



# 2021.07.14 - What is Git?

## Git

- Git 사용이 능숙한 분을 찾습니다 라는 공고가 많이 보임
- Git 그냥 쓰면 되는 거 아닌가요?
  - 싱글(개인)플레이 - 코드 관리/백업
  - 멀티(팀)플레이
    - 협업 Workflow
    - 서비스/제품 이력 관리
    - 커밋 규칙과 정확한 메시지
    - 코드 리뷰
- 효과적으로 리뷰가 이루어질 수 있음 - 모두에게 서로 배우면서 실력을 향상시킬 수 있어요
  - Git과 관련된 역량들이 생각보다 많다는 점 기억!

## Git 구조 및 명령어 살펴보기

- Local : working Directory, Staging Area, LocalRepo
- Remote : remote repo
- Git 명령어 모음!

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/98cb7b0c-6614-4a33-a9d4-68fc8dfc408d/git-cheat-sheet-education.pdf>

## 🎄 Git Branch

- 커밋이 제일 중요한 단위
- 브랜치?
  - 코드 관리를 위해서 가지치기를 위한 분기를 만들어 내기 위한 용도로 사용
  - 특정 커밋을 가지치고 있어 포인터, alias 같은 느낌
  - 주로 브랜치 이름을 사용해서 작업을 하게 됨
  - feature 브랜치를 따서 개발을 시작해 보자
- 개발자가 Git을 잘 다루게 된다면, 소스트리를 잘 읽어서 빠르고 정확히 파악해 문제가 발생한다면, 쉽게 해결 가능
  - Head를 바꾸어 가면서 과거 시점에 돌아가서 상태를 확인하거나, 마음대로 가지고 놀 수 있게 함
    - 조심해야 할 부분(충돌이 발생할 수 있기 때문에!)

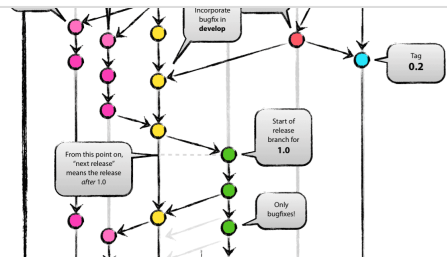
## 🔧 Git Flow

- Git Flow 설명 및 GitHub repo

### A successful Git branching model

This model was conceived in 2010, now more than 10 years ago, and not very long after Git itself came into being. In those 10 years, git-flow (the branching model laid out in this

» <https://nvie.com/posts/a-successful-git-branching-model/>



### nvie/gitflow

A collection of Git extensions to provide high-level repository operations for Vincent Driessen's branching model. For the best introduction to get started with git flow,

» <https://github.com/nvie/gitflow>

### nvie/gitflow

Git extensions to provide high-level repository operations for Vincent Driessen's branching model.



37 Contributors 223 Issues 25k Stars 3k Forks

### Gitflow Workflow | Atlassian Git Tutorial

Gitflow Workflow is a Git workflow that helps with continuous software development and implementing DevOps practices. It was first published and made popular by Vincent Driessen at nvie. The Gitflow Workflow defines a strict branching model designed around the project release.

» <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

- 나름 모범적인 전략
- 이 방식으로 일하는 Workflow 방식
- CI/CD 자동 배포 환경이 갖추어진다면, 배포되는 개발기, 테스터기, 스테이지기, 운영기를 통한 서버로 자동배포가 이루어짐
- 목적에 의해 시스템들이 연동되어 사용되는 것을 확인할 수 있음

## Trunk Based?

- Keep Fresh main branch, short-lived branches
- 최소한의 release 브랜치를 사용하기도 함
- master와 release의 용도가 뒤바뀐 것을 알 수 있음
- 어느 때 사용하면 좋을까?
  - 성숙한 개발자들이 많을 때
  - 제품에 버그가 있더라고, 빠른 출시가 필요할 때 사용하기 좋은 전략
- 우리는 어떠한 전략이 좋을까?
  - develop에서 QA를 지원한다거나, master에서 버전을 tagging해서 사용해 보는 것도 좋음

## Sprint 과정

- 보통 Sprint 단위로 개발을 진행하게 됨
1. Master에 코드가 있음 ⇒ Develop 브랜치 따서 작업
    - Develop 브랜치에서 feature/~~로 기능 브랜치를 따서 개발 후 MR
      - 코드 리뷰 후 통과되어 Develop에 Merge가 됨
    - Sprint 끝날 쯤 release 브랜치를 만들어 확인 후, QA 팀에서 확인
      - 버그가 발생했다면?
      - release 브랜치에서 bugfix 브랜치를 따서 작업 후 MR
      - 병합 후, 다시 한 번 release 브랜치에서 확인 후 이상 없으면 master에 v0.1.0으로 tagging해서 배포

- 주의! release 브랜치 내용을 develop에 반영해 주어야 버그 fix된 내용이 develop에 반영
2. 다시 Develop에서 feature 브랜치를 따서 작업
    - conflict가 나면 수정 후 다시 develop에 MR
    - 그리고 바로 출시 release에서, master에서 v0.2.0
  3. 갑자기 master 운영 중 버그 발생?
    - hotfix 브랜치 만들어 빠르게 패치 버전을 만들어 바로 Merge
    - hotfix 브랜치에 고쳐진 내용을 release, develop 브랜치에 업로드 해준다.

## GitLab과 로컬 설정 맞추는 방법

- 전역적인 방법

```
git config user.name ~~
git config user.email ~~~

git config -l
```

- 글로벌로 설정해서 사용하기 힘든 경우
  - 각 브랜치 별로 이름을 확인 가능
    - 5명이 개발했는데, 8명이 개발한 경우처럼 될 수 있음
    - 커밋용 사용자 정보는 마음껏 수정해서 확인할 수 있음
  - Repository Manager에서 확인

```
git config -l
## branch ~~~
```

## 커밋 메시지!

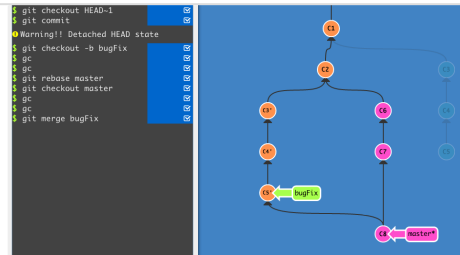
- 불친절한 메시지는 팀원들의 시간도 낭비될 수 있음
- 컨벤션을 맞추어서 깔끔하게 하는 것을 목표로 작성해야 함

## Git 연습!

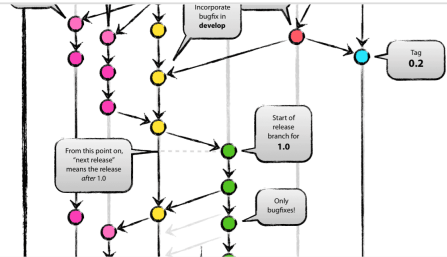
### Learn Git Branching

An interactive Git visualization tool to educate and challenge!

<https://learngitbranching.js.org/?locale=ko>



우린 Git-flow를 사용하고 있어요 | 우아한형제들 기술블로그  
안녕하세요. 우아한형제들 배민프론트개발팀에서 안드로이드 앱 개발을 하고 있는 나동호입니다. 오늘은 저희 안드로이드 파트에서 사용하고 있는 Git 브랜치 전략을 소개하려고 합니다. '배달의민족 안드로이드' <https://techblog.woowahan.com/2553/>



### Explain Git with D3

We are going to skip instructing you on how to add your files for commit in this explanation. Let's assume you already know how to do that. If you don't, go read some other tutorials. Pretend that you already have your files staged for commit and enter `git commit` as many times as you like in the <https://onlywei.github.io/explain-git-with-d3/#commit>

## 정리

- 개발해 방해되지 않을 정도까지는 Git 기본 개념 / 명령어 숙지
- Git Branch를 사용한 코드관리의 흐름(flow) 이해
- 회사/팀별 상황에 맞는 Git Branch 전략 선택
- 팀 별 합의된 정책에 맞춰 일관된 Git 사용 노력 필요
- 궁금하거나 자주 마주하는 케이스에 대한 Git 사용법은 꼭 연습