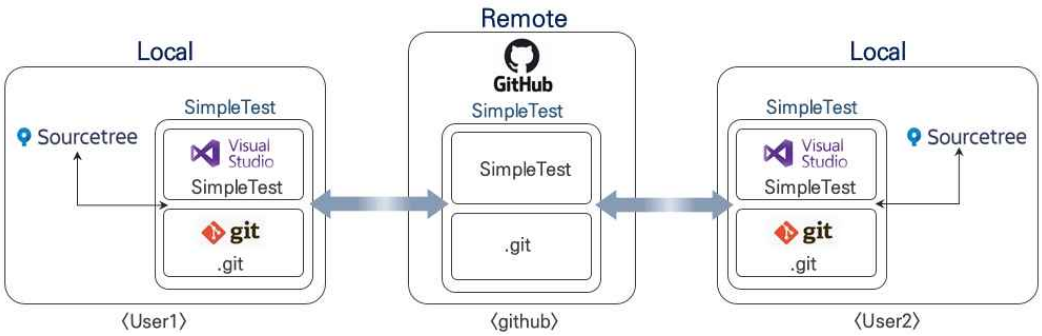
Git을 사용한 협업

1. Git을 이용한 협업이란
   - how to git clone
   - must be pull before work
2. 협업시 충돌 해결방법

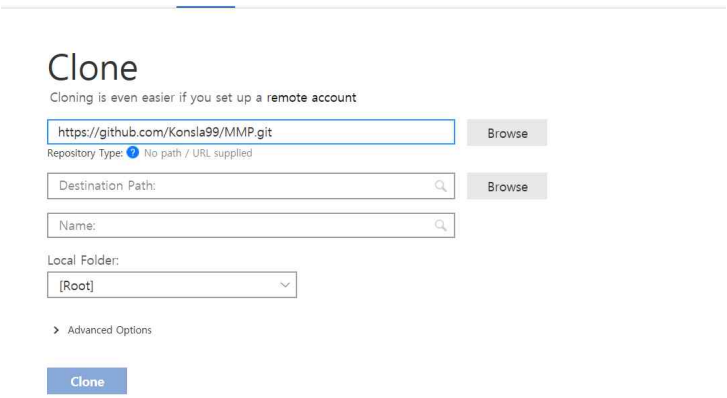
Git 협업

remote repository에는 여러명이 접속하여 협업하는 것이 가능하다.

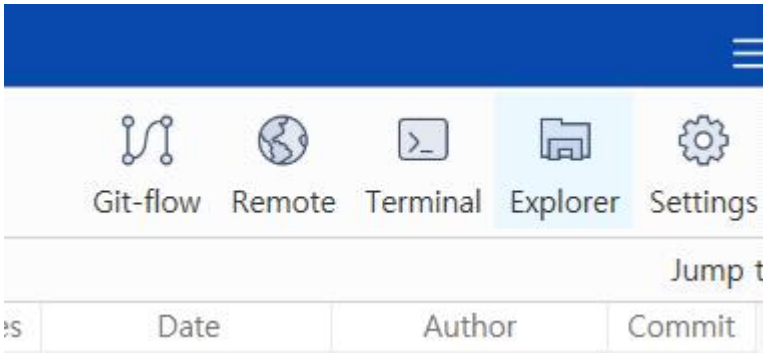


저장소 복제

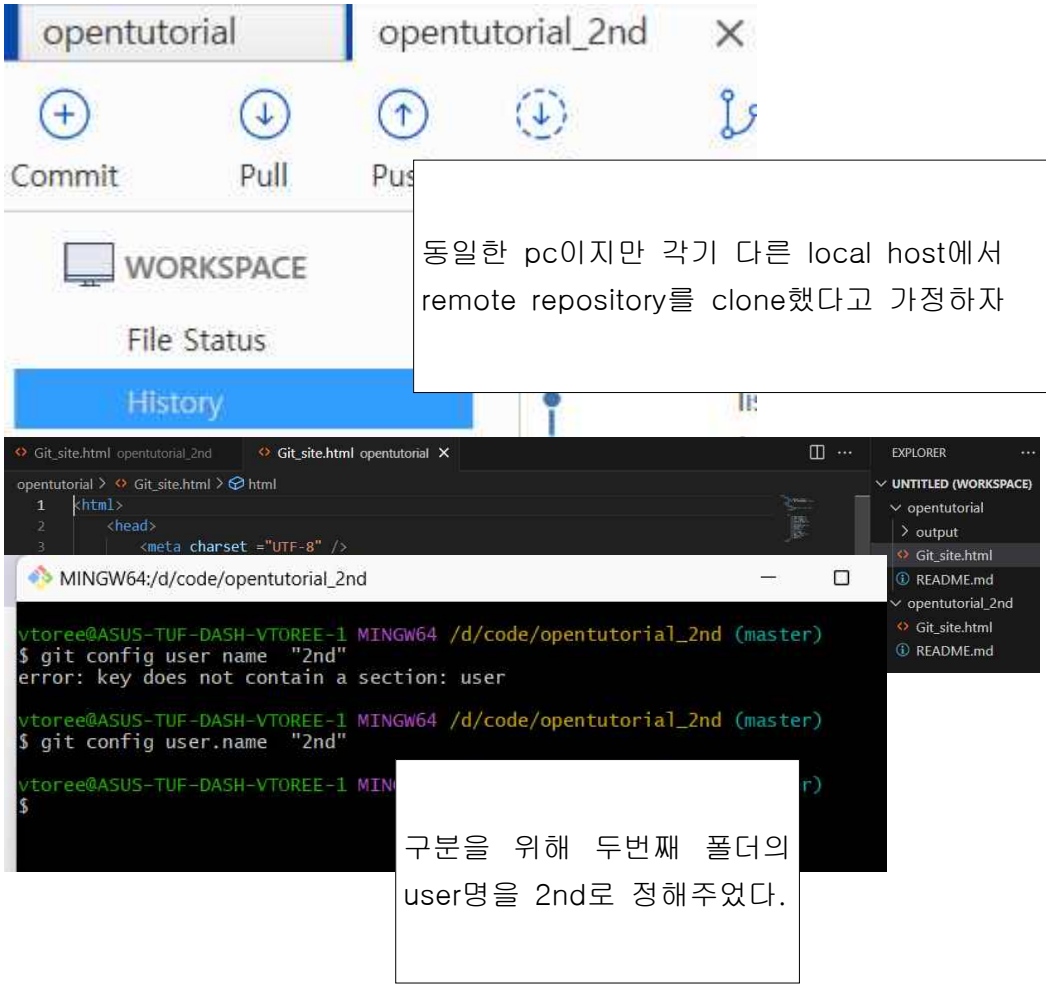
새로추가된 멤버는 우선적으로 본인의 local 환경으로 remote 환경의 내용을 가져올 필요가 있다.



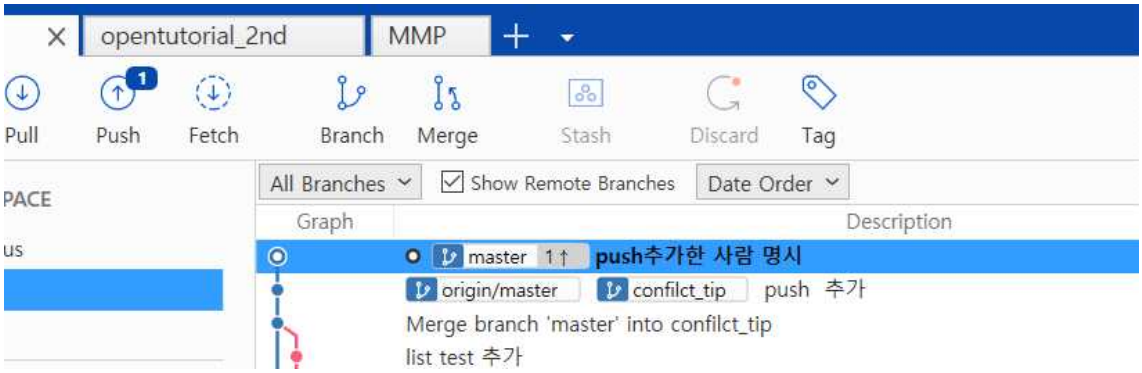
clone을 통해 remote repository를 가져온다



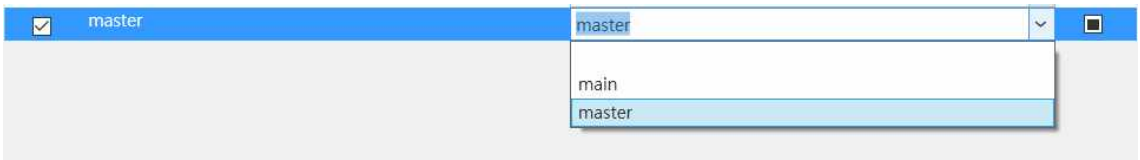
explorer선택시 clone한 파일을 확인 할 수 있다.



Konsla 유저에서 수정



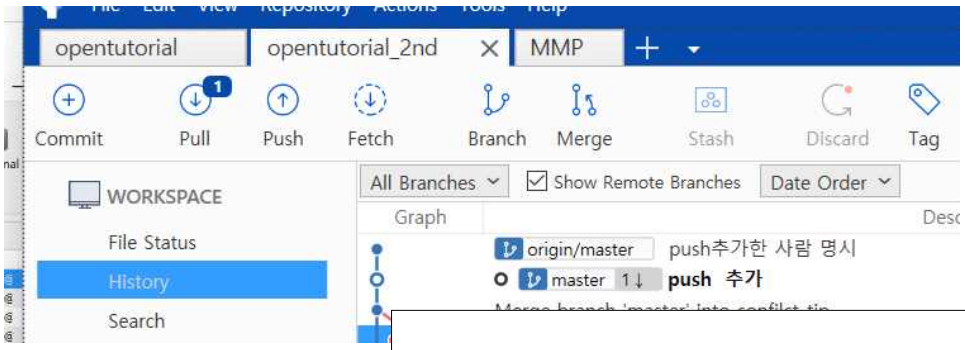
remote와 현재commit사항의 차이가 1단계 존재하는것을
Branch의 위치, push 옆의 숫자를 통해 확인 할 수 있다.



local의 master branch에서 remote의 master로 push

remote저장소에서 확인 가능하다.

습관적인 PULL



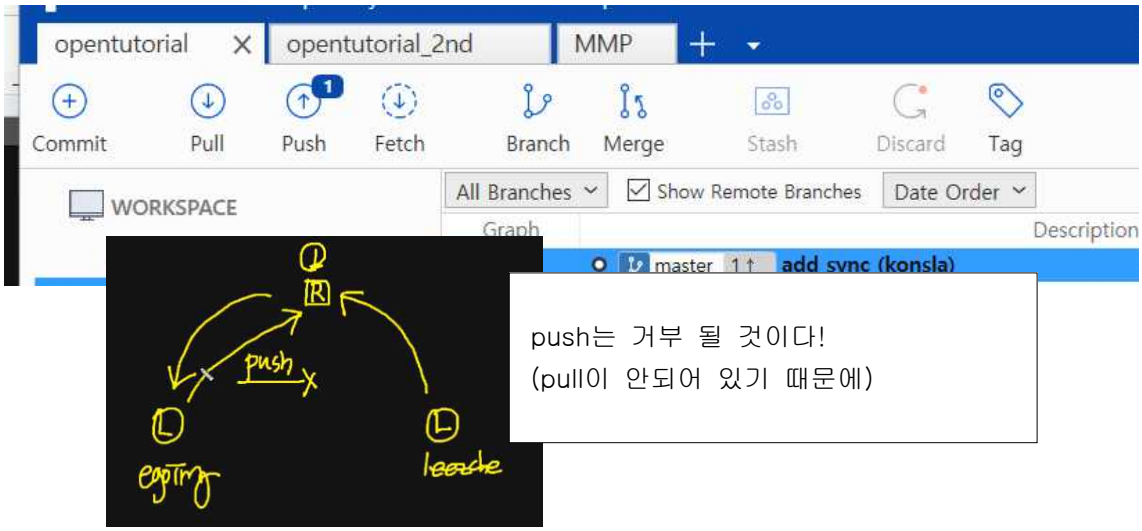
2nd 유저가 추가적인 작업을 진행하려 할 때 pull을 항상 해주는 습관이 필요하다.

pull을 하여 remote저장소의 내용과 일치하는 상태에서 작업을 진행해 주어야 한다.

```
        <li>충돌해결</li>
        <li>원격 저장소!</li>
        <li>version</li>
        <li>push</li><!--konsla-->
        <li>pull</li><!--2nd-->
    </ul>
</body>
```

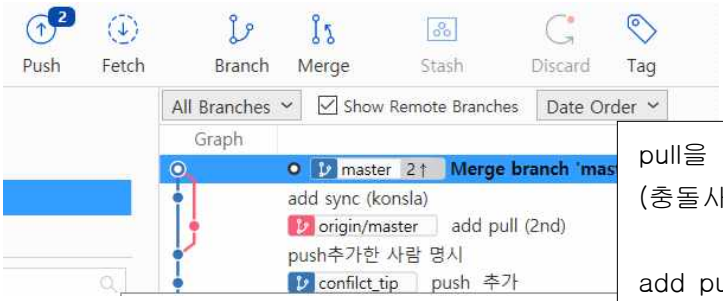
2nd 유저가 pull 이라는 내용을 추가 commit후 push까지 완료

이상황에서 konsla가 pull없이 내용을 추가한경우



push는 거부 될 것이다!
(pull이 안되어 있기 때문에)

pull을 안했기 때문에 에러



pull을 해주면 자동 병합이 된다.
(충돌사항이 없다면)

add pull은 2nd의 작업
add sync는 konsla의 작업

원격 저장소가 존재하고 협업을 한다면

1. 반드시 작업하기 전 pull을 해 동기화 해줄 것!
2. 본인의 작업을 하고 commit을 한다.
3. push하기전 다시 pull을 한다(동료의 추가자료가 있는지 확인하기 위해)

협업시 충돌 해결

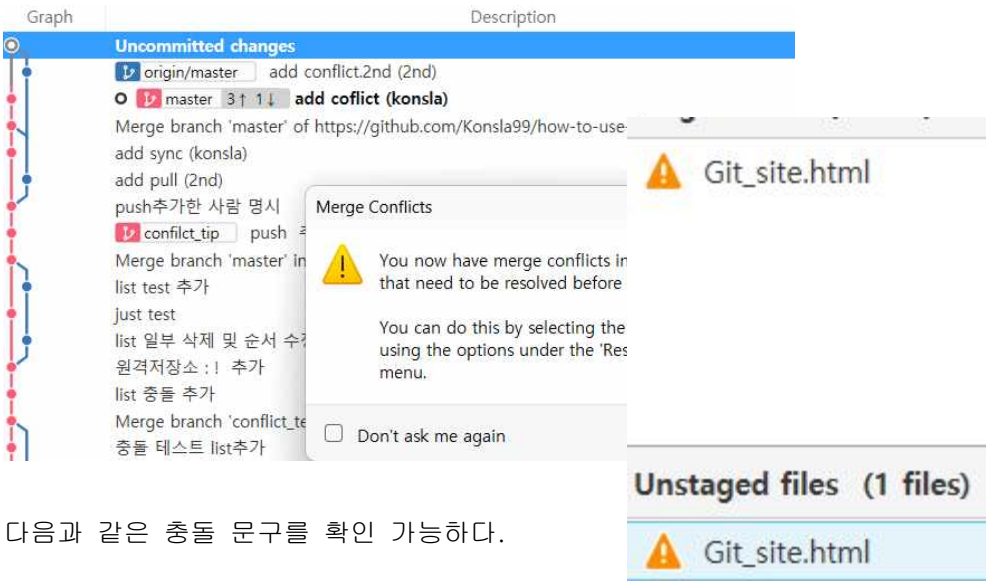
혼자 local에서 작업할 때 동일한 부분을 수정하면 conflict가 발생한것과 동일하게
협업하는 co-worker와 동일한 부분을 수정하게 되었을 경우
자동으로 merge되지 않고 conflict가 발생한다.

```
<li>version</li>
<li>push</li><!--konsla-->
<li>pull</li><!--2nd-->
<li>conflict</li><!--konsla-->
```

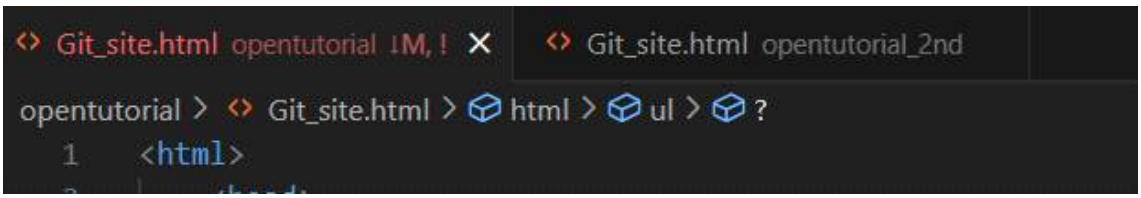
```
<li>version</li>
<li>push</li><!--konsla-->
<li>pull</li><!--2nd-->
<li>conflict.2nd</li><!--2nd-->>
```

두 작업자가 수정한 part가 겹치는 상황
2nd가 먼저 push했다고 가정

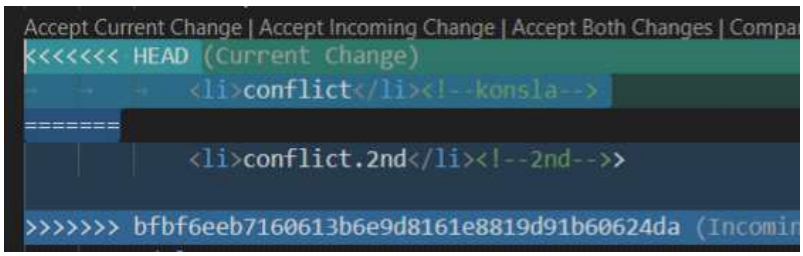
2nd의 push이후
해당 사항을 모르는 konsla는 커밋한 내용을 push하기위해 pull을 진행



다음과 같은 충돌 문구를 확인 가능하다.

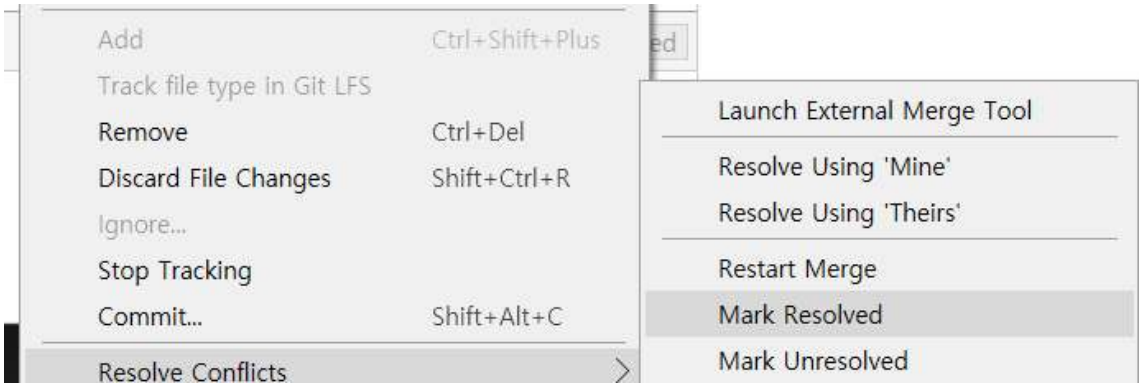


2nd와 다르게 konsla의 작업환경은 충돌표시가 확인된다.

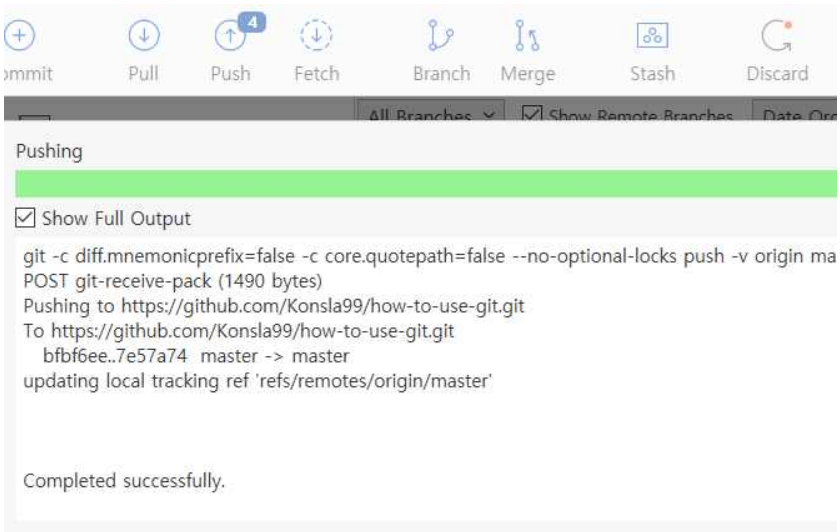


==========를 기준으로 위쪽의
<<<<<<<부분의 본인의 내용

>>>>>> 의 내용은 their



기존과 동일하게 conflict를 해결한다.



conflict를 해결하니 비로소 push가 진행되는 것을 확인 할 수 있다.

conflict를 해결할 책임은 push를 늦게 한사람에게 온다,
이를 유의하자!

자주 동기화를 해주어 conflict를 적게, 작은 단위에서
발생 하도록 하자!