

Git을 활용해보기

1. 비교, 병합 외부 도구 연결

2. 협업시 충돌 해결방법

3. Tag

4. 보안, 환경 파일의 관리

1. 비교, 병합 외부 도구 연결

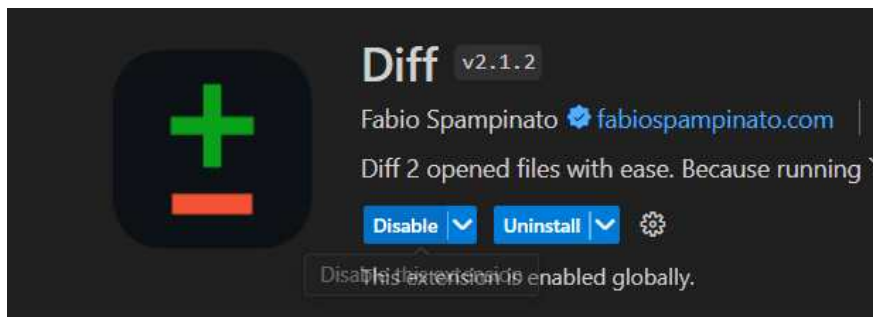
```
Hunk 1 : Lines 25-32
<li>version</li>
<li>push</li><!--konsla-->
<li>pull</li><!--2nd-->
- <li>conflict</li><!--konsla-->
+ <li>conflict.2nd</li><!--2nd-->
+
</ul>
</body>
```

source tree에서 기본으로 제공하는 비교 도구

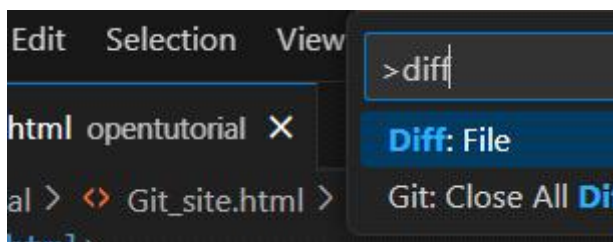
복잡한 차이점이나 conflict 발생시 전문적인 도구를 사용하는 것이 좋을 수 있다.

diff

difference의 약자 파일간의 차이점을 비교해주는 S/W



vscode전용 tool 설치



ctrl + shift + p

-> diff

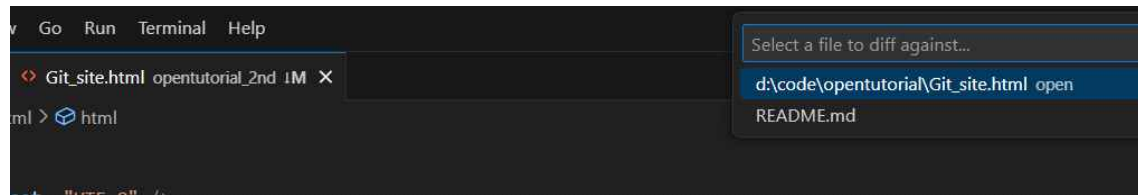
명령 팔레트 활성화

주의사항!

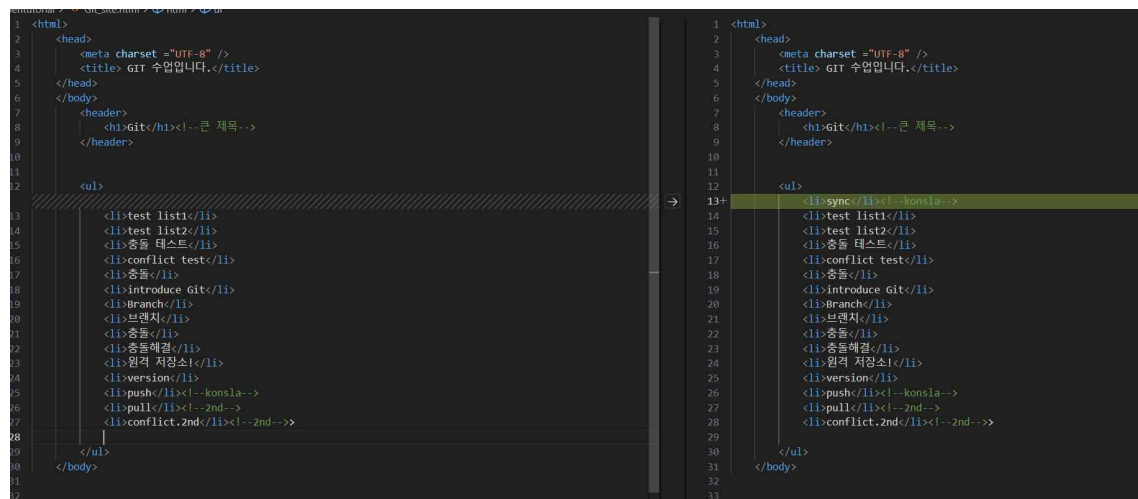
비교하고자 하는 대상이 활성화 되어 있어야 한다.

ex) A,B두개의 파일을 비교하려고 할 때 둘중 하나의 파일은 vscode에 열려있어야 한다.

경로지정



ex) 2nd의 작업 환경에서 konsla의 작업물을 열어 비교



다른 내용을 확인 가능

Red = 삭제

Green = 추가

.... to be continue

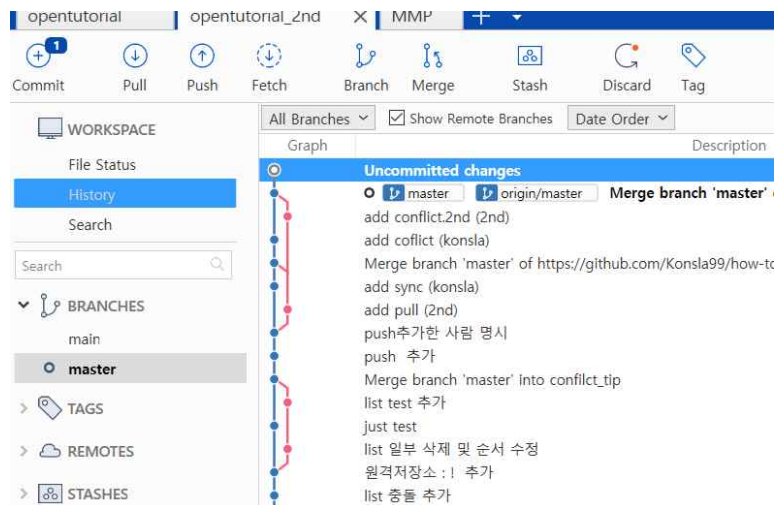
2. Stash<안전한 곳에 보관>

끝나지 않은 작업을 안전한 곳에 보관

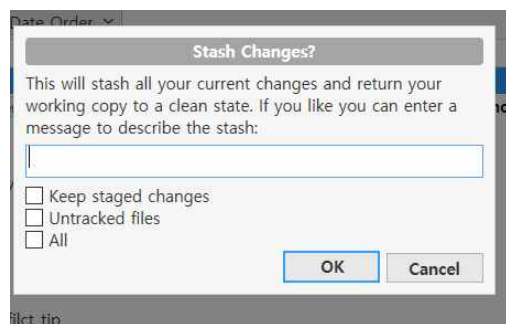
if 작업을 진행도중 bug 발생

완성하지 않은 코드로 인해 bug해결 불가능한 상황 발생 가능

기존에는 해당 문제를 해결하기 위해
branch를 만들어 작업후 merge하는 방법을 사용함



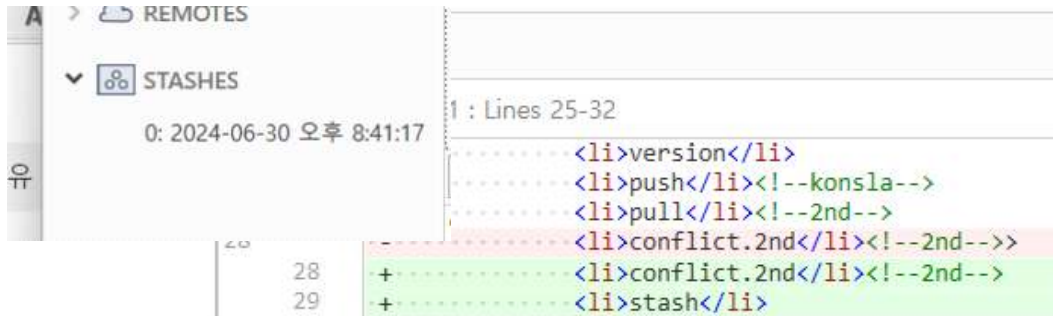
커밋하지 않은 stage에 올라가지도 않은 미완성의 작업



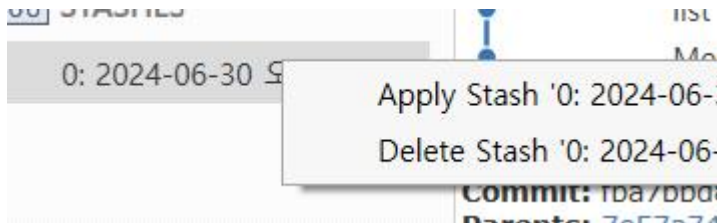
커밋처럼 임시저장 내용적어서 저장가능



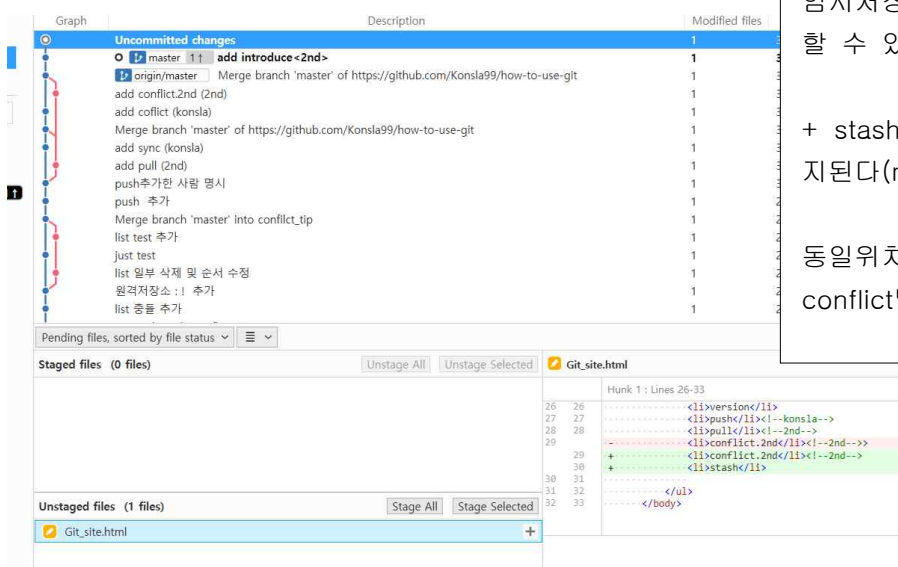
uncommitted change 삭제



stash에서 확인이 가능하다!



우클릭하면 볼 수 있는 2가지 option



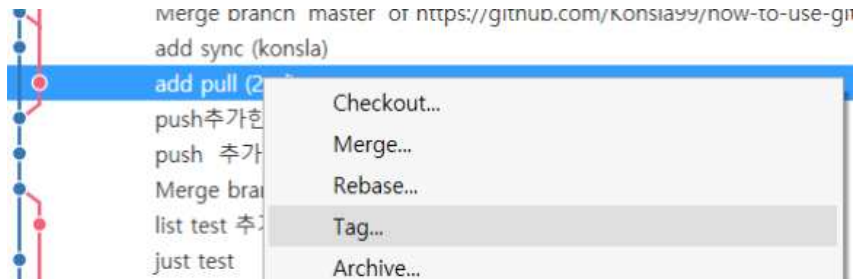
임시저장한 내용이 불러와진 것을 확인할 수 있다.

+ stash 이후 수정한 내용도 그대로 유지된다(merge된 것으로 추정)

동일위치를 수정하고 stash를 apply하면 conflict발생

3. Tag <버전들 안에서 의미있는 내용을 표기>

내부의 버전에서 별도의 추가적인 내용을 기술하여 특별히 관리할 필요가 있는 경우



원하는 커밋 버전에서 우클릭하여 tag클릭

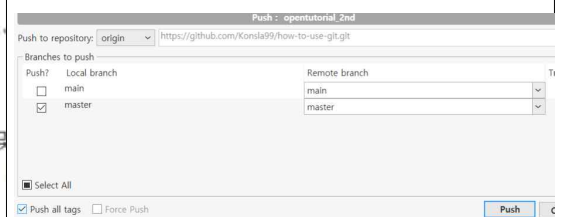


specified commit
커밋 아이디



Tag 추가된 것을 커밋내용과
Tags 메뉴에서 확인 가능하다.

Tag는 push가 안된다!



**반드시 push all tags를 check 해야
push가능**

해당 버전을 이후 추가관리 할 때

해당 버전용 branch를 만들어 주는 것이 추천된다!

4. 보안, 환경 파일의 관리

