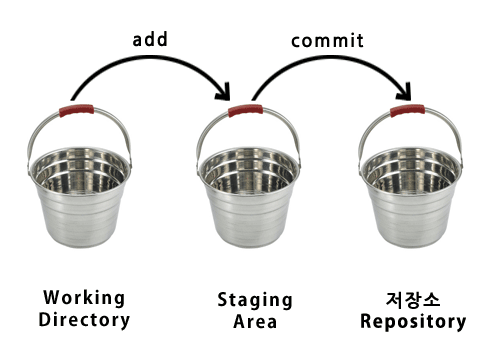
**Git & GitHub**

**1. 개념**

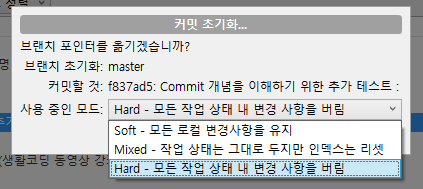


**2. Discard(폐기)**

**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!)**

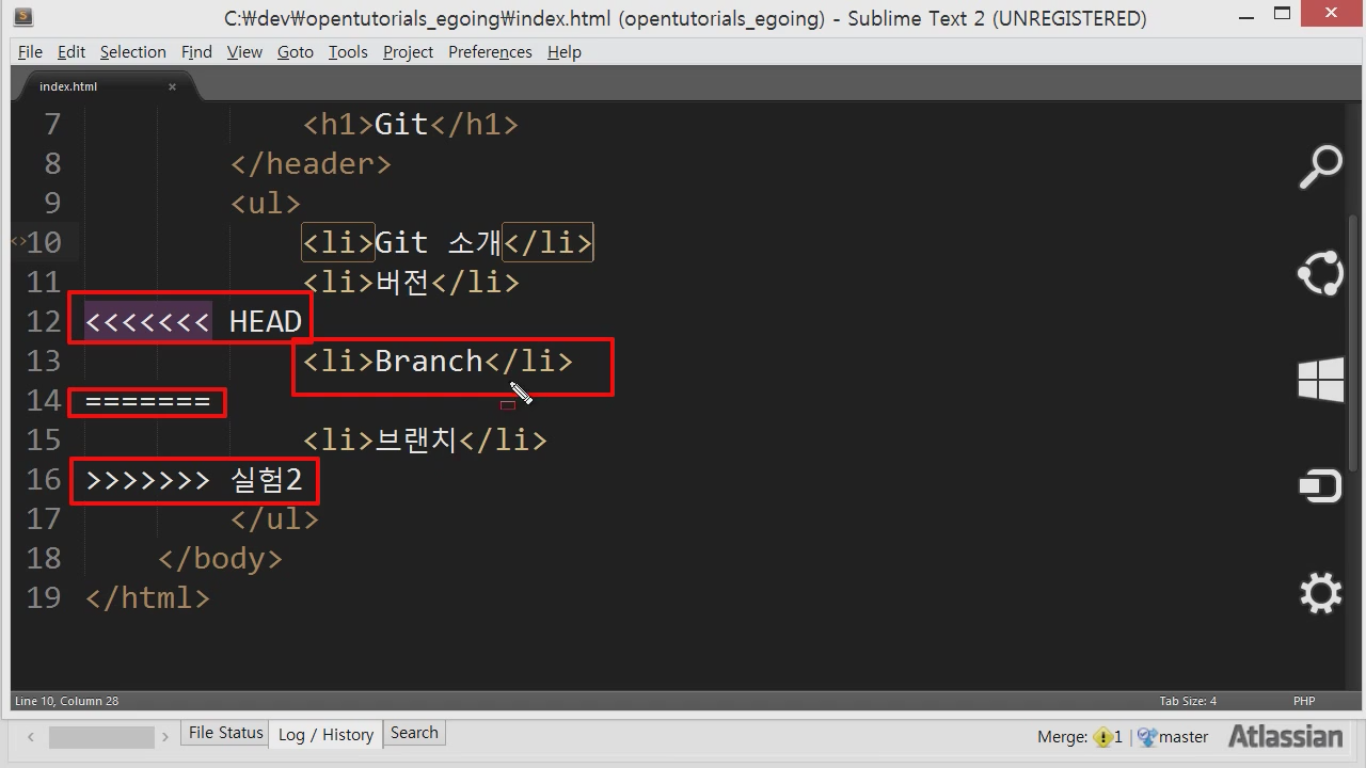
**Mixed : 직전의 Commit Version만이 문제인 경우 유용할 수 있다. 현재 새로 Commit할 변경 작업 내용은 유지됨.**

**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**

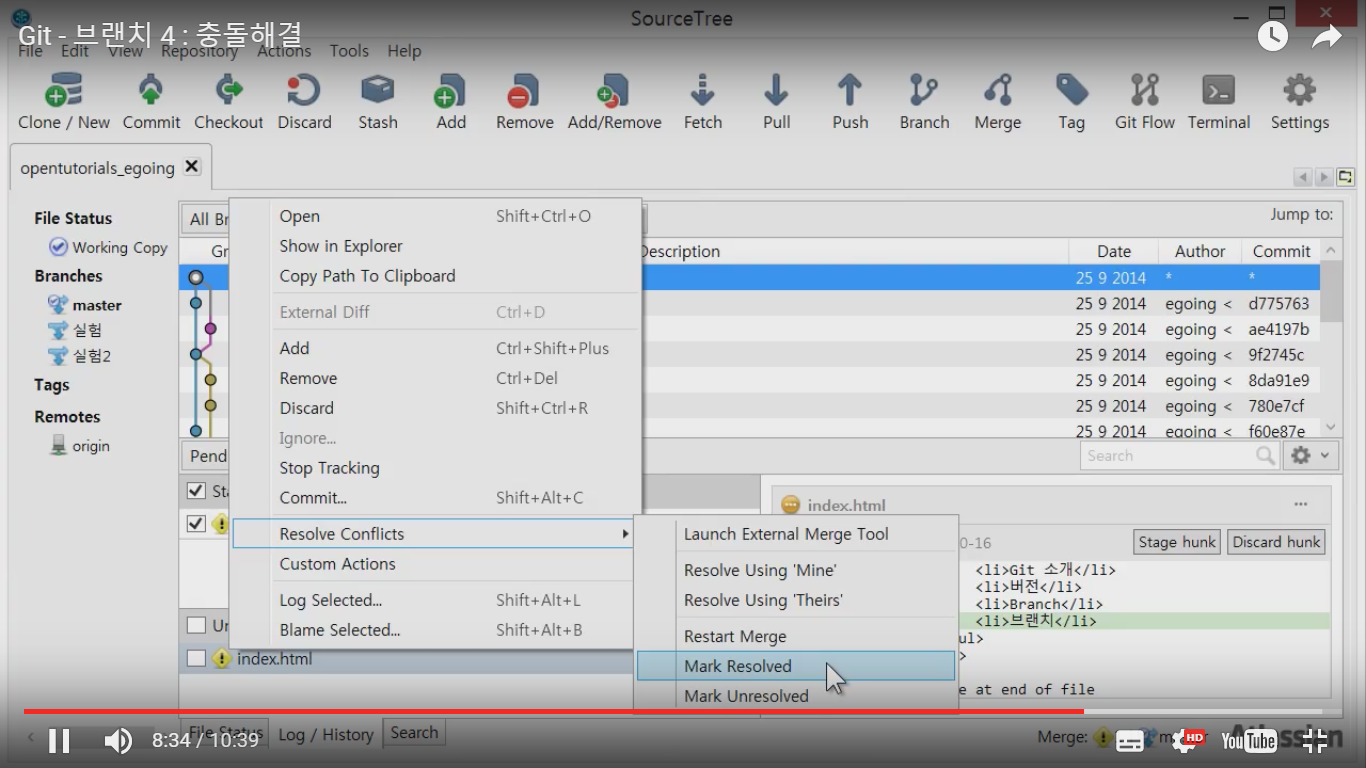
****

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

**====**

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

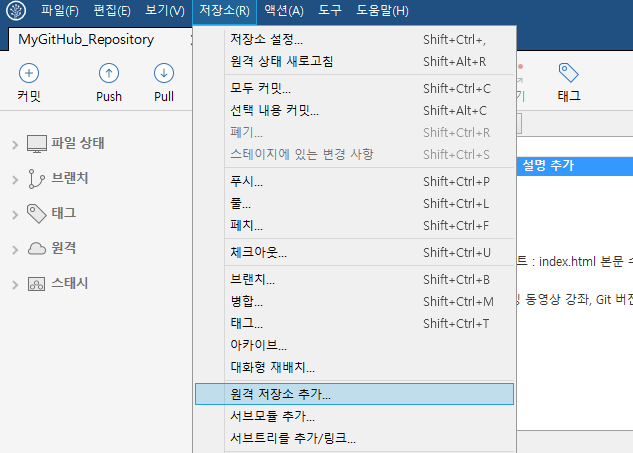
충돌 Text에서 편집 작업 후 저장하면 된다.



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 Yobi(네이버에서 제공🡪서비스 폐기됨)도 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.

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

# 

# 

# Branch를 여러 개 만든 경우 추가로 나열된다.

# 

# 최초 연동시

# GitHub Login이 필요하다.

# 연동된 상태에서 Local 저장소의 변화 발생하면

# 

# 이렇게 표시된다. Push해야 할 변경사항(누적분), Remote와 Local master의 버전 차이(최신 버전 상태) 정보

# 이런 식으로 동기화해서 관리한다.

# <https://github.com/Kilsan7/gitMy1stProject>에서도 동일한 버전 변경 이력을 확인할 수 있다.

# 현재 github의 원격저장소는 두 개(gitMyRepository와 Nexus7\_GitHub\_Repository)

# GitHub에서 SSH Key Pair 생성(Private Key, Public Key 🡪 이걸 Copy해서 Git에 등록함), Git은 SSH방식을 권장한다고 함.

# 

# 1. Git bashk 실행 🡪 Key Generate하여 Public key를 GitHub에 등록한다.

# 

# 암호문(passphrase)을 추가할 수 있다. 안 해도 된다. 안 넣었다.

# 

Public Key 생성됨

Private Key 생성됨

# 2. Public Key 내용을 Copy하여 GitHub에 등록한다.

# Public Key 위치 : 윗 화면 참조

# 

# 메모장으로 읽어서 Copy한다.

# 

# 다시 GitHub으로 와서 Key를 등록한다. Title은 임의로 넣어준다.

# 

# GitHub에 가입할 때 입력한 암호로 확인을 한다.

# 

# 등록 완료(!)

# 

# SourceTree라는 간접적인 방식이 아니라, Git이 제공하는 직접적인 Tool인 GitGui(Git CMD, Git Bash는 Command Line Interface)를 이용하여 GitHub의 원격저장소를 관리하는 작업을 시작해 보자.

# 먼저 GitGui를 열고

# 

# Open Existing Repository를 Click한다.

# 

# 로컬 저장소를 추가한다.

# 

# 또 하나의 로컬 저장소를 추가한다.

# 그러면

# 

# 이렇게 세 개의 로컬 저장소가 표시된다.

# 이 중 첫 번째를 클릭하면

# 

# Working Copy Zone(Unstaged)에 아직 업로드하지 않은 파일들이 리스팅되어 있는 게 보인다.

# 

# 한글은 깨진다. Commit 메뉴에서 Stage To Commit을 눌러 Working Copy 영역의 파일들을 Stage에 올린 후 Commit한다. Commit Message는 필수.

# 

# 그러면

# 

# 이렇게 화면엔 아무것도 남지 않는다.

# GitHub로 가서 보면

# (1) 선택가능한 Repository 중 gitMy1stProject Click

# 

# (2) 응…? GitGui에서 Commit한 것이 보이지 않는다.

# 

# (3) 아! Push를 해줘야 원격저장소에 업로드가 끝난다!

# 

# 성공!

# (4) GitHub에서 확인하면

# 

# 성공적으로 업로드되고 버전관리가 됨을 알 수 있다!

# 이제 Binary File이나 png, project 설정 파일 등은 제외하고 Source Code나 문서들만 관리하도록 제한을 가하면 되겠다.