3-3. 브랜치 정보 확인하기

이번 장에서는 **여러 branch에서 각각의 commit이 이루어질 때 commit끼리 어떤 관계를 하고 있는지 확인하는 방법**과 **branch 사이의 차이점을 확인하는 방법**을 알아보자.

**\* 새 branch에서 commit하기**

apple branch에 새로운 commit을 만들어보자.

1. work.txt파일에 **‘apple content 4’라는 텍스트를 추가하고 저장**하자.

그리고 **apple.txt라는 새로운 파일을 만들고 여기에도 똑같이 ‘apple content 4’라는 텍스트를 입력**하고 **commit(commit message: apple content 4)을 수행**하자.

2. commit이 어떻게 저장되었는지 **git log 명령을 사용해 확인**해보자.

텍스트이(가) 표시된 사진

자동 생성된 설명

**HEAD->apple**이라고 되어 있으므로 현재 **apple branch에 checkout한 상태**이고, **apple branch의 최신 commit은 ‘apple content 4’이다.**

3. git log 명령을 사용할 때 **--branches 옵션**을 사용하면 **각 branch의 commit을 함께 볼 수 있다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

**branch와 commit의 관계를 더 쉽게 그래프 형태로 표시**하려면 git log 명령에 **--graph 옵션**을 함께 사용한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

commit 내역의 왼쪽에 **수직선은 commit과 commit의 관계를 보여주는 것**이다. graph를 살펴보면 **apple branch의 최신 commit은 ‘apple content 4’인데, 점선을 따라 commit의 부모를 찾아가 보면 ‘work3’ commit을 만나게 된다.** 즉, apple branch에서는 ‘work3’ commit 다음에 ‘apple content 4’ commit이 만들어졌다는 뜻이다.

master branch의 최신 commit은 ‘master content 4’이다. 그리고 수직선을 따라가 부모 commit을 찾아보면 ‘work3’ commit이 나온다. **apple branch의 commit과 master branch의 commit이 같은 부모 commit을 가진다.** **즉, master branch나 apple branch는 ‘work3’ commit까지는 같고 그 이후부터 branch마다 다른 commit을 만들었다는 사실을 알 수 있다.**

**\* branch 사이의 차이점 알아보기**

**branch마다 commit이 쌓여갈수록 branch 사이에 어떤 차이가 있는지 일일이 확인하기 어렵다.** 이럴 때는 **branch 이름 사이에 마침표 두 개(..)를 넣는 명령**으로 차이점을 확인할 수 있다. **마침표 왼쪽에 있는 branch를 기준으로 오른쪽 branch와 비교한다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

이렇게 하면 **master branch에는 없고 apple branch에만 있는 commit을 보여준다.** 따라서 위에서는 ‘apple content 4’를 보여준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

반대로 apple branch를 기준으로 master branch와 비교하면 ‘master content 4’ commit을 보여준다.