트폴리오 링크로 활용하는 것에 대해 진행하겠습니다.

1) 기본 용어 설명

반복적으로 사용하는 몇 개의 용어를 이해하는 것이 필요합니다.

1-1) Command Line(명령 라인)

Git 명령어를 입력할 때 사용하는 컴퓨터 프로그램입니다. Mac에선 Terminal이라고 합니다.

마우스를 사용하는 것이 아닌 프롬프트로 알려진 텍스트 기반 명령어를 입력합니다.

GitHub에서는 Git Bash 앱을 통해 텍스트 형식으로 입력합니다.

1-2) Repository(저장소)

프로젝트가 있는 디렉토리나 저장 공간입니다. GitHub 사용자는 종종 'repo'로 줄여서 사용합니다.

컴퓨터 안의 로컬 폴더가 될 수도 있고, GitHub나 다른 온라인 호스트의 저장 공간이 될 수도 있습니다. 저장소 안에 코드 파일, 텍스트 파일, 이미지 파일을 저장하고, 이름을 붙일 수 있습니다.

1-3) Version Control(버전 관리)

기본적으로 Git이 서비스되도록 고안된 목적입니다. MS 워드 작업할 때, 저장하면 이전 파일 위에 겹쳐 쓰거나 여러 버전으로 나누어 저장합니다.

Git을 사용하면 그럴 필요가 없습니다. 프로젝트 히스토리의 모든 시점의 버전을 유지하므로, 결코 잃어버리거나 겹쳐 쓰지 않을 수 있습니다.

1-4) Commit(커밋)

파일 및 폴더의 추가, 변경 사항을 저장소에 기록하려면 커밋이 필요합니다. 커밋을 하면 그 시점을 저장소의 버전으로 만들어서 프로젝트를 이전의 어떠한 상태로든 재평가하거나 복원할 수 있는 체크 포인트를 가질 수 있는 것입니다.

2) Git Bash 설치

Git Bash는 Git 명령어를 입력할 수 있는 프롬프트 입력창입니다.

Git SCM(Souce Code Management)을 설치합니다.

참고 URL

<https://gitforwindows.org>

3) GitHub 가입

새로운 저장소를 생성하기 위해 GitHub에 가입합니다.

참고 URL

<https://github.com>

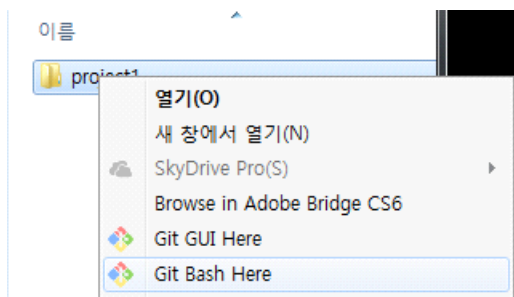
4) 실습

4-1) 로컬 폴더 생성

제작할 내용이 있는 폴더를 생성합니다. 업로드 할 포트폴리오 소스를 추가합니다.

폴더를 오른쪽으로 클릭하고, [Git Bash here]를 선택합니다.

Git Bash 프롬프트 창이 나타납니다.



4-2) 로컬 폴더 초기화

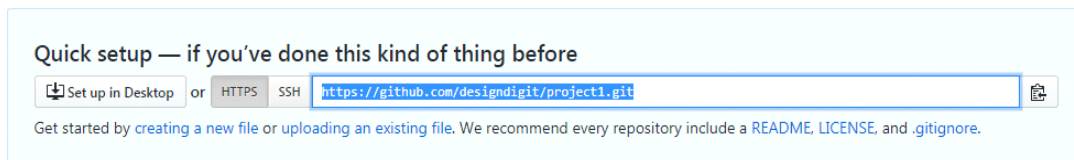
command : **git init**

Git에 연결될 로컬 폴더를 초기화합니다. 숨김 파일 형식으로 .git 폴더가 생성됩니다.

4-3) 로컬 폴더와 연결될 저장소 생성

GitHub 사이트에서 새로운 저장소를 생성합니다.

아래의 영역이 Git 저장 경로입니다.



4-4) Git 저장소와 로컬 폴더 연결

command : **git remote add origin repository address**

Git 저장소(origin repository address)와 현재의 로컬 폴더를 연결(remote)합니다.

예시 : **git remote add origin https://github.com/designdigit/project1.git**

4-5) Git 저장 경로에 있는 자료 가져오기

command : **git pull origin master**

가져올 자료가 없을 경우에는 치명적인 오류(fatal)가 발생합니다. 기존 자료가 있을 경우 업로드하기 전에 반드시 pull 명령어를 실행시켜야 자료가 모두 없어지는 경우가 발생되지 않습니다.

4-6) 로컬 폴더와 스테이징 영역 간의 싱크 확인

command : **git status**

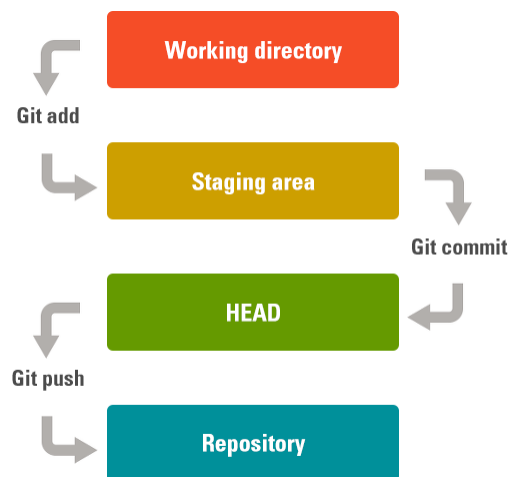
로컬 폴더와 스테이징 영역 간의 소스를 비교하여, 올라갈 여지가 있는 소스를 확인합니다.

4-7) 변경된 파일을 모두 트래킹

command : **git add .**

트래킹(Tracking)이란 새로 만들어지거나 수정된 소스를 스테이징 영역에 올리는 것을 의미합니다.

작업 트리와 인덱스



Git에서는 우리가 흔히 말하는 폴더를 Working directory(작업 트리)라고 부릅니다.

그리고 커밋을 실행하기 전의 저장소와 작업 트리 사이에 존재하는 공간을 Index(인덱스) 혹은 Staging area(스테이징 영역)이라고 합니다.

트래킹은 스테이징 영역에 올라가지 않거나, 수정된 소스를 올리는 과정을 의미합니다.

스테이징 영역에 올리지 않은 소스는 커밋될 수 없습니다.

4-8) 변경된 파일을 모두 커밋

```
command : git commit -m "message"
```

스테이징 영역에 올라가 있는 파일들에 대해 Head에 업로드하며 버전을 만들어 줍니다.

하지만 Repository에는 업로드 되지 않았습니다.

예시 : `git commit -m "project1"`

4-9) 변경된 파일을 Git 저장 경로에 업로드

```
comman : git push origin master
```

push는 Git 저장소에 업로드하는 과정입니다. 처음에는 로그인 창이 다시 뜨는 것을 확인할 수 있습니다.

The screenshot shows the GitHub interface for a repository named 'project1' under the user 'designndigit'. At the top, there are buttons for 'Unwatch', 'Star' (1), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area shows a message 'No description, website, or topics provided.' with an 'Edit' button. Below this is a summary bar indicating '1 commit', '1 branch', '0 releases', and '0 contributors'. A section for 'Branch: master' includes a 'New pull request' button and links for 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. A table lists the files in the repository: 'css', 'images', 'script', and 'index.html', all committed to 'project1_v1' 36 seconds ago. At the bottom, there is a prompt to 'Add a README' to help people understand the project.

File	Commit	Time
css	project1_v1	36 seconds ago
images	project1_v1	36 seconds ago
script	project1_v1	36 seconds ago
index.html	project1_v1	36 seconds ago

5) GitHub로 웹 링크 사용

참고 URL

<https://opentutorials.org/course/2473/16117>

[Settings] - [GitHub Pages] - [Source] - [Master Branch]

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at <https://designdigit.github.io/project1/>.

Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▼

Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

예시 : <https://github.com/designdigit/project1>

예시 : <https://designdigit.github.io/project1>

6) Git Command 일괄 정리

git init : Working director(로컬 경로)를 초기화

git remote add origin repository address : Git Repository(저장소)와 연결

git pull origin master : Git Repository(저장소) 백업, 처음 업로드 시에는 생략 가능

git status : Staging area(인덱스)에 올라가지 않은 파일을 체크

git add . : Staging area(인덱스)에 업로드

git commit -m "message" : HEAD(헤드)에 업로드하며 버전 저장

git push origin master : Git Repository(저장소)에 업로드