

# GitHub 포트폴리오

7주차. 오픈 소스 프로젝트 개요

Github Pages 개념과 활용	Github Pages는 정적 웹사이트를 무료로 호스팅할 수 있는 도구
	일정한 파일 형식을 지원하며 쉽게 웹사이트 배포 가능
Github Pages 웹 포트폴리오 설정	기본 도메인 및 커스텀 도메인 설정 가능
	기본 도메인은 일반적으로 "<계정>.github.io" 로 설정
	첫 페이지는 index.html 또는 index.md 파일명으로 작성

# 학습개요



## 학습목표

- 01.** 오픈 소스 프로젝트의 개요와 기여 방법을 설명할 수 있다.
- 02.** Github에서 Fork와 Clone의 차이점을 명확히 이해하고 실습을 통해 적용할 수 있다.

## 학습내용

- 오픈 소스 프로젝트
- Fork와 Clone의 차이점

## 오픈 소스 프로젝트

## Fork와 Clone의 차이점

- 오픈 소스 프로젝트는 소프트웨어 개발 커뮤니티의 핵심으로, 많은 개발자들이 협업하여 소프트웨어를 개발하고 개선해 나가는 중요한 방법임
- 오픈 소스 프로젝트에 기여하는 방법을 배우고 직접 실습해 봄으로써 협업 능력을 키울 수 있으며, 이는 실제 개발 환경에서도 매우 유용한 경험이 될 것임
- 또한, 오픈 소스 프로젝트에 기여한 경험은 포트폴리오에 추가하여 자신을 어필하는 데 큰 도움이 됨
- 이번 학습을 통해 Github에서 오픈 소스 프로젝트를 Fork하고 Clone하는 방법을 배우고, 실제 기여를 통해 자신의 코딩 실력과 협업 능력을 증명할 수 있음



# 오픈 소스 프로젝트

UNIT



## 오픈 소스 프로젝트란 ?

### 오픈 소스

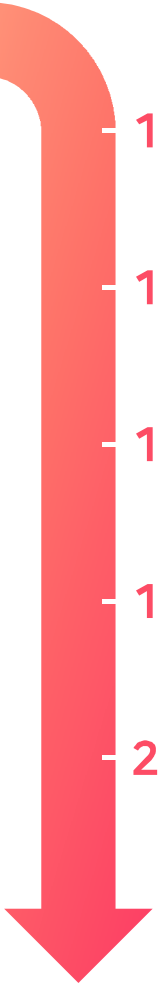
소스 코드가 공개되어 있어 누구나 그 코드를 열람, 수정, 재배포할 수 있는 소프트웨어

- 오픈 소스 이니셔티브(Open Source Initiative, OSI)의 오픈소스 정의(Open Source Definition)
  - 무제한적인 재배포 가능
  - 소스 코드 공개
  - 파생 작업물 허용
  - 개작 및 역공학 용인
  - 개인, 단체, 사용 목적에 대한 차별 금지

# 오픈 소스 프로젝트 개요



## 오픈소스 프로젝트의 역사


- 
- 1983년
    - 리처드 스톨만이 GNU 프로젝트 시작 (GNU's Not Unix)
  - 1991년
    - 리누스 토르발스가 리눅스 커널 최초 공개
  - 1997년
    - 에릭 레이먼드의 "성당과 시장" 에세이로 "오픈 소스" 용어 대중화
  - 1998년
    - 오픈소스 이니셔티브(OSI) 설립, 오픈소스 정의 발표
  - 2000년대
    - Apache, Mozilla, Linux, Android 등 오픈 소스 프로젝트 대활성

## 기여할 오픈 소스 프로젝트 선택

- 어떤 프로젝트인지 충분히 알아야 함(한번 사용해 본 경험 필요)
- 기여활동이 활발한 프로젝트 (마지막 커밋이 최근인 프로젝트)

## 기여 주의사항

- 기여할 대상 오픈 소스 프로젝트의 현행화
  - `git fetch <오픈 소스 프로젝트>` # 최신코드 받아오기
  - `git checkout <로컬 리포지토리 브랜치>` # 내 작업용 브랜치에 checkout
  - `git merge <오픈 소스 프로젝트>/<로컬 리포지토리 브랜치>` # 코드 병합
  - `git push origin master` # 내 github 계정의 저장소 최신화

 > 내가 기여한 것에 대해 생각보다 반응(댓글 등)이 적은 것에 낙담하지 않도록 함



A grayscale background image of a person with glasses working at a desk. A computer monitor is in the foreground, displaying a 'TEST UPDATE' notification with a circular arrow icon and a button that says 'UPDATE'. The person's hands are on a keyboard, and a mouse is on the desk. The overall image has a dark, textured overlay.

# Fork와 Clone의 차이점

UNIT



# Fork의 개념 및 사용법



## Fork 개념

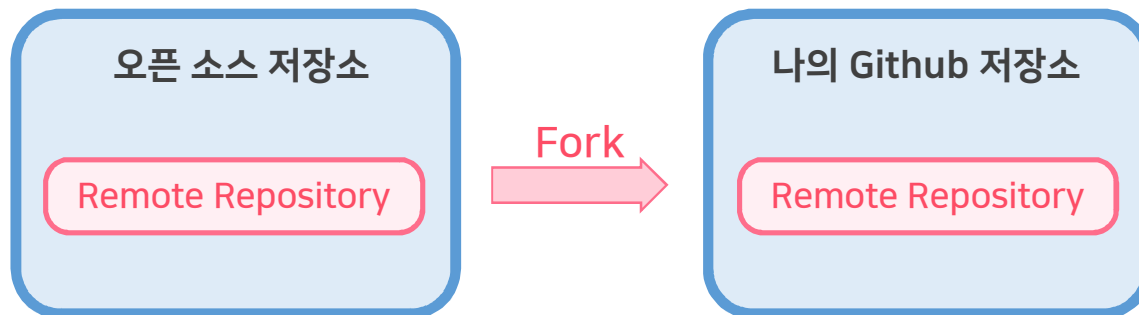
Fork

오픈소스 프로젝트의 복사본을 자신의 GitHub 계정으로 가져오는 것

- 원본 프로젝트의 전체 복사본이 자신의 저장소에 생성

## Fork 목적

- 프로젝트 수정 :  
원본 프로젝트를 직접 수정할 권한이 없더라도, 복사본에서 코드를 변경, 새로운 기능 추가 가능
- 이슈 해결 :  
프로젝트의 버그 발견 시, Fork한 복사본에서 수정한 후 Pull Request로 원본 프로젝트 반영 가능
- 아이디어 실험 :  
Fork한 프로젝트에서 새로운 아이디어나 실험 가능



# Clone의 개념 및 사용법



## Clone 개념

Clone

원격 저장소의 소스를 자신의 로컬 컴퓨터로 가져오는 것

- 복사본 또는 원본 프로젝트의 전체 복사본이 자신의 로컬 저장소에 생성

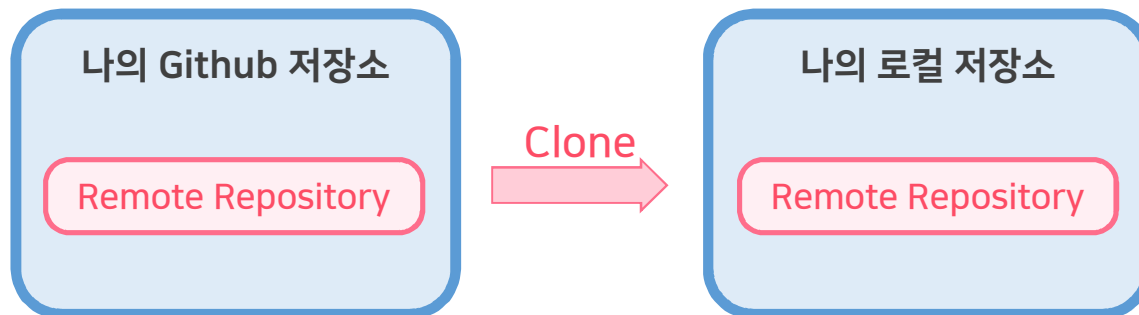


# Clone의 개념 및 사용법



## Clone 목적

- 로컬 개발 환경 구축 : 새로운 개발, 기존 프로그램 수정, 버그 수정
- 프로젝트 백업
- 오픈 소스 프로젝트 기여 준비



# 저장소와 Git 명령



## 저장소 관점 전체 구성

