

170110 – 프로그램의 기본 , git

소스 저장소란 무엇인가?

- 버전 관리 시스템(Version Control System)

간략히 말해서 파일의 변경이력을 기록하여 관리를 용이하게 해주는 것이다.

구체적 이점

변경 이력을 기록하여 변경된 내용을 공유 가능

타인이 작업한 내용을 쉽게 병합

과거 상태로 쉽게 복구 가능

여러 분기(Branch)를 통해 병렬 관리 가능

git 에서의 작업 흐름은

working directory – staging area – git directory(repository)입니다

실제작업공간 working directory, 정류장 staging area, 저장소 git directory(repository)로 분류할 수 있습니다.

정류장인 staging area 에서 이걸 저장소에 올릴지 말지 고민하는게 중요합니다. 왜냐하면 이걸 모든 기록이 남기 때문입니다.

또한 이 정류장에서 어떤 변경 내용만 올릴지 선별적으로 선택이 가능하기 때문도 있습니다.

또한 repository 는 로컬과 리모트로 나뉘게 되는데요.

로컬은 내 컴퓨터안에, 리모트는 인터넷에 저장이 된다고 보면 됩니다.

그리고 gitignore 파일로 stage 영역에 올라가지 않도록 할 수 있는 기능이 있습니다. 어떤 언어를 쓰든, 어떤 프로그램을 쓰든 간에 반드시 무시해야할 파일이 있습니다. 이러한 것을 알기 위해서는 <https://www.gitignore.io/>으로 접속하여 찾도록 합니다.

이것을 올리는 과정은 다음과 같습니다.

working directory – staging area – local repository – remote repository
에서 add – commit – push 과정이고 반대로 remote 에서 local 로 오는 과정은 pull 이라고 합니다.

* 충돌을 해결하는 법

내가 pull 해와서 고치고 있는데, 다른 사람이 그 사이에 내가 고친 부분을 같이 고쳐서 push 해놨습니다. 그래서 이것을 올리기 바로 전에 pull 을 하려고 하면, 이 때 충돌이 발생합니다. 다행인 것은 git 에서 이것을 알아서 관리를 해주기

때문에 당황하지 않고 다시 폴더로 들어가 내 파일을 수정한 뒤, 고치고 다시 commit, 또 다시 pull 뒤에 push 해주면 충돌이 일어난 부분을 부드럽게 넘길 수 있습니다.

독학부분

브랜치! branch 란 무엇일까요?
영어로 나뉘가지입니다.

가지는 안전하게 격리된 상태에서 무언가를 만들 때 사용해요.
여러분이 저장소를 새로 만들면 기본으로 master 가지가 만들어집니다.
이제 다른 가지를 이용해서 개발을 진행하고 나중에 개발이 완료되면 master 가지로 돌아와 병합하면 돼요.

master branch 에서 추가 작업을 해봅시다.

현재 master 와 feature/hotdog 브랜치는 다른 파일 상태를 가지고 있습니다.

서로 격리되어 작업된 두 branch 를 합쳐볼까요??

1. master branch 에서 merge 버튼을 누릅니다.
2. feature/hotdog 만 선택할 수 있습니다.
3. merge 는 두 파일을 자동으로 합쳐주는데 모호성으로 인해 그렇게 하지 못할 때 conflict 가 발생할 수 있습니다.
4. conflict 가 발생한 소스코드는 remote 에서 충돌이 발생했을 때와 똑같은 구조로 가이드가 나옵니다.

따라서 파일을 정리하면 됩니다.

5. 그리고 소스트리로 가보면 이제 commit 할 수 있습니다.

6. commit 이 완료되면 branch 가 이쁘게 합쳐진 것을 확인할 수 있습니다.

branch 정리

독립적으로 작업을 진행하기 위한 개념

기능별로 나누어 협업 가능

각각의 브랜치는 다른 브랜치의 영향을 받지 않기 때문에, 여러 작업을 동시에 진행할 수 있음.

merge 는?

branch 를 합쳐주는 것

하나의 브랜치에 다른 브랜치를 합치는 과정

협업을 위한 필수 기능

명령어 : git merge 브랜치 이름

간단한 터미널 명령어

\$ pwd 현재 경로

\$ ls 해당 디렉토리 안에 무슨 파일이 있는지 list 를 보여줌

\$ mkdir 디렉토리(폴더)만들기

\$ cd 디렉토리 이동

\$ vi 파일추가,또는 수정

git 기본 명령어

git init : 현재 있는 워킹 디렉토리에 .git 폴더 생성

git status : 현재 워킹 디렉토리 상태 확인

git clone : remote 저장소를 통째로 워킹 디렉토리에 복사

git pull : remote 의 정보를 가져와 워킹 디렉토리와 병합

git add : 새로 만들어졌거나 변경된 파일을 워킹 디렉토리에서 stage 에 올리기

git add . : 전부 올리기

git commit : stage 에 올라온 파일을 local repo 에 올리기 +comment 를 남기기(-m "")

git push : remote 로 파일, comment 보내기

* 만약 comment 를 안하면 코멘트 입력하라는 란이 나옵니다. 첫번째줄에 제목, 밑줄 밑에 내용을 쓰면 됩니다.

커맨드 -

:w 저장

i = insert

esc = 나가기

:q 나가기

:q! 저장하지 않고 강제 나가기

:wq 저장하고 나가기

질문

마크다운 문법이 여러 종류가 있는데 어떻게 유효한 것인가? (예 : 제가 본 마크다운문법에서는 ***이 가로줄로 되어있는데 적용이 안되는 경우)

답 - 깃허브에서 지원하는 것을 쓰면 된다.

readme 파일을 수정할 때 md 파일 형식을 어떻게 여는지 몰라서 텍스트에서 수정을 했는데 터미널에서 수정한것을 텍스트로 켜보면 조금씩 다르게 나온다.

어떤 것으로 수정을 하는 것이 맞는 것인지

답 - 마크다운 에디터로 보면 편할 것이다.

터미널로 푸쉬를 할 때 왜 -u origin master 인지

답 - 브랜치 개념을 좀 더 이해하면 될 것이다(나중에 배우자)

왜 모두다 master 브랜치에 있는지, 그럼이게 다 합쳐진 것인지?

답 - 이것도 나중에 알 수 있는데, 나중에 선별적으로 합칠 수 있다고 알면 된다

터미널에서 한 것을 소스트리에서 보려면?

답 - 오픈으로 열면 된다