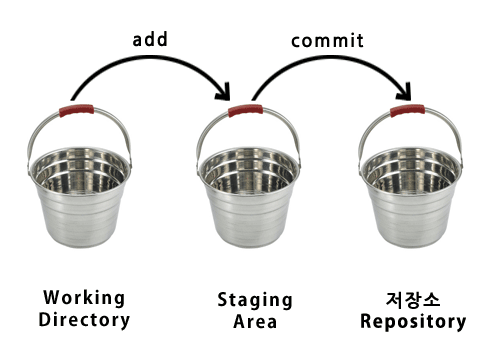
**Git**

**1. 개념**

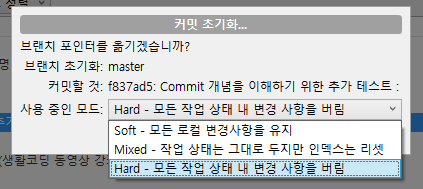


**2. Discard(폐기)**

**3. 위험한 기능(Reset)**

**(영문 : Reset current branch to this commit)**

**선택하면 다음 창이 뜸**

****

**극악의 모드가 Hard!!! 🡪 그러면 이 프로젝트는 날아간다!(따로 Backup이 필요한 이유!)**

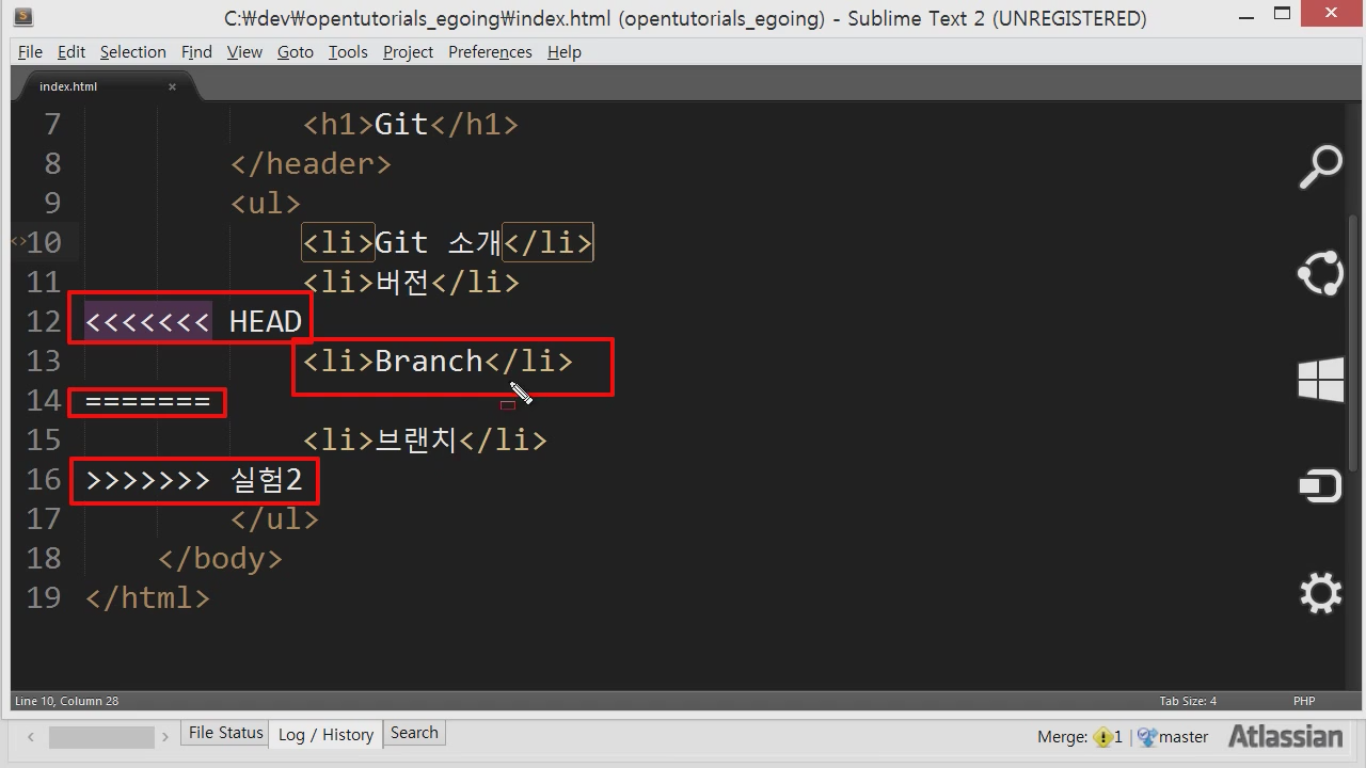
**Mixed : 직전의 Commit Version만이 문제인 경우 유용할 수 있다. 현재 새로 Commit할 변경 작업 내용은 유지됨.**

**4. Revert는 버전별 변경사항을 삭제하지 않고 특정 버전까지 역순으로 돌아갈 수 있는 기능임**

**5. Merge Conflict 처리**

**동일한 부분에 두 Branch에서 모두 수정 작업을 진행한 경우(비일비재!!!)**

**Merge Conflict 경고창 🡪 Uncommitted changes 🡪 Source Code**

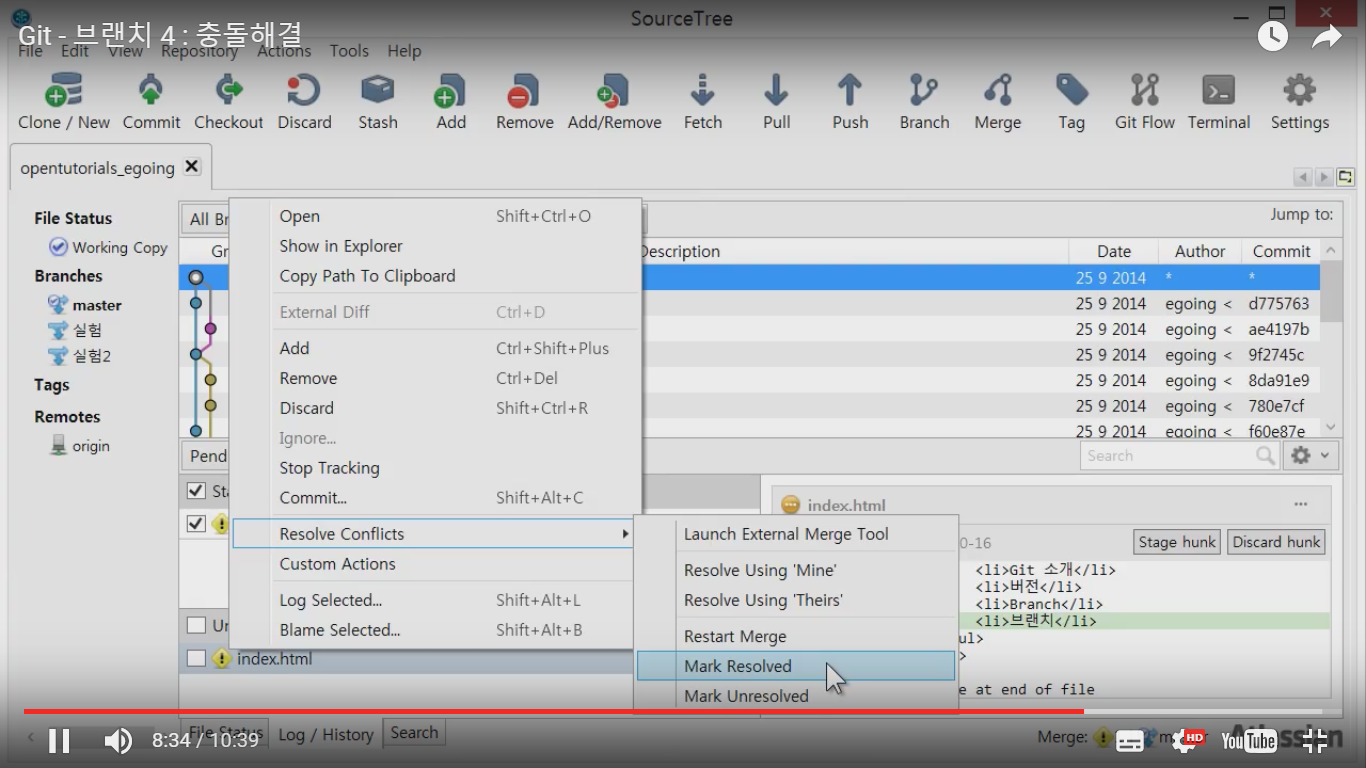
****

**<<<<< Head : 현재의 Main Branch(Check out branch)**

**====**

**>>>>> 실험2 : 2nd Branch(Merge AAA into current branch)**

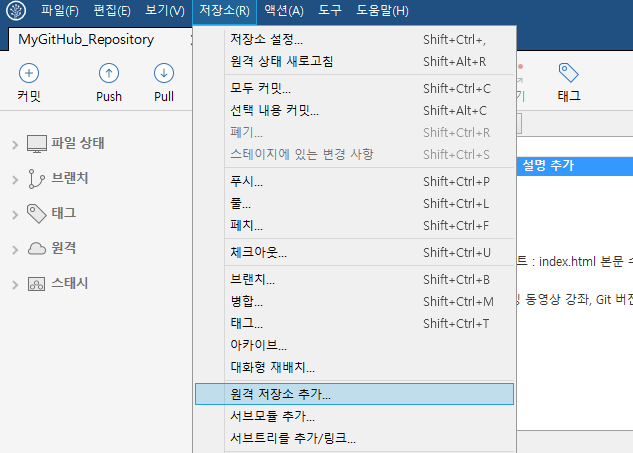
충돌 Text에서 편집 작업 후 저장하면 된다.



그리고…Git에 충돌 해결을 알려준다.

R-Click 🡪 Resolve Conflicts 🡪 Mark Resolved 확인

**6. 현재 PC에 설치된 버전관리시스템(Git)을 GitHub과 연동시켜 웹에 원격저장소를 만들고(GitHub.com) 저장/동기화한다.**

****

# GitHub에 가입하여 1st Project로 만든 저장소(Repository)는 gitMy1stProject

# Source Tree의 원격 저장소 추가(Add remote…)

# 

# 성공!!!(추후 Bitbucket이나 Yobi도 추가할 수 있다)

# 

# 원격 메뉴(Remotes) 아래에 origin 이름의 원격 저장소 생성된 것을 볼 수 있다.

# 동기화 위해선 Push(Local -> Remote), Pull(Remote -> Local)가 필요하다.

# (저장소 > 푸시/풀)

# 에러가 떨어지면 먼저 pull을 한 후에 push를 한다.

# 성공!!!

# 원격 저장(Backup)보다 더욱 중요한 것은 문서, 소스코드별 버전 관리가 가능해졌다는 점이다.

# 추가 기능들도 계속 확인 요(Code, Wiki, Pulse, Graphs 등)

# 

# 

# Branch를 여러 개 만든 경우 추가로 나열된다.

# 

# 최초 연동시

# GitHub Login이 필요하다.