Week3 Git&Python

이정수

1. Git

- Git은 소프트웨어를 개발하는 기업의 핵심 자산인 소스코드를 효과적으로 관리할 수 있게 해주는 무료, 공개소프트웨어.(형상 관리 도구의 일종)

- 형상 관리 도구를 사용하면 변경을 쉽게 되돌릴 수 있다. 소스코드를 과거의 특정 시점으로 되돌리거나, 특정 시점의 변경 사항을 취소하거나, 두 버전의 소스 코드를 비교하는 등의 일이 가능하다.

2. Git의 특징

- 팀 프로젝트가 아닌, 개인 프로젝트일지라도 GIT을 통해 버전 관리를 하면 체계적인 개발이 가능해지고, 프로그램이나 패치를 배포하는 과정도 간단해짐. (pull을 통한 업데이트, patch 파일 배포)

2.1 Distributed development

- 전체 개발 이력을 각 개발자의 로컬로 복사본을 제공하고 변경된 이력을 다시 하나의 저장소로 복사한다.

- 이러한 변경은 추가개발지점을 가져와, 로컬개발 지점과 동일하게 병합(merge)할 수 있다.저장소는 Git protocol 및 HTTP로 쉽고 효율적(특별한 웹서버 구성없이)으로 접근할 수 있다.

- 분산 버전관리이기 때문에 인터넷이 연결되지 않은 곳에서도 개발을 진행할 수 있으며, 중앙 저장소가 날라가버려도 다시 원상복구할 수 있습니다.

2.2 Strong support for non-linear development

- 신속하고 편리항 branch 및 merge 지원, 비선형(여러갈래) 개발 이력을 시각화하고 탐색 할 수 있는 강력한 도구를 제공한다.

2.3 Efficient handling of large projects

- Git은 매우 빠르고, 대형프로젝트나 이력이 많은 작업에 매우 합리적이다. Git은 대부분의 다른 버전관리시스템보다 빠르게 요청한다. 그리고 일부 작업에서는 더 빠르게 진행한다.

- 또한, 최근의 정상급 오픈소스 버전관리 시스템보다 장기간의 수정내역을 매우 효율적인 압축 방법을 사용한다.

2.4 Cryptographic authentication of history

- GIt의 이력은 성공한 개발이력의 commit에 의해 개정명으로 저장된다. 일단 그것이 배포되면, 그것을 모르고 예전버전으로 변경하는것은 불가능하다. 또한, 그것들을 암호화 할 수 있다.

3. Git과 Github

-Git은 형상 관리 도구, Github는 웹 호스팅 시스템의 하나

- 협업하고 잇는 코드를 저장하고 버전 관리 시스템을 지원하는 웹호스팅 서비스의 기능을 통해, push, pull request같은 이벤트에 반응하여 자동으로 작업(배포 등)을 실행하게 할 수 있다.

- 웹호스팅 시스템에는 GitHub, GitLab, BitBucket등이 있다.

4. Git관련 기초 용어들

- Repository : 저장소를 의미하며, 저장소는 히스토리, 태그, 소스의 가지치기 혹은 branch에 따라 버전을 저장한다. 저장소를 통해 작업자가 변경한 모든 히스토리를 확인 할 수 있다.

- Working Tree : 저장소를 어느 한 시점을 바라보는 작업자의 현재 시점.

- Staging Area : 저장소에 커밋하기 전에 커밋을 준비하는 위치.

- Commit : 현재 변경된 작업 상태를 점검을 마치면 확정하고 저장소에 저장하는 작업.

- Head : 현재 작업중인 Branch를 가리킨다.

- Branch : 가지 또는 분기점을 의미하며, 작업을 할때에 현재 상태를 복사하여 Branch에서 작업을 한 후에 완전하다 싶을때 Merge를 하여 작업을 한다.

- Merge : 다른 Branch의 내용을 현재 Branch로 가져와 합치는 작업을 의미한다.

git init : 버전관리 하고싶은 폴더에서 초기화를 하는 준비

git branch

- 독립적인 공간을 만든다.

- 새로 만든 branch lab1은 master와 완전히 동일한 상태를 가진 공간.

- 브랜치에서 수정을 한 후 커밋하면 lab1에만 기록되며 master 브랜치에는 어떤 영향도 주지 않는다.

- 원하는 만큼 빠르게 branch를 만들 수 있다.

- 실험 중 다른 브랜치로 돌아가야 할 때 : checkout master 로 head를 옮겨야 한다.

(cf > 작업 중인 위치를 가르키는 가상의 커서가 존재하는데 이를 git에서는 HEAD라 한다.)

- 실험 성공 : lab1 브랜치의 내용을 마스터 브랜치와 병합(Merge) 한다.

- 실험 실패 : lab1 브랜치를 삭제한다.

checkout

- 독립된 작업 공간인 브랜치를 자유롭게 이동할 수 있다

git commit

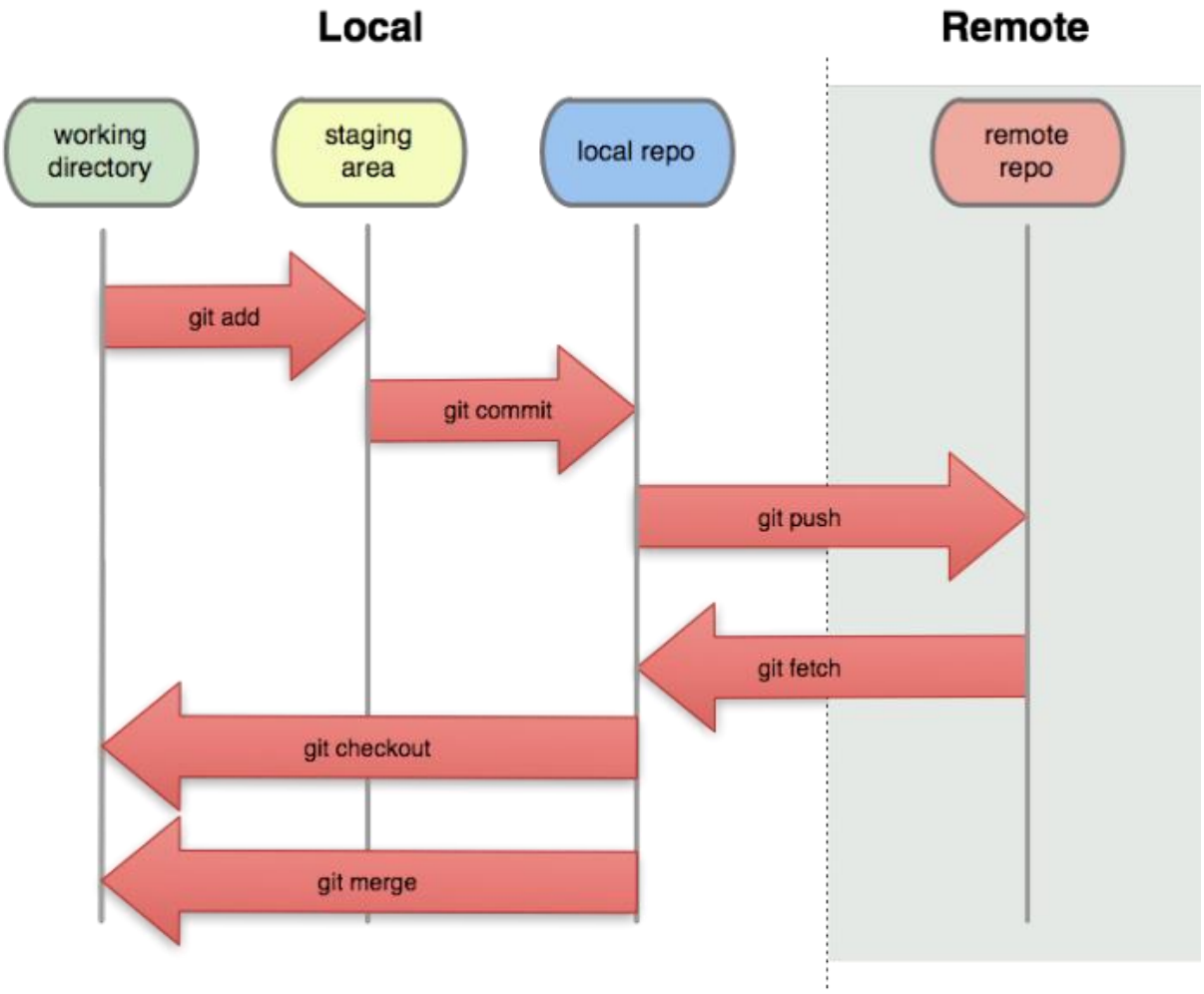
- 의미있는 수정 작업이 끝났을 때 마침을 알리는 작업

pull

- 리모트 저장소의 변경된 내용을 로컬(내 컴퓨터) 저장소에 적용하는 작업을 pull이라 한다

master

- git init을 했을 때, default로 만들어지는 가지가 'master'이다.


Local
Remote
working directory
staging area
local repo
remote repo
git add
git commit
git push
git fetch
git checkout
git merge

Git에 대한 내용

https://goddaehee.tistory.com/91