3-2. 브랜치 만들기

\* 실습 상황 설정하기

먼저 상황을 설정해보자. 먼저 디렉터리를 만들고 그 안에 사용 설명서라고 생각하고 텍스트 파일을 하나 만들자.

1. 터미널 창을 열어 홈 디렉터리에 manual이라는 새 디렉터리를 만들고 해당 디렉터리로 이동한다.

2. manual 디렉터리를 저장소로 만들고 ls -al 명령을 사용해 .git 디렉터리가 만들어 졌는지 확인한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

3. manual 디렉터리 안에 work.txt 파일을 만들어 보자. 그리고 ‘content1’이라는 내용을 입력하고 commit(commit message: work1)을 수행한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

4. 이후 ‘content2’, ‘content3’를 추가해 각각 commit(각 commit message: work2, work3)을 수행해 준다.

5. git log를 이용해 commit 내역을 확인해보자.

텍스트이(가) 표시된 사진

자동 생성된 설명

master branch가 가장 최신 commit인 ‘work3’를 가리키고 있고, HEAD가 master branch를 가리키고 있다. **HEAD는 여러 branch중에서 현재 작업 중인 branch를 가리킨다.**

\* 새 branch 만들기

이제부터 직접 branch를 만들면서 위와 같은 문제를 어떻게 더 세련된 방법으로 처리하는지 알아보자. apple, google, ms라는 고객사가 있다고 가정하자.

1. 깃에서 **branch를 만들거나 확인하는 명령**은 **git branch**이다.



위처럼 master라고 나타난다. master는 저장소를 만들 때 기본적으로 만들어진다. 우리가 지금까지 작업했던 branch가 master branch이다.

2. **새로운 branch를 만들려면 git branch 명령 다음에 만들려는 branch 이름을 적는다.**



apple branch가 만들어져도 화면에는 아무것도 출력되지 않는다. git branch 명령을 이용해 확인할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**master branch 위에 apple branch가 추가된 것을 확인할 수 있다.** master앞에 \*표시는 우리가 master branch에서 작업하고 있다는 의미이다.

3. **branch가 추가된 후에는 commit log 화면도 다르게 나타난다.** **git log 명령**을 사용해 commit을 확인해보자.

텍스트이(가) 표시된 사진

자동 생성된 설명

HEAD -> master라고 표시되었던 부분이 **HEAD -> master, apple**로 바뀌었다. 이 표시는 **저장소에 master, apple 2개의 branch가 있고**, **HEAD -> master이므로 현재 작업 중인 branch는 master branch라는 의미**이다.

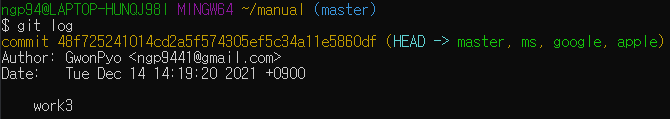
4. 이전과 같은 방법으로 google branch와 ms branch를 만들어 보자. 그리고 git branch 명령으로 저장소 안에 있는 모든 branch를 확인하자.

텍스트이(가) 표시된 사진

자동 생성된 설명

**\* branch 사이 이동하기 - git checkout**

1. git log로 확인해 보면 apple, google, ms branch가 ‘work3’ commit 상태에서 만들어진 것을 알 수 있다. 즉, master branch 뿐만 아니라 ms, google, apple branch에도 최신 commit이 ‘work3’라는 뜻이다.



2. **현재 우리는 master branch에 있다. commit을 하나 만들어서 어떻게 달라지는지 확인해보자.** work.txt파일 안에 ‘master content 4’라는 내용을 추가하고 commit(commit message: master content 4)을 해보자.

3. git log 명령에 **--oneline 옵션**을 추가해서 확인해보자. --oneline 옵션은 **한 줄에 한 commit씩 나타내 주기 때문에 commit을 간략히 확인할 때 편리하다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

최신 commit인 ‘master content 4’는 master branch에만 적용되어 있다. ms, google, apple branch는 아직 ‘work3’ commit 상태이다.

4. **현재 branch에서 다른 branch로 이동하면 어떻게 변하는지 살펴보자.** **다른 branch로 이동하려면 git checkout 명령 다음에 branch 이름을 사용**한다. 그래서 apple branch로 이동하는 것을 **‘apple branch로 checkout한다’**고 말한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

git checkout 명령을 사용하고 나면 $위에 나타나는 **파일 경로 끝에 (apple)이라고 표시**된다. 현재 branch가 apple이라는 뜻이다.

5. 작업 branch를 apple branch로 바꿨을 때 어떤 변화가 생기는지 확인해 보자.

텍스트이(가) 표시된 사진

자동 생성된 설명

master branch에서 apple branch로 **분기하기 전까지 master branch에 있던 commit들은 그대로 apple branch에 복사된 것을 확인할 수 있다.** 그리고 최신 commit hash에서 **HEAD가 apple을 가리키고 있다.**

6. apple branch의 최신 commit은 처음 분기될 때인 ‘work3’ commit이다. 그럼 work.txt 파일에는 무엇이 있을까?

텍스트이(가) 표시된 사진

자동 생성된 설명

최신 commit이 work3이기 때문에 content1부터 content3까지 세 개의 행만 가지고 있는다. 즉, apple branch가 master branch에서 **분기된 이후에 master branch에 추가된 commit은 apple branch에 영향을 미치지 않는다.**