

Git Flow

Date	@July 14, 2021
Tags	

Git 개요

- 채용 공고에서 `Git` 등의 분산 버전 관리 이용에 능숙한 분을 찾습니다 라는 문구를 심심치 않게 볼 수 있다.

싱글 플레이

코드 관리/백업

멀티(팀) 플레이

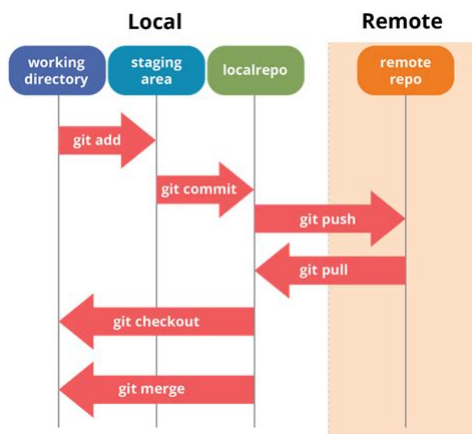
협업 Workflow

서비스/제품 이력 관리

커밋 규칙과 정확한 메시지

코드 리뷰

Git Branch



<https://about.gitlab.com/images/press/git-cheat-sheet.pdf>

버전 관리에서 가장 중요한 것은 "커밋"이다. 업무에 따라서 Branch의 이름으로 작업하고 정리하는 방식이다.

Git Branch Flow

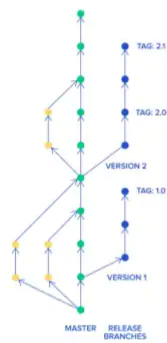


- 기본 브랜치 : Master(출시), Develop(개발)
- 부가 브랜치 : Feature, Bugfix, Release(CI/CD등의 베타), Hotfix 등

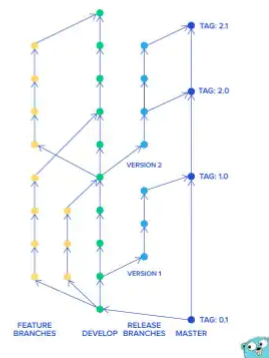
`git flow init` 라는 명령어로 flow가 나뉘어 지고, 부가적인 분리는 `git flow feature start sign-up` 등으로 branch 생성/변경이 가능하다. finish라는 기능은 자동으로 머지를 해버리기 때문에, 이후 Pull Request 등의 코드 리뷰 과정을 지나야 때문에 이는 진행하지 않는다.

Trunk based

TBD vs Gitflow



vs



Trunk Based는 능숙한 개발자들이 하루에도 수십번 배포하는 과정의 비즈니스에서 자주 사용된다.

따라서 우리는 Tagging을 달아서 배포하는 내용으로 진행하는 것이 좋을 것!

Git commit 사전 작업

사전 컨벤션 작업과 히스토리 축적과정을 잘 이해해서 진행해야 합니다!

gitignore

gitignore.io

Create useful .gitignore files for your project by selecting from 522 Operating System, IDE, and Programming Language .gitignore templates

<https://www.toptal.com/developers/gitignore>

gitignore.io

Create useful .gitignore files for your project

Source Code | Downloadable Files | About Gitignore

Toptal

branch 연습하기!

Learn Git Branching

An interactive Git visualization tool to educate and challenge!

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

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 terminal box.

<https://onlywei.github.io/explain-git-with-d3/>

Book

The entire Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, is available here. All content is licensed under the Creative Commons Attribution Non Commercial Share Alike 3.0 license. Print versions of the book are available on Amazon.com.>

<https://git-scm.com/book/ko/v2>



- 개발에 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 이로간된 Git 사용 노력 필요
- 궁금하거나 자주 마주하는 케이스에 대한 Git 사용법은 꼭 연습