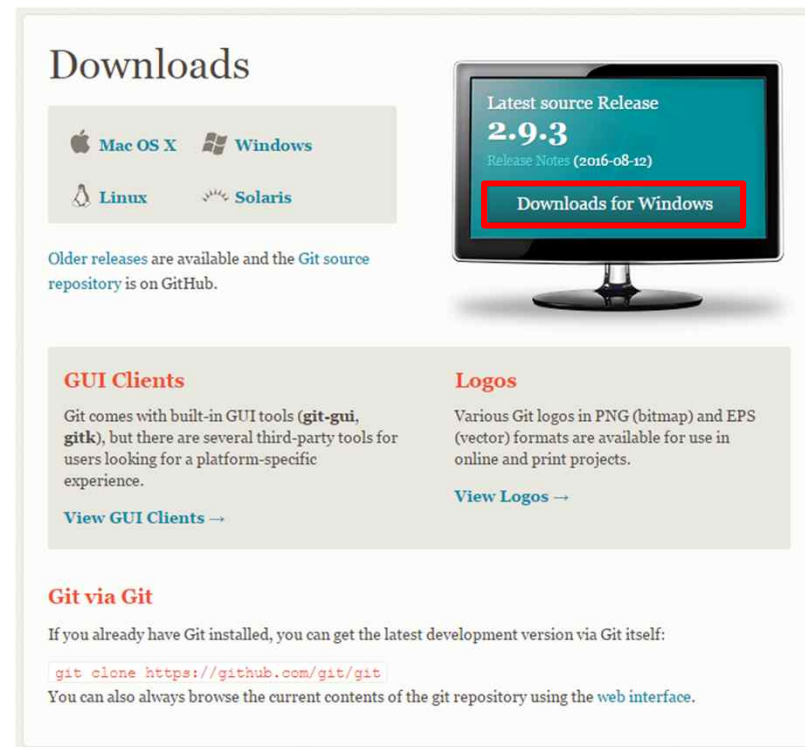


SourceTree를 이용한 Git 사용법

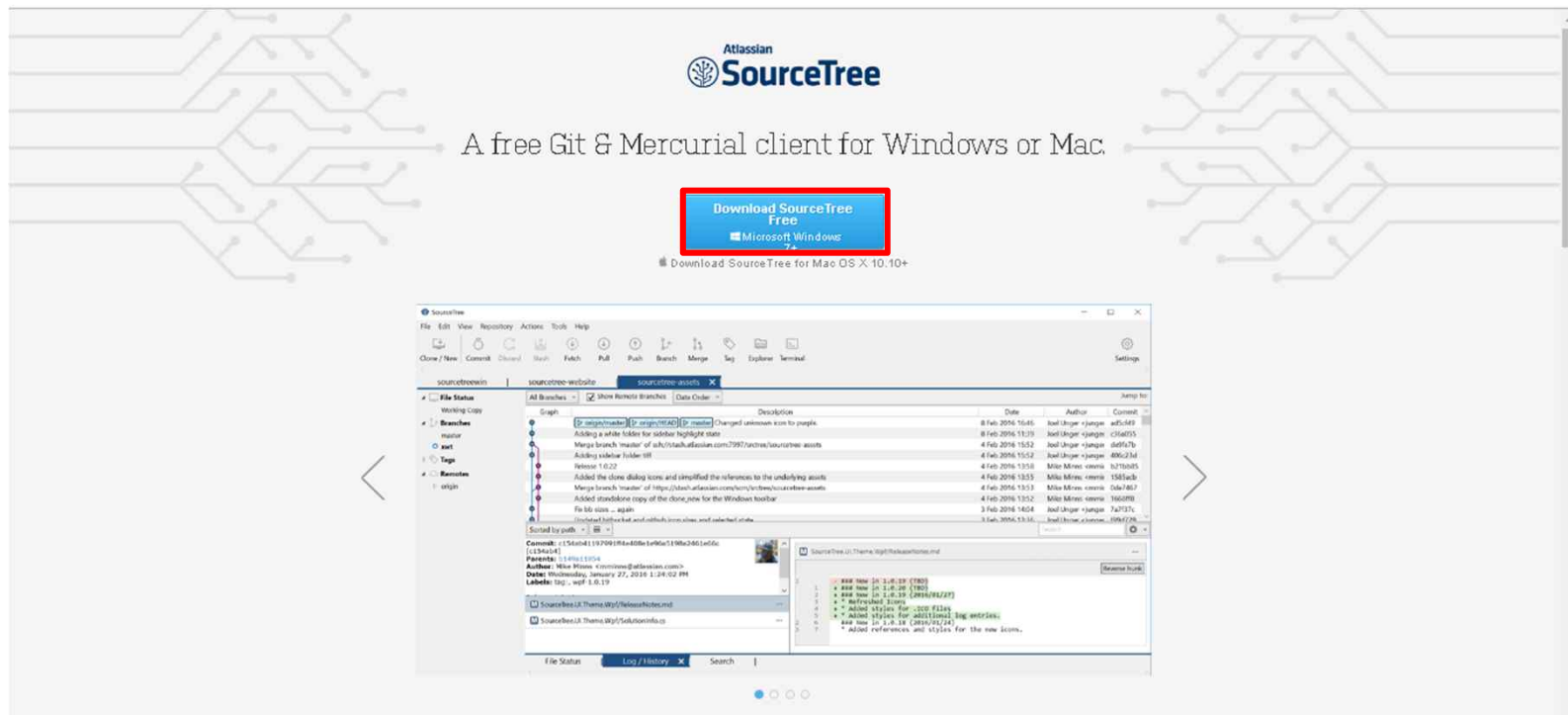
GIT 설치방법

- <https://www.git-scm.com/downloads> URL로 접속
→ 다운로드 클릭 → 설치



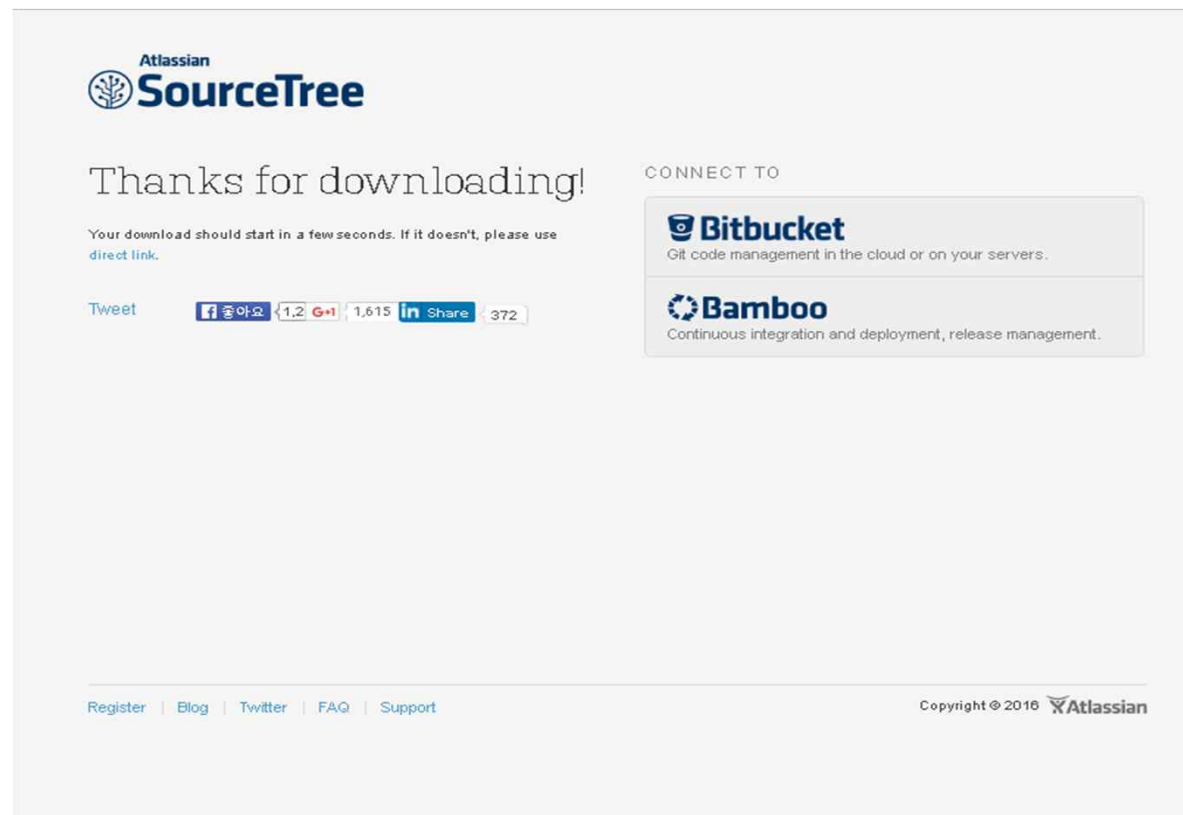
SourceTree 설치방법(1)

- <https://www.sourcetreeapp.com/> URL로 접속
→ 다운로드 클릭



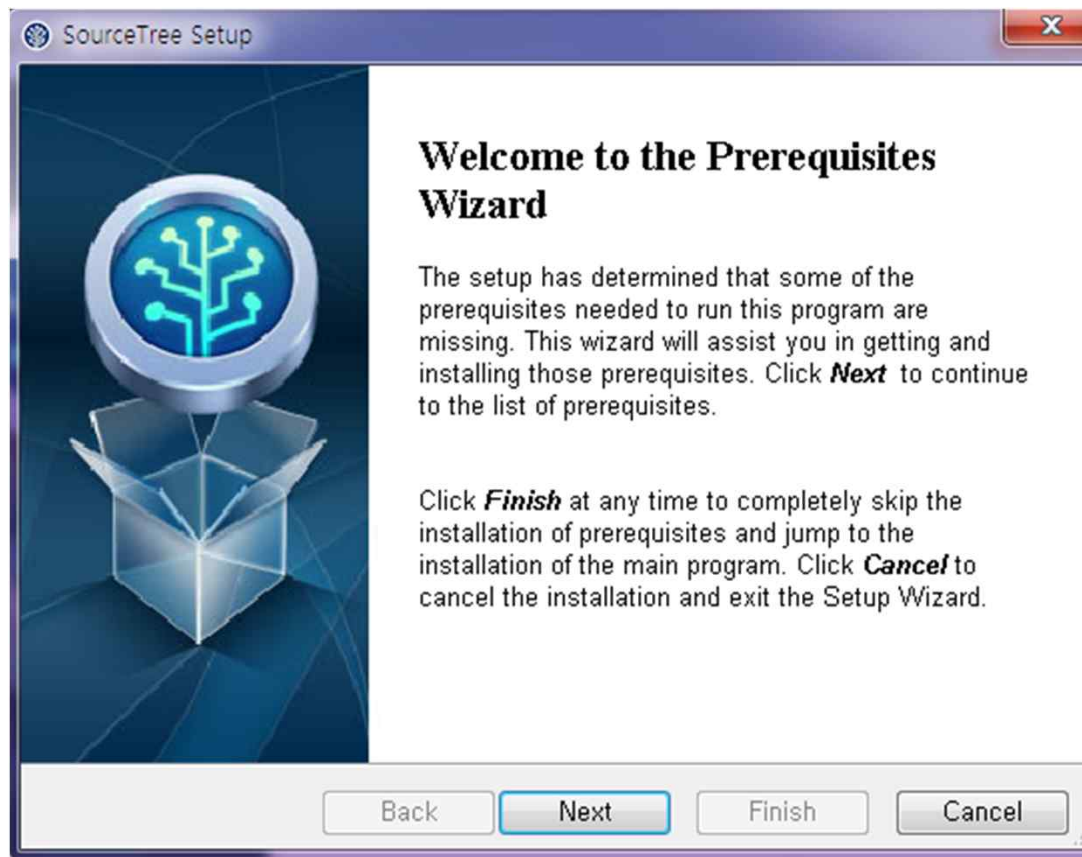
SourceTree 설치방법(2)

- Download 버튼을 클릭하면 자동으로 아래와 같은 창으로 넘어가면서 Install File Download가 시작



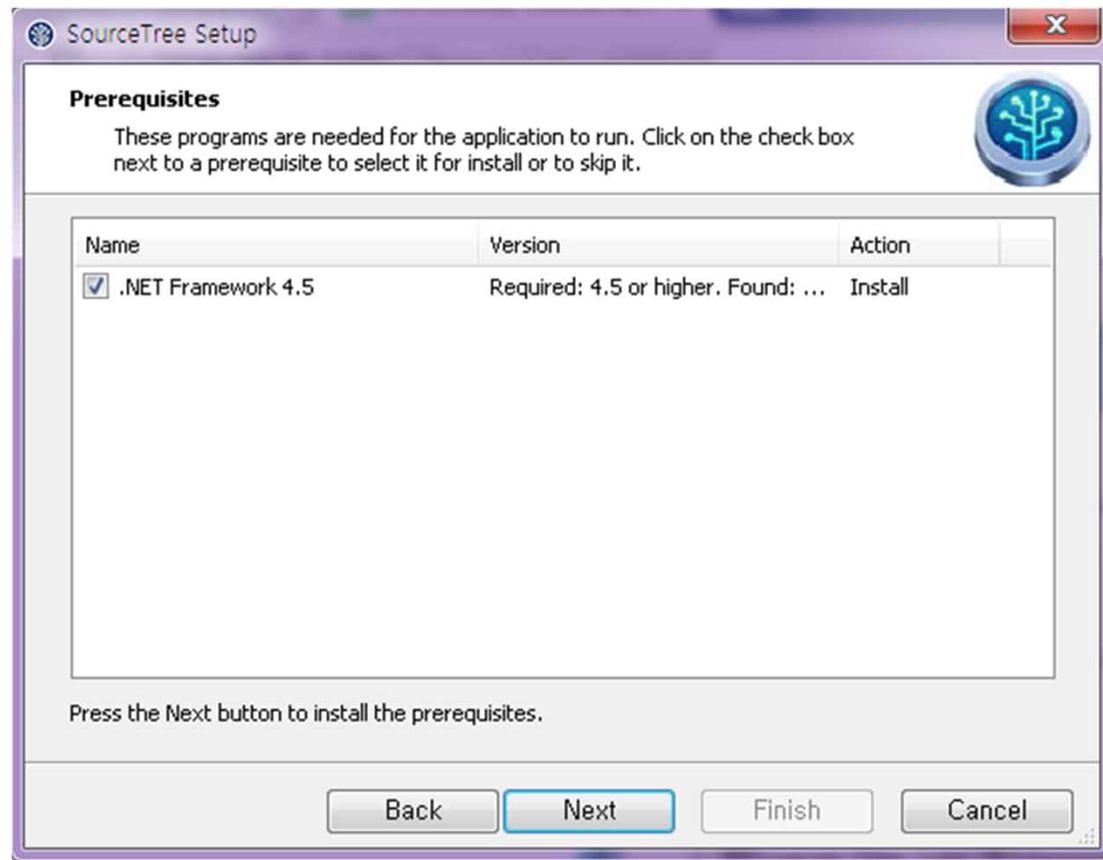
SourceTree 설치방법(3)

- 소스트리를 설치하기 전 .NET Framework 4.5를 설치단계



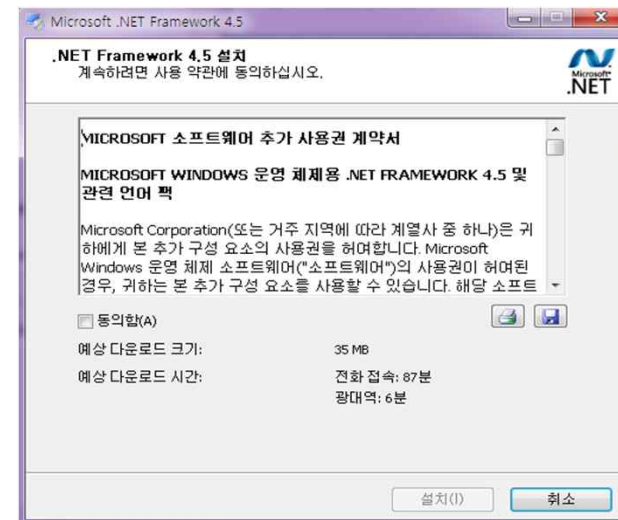
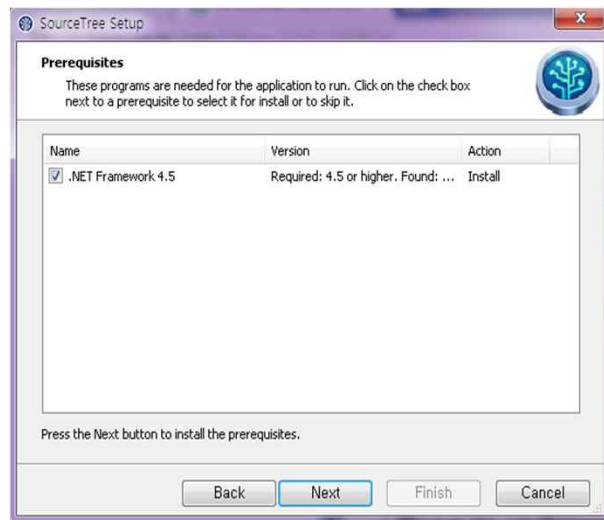
SourceTree 설치방법(4)

- 다음과 같은 화면이 나타나면 .NET Framework 4.5 설치



SourceTree 설치방법(5)

- 다음과 같은 화면이 나타나면 .NET Framework 4.5 체크 후 Next 클릭
이후 동의함 체크 후 설치 클릭



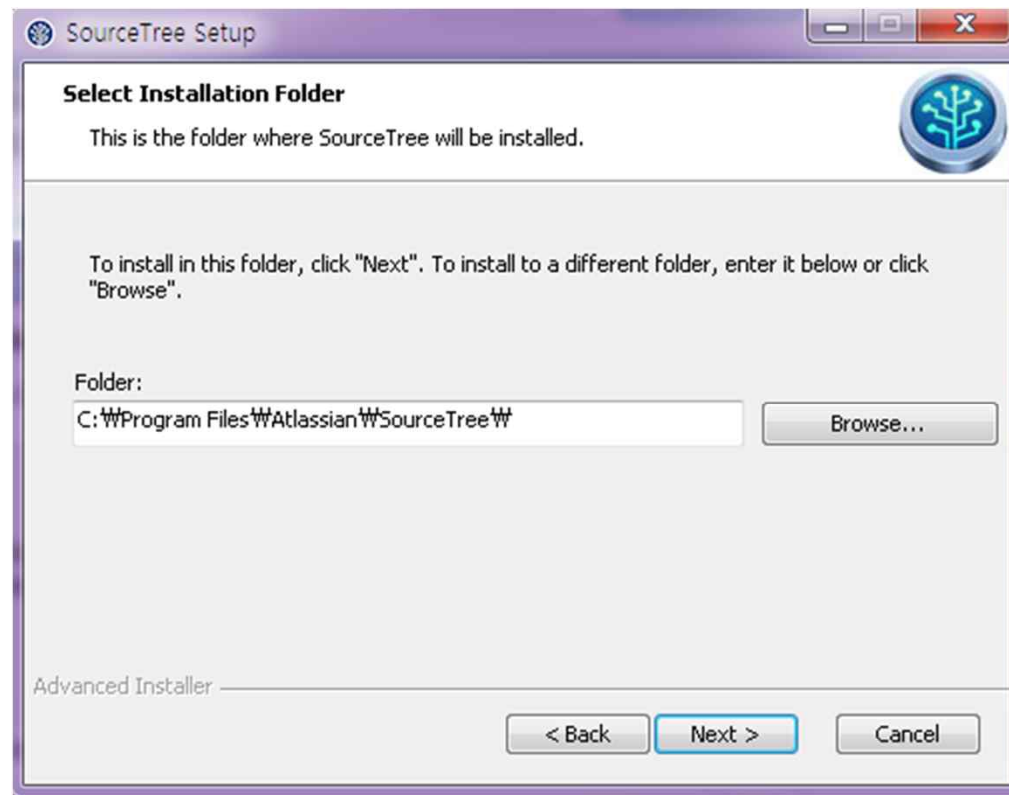
SourceTree 설치방법(6)

- Install File을 클릭하여 실행하면 위와 같은 창이 나타남



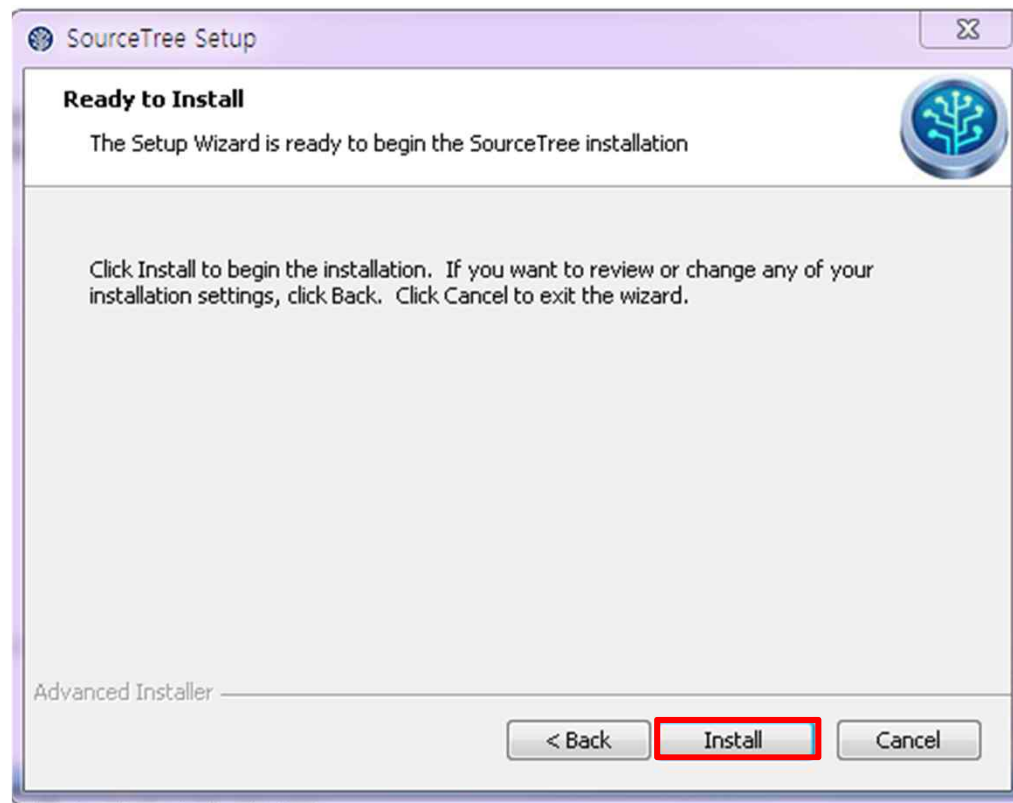
SourceTree 설치방법(7)

- SourceTree를 어디에 Install 할지 Path를 설정



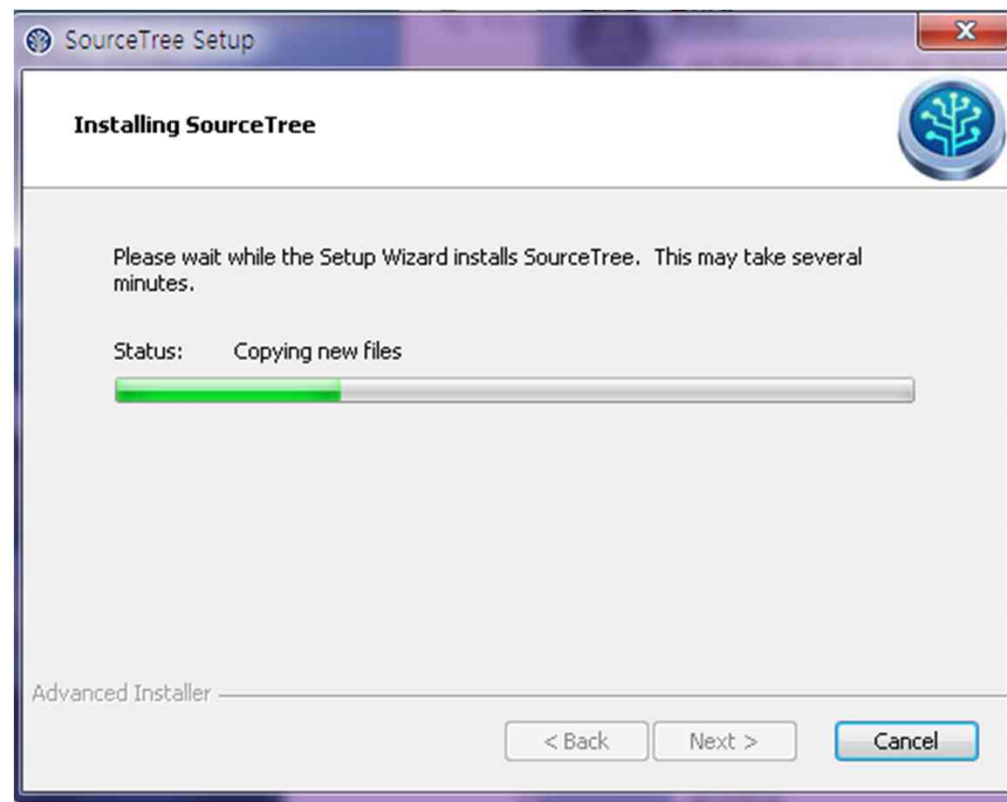
SourceTree 설치방법(8)

- Install 클릭



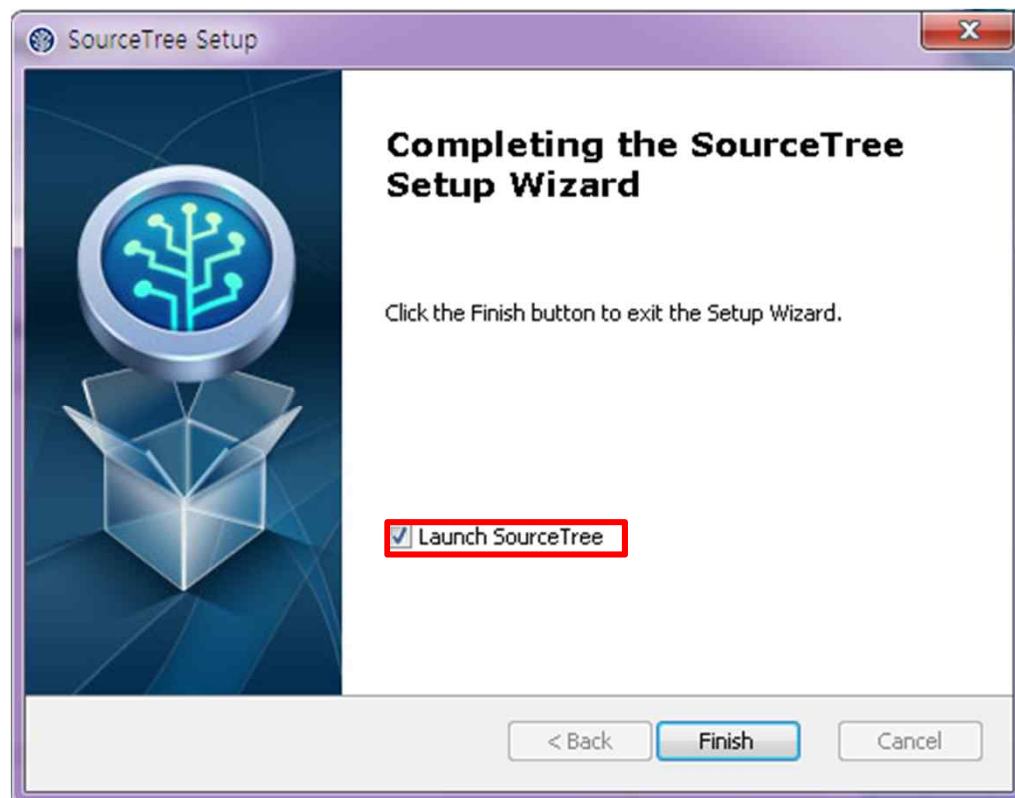
SourceTree 설치방법(9)

- Install 진행과정



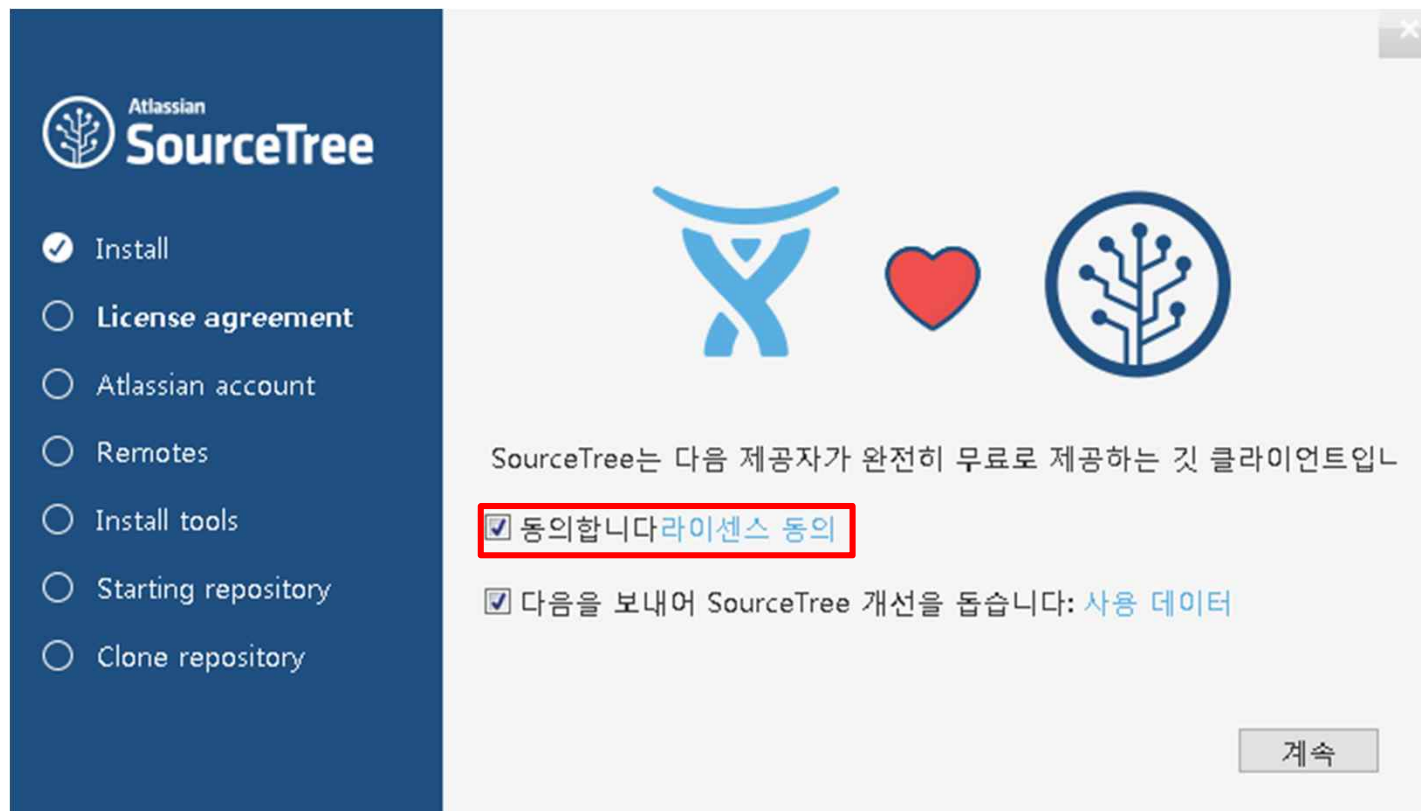
SourceTree 설치방법(10)

- 붉은색 박스 체크



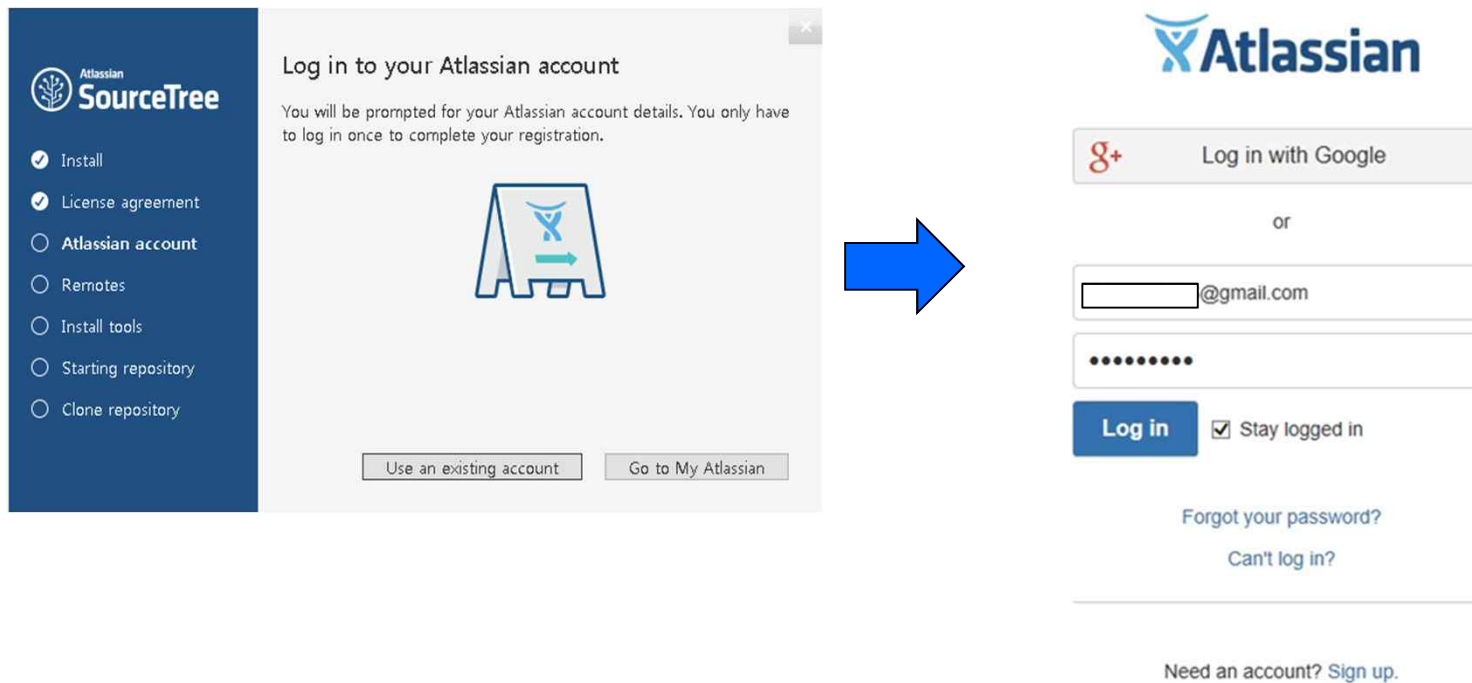
SourceTree 설치방법(11)

- 라이선스에 동의 후 계속



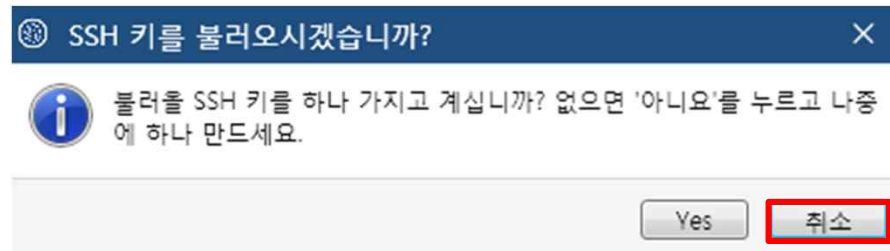
SourceTree 설치방법(12)

- 계정이 있다면 **Use an existing account** 버튼을 선택하고
계정이 없을 경우 **Go to My Atlassian** 버튼을 눌러서
계정을 만든 뒤 설치 진행
(Atlassian account 계정부분에서 설정 건너뛰기 클릭)



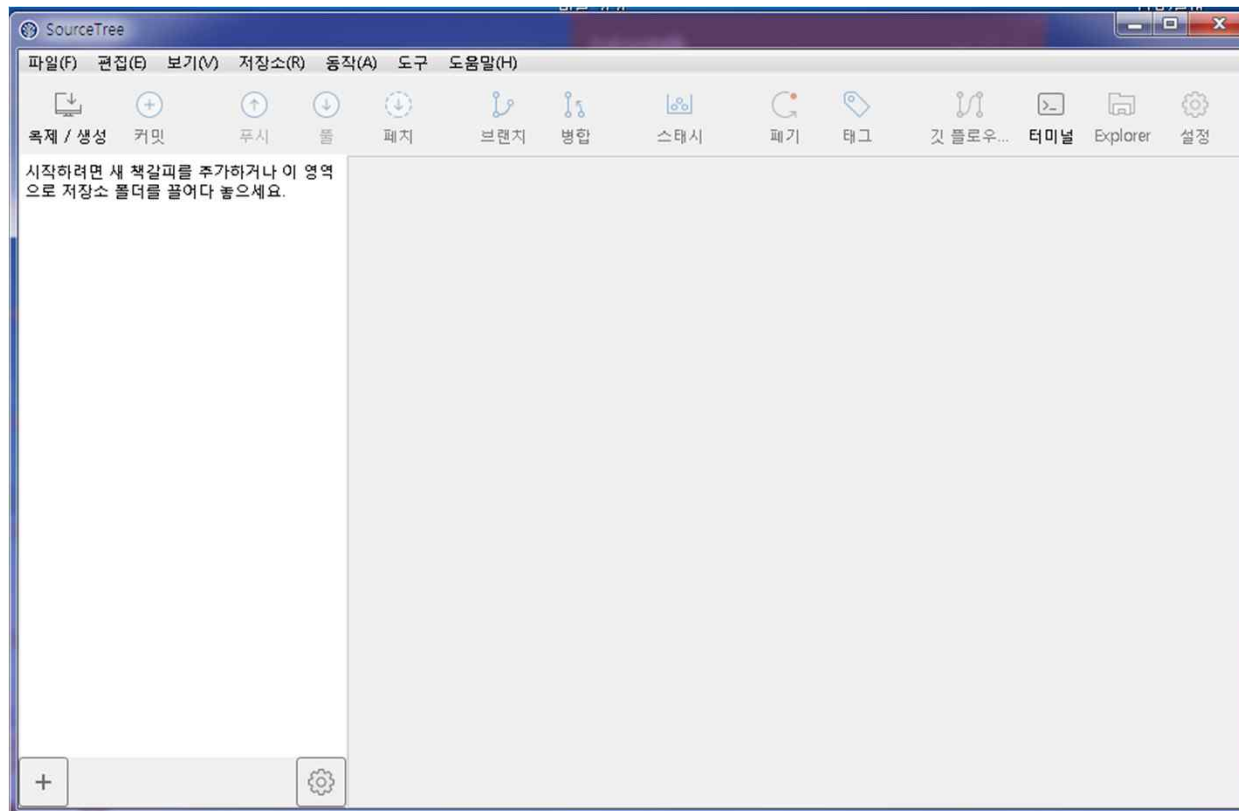
SourceTree 설치방법(13)

- SSH가 있는 경우 Yes, 없는 경우 No



SourceTree 설치방법(14)

- Git을 Gui로 이용하기 위한 SourceTree 설치 완료



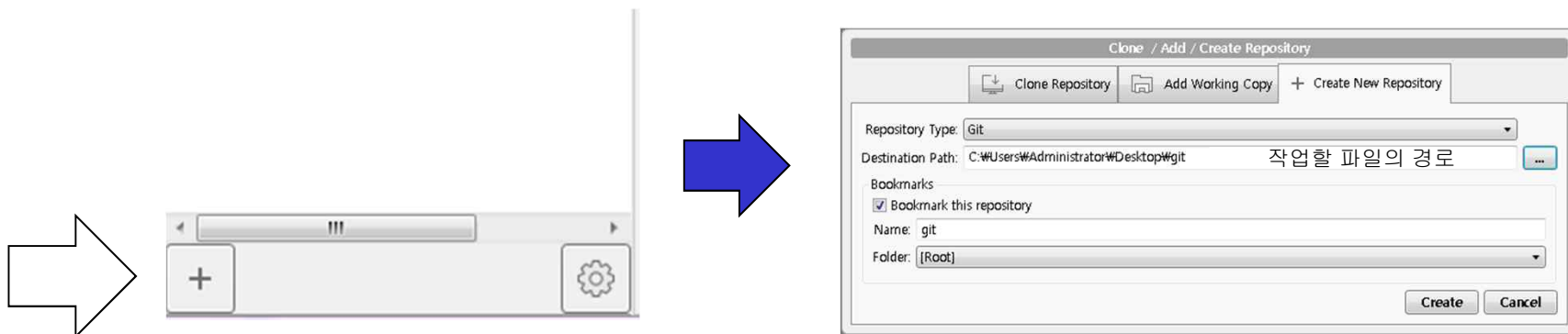
Sublimetext 설치

- Sublimetext 설치

저장소 만들기(**init**)

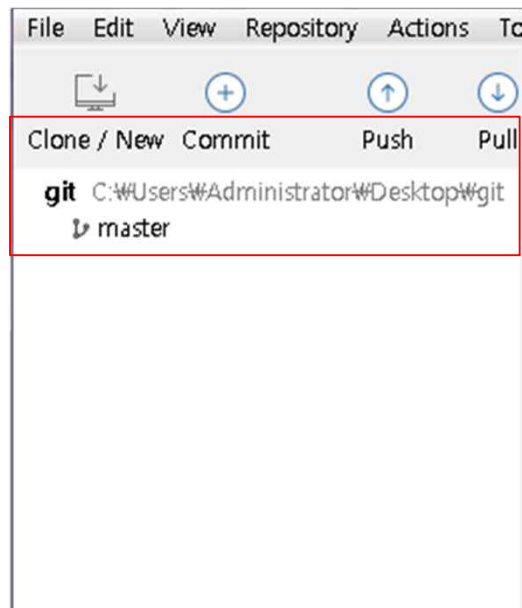
저장소 만들기(init)

- 바탕화면에 githtml 폴더 생성 → Add repository 클릭 → Create New Repository 탭을 누르고 Destination Path에 git 폴더로 경로 지정 → Create



저장소 만들기(init)

- 생성된 저장소



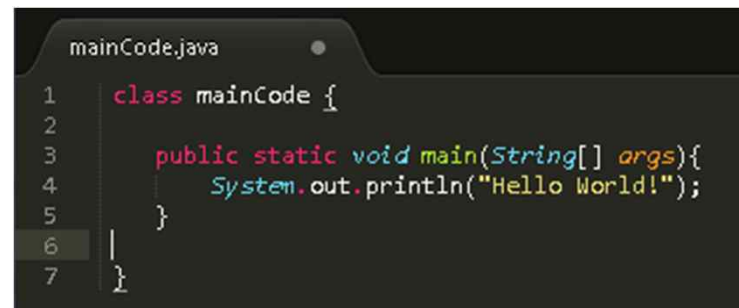
SourceTree를 활용한 버전 만들기(commit)

Commit

- 프로젝트를 진행하다보면 하나의 기능을 완성하거나 더 이상 수정을 할 필요가 없다고 생각할 때 완성된 기존의 소스를 다른 곳에 백업한다.
이 것을 **Git**에서는 **commit**이라 한다.

Commit

- Git 리포지토리(자신의 프로젝트 폴더)에 소스 파일을 작성하고 생성

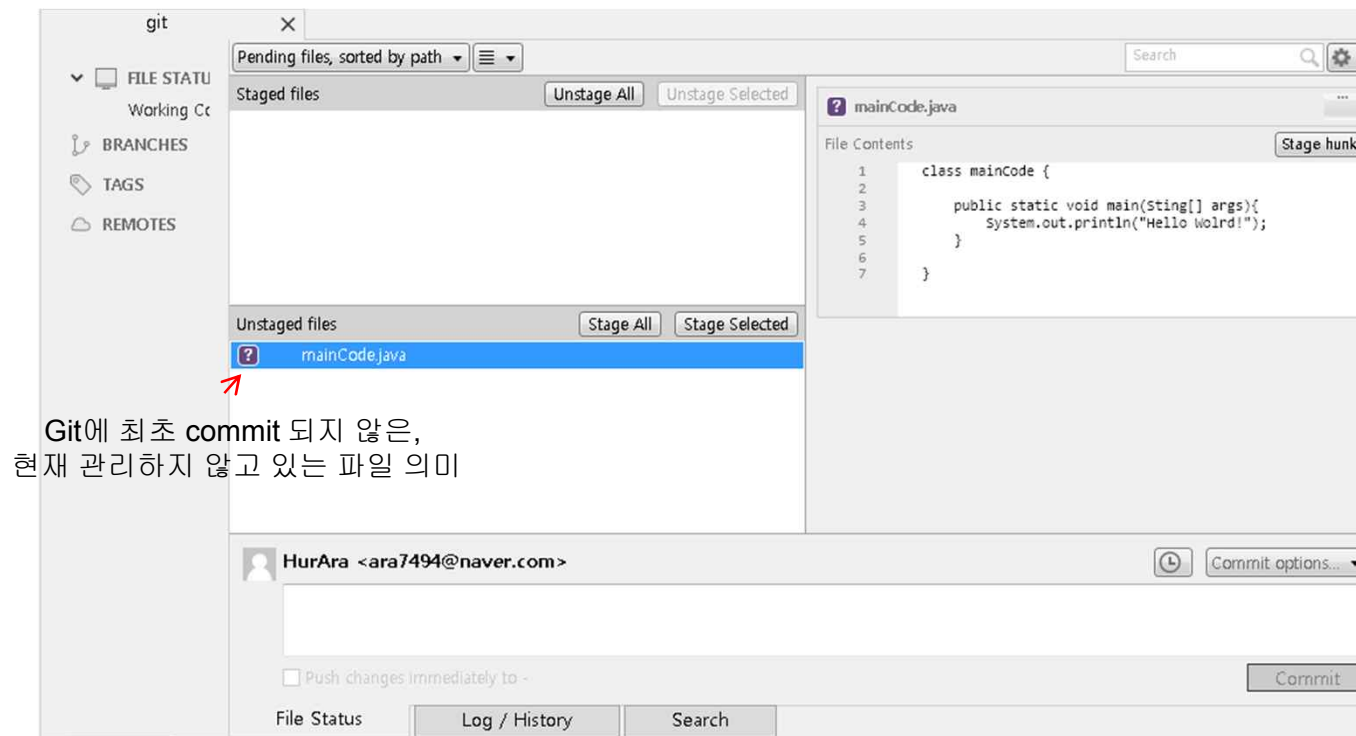


```
mainCode.java
1  class mainCode {
2
3      public static void main(String[] args){
4          System.out.println("Hello World!");
5      }
6  |
7  }
```



Commit

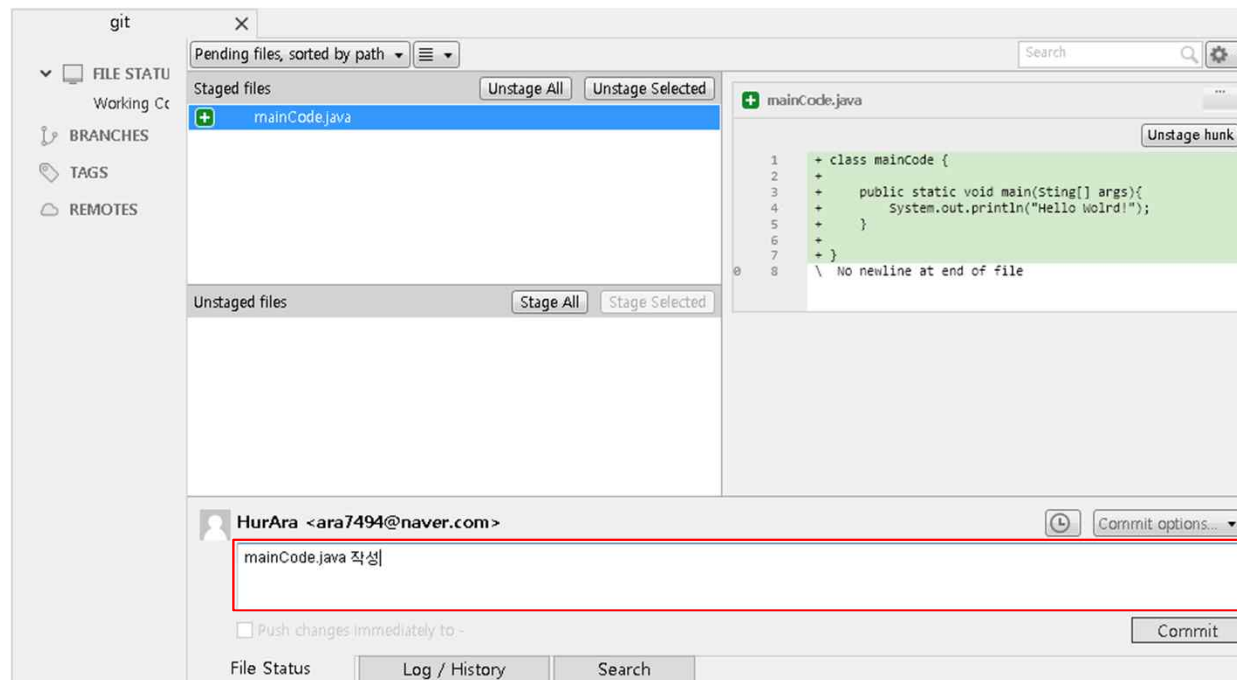
- 파일을 저장한 후 **SourceTree**를 확인해보면 다음과 같이 새롭게 생성하거나 수정한 파일들이 **SourceTree**의 **Unstaged files** 목록에 표시
→ 이 목록에서 **add** 하고자하는 파일을 선택 후 “**Stage Selected**” 클릭



Git에 최초 commit 되지 않은,
현재 관리하지 않고 있는 파일 의미

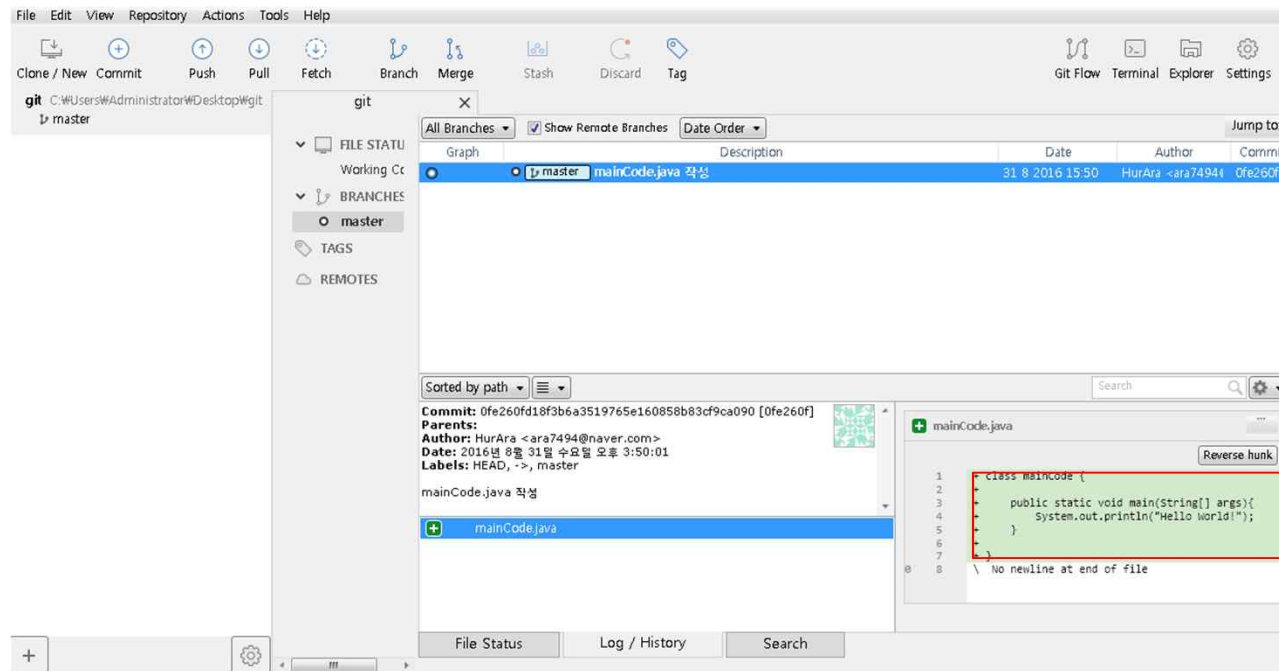
Commit

- **Unstaged files** 목록에서 **add**를 하면 파일이 상단의 **Staged files**로 이동 (commit 하기 전의 임시영역에 해당 파일들이 **commit** 될 준비가 되었음을 나타냄)
- **Staged files**를 커밋 하기 위하여 맨 하단에 커밋 메시지로 'mainCode.java 작성' 이라고 입력 → **commit** 버튼 클릭



Commit

- 하나의 파일 커밋 로그 생성
(커밋된 파일을 클릭하면 파일의 내용을 확인할 수 있음
초록색 영역은 추가되거나 수정된 부분을 의미하며 빨간색 영역은 예전 버전에서 삭제된 부분을 의미)



SourceTree를 활용한 commit 전 수정사항 취소하기(discard)

discard

- 만약 소스코드를 수정을 하고 **commit**을 하기 직전의 상황에서 수정한 내용에 문제점이 발견되고 수정한 부분을 되돌릴려고 한다.
직접 이 부분을 사람이 되돌리고자 하면 하나하나 원 상태와 똑같이 지우고 수정해가며 이전의 상태로 되돌릴 수 있다. 이 때 **Git**의 **discard**를 통해 자동으로 수정하기 전의 **commit**한 버전으로 돌아갈 수 있다.

discard

- Sublimetest 파일로 아래와 같이 'html'이라는 파일 저장

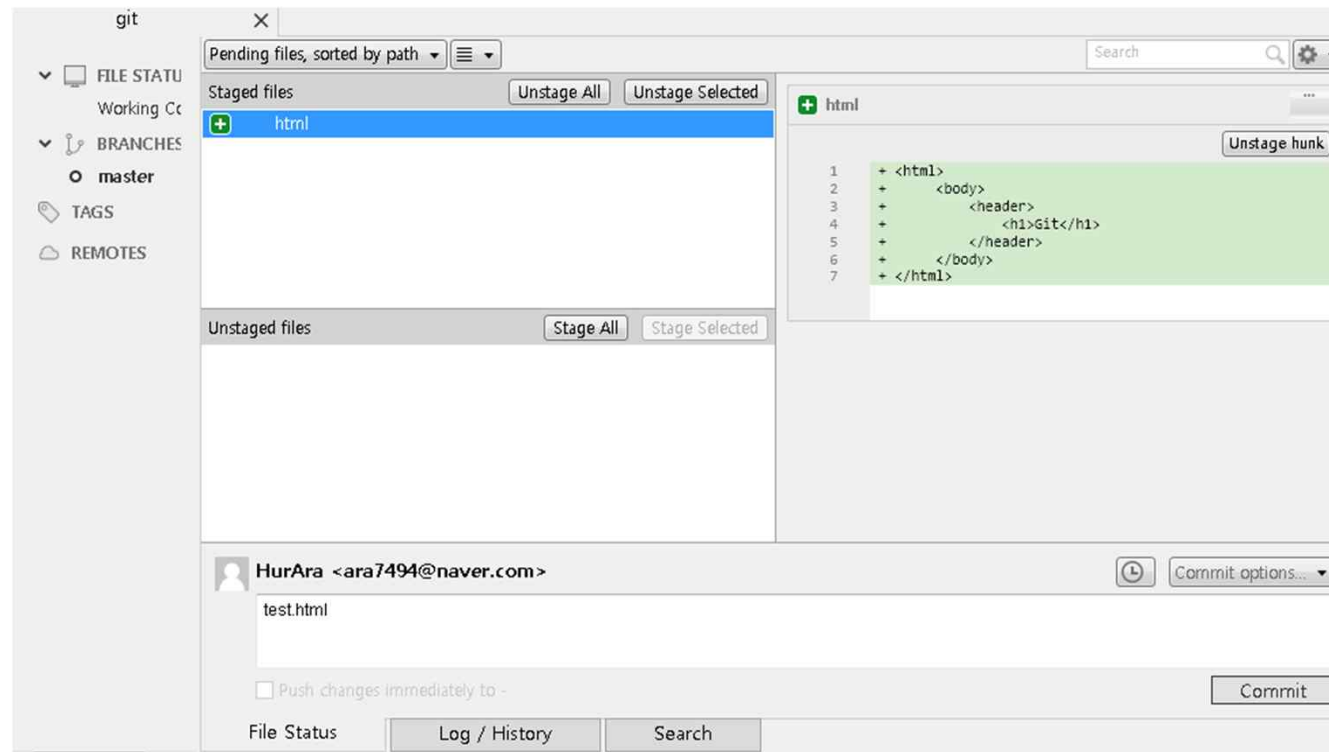


```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5     </header>
6   </body>
7 </html>
8
```

<수정 전>

discard

- 다음과 같이 commit



discard

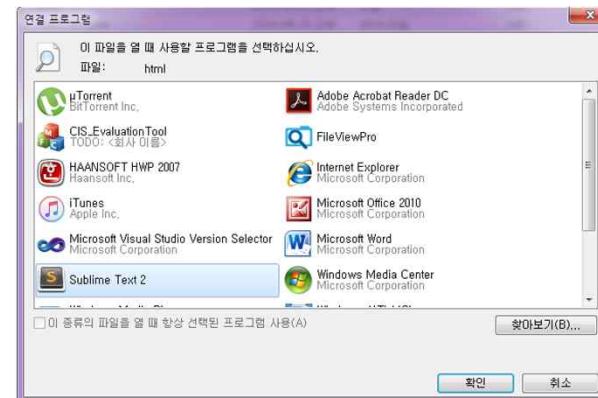
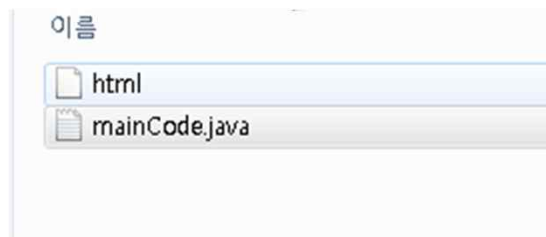
- Html 파일을 다음과 같이 수정



```
html
1 <html>
2   <body>
3     <h1>Git</h1>
4   </body>
5 </html>
```

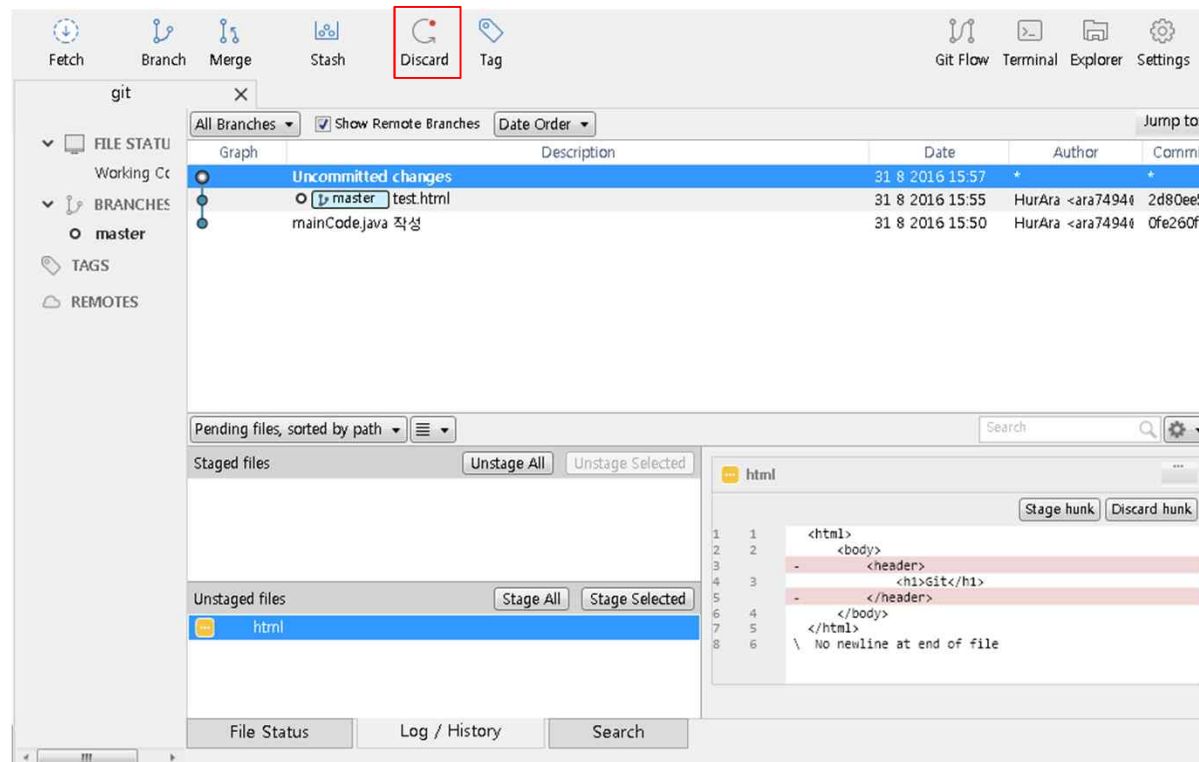
discard

- 파일이 안 열릴 경우 → 연결 프로그램 → Sublime text 2 → 확인



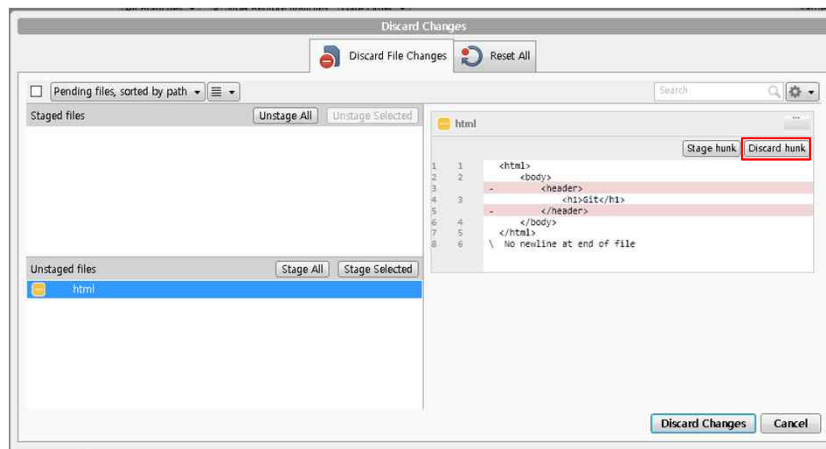
discard

- 파일의 내용이 수정되면 SourceTree에 아래와 같이 표시 → 여기서 **commit**을 하기 전 소스코드에 문제가 발생하여 수정하기 전으로 소스코드를 되돌리고 싶다면 SourceTree의 상단 메뉴에서 **Discard** 클릭



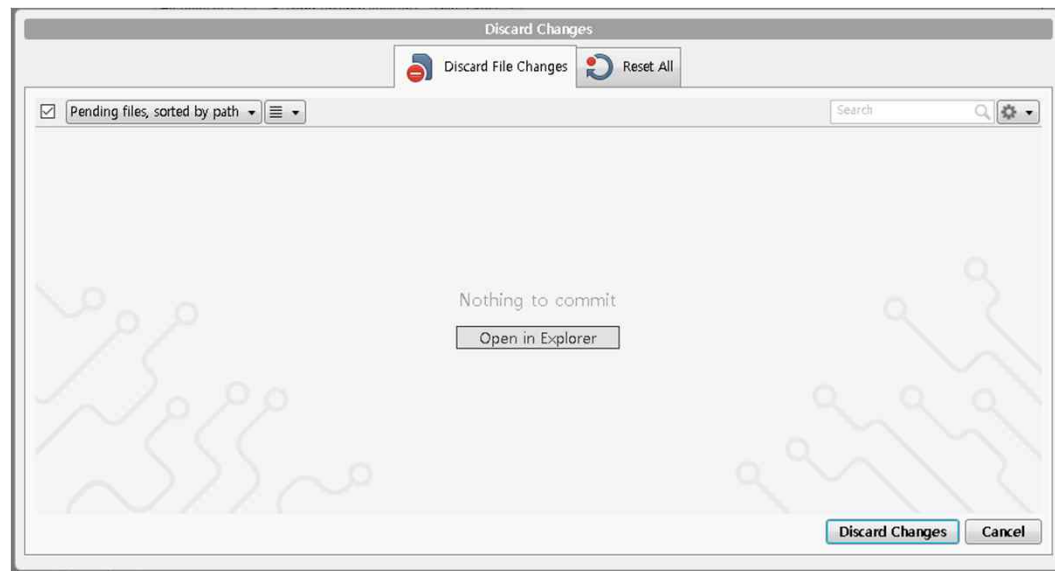
discard

- Html 파일 선택 → 오른쪽 Discard hunk 클릭 → 다음과 같은 창이 뜨면 OK



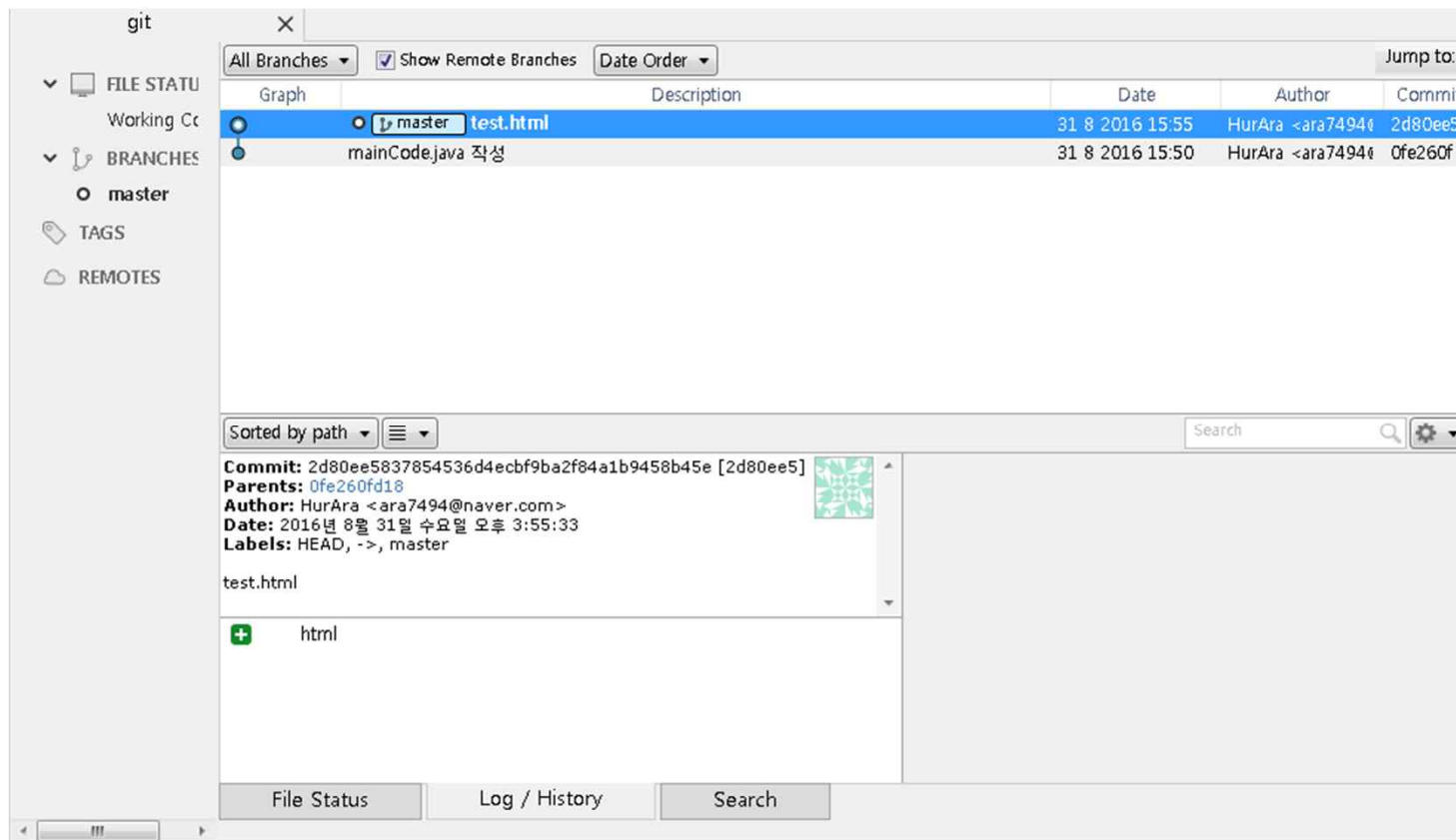
discard

- 이 상태에서 **Cancel** 클릭



discard

- 완료 후 “Uncommitted changes”가 사라지고 작성하던 내용을 확인해보면 수정하기 전의 코드로 돌아가있는 것을 확인할 수 있음



discard

- 결과



```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5     </header>
6   </body>
7 </html>
```

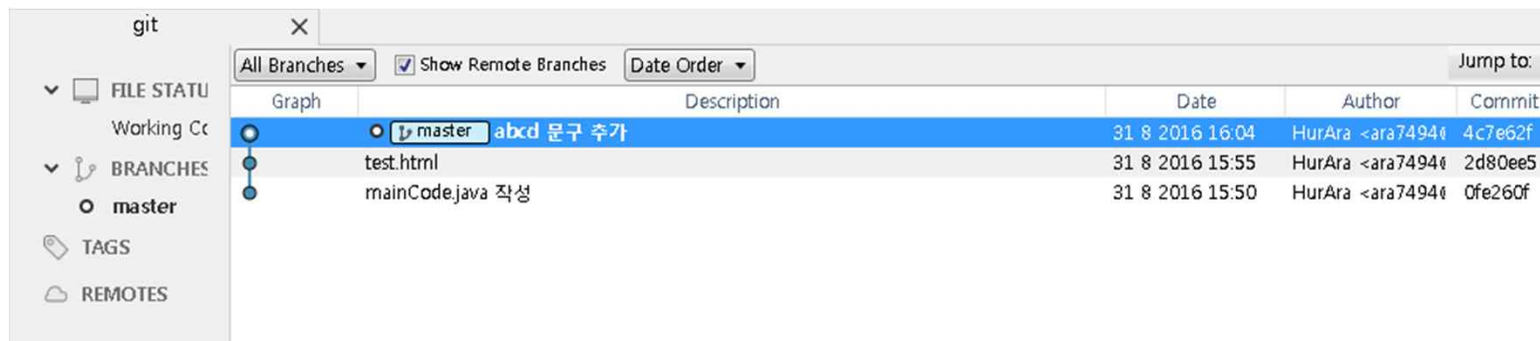
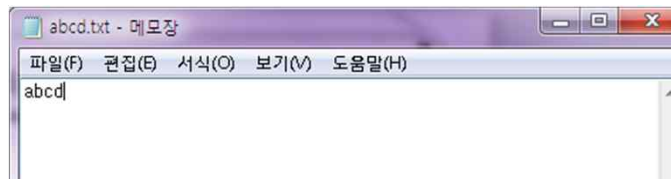
**SourceTree를 활용한 commit 후
수정사항 되돌리기(reset)**

reset

- 프로젝트가 진행되던 도중 문제가 발생하여 이저의 버전으로 돌아가야 하는 상황이 발생된다. 이때 돌아가길 원하는 버전으로 **reset**을 통해 되돌릴 수 있다.

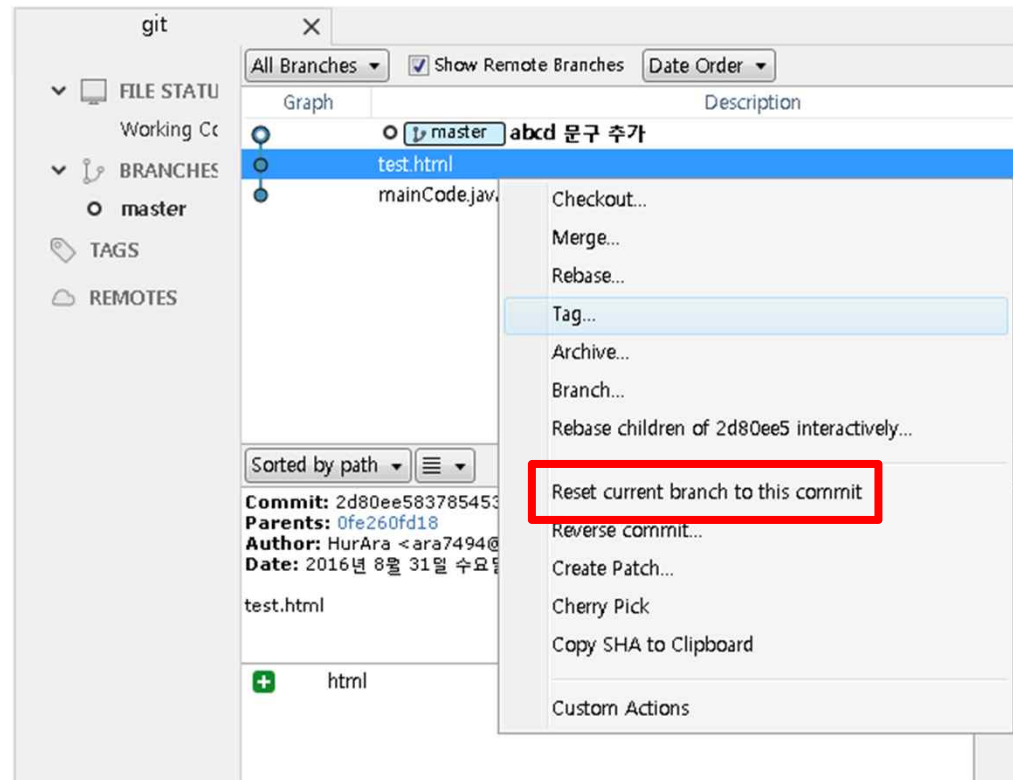
reset

- Abcd.txt 파일을 만들고 저장 → SourceTree에서 commit
(커밋메시지로 'abcd 문구 추가')



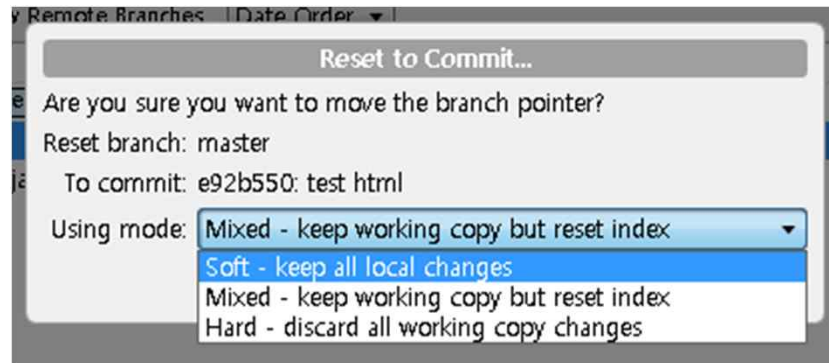
reset

- 되돌아갈 커밋 로그를 선택하고 마우스 오른쪽 버튼을 클릭하고
“Reset current branch to this commit” 클릭



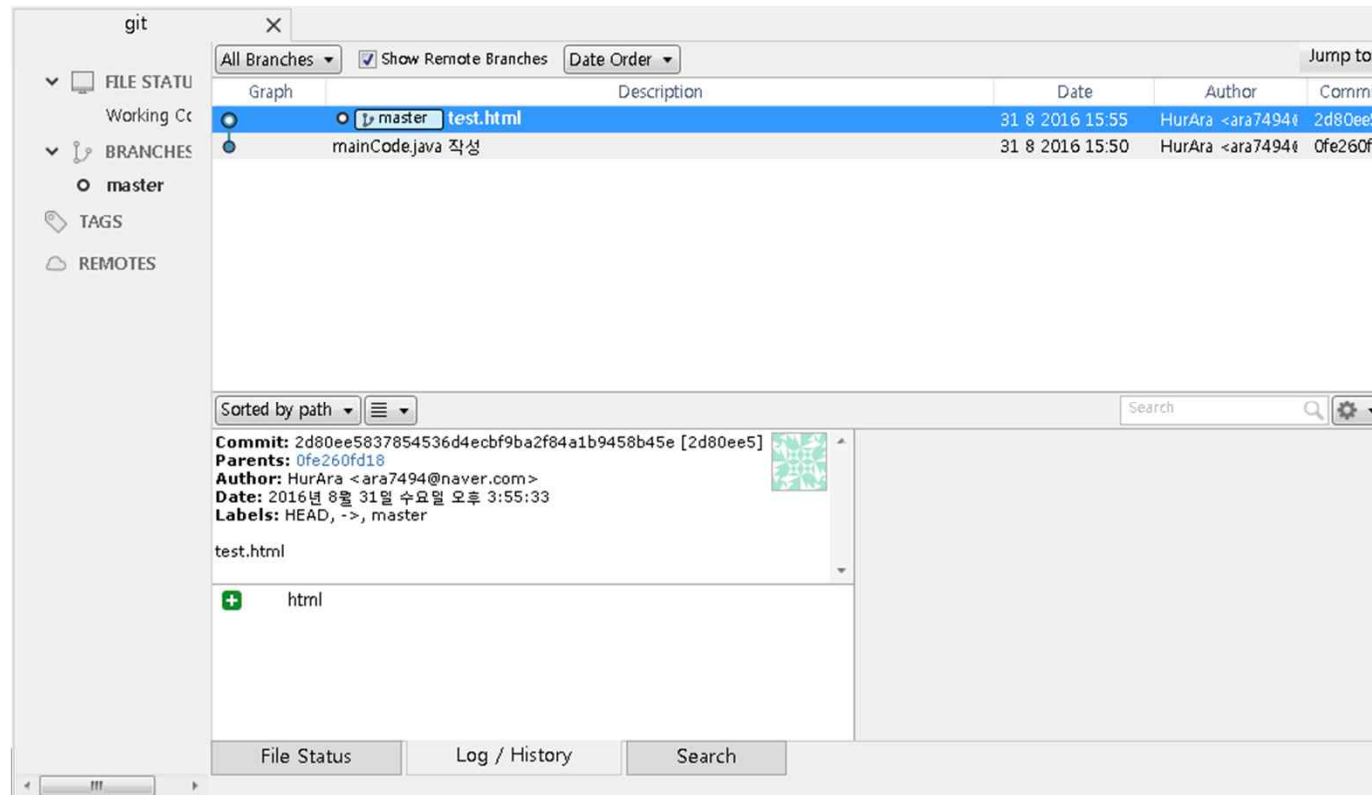
reset

- Reset to Commit 창은 아래와 같음 → 원하는 모드 선택 후 ok
 - 1) soft: index에 올라온 수정사항 보존, working tree에 올라온 수정사항 보존
 - 2) Mixed: index에 올라온 수정사항 취소, working tree에 올라온 수정사항 보존
 - 3) Hard: index에 올라온 수정사항 취소, working tree에 올라온 수정사항 보존



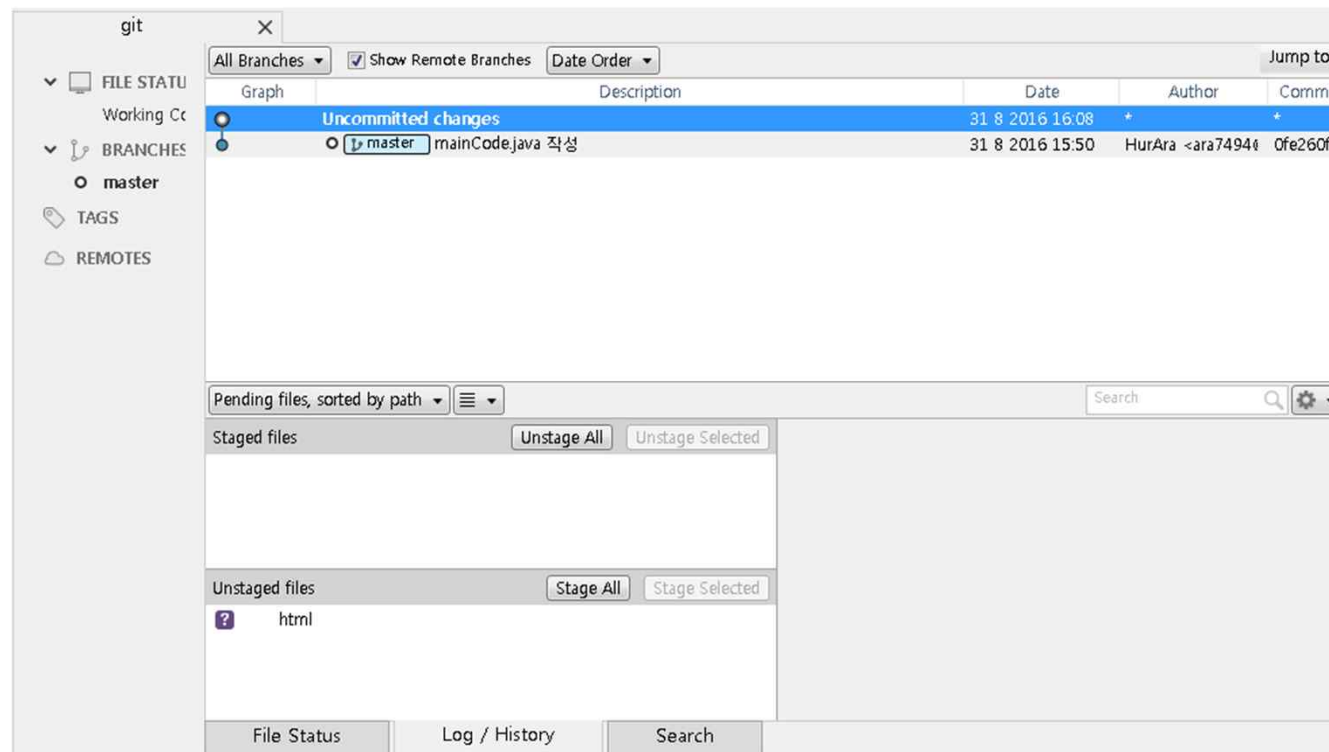
reset

- **Hard mode**를 선택할 경우, **Reset**한 버전이후의 버전이 모두 삭제되며 커밋 로그도 해당버전의 로그로 수정 됨



reset

- **Mixed mode**를 선택할 경우, **Reset** 한 버전이후의 버전이 모두 삭제 되었으나 **working tree**의 소스코드는 유지되어 **Uncommitted changes** 항목이 표시 됨



SourceTree를 활용한 브랜치 만들기(branch)

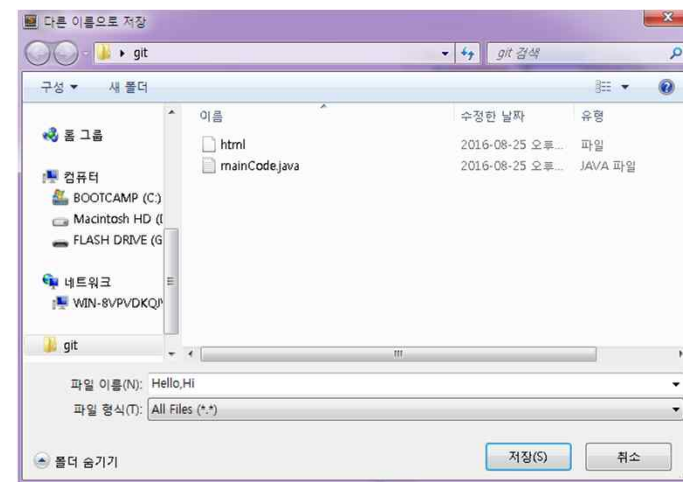
branch

- 프로젝트에 새로운 기능을 추가하고자 한다. 동시에 기존의 프로젝트에서 발생하는 버그와 같은 수정사항을 고쳐야 하는 상황이다. 그렇다면 다른 방향의 두 가지의 작업을 함께 진행하여야 하는데 만약 새로운 기능을 추가하는 작업이 취소되거나 수정되어 기능을 추가하기 전의 상태로 돌아가야 하는 상황이 발생할 수 있다. 하지만 이 때 **Git**을 통해 이전의 상황으로 돌아갈 경우 동시에 작업하던 프로젝트의 수정 작업 역시 사라지고 이전 상태로 돌아가는 문제가 발생한다. 이때 **Git**의 **branch** 개념이 필요하다.

branch

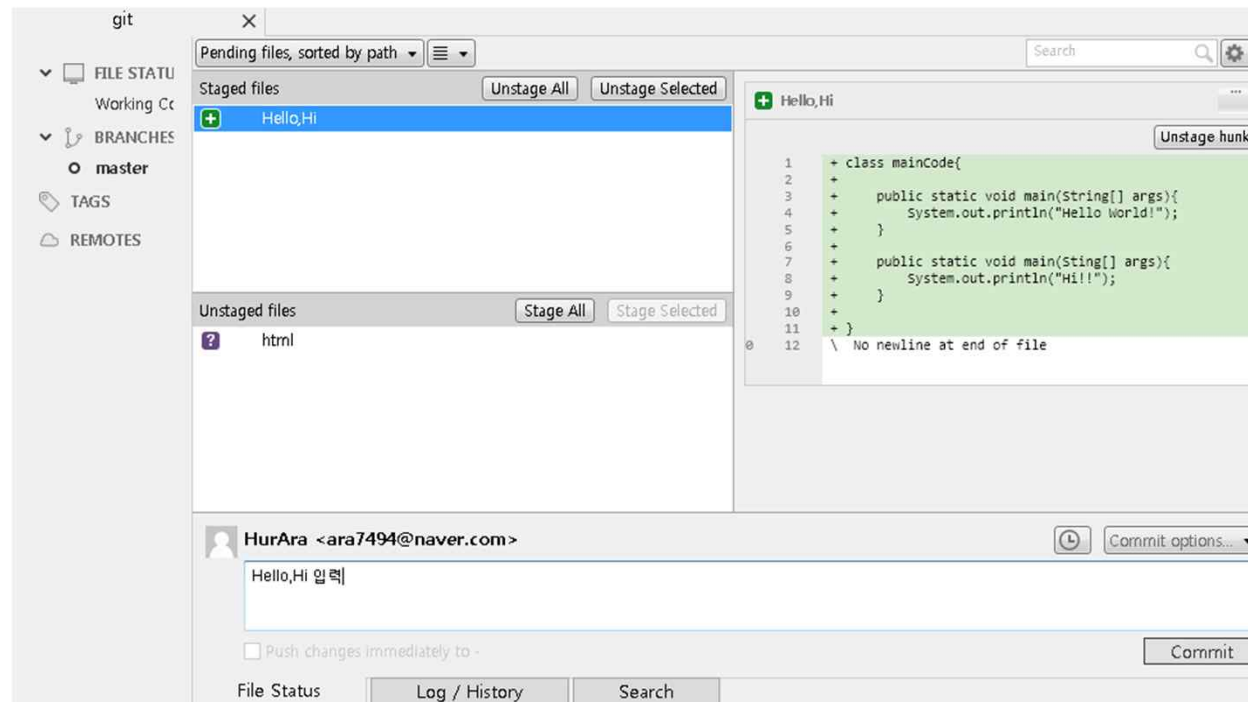
- Sublime text에 다음과 같이 내용을 입력 → Hello, Hi 라고 저장

```
Hello, Hi
1  class mainCode{
2
3      public static void main(String[] args){
4          System.out.println("Hello World!");
5      }
6
7      public static void main(String[] args){
8          System.out.println("Hi!!");
9      }
10
11 }
```



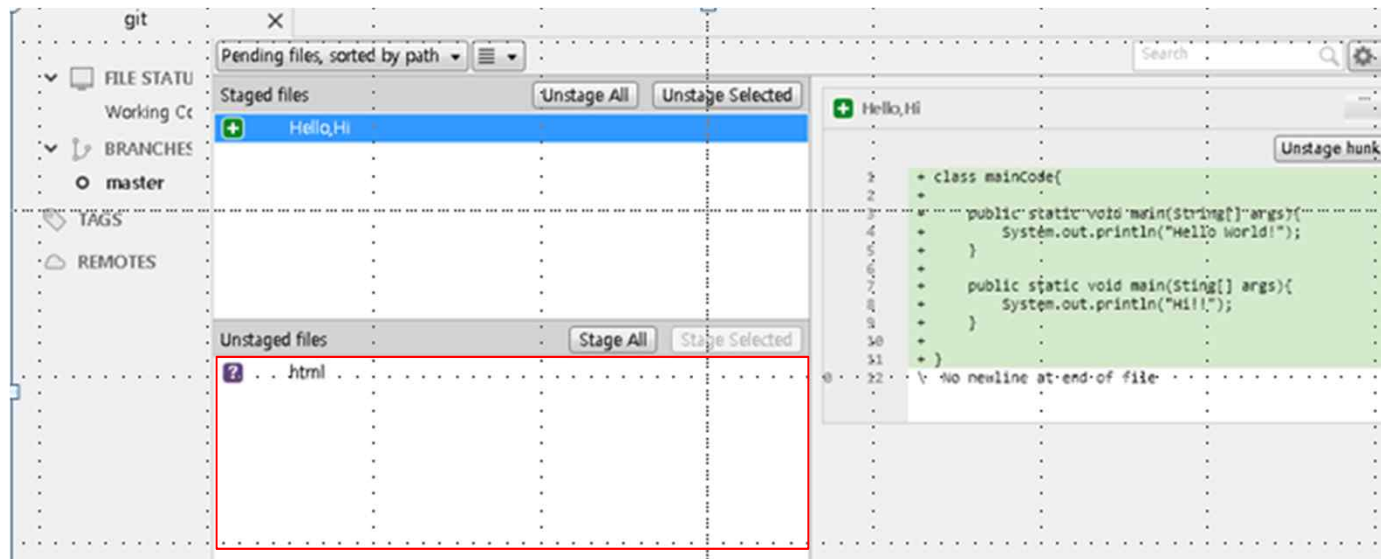
branch

- 커밋 메시지를 아래와 같이 입력하고 **commit**



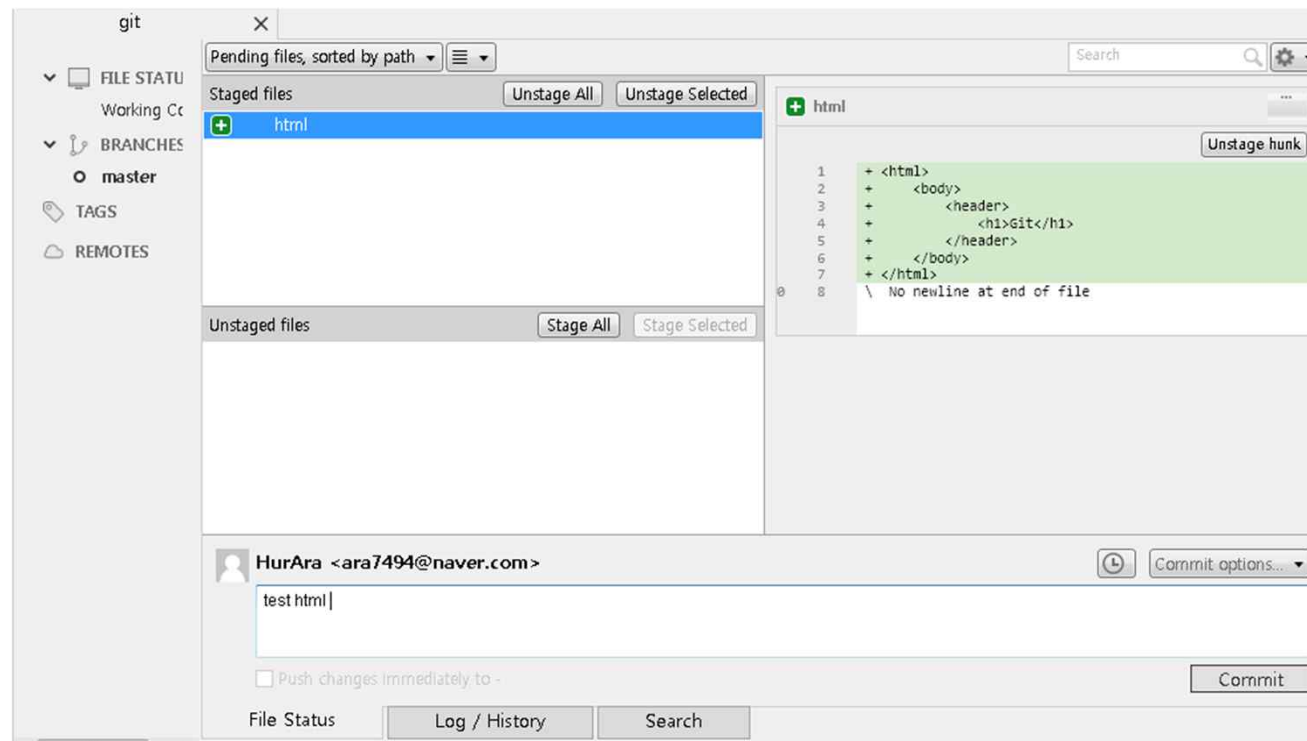
branch

- 아래 빨간 박스의(Unstaged files)에 있는 파일을 체크(Stage Selected)



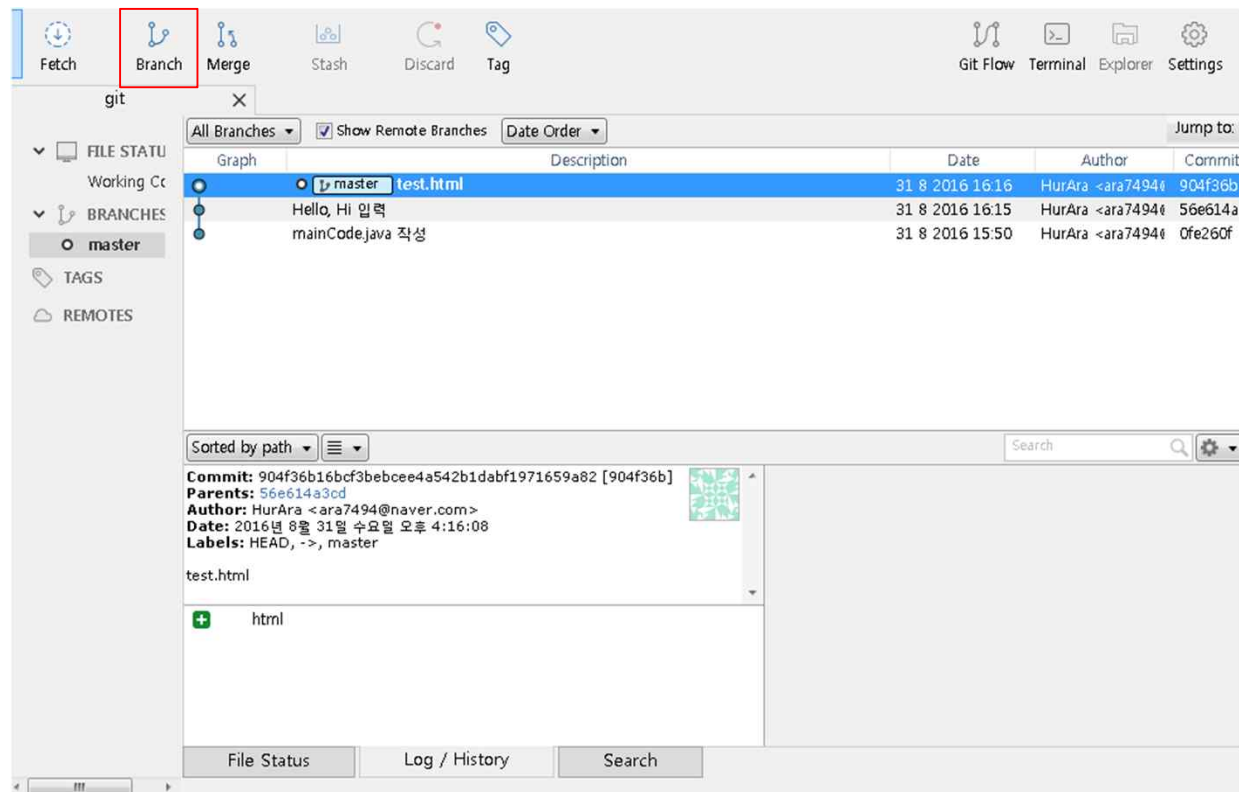
branch

- 커밋 메시지는 아래와 같이 입력 → commit



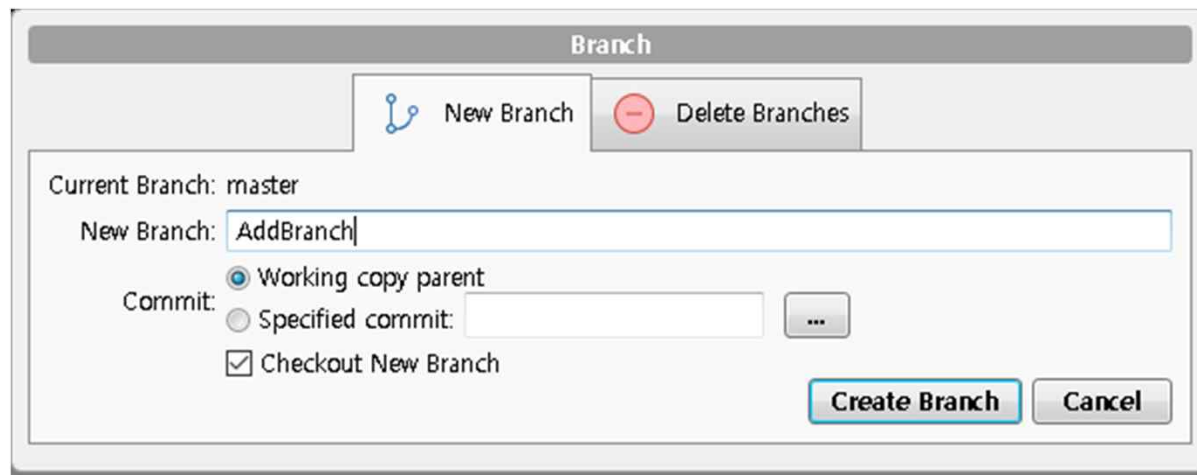
branch

- 상단 메뉴의 Branch 클릭



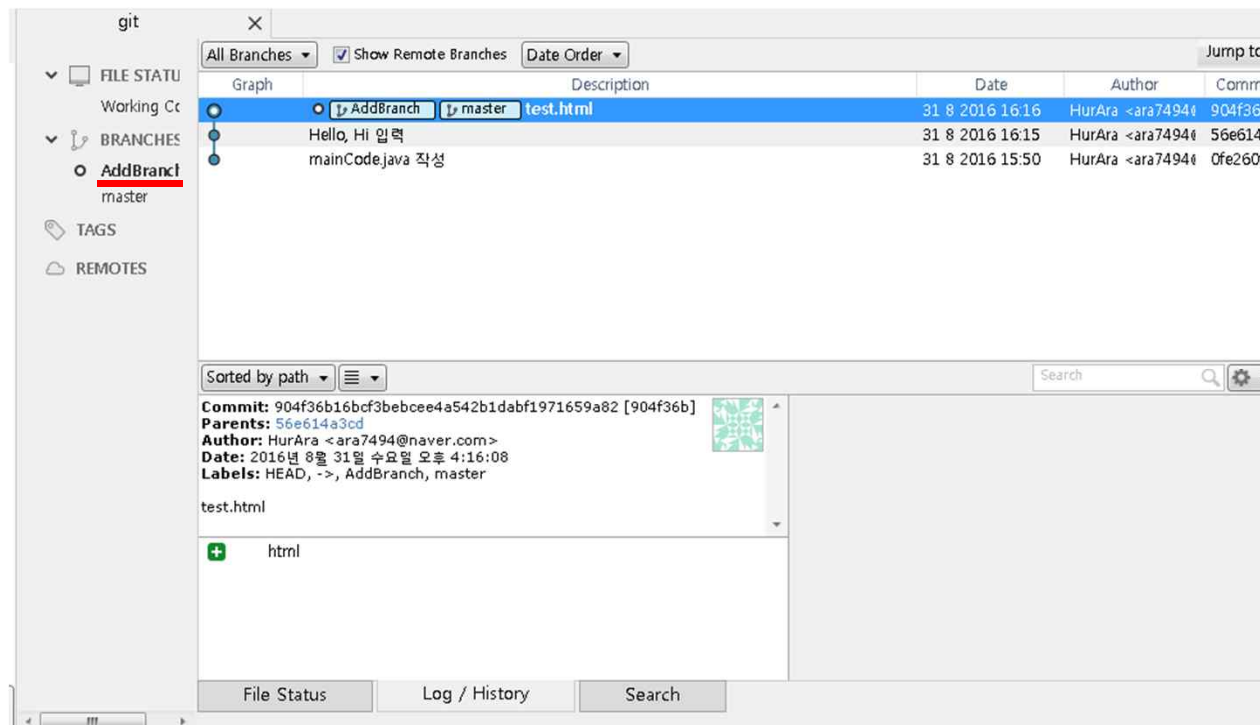
branch

- Branch의 이름을 입력하고 Create Branch 클릭



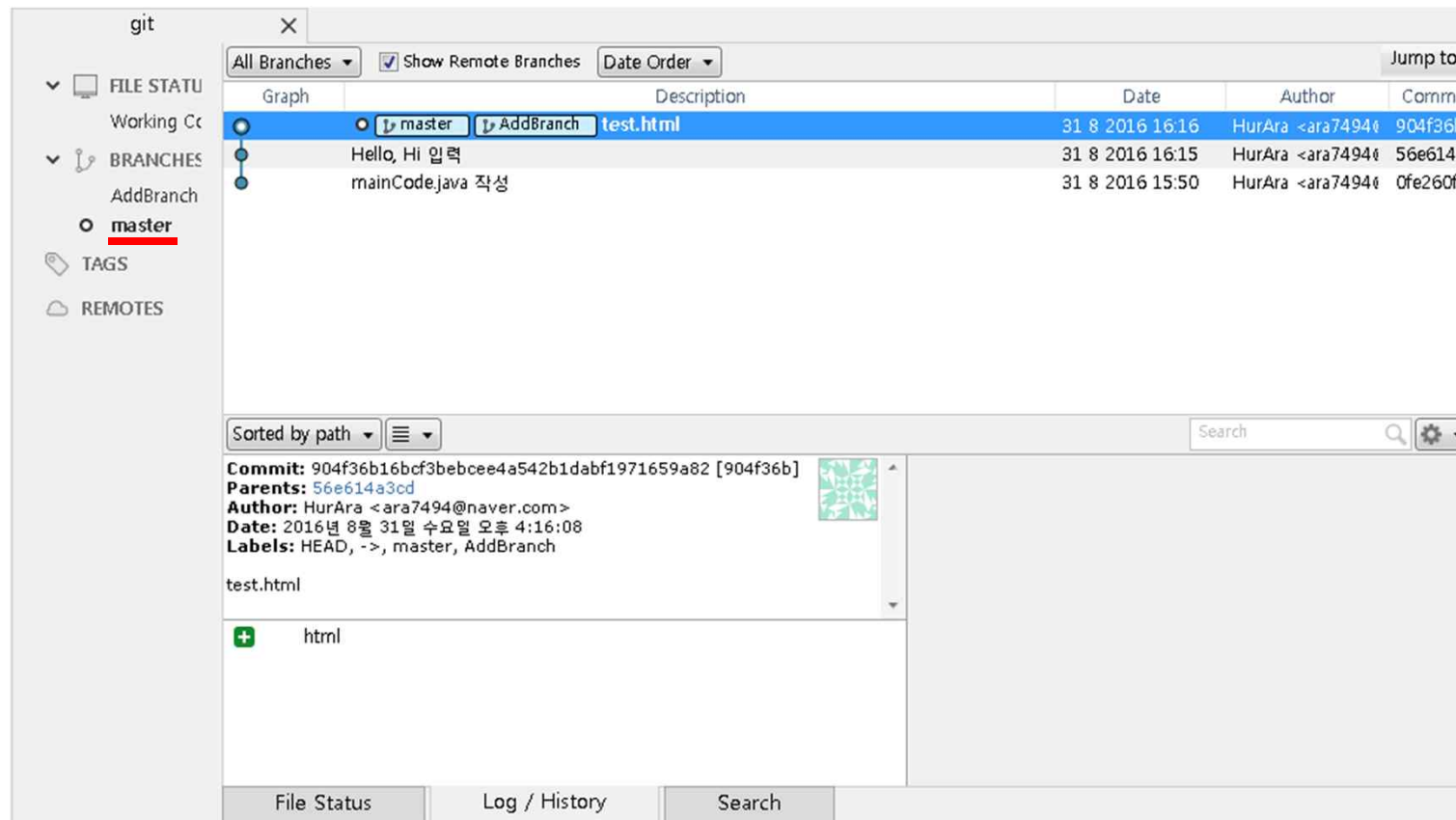
branch

- 다음과 같이 하나의 리포지토리에 두 개의 브랜치(master, AddBranch)가 생성



AddBranch의 브랜치

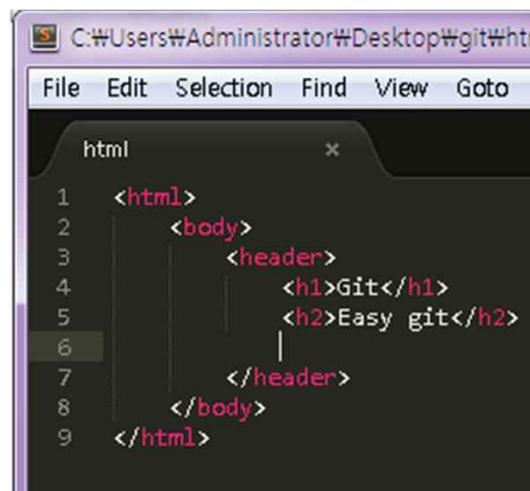
branch



Master 브랜치

branch

- 먼저 SourceTree에서 AddBranch의 브랜치를 선택한 후 내용을 수정하고 commit



The screenshot shows a text editor window titled 'C:\Users\Administrator\Desktop\git\html'. The editor displays an HTML document with the following structure:

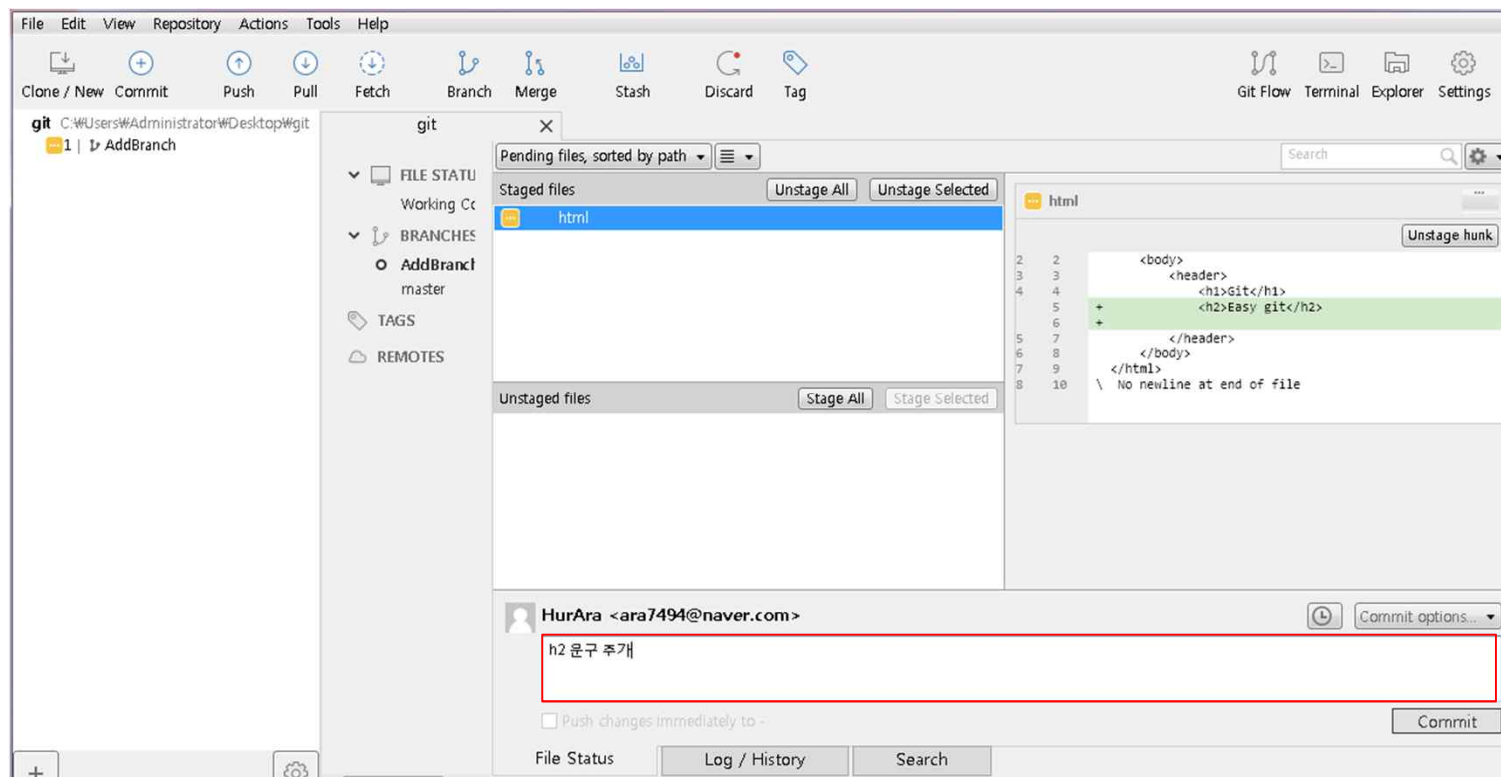
```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5       <h2>Easy git</h2>
6     </header>
7   </body>
8 </html>
```

The line containing the new `<h2>Easy git</h2>` tag is highlighted, indicating it has been added to the document.

<h2> Easy git<h2> 문구 추가

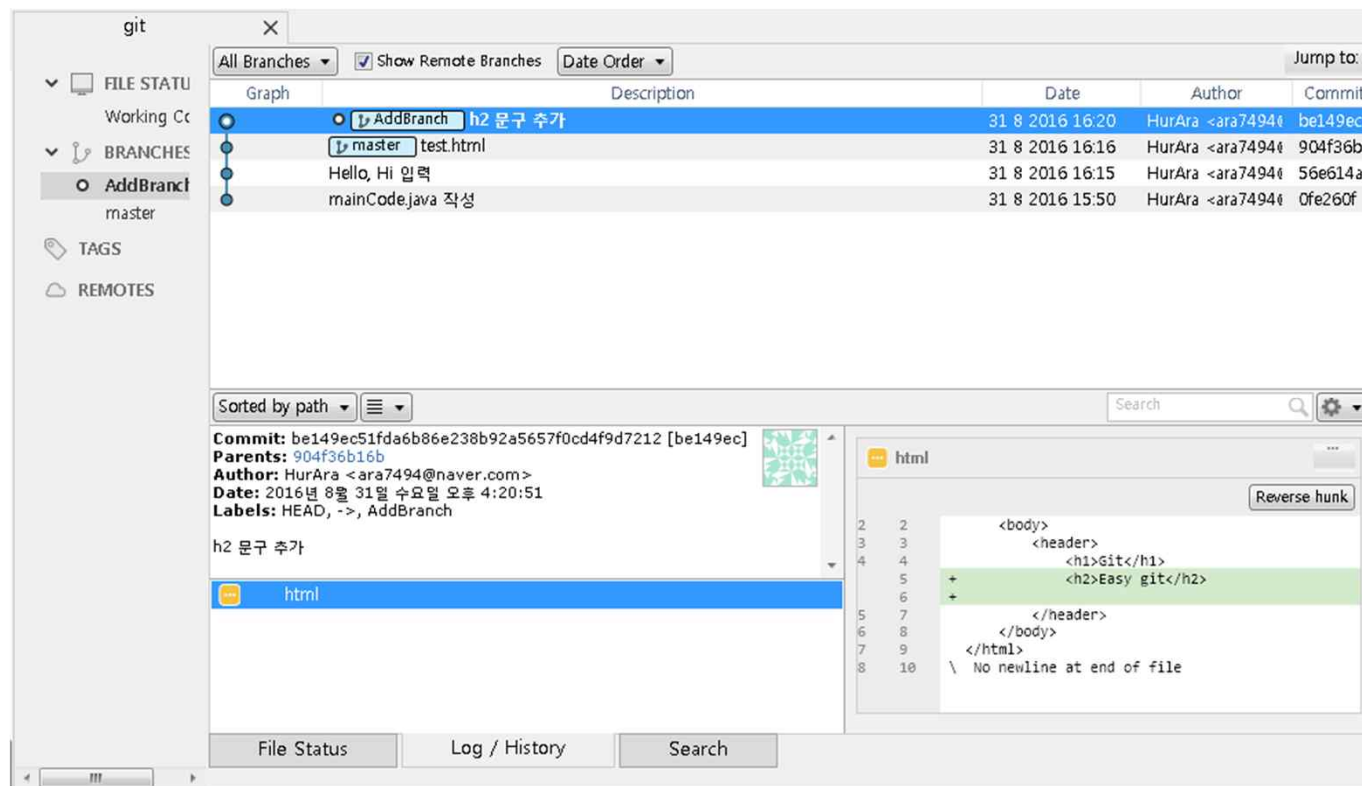
branch

- 다음과 같이 commit




branch

- AddBranch 브랜치에 새로운 커밋 로그 생성



branch

- 다음으로 **master** 브랜치를 선택

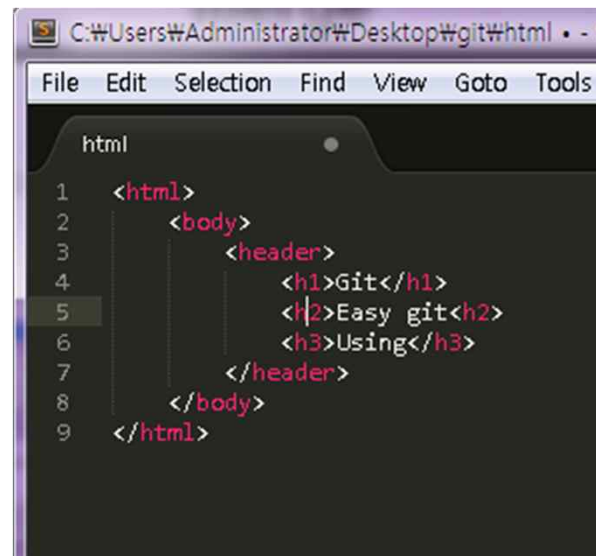


```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5     </header>
6   </body>
7 </html>
```

Master 브랜치 선택 후 사라진 **<h2>** 구문

branch

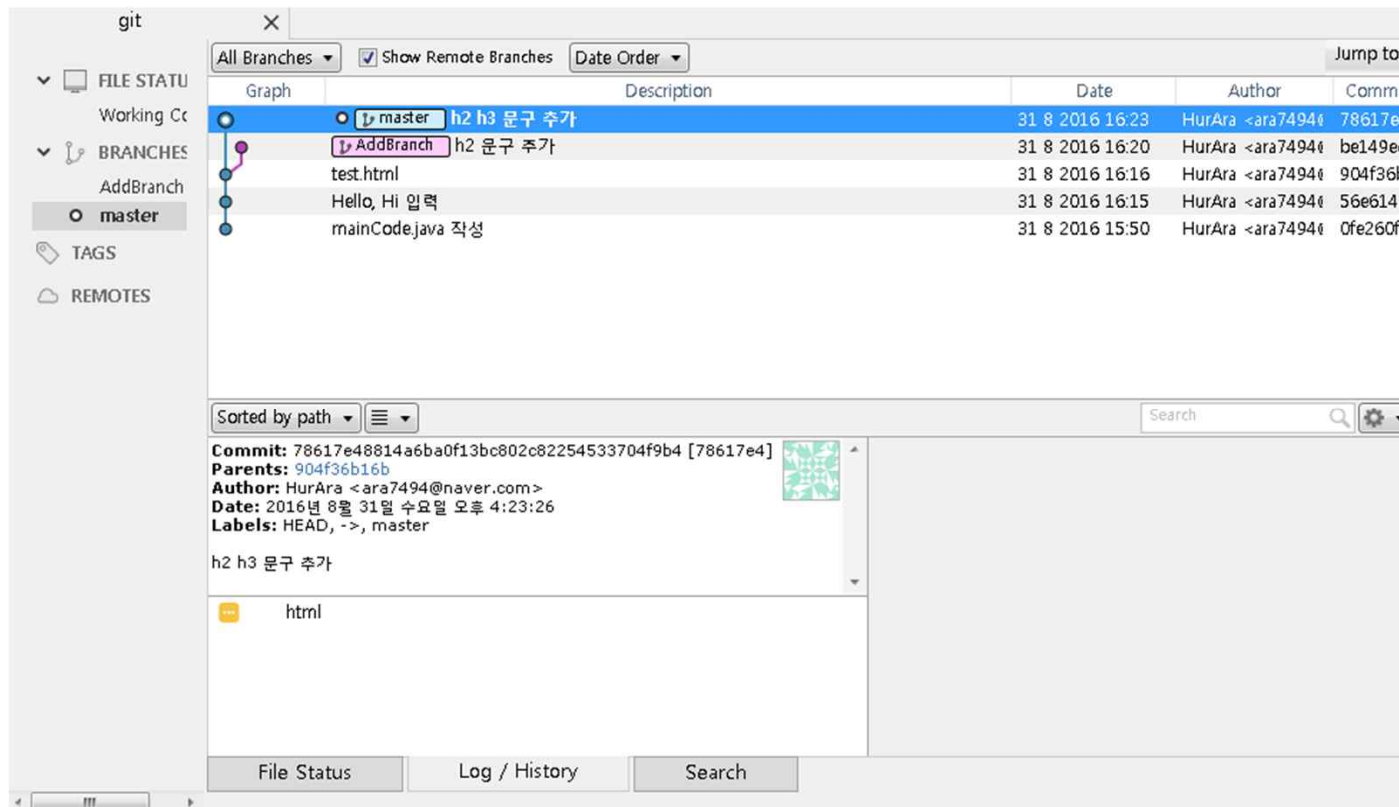
- Master 브랜치 선택 후 다음과 같이 문구 추가

A screenshot of a text editor window titled "C:\Users\Administrator\Desktop\git\html - S". The editor has a menu bar with "File", "Edit", "Selection", "Find", "View", "Goto", and "Tools". The code is written in HTML and is as follows:

```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5       <h2>Easy git</h2>
6       <h3>Using</h3>
7     </header>
8   </body>
9 </html>
```

branch

- Master 브랜치에 새로운 커밋 로그 생성 함
(2개의 다른 브랜치에 각각의 커밋 로그가 생성)



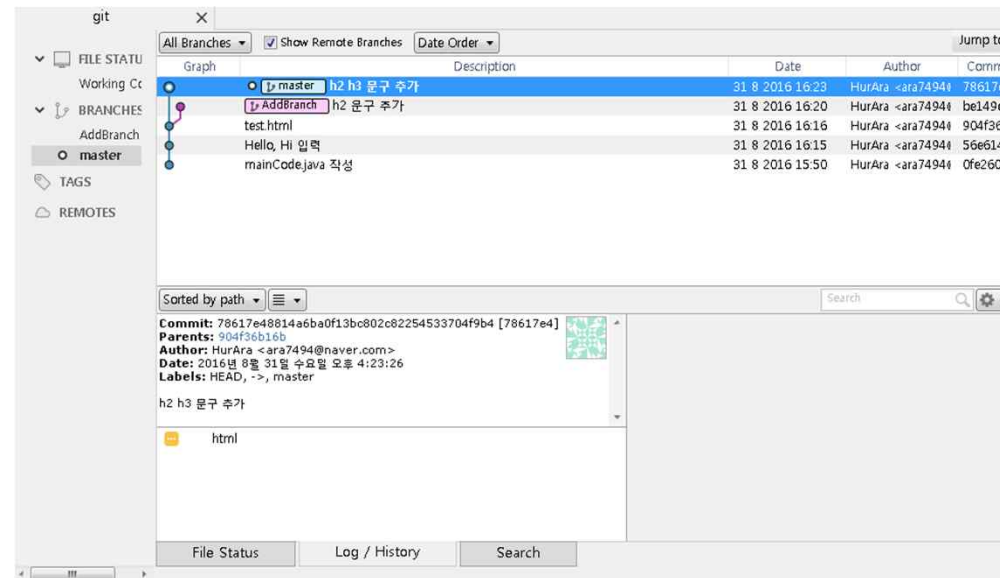
SourceTree를 활용한 브랜치 합치기(merge)

merge

- 두 개의 작업을 하나의 버전으로 만들어 서비스를 제공하는 것이다.

Merge 충돌

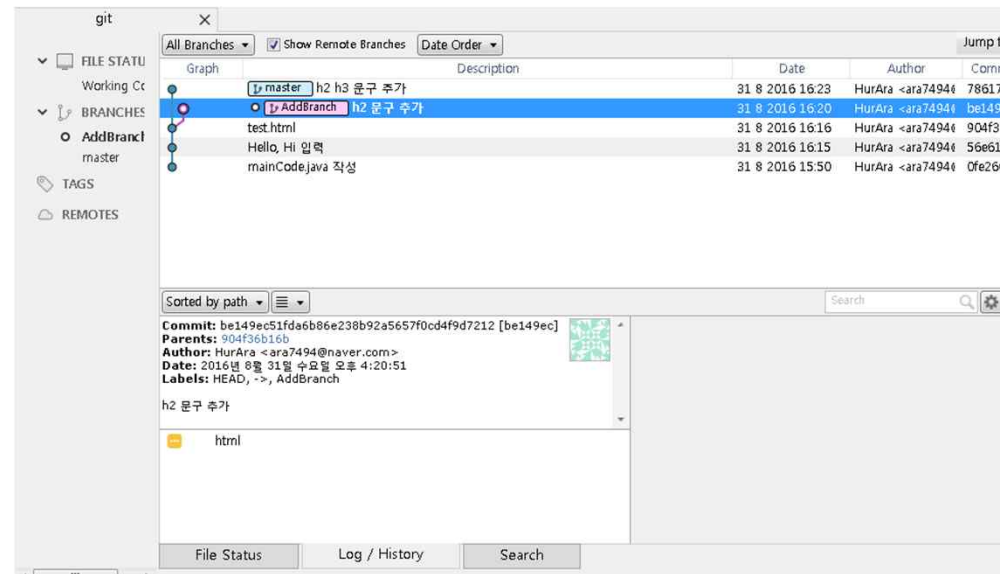
- Master 브랜치의 상태



```
html
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5       <h2>Easy git<h2>
6       <h3>Using</h3>
7     </header>
8   </body>
9 </html> |
```

Merge 충돌

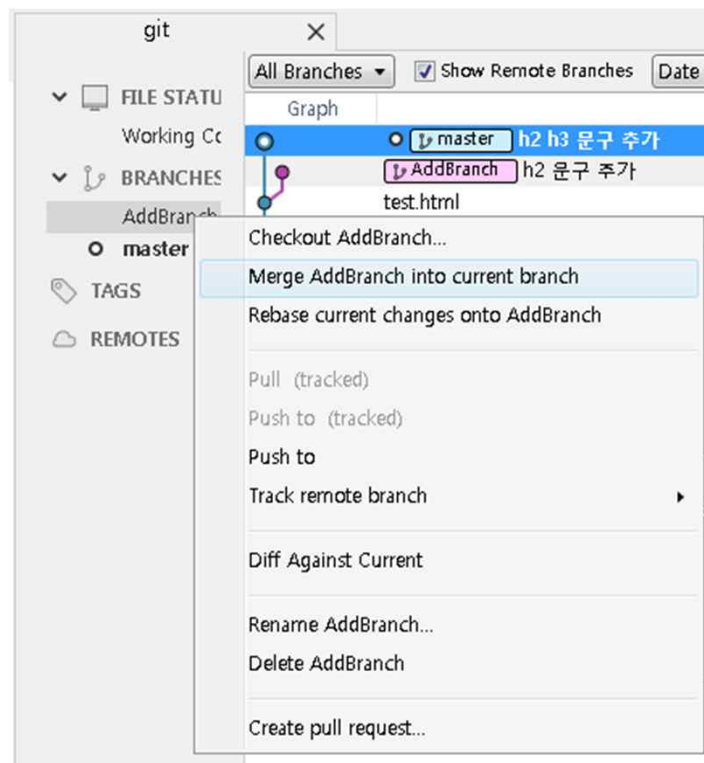
- AddBranch 브랜치의 상태



```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5       <h2>Easy git</h2>
6     </header>
7   </body>
8 </html>
```

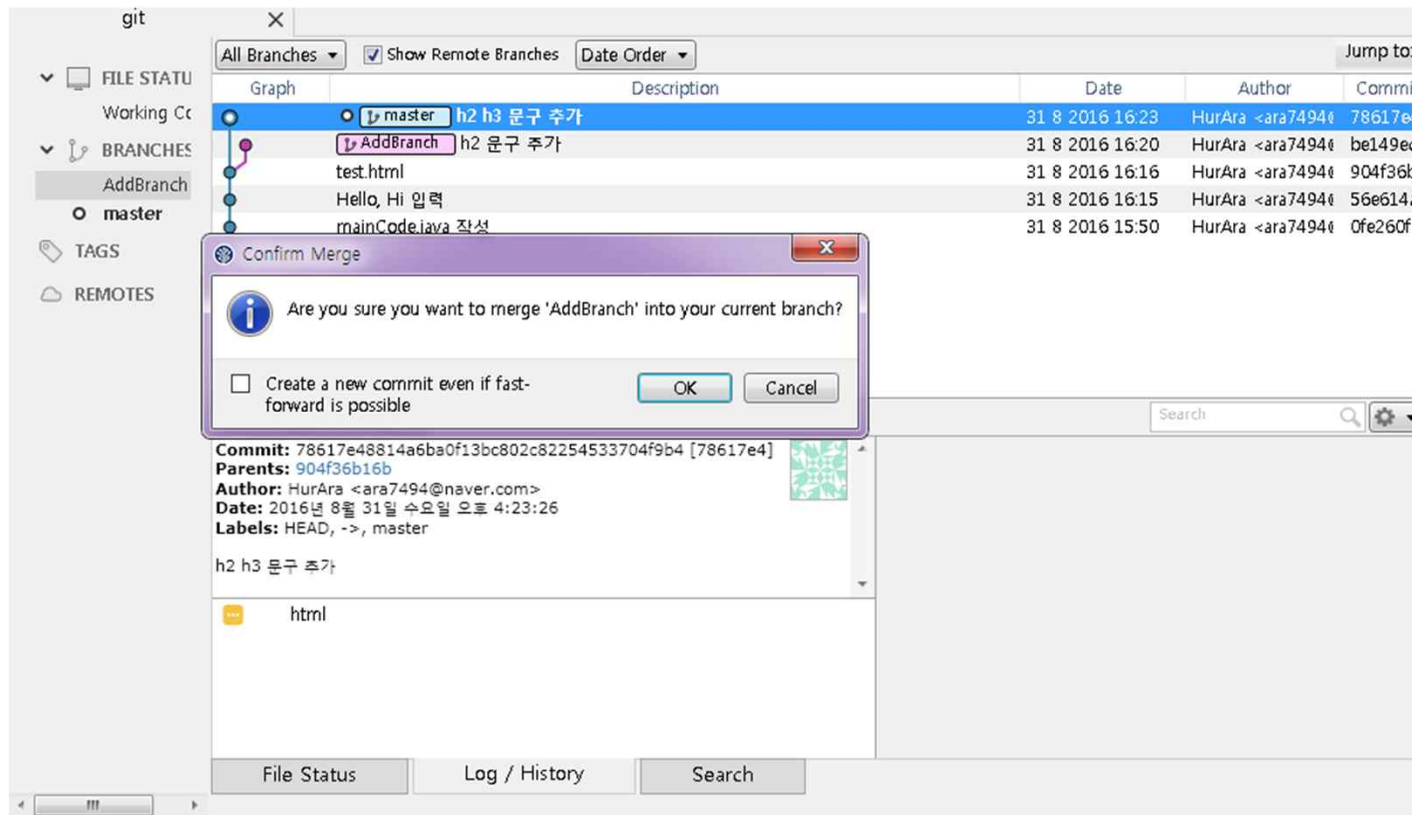

Merge 충돌

- 가져올 branch를 마우스 오른쪽 클릭을 하고 “Merge AddBranch into current branch” 클릭



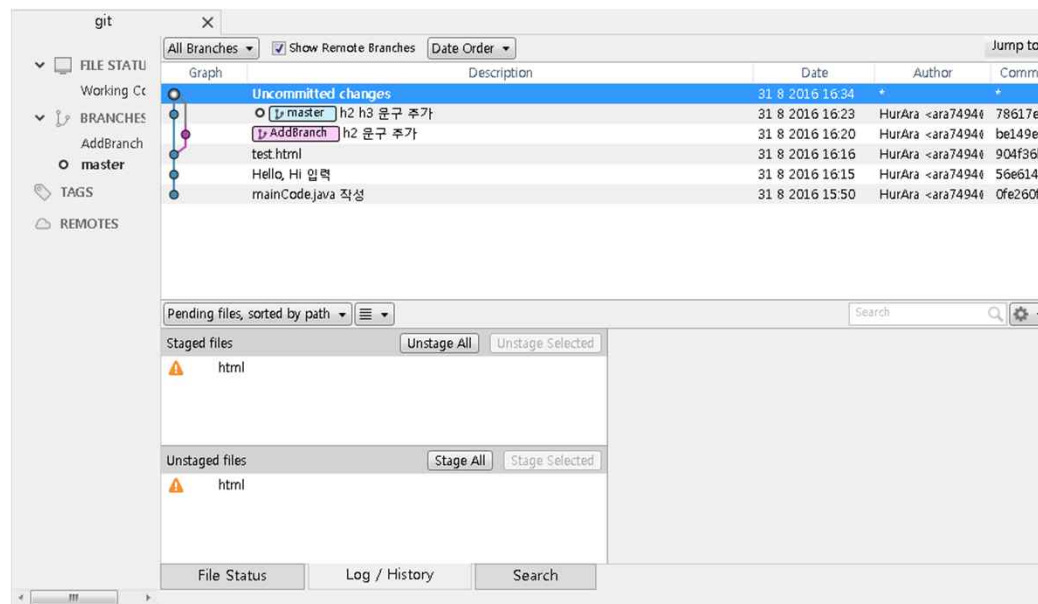
Merge 충돌

- Confirm Merge 팝업이 뜨면 “OK”를 클릭



Merge 충돌

- 여기서 master branch에 AddBranch를 merge 할 경우 아래와 같이 목록에 “Uncomitted changes”가 됨



충돌된 결과

Merge 충돌

- 충돌의 원인은 두 개의 **branch**의 내용을 수정할 때, 동일한 위치의 코드를 추가하거나 수정하였기 때문이다.



```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5       <h2>Easy git</h2>
6       <h3>Using</h3>
7     </header>
8   </body>
9 </html> |
```

Master branch

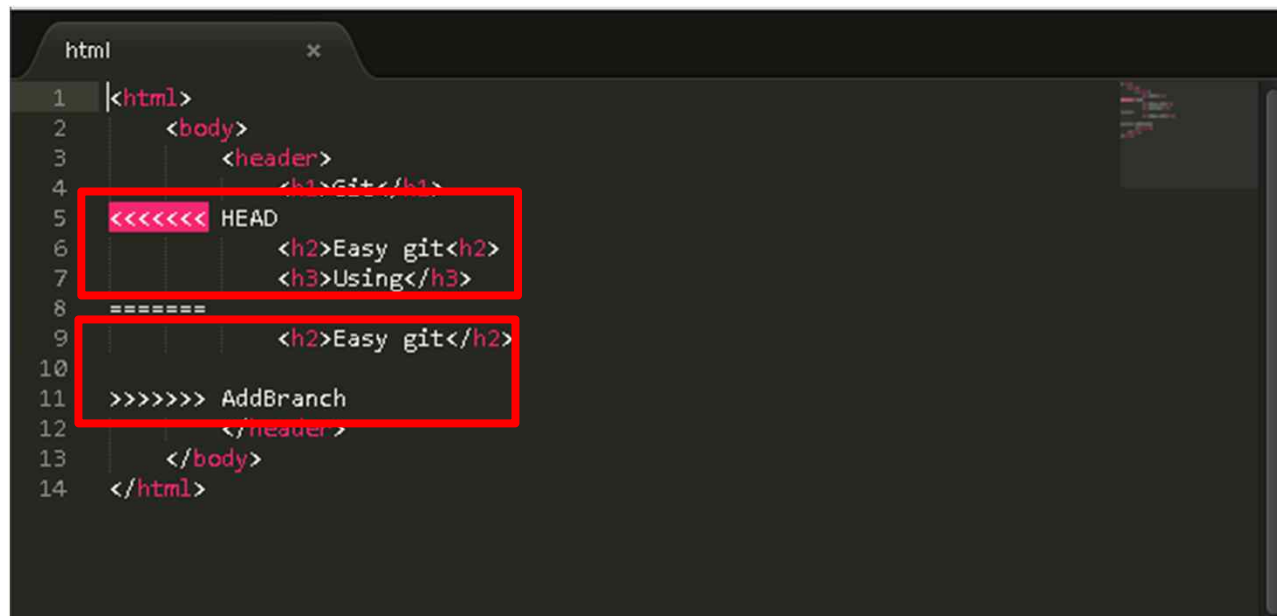


```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5       <h2>Easy git</h2>
6     </header>
7   </body>
8 </html> |
```

AddBranch branch

Merge 충돌

- 충돌이 발생한 후 master branch의 내용은 아래와 같이 두개의 branch의 내용을 모두 포함하면서 사용자에게 코드의 수정을 요구한다.

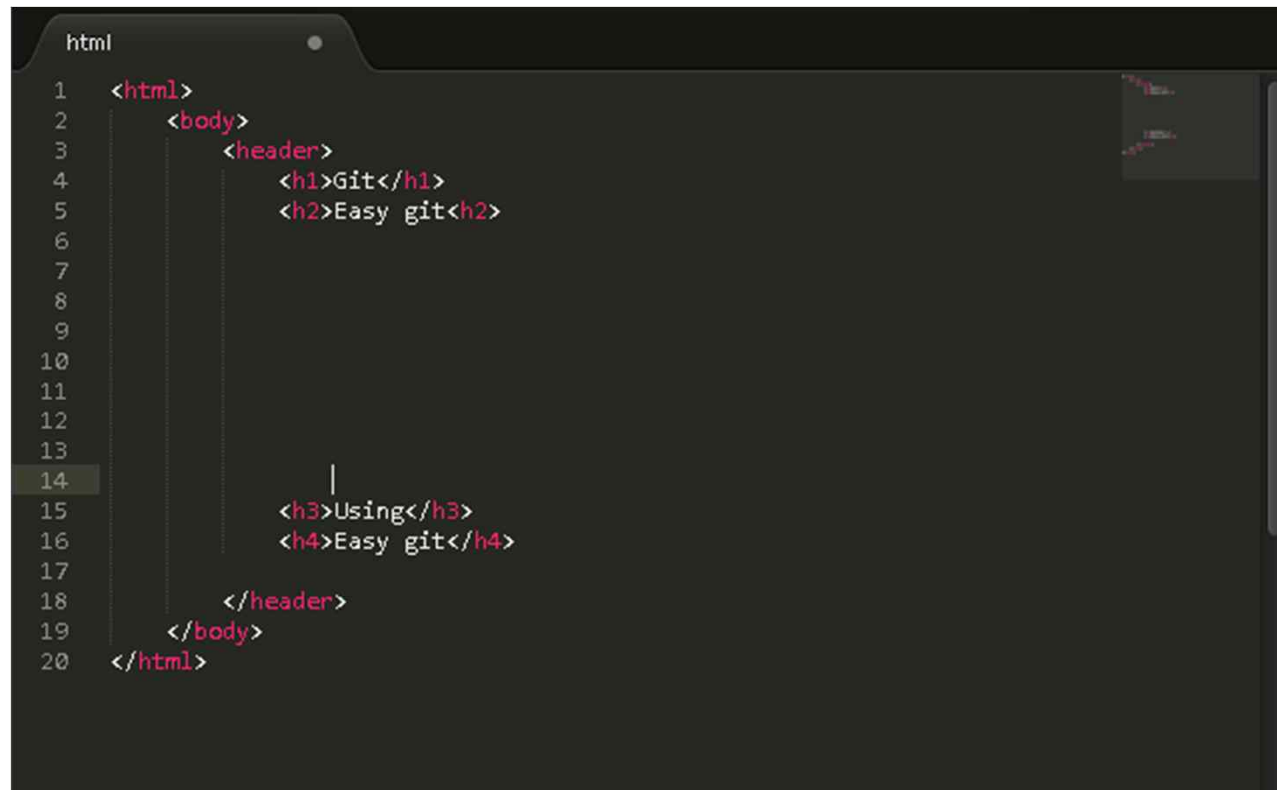


```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5     <<<<<<< HEAD
6       <h2>Easy git</h2>
7       <h3>Using</h3>
8     =====
9       <h2>Easy git</h2>
10    >>>>>>> AddBranch
11    </header>
12  </body>
13 </html>
```

Merge Conflict가 발생

Merge 충돌 해결방법

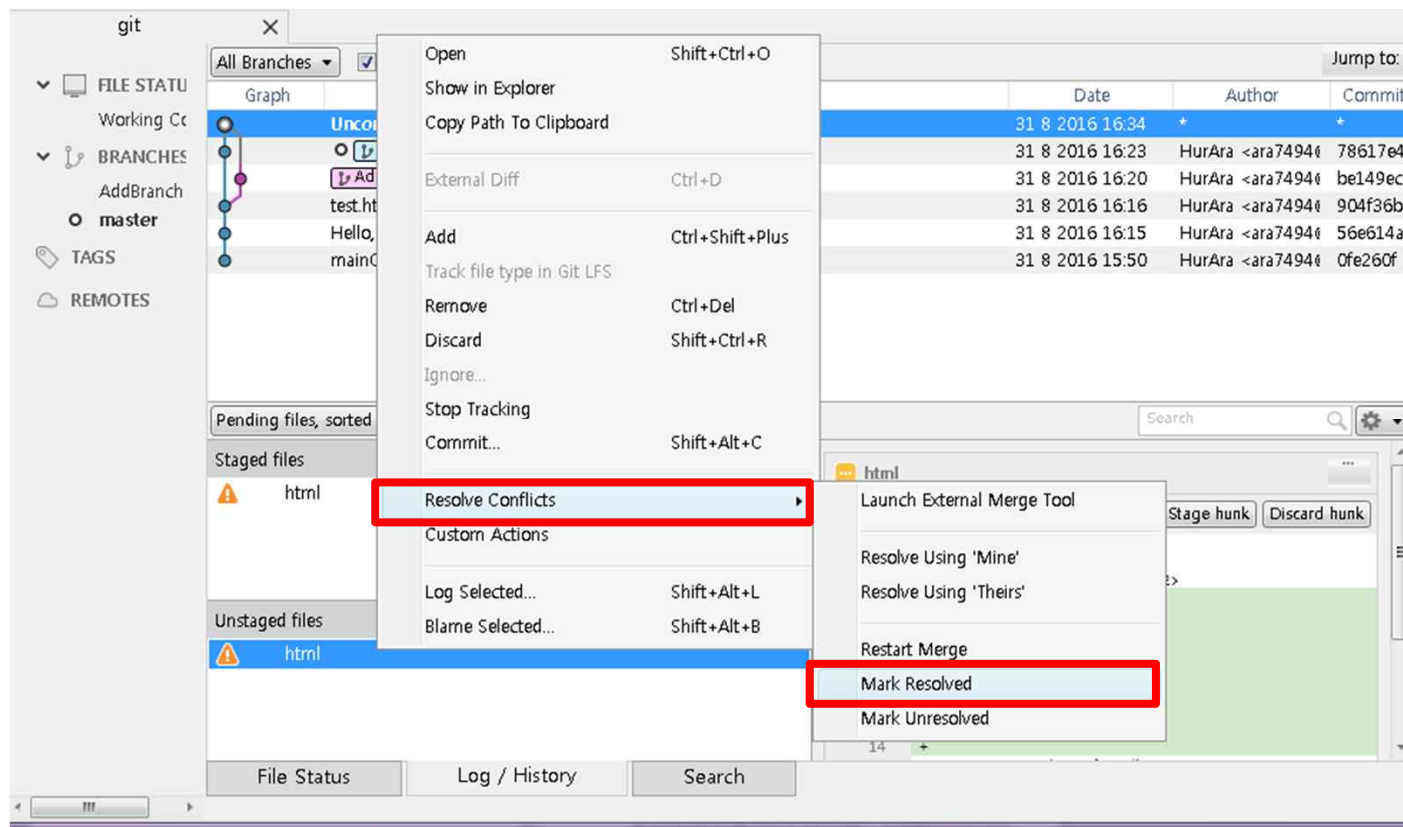
- 이전의 merge의 결과로 나온 내용을 다음과 같이 수정



```
1 <html>
2   <body>
3     <header>
4       <h1>Git</h1>
5       <h2>Easy git</h2>
6
7
8
9
10
11
12
13
14
15       <h3>Using</h3>
16       <h4>Easy git</h4>
17
18     </header>
19   </body>
20 </html>
```

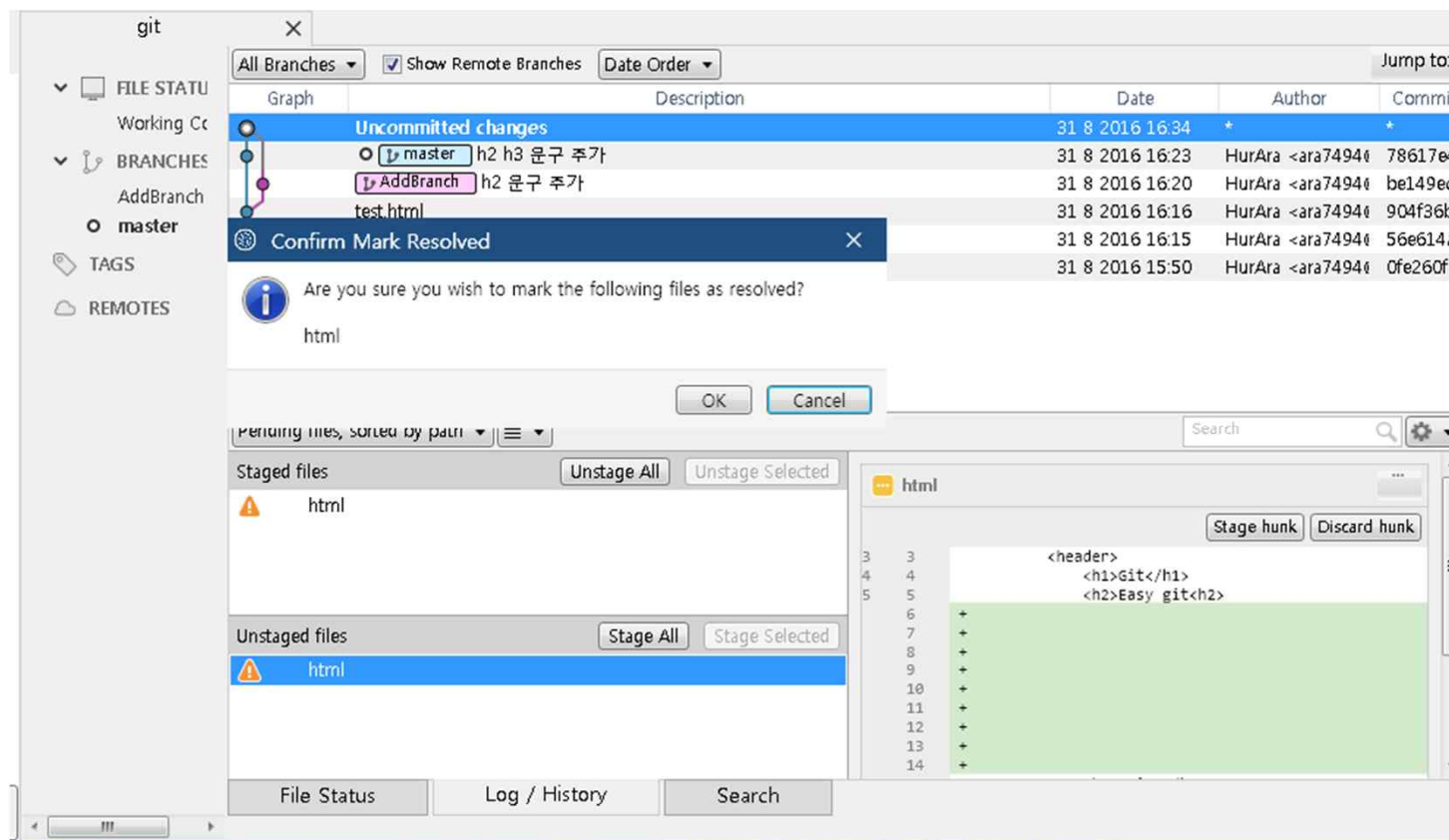
Merge 충돌 해결방법

- SourceTree에서 Resolve Conflicts 항목에 마우스 오른쪽 클릭을 하고 아래 그림과 같이 Mark Resolved를 클릭



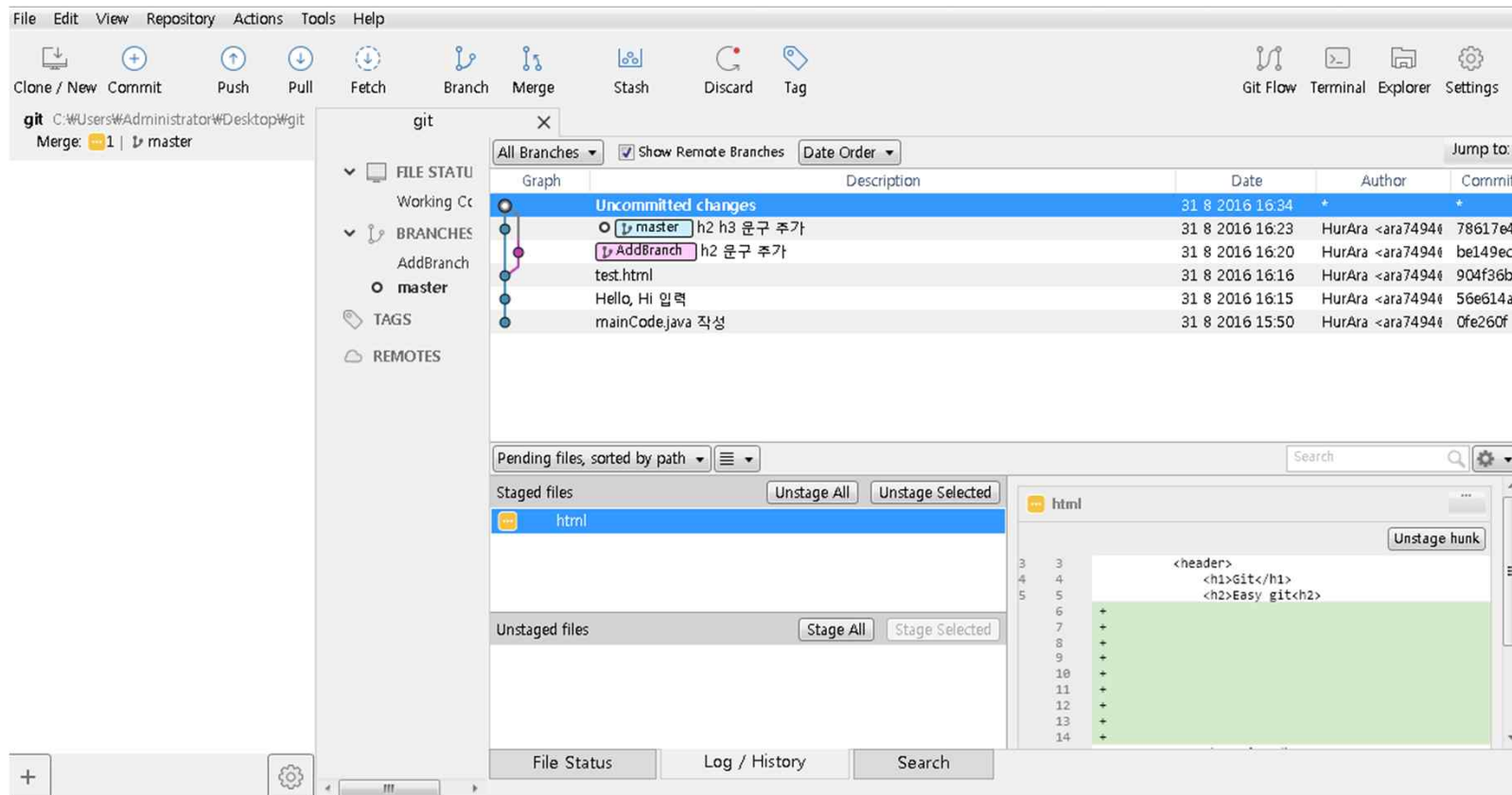
Merge 충돌 해결방법

- OK 클릭



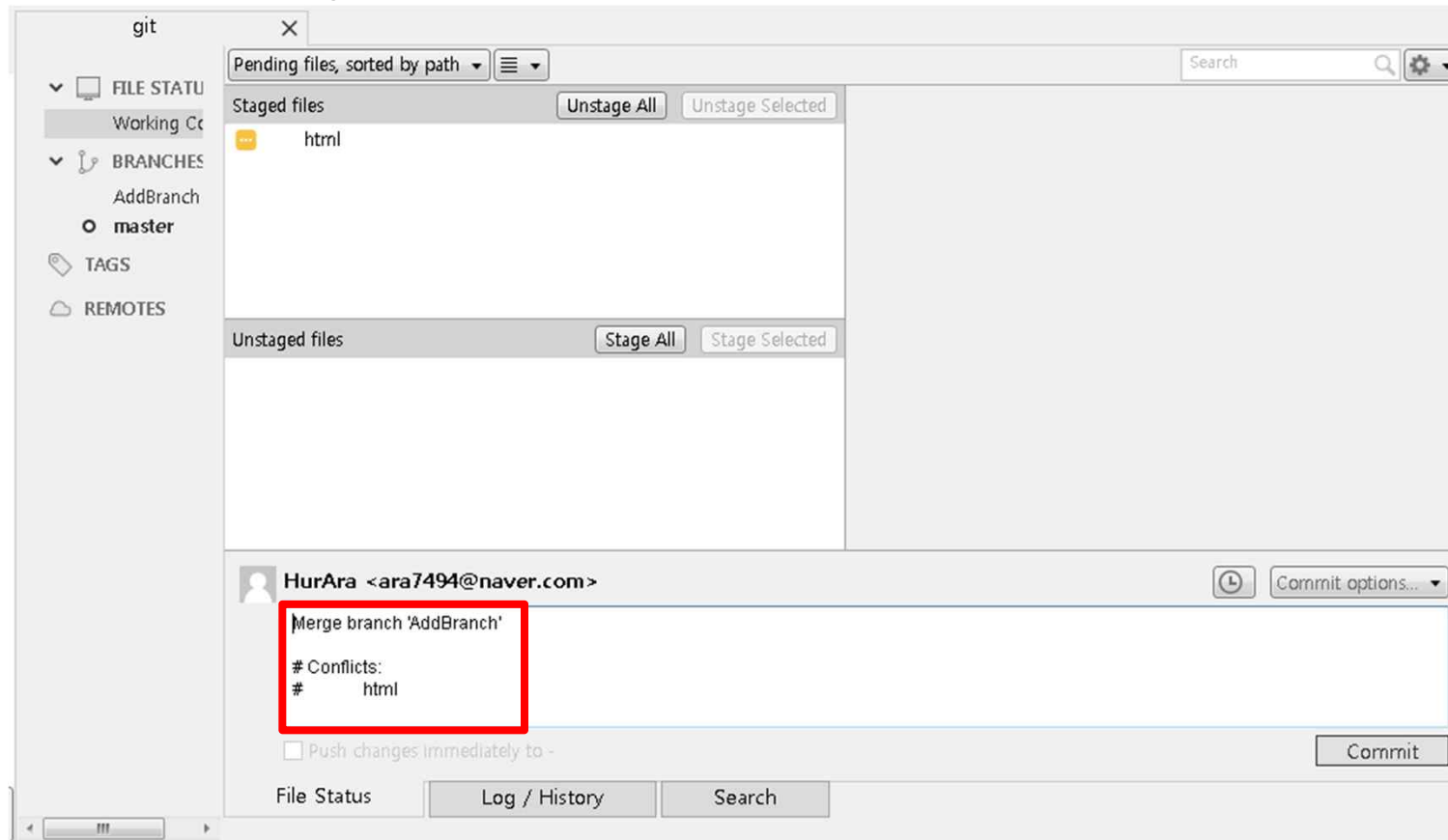
Merge 충돌 해결방법

- 결과 아래와 같이 화면이 나타남



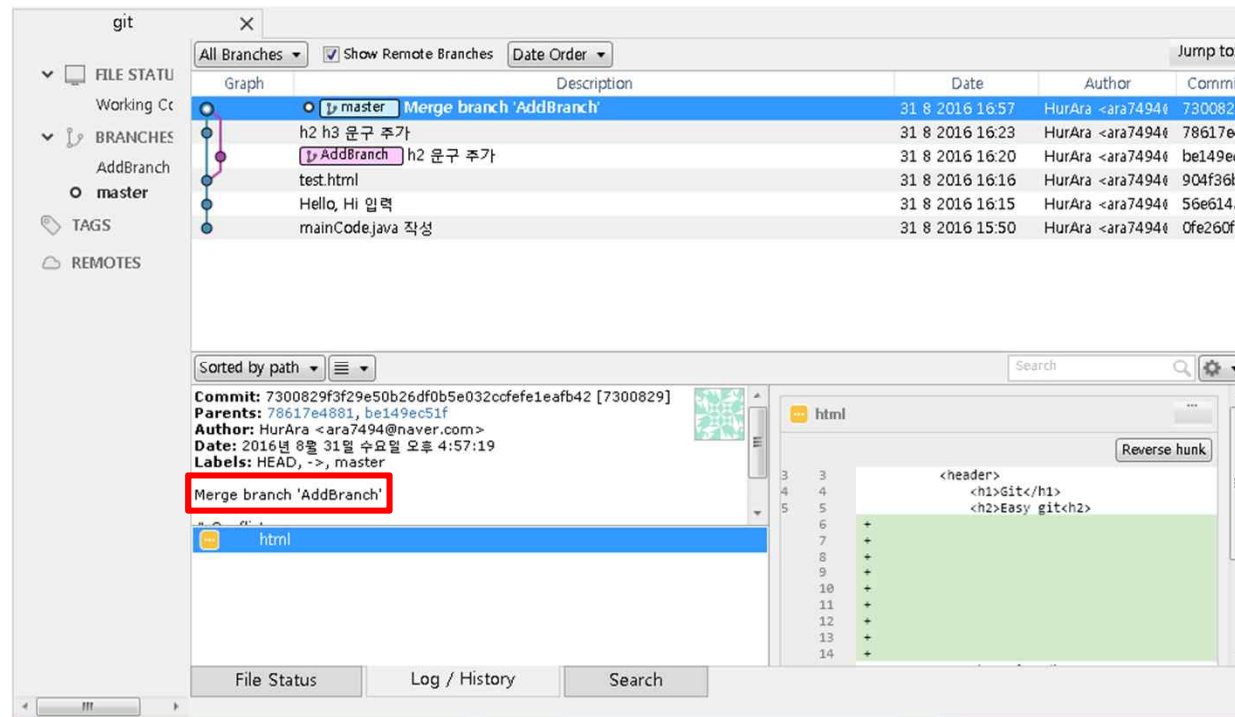
Merge 충돌 해결방법

- 이 부분을 커밋을 해주게 되면 자동으로 커밋 메시지가 생성
- 커밋을 해주면 merge 됨



Merge 충돌 해결방법

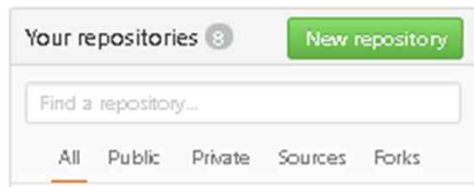
- 충돌을 해결한 master branch



SourceTree, GitHub를 활용한 원격 저장소 만들기

원격저장소 만들기

- <https://github.com/> 에 접속하여 회원가입을 하고 로그인
- New repository 클릭



원격저장소 만들기

- Repository name 란에 저장소의 이름을 쓰고 하단의 “Create repository” 클릭

Create a new repository
A repository contains all the files for your project, including the revision history.

Owner: testmju / Repository name: git.test 저장소 이름

Great repository names are short and memorable. Need inspiration? How about musical-waddle.

Description (optional)

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

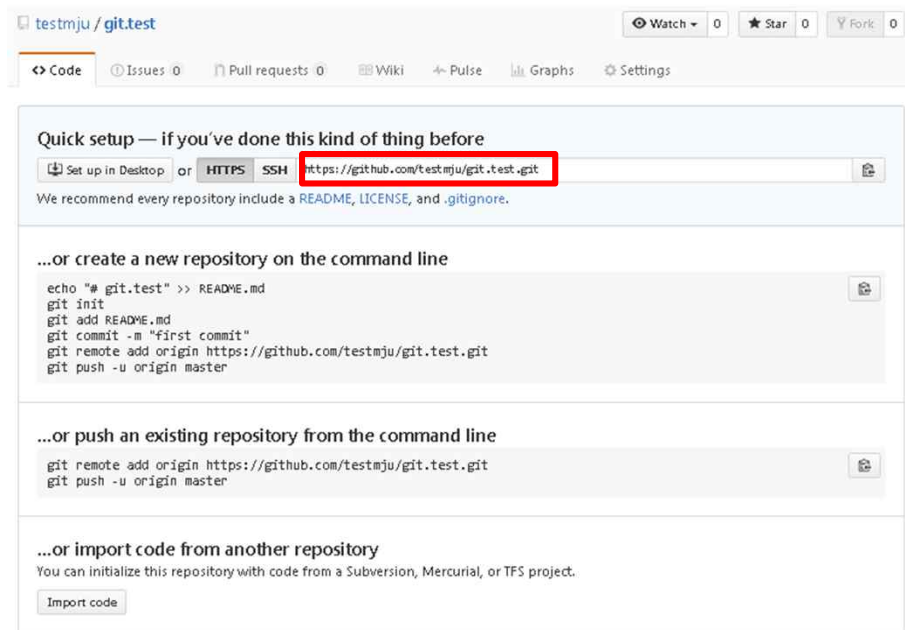
Add .gitignore: None | Add a license: None ⓘ

Create repository

원격저장소 만들기

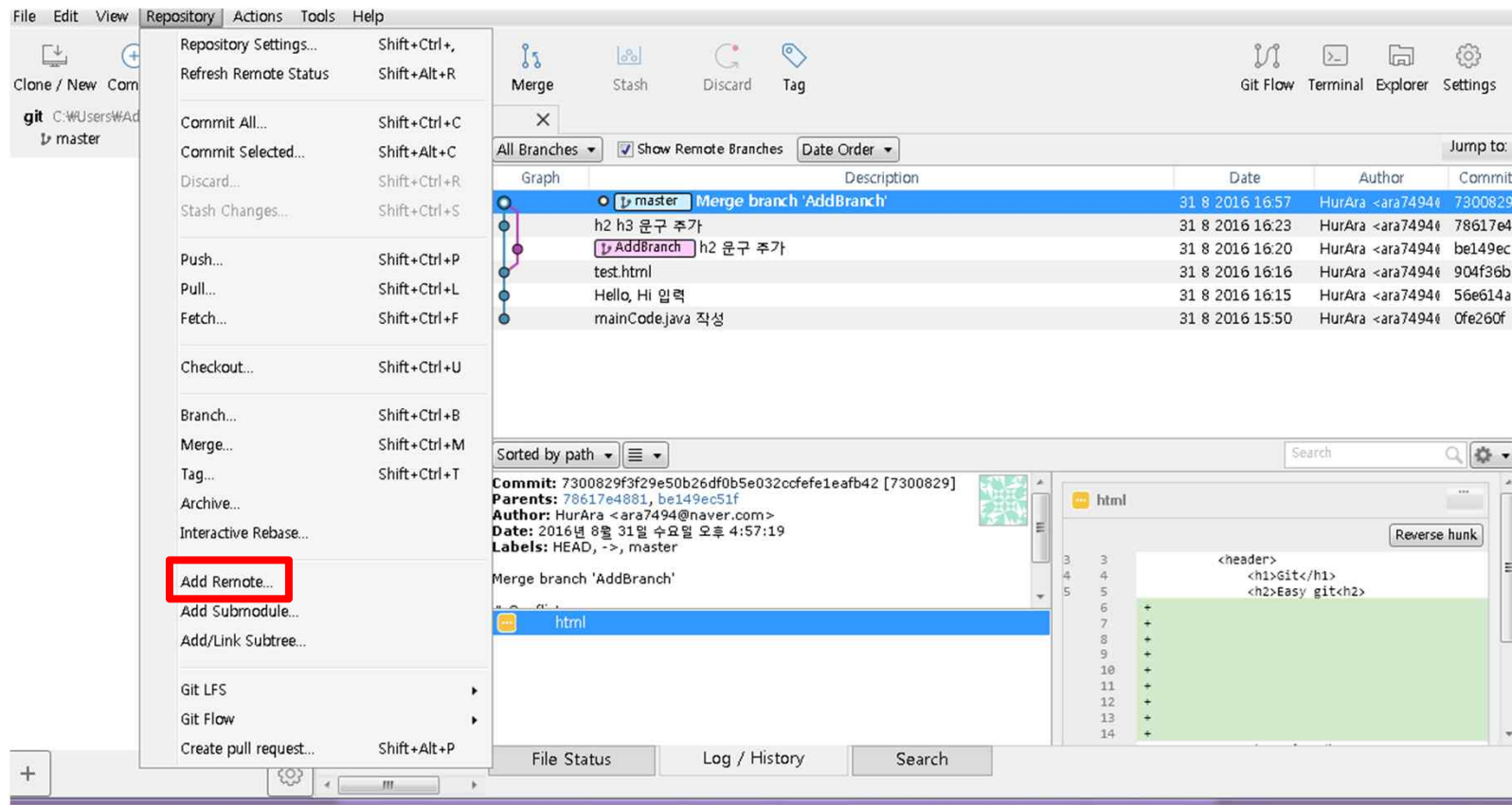
- 생성된 저장소에 자신의 컴퓨터에 저장되어 있는 장소를 복사하기 위해 상단의 HTTPS 항목의 주소 값을 복사

※ 아래 'or create a new repository on the command line' 이하의 항목은 SourceTree와 같은 유저 인터페이스 기반이 아닌 텍스트 기반 인터페이스에서 원격 저장소를 추가하기 위해 사용 됨



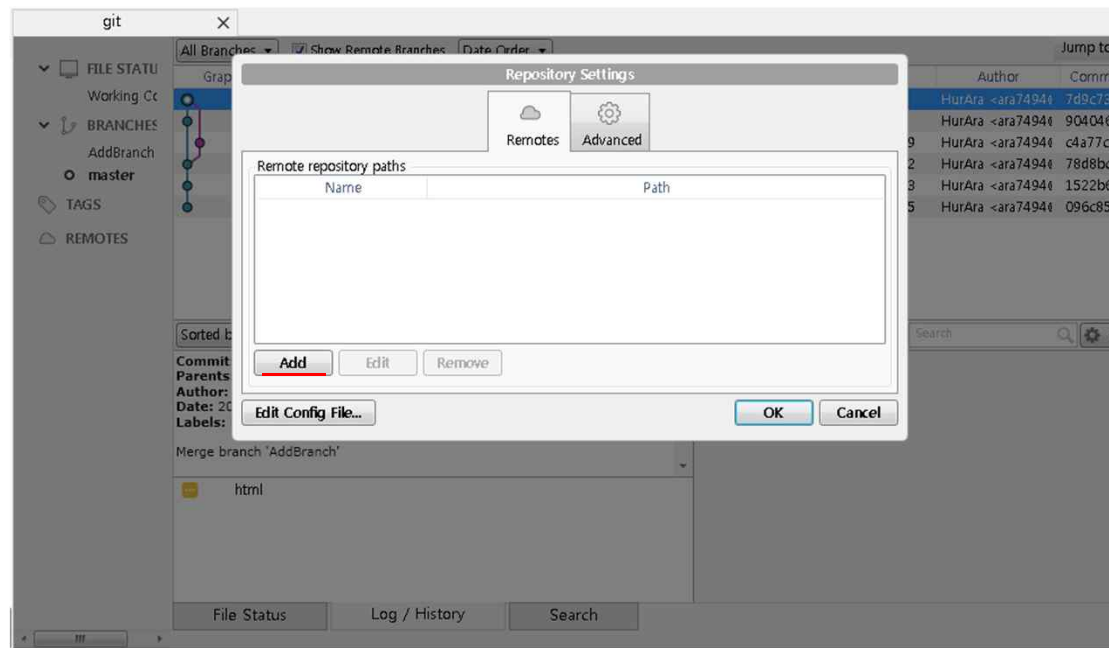
원격저장소 만들기

- SourceTree 상단 메뉴의 “Repository → Add Remote” 클릭



원격저장소 만들기

- 아래의 그림과 같이 해당 로컬 저장소의 원격 저장소를 관리하는 창이 나오면 “Add” 클릭



원격저장소 만들기

- 앞서 복사해두었던 GitHub의 원격 저장소 “HTTPS” 항목의 값을 “URL / Path” 란에 입력
- 원격 저장소가 로컬 저장소의 첫번째 원격 저장소일 경우 “Default remote” 체크하여 기본 원격 저장소로 지정하고 “OK” 클릭

Remote details

Required information

Remote name: ☒ Default remote

URL / Path:

Optional extended integration

Host Type:

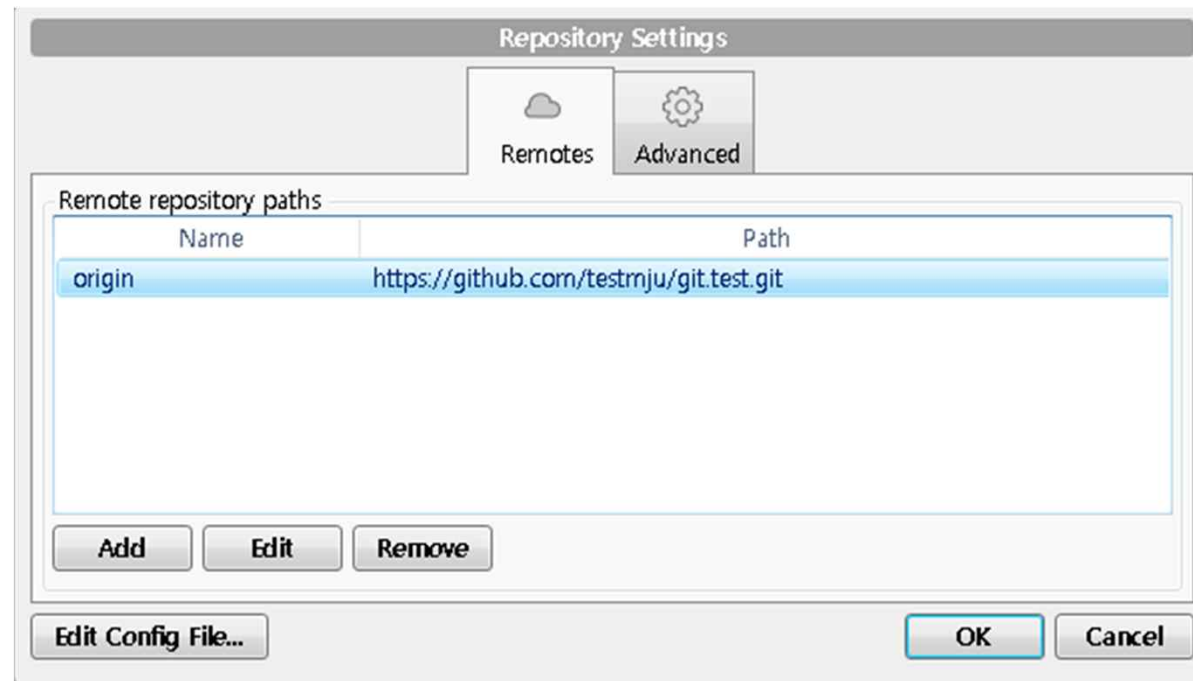
Host Root URL:

Username:

Extended integration is used to enable deeper integration with hosting providers such as Bitbucket, including locating existing clones when following links from sites and creating pull requests.

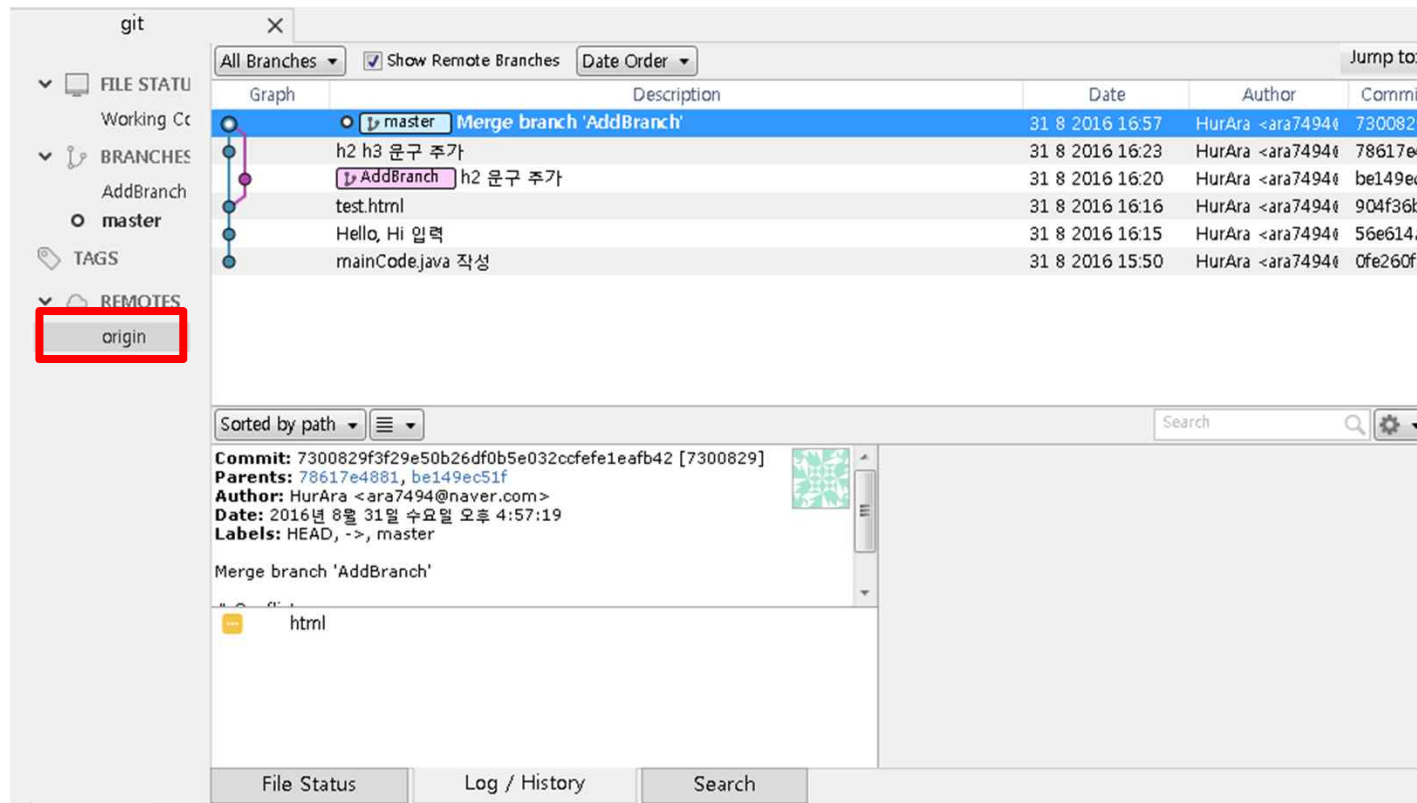
원격저장소 만들기

- 아래와 같이 원격 저장소 목록에 연결한 “origin” 생성이 됨 → “OK” 클릭



원격저장소 만들기

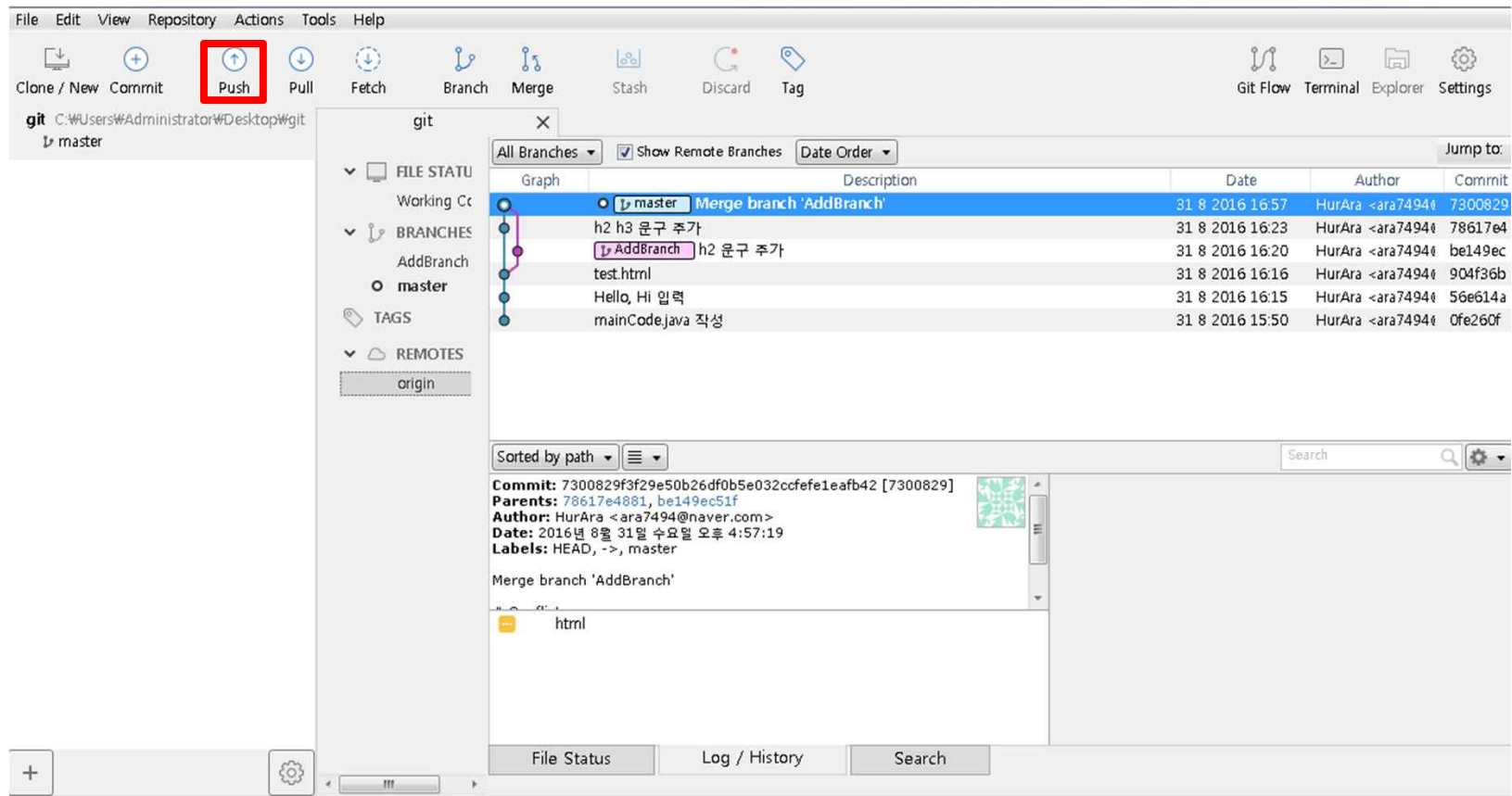
- SourceTree의 메인화면 Remotes 항목에 원격 저장소 origin이 표시 됨



SourceTree, GitHub를 활용한 Push

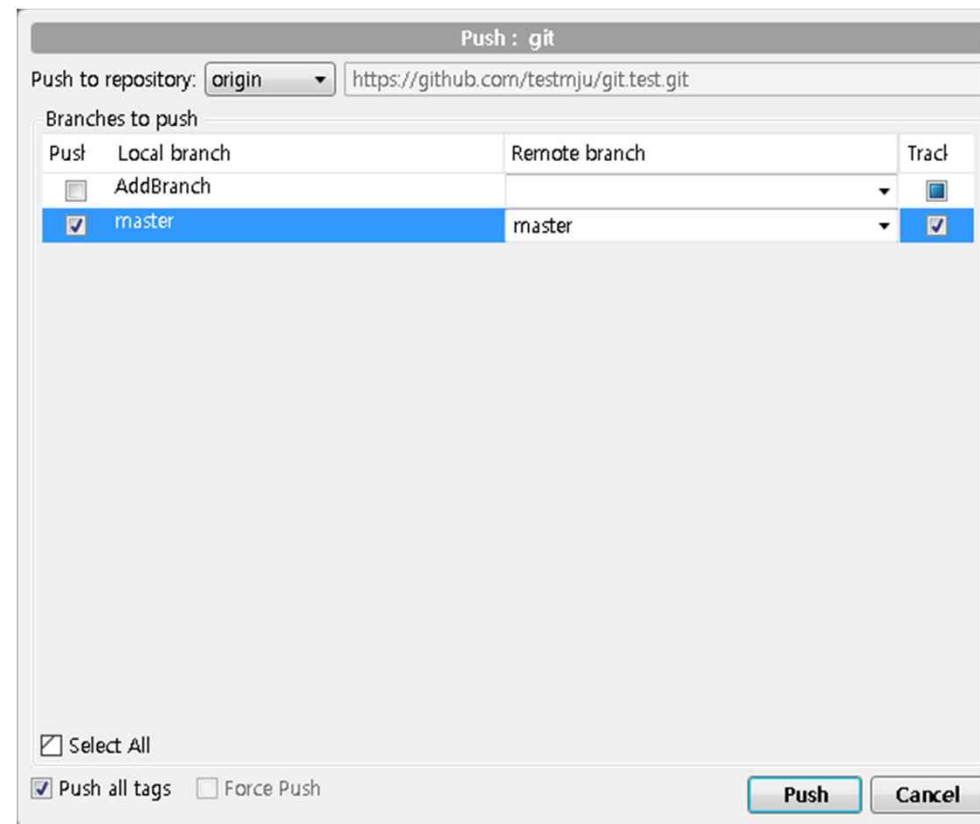
push

- SourceTree에서 해당하는 로컬 저장소를 선택한 뒤 상단 메뉴의 “Push” 클릭



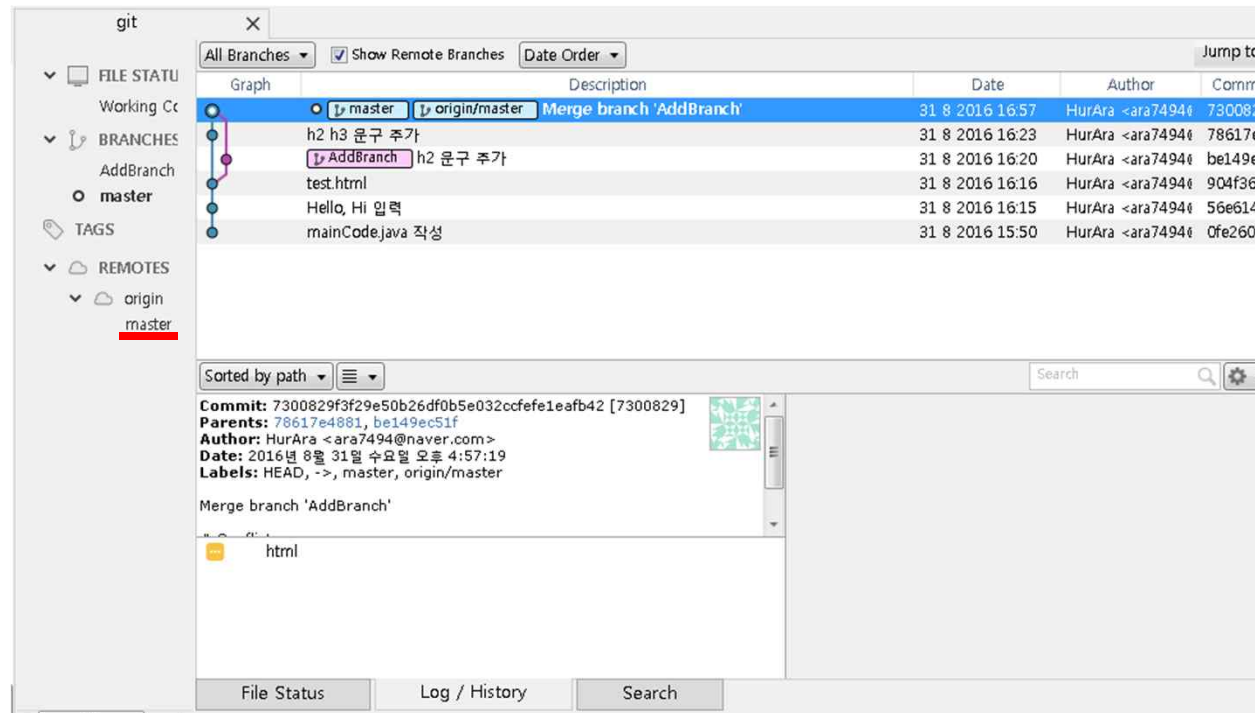
push

- 다음과 같이 선택하고 올리고자 하는 **branch**를 선택 → “Push” 클릭 → Username과 Password를 입력하라는 창이 나오면 자신의 GitHub 계정 입력



push

- Push가 완료되면 아래와 같이 원격 저장소 **origin**에 Push 과정에서 선택한 **master** 브랜치가 등록 됨
- 또한 사이트에 접속하여 해당 원격 저장소를 확인해보면 로컬 저장소의 프로젝트가 동기화되어 있음



원격 저장소 **origin**에 등록된
master 브랜치

push

- Push가 완료되면 아래와 같이 원격 저장소 origin에 Push 과정에서 선택한 master 브랜치가 등록 됨
- 또한 사이트에 접속하여 해당 원격 저장소를 확인해보면 로컬 저장소의 프로젝트가 동기화되어 있음

No description or website provided. — Edit

5 commits

1 branch

0 releases

1 contributor

Branch: master ▾ New pull request

Create new file Upload files Find file Clone or download ▾

testmju Merge branch 'AddBranch' ... Latest commit 730829 12 minutes ago

Hello, Hi

Hello, Hi 입력

an hour ago

html

Merge branch 'AddBranch'

12 minutes ago

mainCode.java

mainCode.java 작성

an hour ago

Help people interested in this repository understand your project by adding a README. Add a README

SourceTree, GitHub를 활용한 clone

clone

- 백업해둔 프로젝트를 내컴퓨터로 받아오거나 협업을 위해 새로운 개발자가 프로젝트를 로컬 저장소로 받아와야하는 경우가 있다. 이때 **clone**을 통해 로컬 저장소에 원격 저장소에 있는 프로젝트를 가져올 수 있다.

clone

- GitHub에 있는 가져올 원격 저장소에 접근하여 HTTPS 값을 복사

No description or website provided. — Edit

The screenshot shows a GitHub repository interface. At the top, it displays repository statistics: 6 commits, 1 branch, 0 releases, and 1 contributor. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The 'Clone or download' button is highlighted with a red rectangle. A dropdown menu is open, showing 'Clone with HTTPS' as the selected option. The resulting URL, 'https://github.com/testmju/git.test.git', is highlighted with a red rectangle. Other options in the dropdown include 'Use SSH', 'Open in Desktop', and 'Download ZIP'. Below the dropdown, there is a section for adding a README file.

6 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

testmju Merge branch 'AddBranch'

Hello, Hi	Hello, Hi 입력
html	Merge branch 'AddBranch'
mainCode.java	mainCode.java 작성

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL.

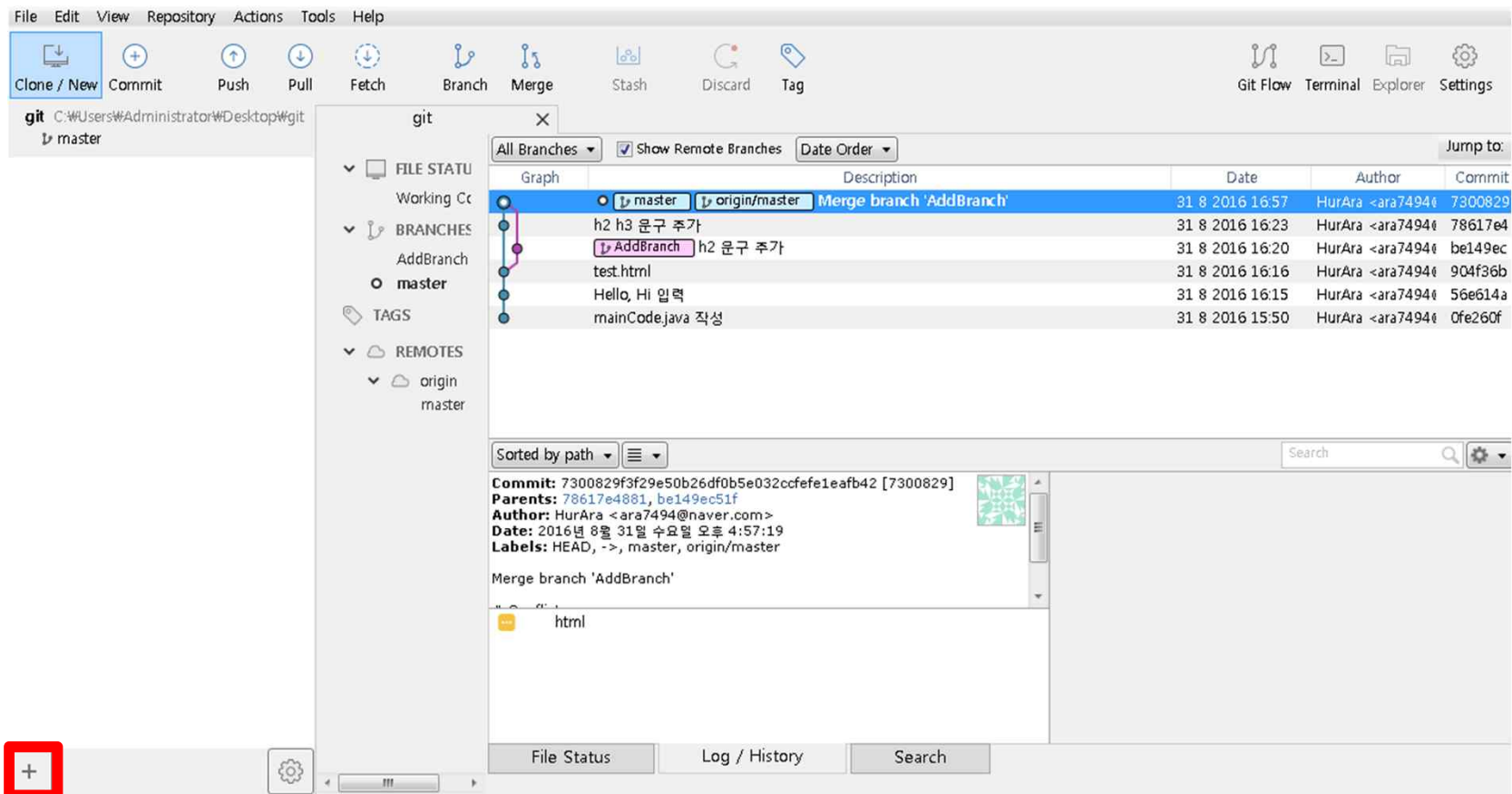
`https://github.com/testmju/git.test.git`

Open in Desktop Download ZIP

Help people interested in this repository understand your project by adding a README. Add a README

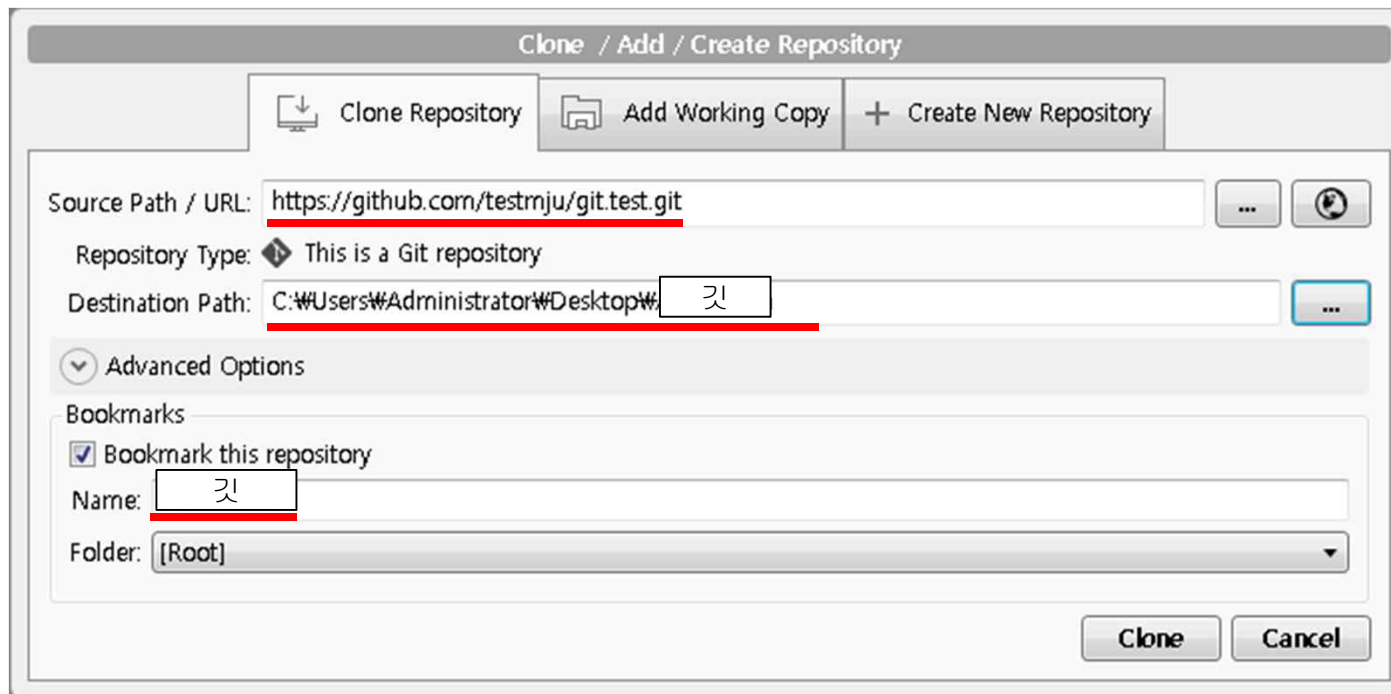
clone

- SourceTree에서 왼쪽 하단의 “Add Repository” 클릭 (빨간색 줄)



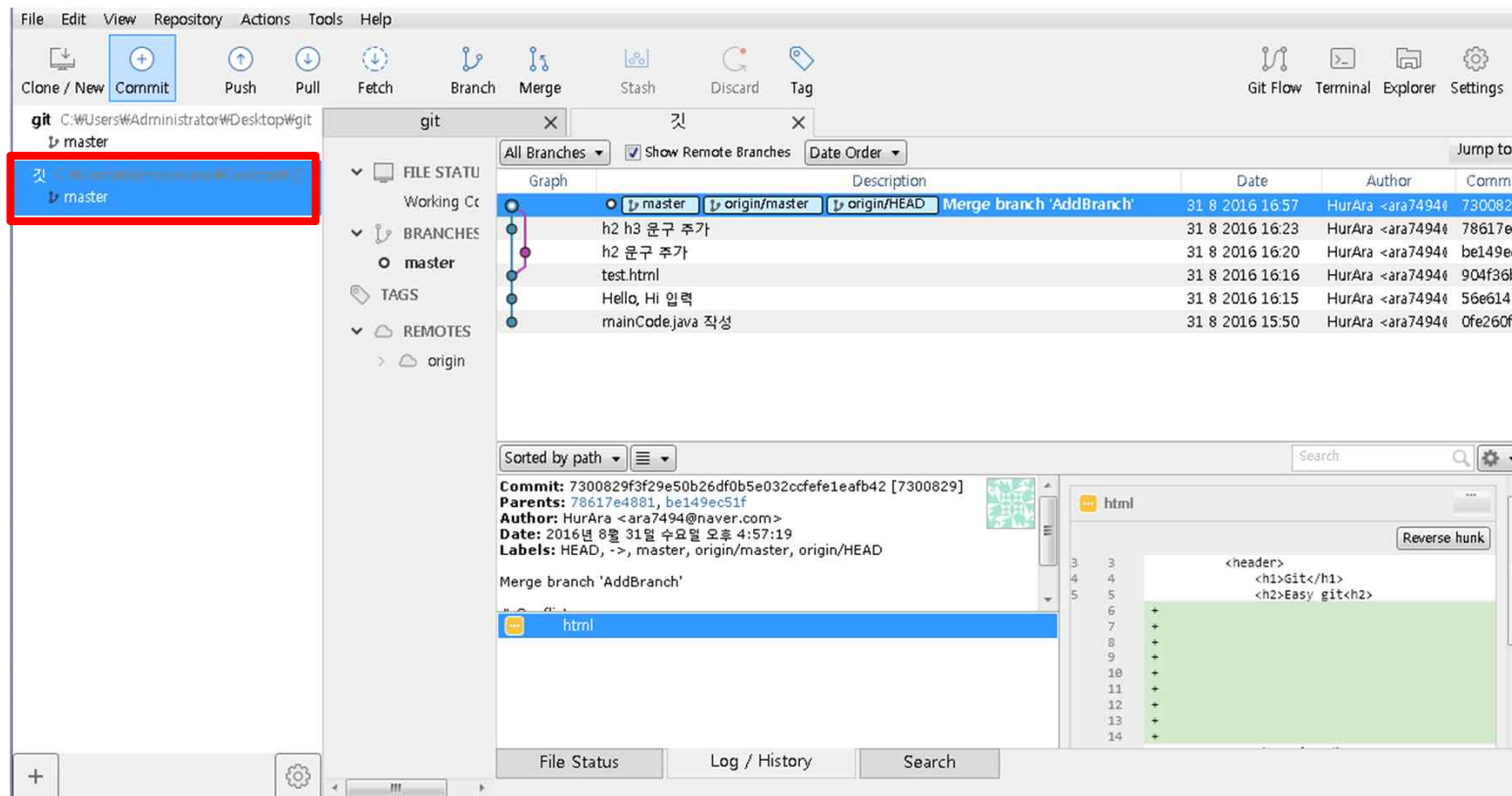
clone

- Clone Repository 탭에서 Source Path/URL 에 복사해둔 HTTPS값을 붙여넣고 Destination Path 에 받아온 프로젝트를 저장해둘 경로를 지정 → Bookmarks에 SourceTree에 표시할 저장소의 이름을 지정해주고 Clone 클릭



clone

- 아래와 같이 SourceTree에 원격 저장소에 존재하던 프로젝트가 생성되었으며 지정해둔 프로젝트 생성 경로에 해당하는 프로젝트 파일이 생성 됨



clone

- 설정한 경로에 GitHub의 원격 저장소에 존재하던 프로젝트가 저장 됨

