

# **Git & GitHub**

202015405 정영준

# 목차

## 1. Git & GitHub

1.1)Git

1.2)GitHub

1.3)차이점

1.4)필요성

## 2. Git Branch

2.1)Master Branch

2.2)Develop Branch

2.3)Feature Branch

2.4)Release Branch

2.5)Hotfix Branch

# 1.1 Git

Git이란?

⇒ 버전 관리 시스템(VCS)

:파일 변화를 **시간에 따라 기록** 후

특정 시점의 버전을 **다시 가져올 수 있는** 시스템



# 1.2 GitHub

GitHub란?

⇒ Git을 무료로 제공하는 사이트

⇒ **저장소(Storage) 역할**



# 1.3 Git & GitHub 차이점

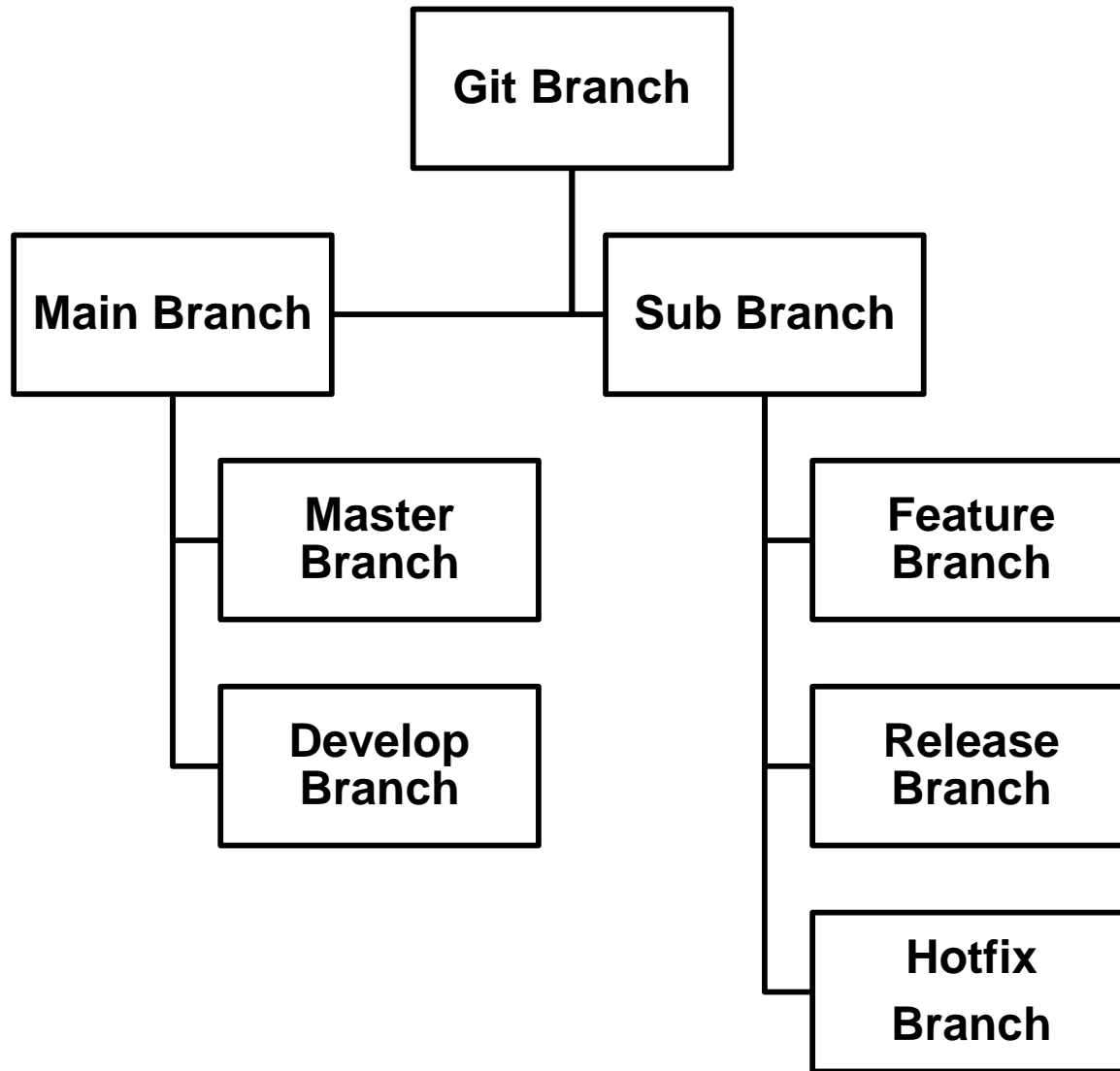
Git	GitHub
버전 관리 시스템(VCS)	코드 저장소(Storage)
변경된 내용 <b>기록</b>	<b>저장</b> 하는 장소

# 1.4 Git & GitHub 필요성

- 여러 사람의 **협업** ⇒ 하나의 프로젝트
- 지속적인 **수정·보완** 필요

## 2. Git Branch

- **Main Branch** : 실수로 commit할 경우 문제가 심각해짐
- **Sub Branch** : main branch에 완벽한 코드가 commit되기 위해서 사용





# 2.1 Master Branch

- 배포(release) 가능한 상태만 관리

## 2.2 Develop Branch

- **합병(merge)**을 위해 사용
- 안정적(모든 기능 추가 + 버그 수정 + 배포 가능)  
⇒ master branch에 합병
- 평시 개발 진행의 기반

## 2.3 Feature Branch

- 새로운 기능 개발, 버그 수정 필요 시  
develop branch에서 분기
- 자신의 로컬 저장소에 관리(공유할 필요X)
- **개발 완료 ⇒ develop branch와 합병 후 공유**

## 2.4 Release Branch

- **개발 단계를 구분** 짓는데 용이
- 최종 버그 수정, 문서 추가

## 2.5 Hotfix Branch

- 배포(release)후 **긴급한 수정**이 필요한 경우  
master branch에서 분기

**감사합니다.**