



제품소프트웨어 패키징 (Git과 GitHub를 활용한 소스코드 관리)

Git push를 활용한 소스코드 적용 : 소스코드 적용



학습내용

- Git push로 변경된 소스코드 적용
- Tortoise git에서 소스코드 변경사항 반영



학습목표

- Git push로 변경된 소스코드를 적용할 수 있다.
- Tortoise git에서 소스코드 변경사항을 반영할 수 있다.

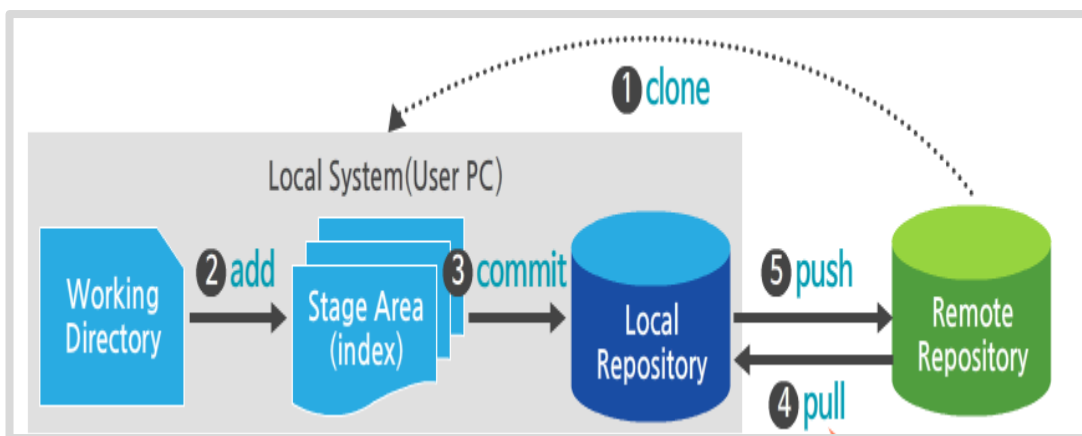


Git push 로 변경된 소스코드 적용

1. 소스코드 적용 절차

1) Git 작업 흐름

- 원격저장소 데이터를 자신의 컴퓨터로 가져오는 방법



- clone : 원격저장소의 데이터를 Local System으로 가져옴
- add : 새롭게 파일을 생성하여 Stage Area에 추가
- commit : 추가된 내용을 로컬저장소에 등록
- pull : 원격저장소의 변경된 데이터를 가져옴
- push : 로컬저장소의 등록된 내용을 원격저장소로 반영

2) Clone 원격저장소 가져오기

- Git을 시작하기 위해, 준비해야 할 것은?
 - 자신의 로컬저장소에 Git 프로젝트를 만들어야 함
 - GitHub에 원격저장소를 생성한 후에 그것을 로컬저장소에 가져오는 작업

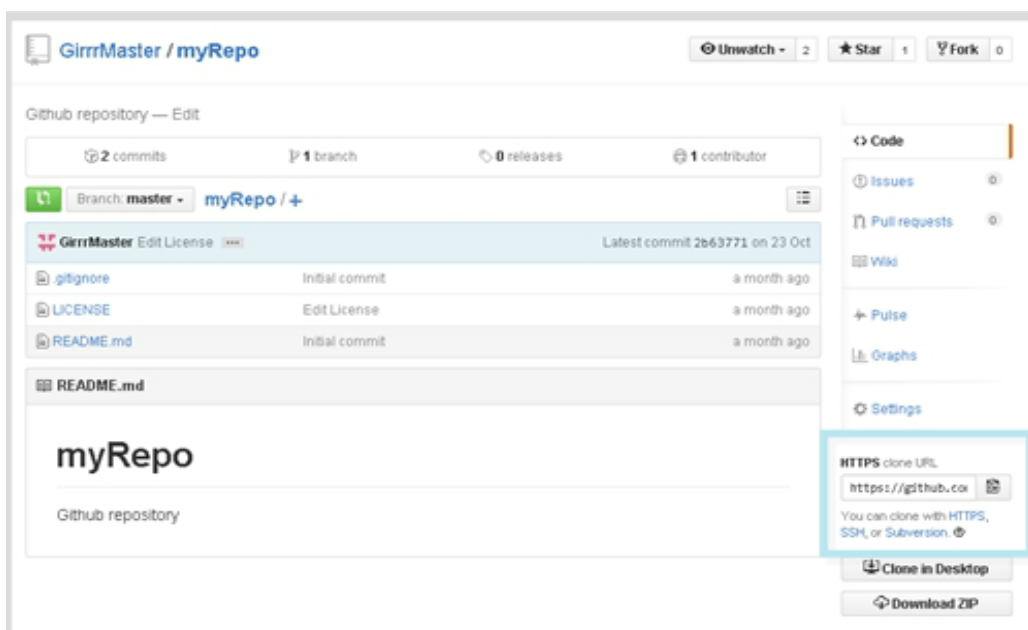


Git push 로 변경된 소스코드 적용

1. 소스코드 적용 절차

2) Clone 원격저장소 가져오기

- myRepo 원격저장소 또는 자신의 로컬저장소에 저장되지 않은 원격저장소를 새로 생성
- 원격저장소 URL 복사



- myRepo HTTPS clone URL : <https://github.com/GirrrMaster/myRepo.git>



Git push 로 변경된 소스코드 적용

1. 소스코드 적용 절차

2) Clone 원격저장소 가져오기

- git clone 명령
 - 내 로컬저장소에 Git을 작업하기 위한 로컬저장소를 생성하는 절차가 함께 포함
 - Git을 활용하기 위해 원격저장소의 데이터를 clone하는 역할을 한번만 수행하면 자동으로 로컬저장소가 생성

```
user@userpc MINGW64 /c/gitFolder
$ git clone https://github.com/GirrrMaster/myRepo.git
Cloning into 'myRepo' ...
remote: Counting objects: 8, done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 8
Unpacking objects: 100% (8/8), done.
Checking connectivity... done.
```

- ls -a 명령으로 파일 확인

```
user@userpc MINGW64 /c/gitFolder
$ cd myRepo

user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ ls -a
./ ../ .git/ .gitignore LICENSE README.md
```

- 원격저장소의 데이터를 제대로 가져왔는지 확인



Git push 로 변경된 소스코드 적용

1. 소스코드 적용 절차

3) pull 원격저장소와 로컬저장소 합치기

- \$ git pull [option] [remote명] [branch명] 명령어를 사용하여 특정 branch를 가져 올 수도 있음
 - 원격저장소 데이터를 로컬저장소와 비교하여 합치는 작업
- myRepo 저장소
 - 혼자 사용하는 저장소
 - clone을 하고 바로 pull을 해도 변화된 데이터가 없음

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git pull
Already up-to-date.
```



Git push 로 변경된 소스코드 적용

1. 소스코드 적용 절차

4) add 파일 추가

- touch 명령을 이용하여 빈 파일 생성

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ touch index.html
```

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ ls
index.html  LICENSE  README.md
```

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ vi index.html
```

- vi 에디터 이용하여 파일 내용 입력

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ touch index.html
```

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ ls
index.html  LICENSE  README.md
```

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ vi index.html
```




Git push 로 변경된 소스코드 적용

1. 소스코드 적용 절차

4) add 파일 추가

- vi 에디터 입력 후 저장 종료

```
<!DOCTYPE html>
<html>
<body>
<p>index.html</p>
</body>
</html>
~
~
~
/c/gitFolder/myRepo/index.html [+]
:wq!
```

- 파일 수정 시, 윈도우 상에서 메모장이나 윈도우용 에디터 사용
- Git Bash 콘솔에서 바로 수정 가능
- add 명령으로 stage area에 추가

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory.

user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git commit -m "add index.html"
[master 6768cbf] add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory.
1 file changed, 6 insertions(+)
create mode 100644 index.html
```

- git 이용 - unix system으로부터 업로드된 코드는 CR이 없음



Git push 로 변경된 소스코드 적용

1. 소스코드 적용 절차

5) commit 로컬저장소에 등록, log 확인

- Stage area에서 로컬저장소로 index.html파일등록

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory.

user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git commit -m "add index.html"
[master 6768cbf] add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory.
1 file changed, 6 insertions(+)
create mode 100644 index.html
```

- git log명령을 이용하여 commit 메시지 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git log
commit 6768cbf3ef2e3be3dd34ea48b96baf1780fc8f9e
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Fri Nov 27 12:00:40 2015 +0900

    add index.html

commit 2b63771733b87bb30193df35b8b6c2a1b9a1cb33
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Fri Oct 23 11:44:36 2015 +0900

    Edit License

    Modify LICENSE

commit 885427a85c96af852e85fc8cca31a20f0313814a
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Thu Oct 22 13:38:31 2015 +0900

    Initial commit
```



Git push 로 변경된 소스코드 적용

1. 소스코드 적용 절차

6) pull 원격저장소와 로컬저장소 합치기

- pull 원격저장소와 로컬저장소 합치기

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)  
$ git pull  
Already up-to-date.
```

- 업데이트된 데이터가 있다면 자신의 로컬저장소로 업데이트 된 내용 반영



Git push 로 변경된 소스코드 적용

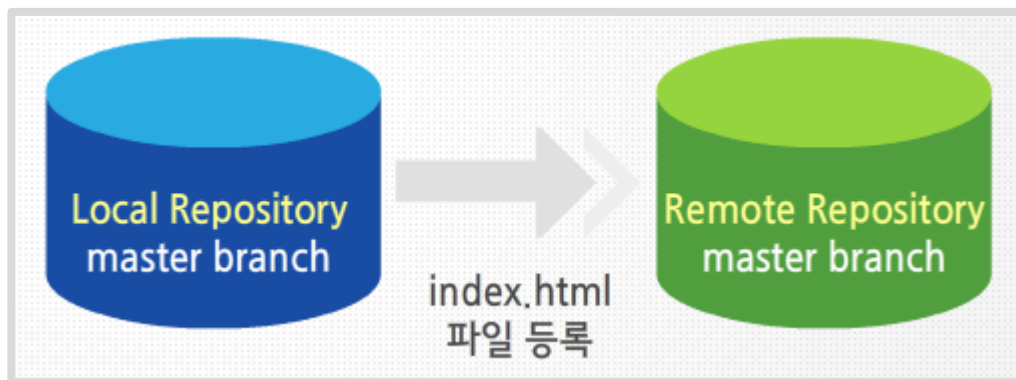
1. 소스코드 적용 절차

7) push 원격저장소로 데이터 등록

- index.html을 원격저장소로 등록
- \$ git push origin master 명령어로 실행

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git push origin master
Username for 'https://github.com': girrrMaster
Password for 'https://girrrMaster@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 327 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/GirrrMaster/myRepo.git
2b63771..6768cbf master -> master
```

- git에서는 origin과 master명 기본 저장
 - clone을 할 때 git에서는 origin과 master 명이 기본적으로 저장됨
- 로컬저장소 master branch가 원격저장소의 master branch로 반영



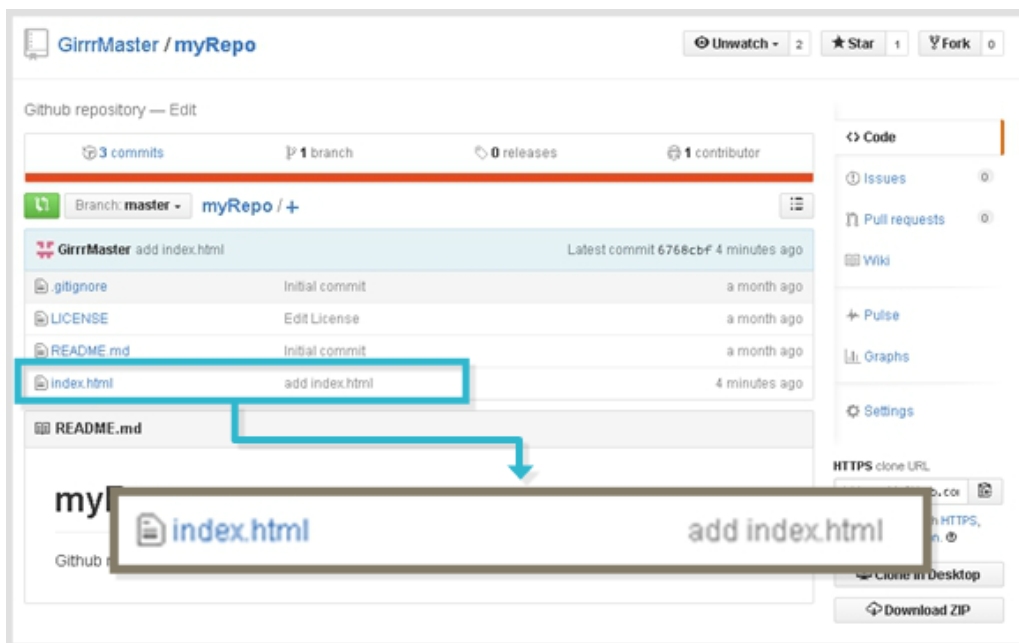


Git push 로 변경된 소스코드 적용

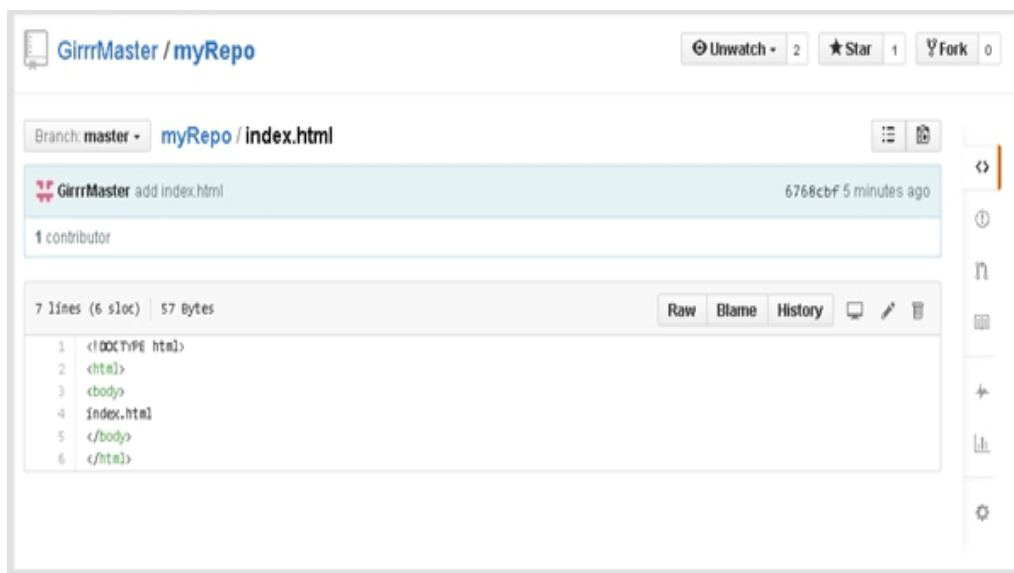
1. 소스코드 적용 절차

7) push 원격저장소로 데이터 등록

- GitHub의 myRepo 저장소에서 추가된 index.html 파일 확인



- index.html 파일 내용 확인



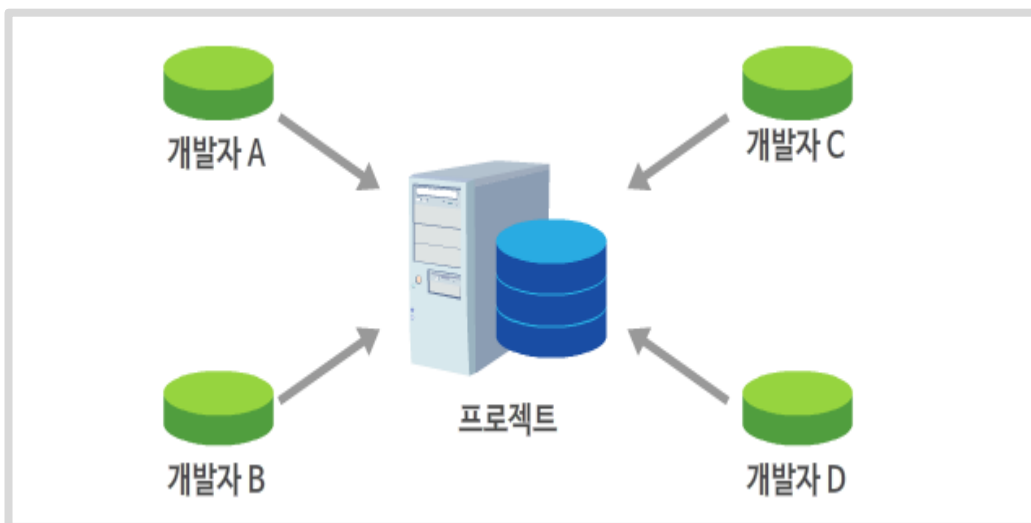


Git push 로 변경된 소스코드 적용

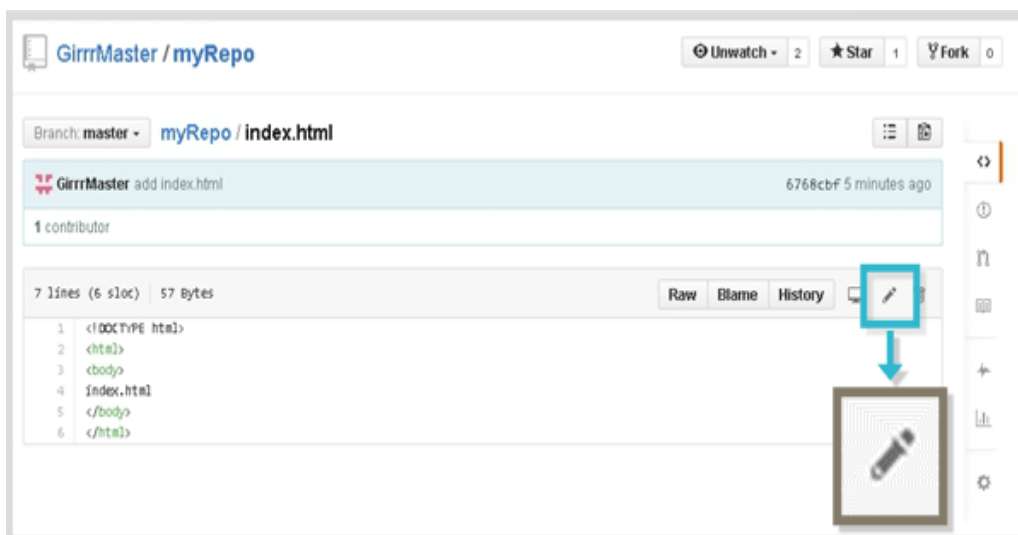
2. 원격저장소의 변경내용을 로컬저장소에 적용

1) GitHub 데이터 수정

- Git을 사용하는 주된 이유는?



- 여러 개발자가 하나의 프로젝트를 동시에 수행하게 되면서 여기저기에서 소스코드를 변경하는 것을 관리하기 때문
- 다른 부분에서의 수정사항을 내 로컬저장소에 반영할 수 있는 방법
 - 원격저장소, GitHub - 데이터 웹 상에서 직접 수정 가능



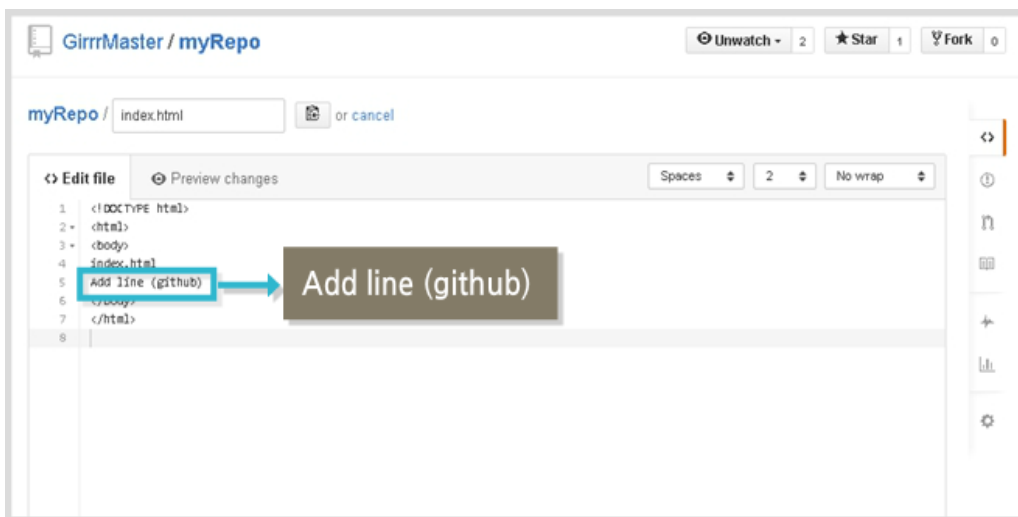


Git push 로 변경된 소스코드 적용

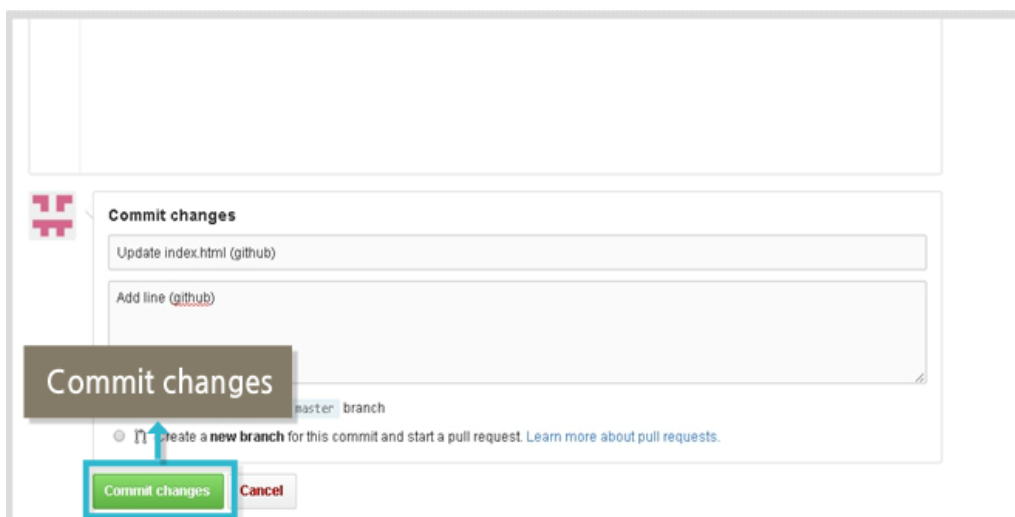
2. 원격저장소의 변경내용을 로컬저장소에 적용

1) GitHub 데이터 수정

- Add line(github) 추가



- commit 수행



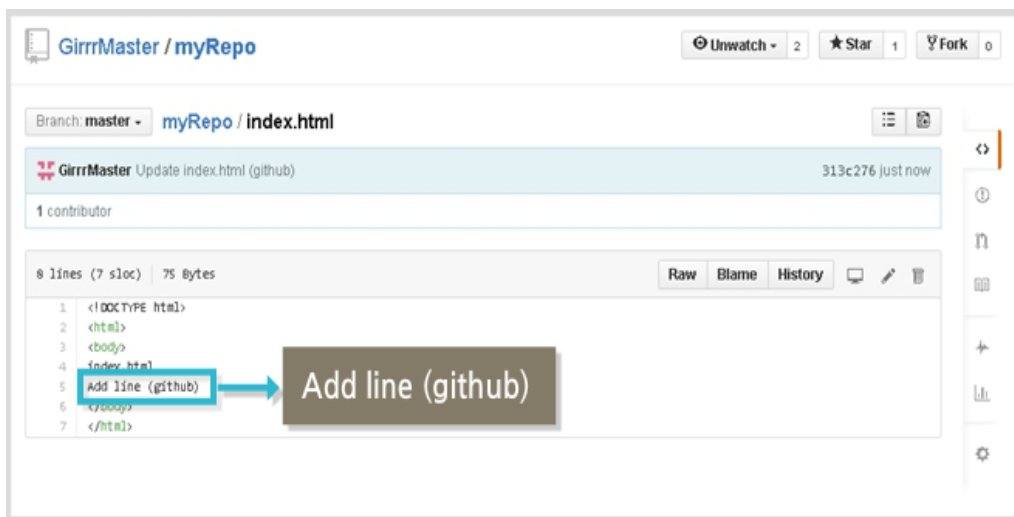


Git push 로 변경된 소스코드 적용

2. 원격저장소의 변경내용을 로컬저장소에 적용

1) GitHub 데이터 수정

- commit 완료 화면 확인



- 로컬저장소의 index.html 파일 데이터는 아직 업데이트 되지 않음

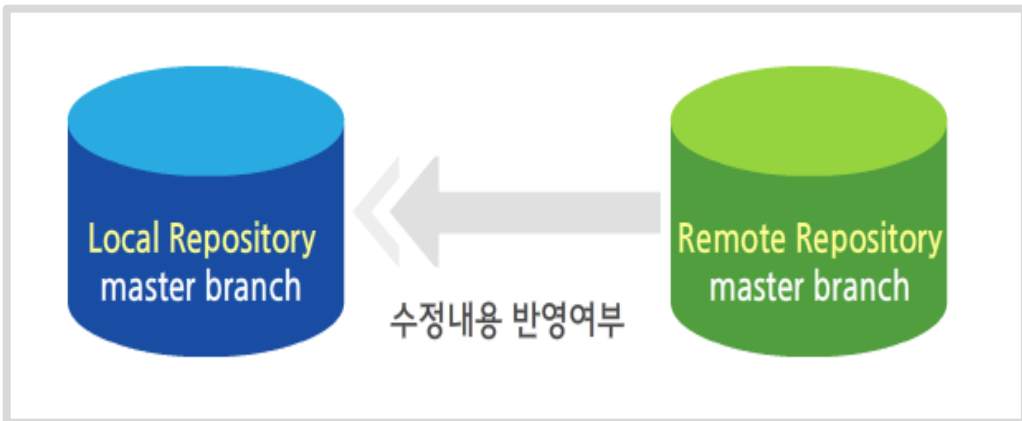


Git push 로 변경된 소스코드 적용

2. 원격저장소의 변경내용을 로컬저장소에 적용

2) 로컬저장소로 업데이트 내용 적용

- Add line(gitHub) 추가



- 수정 내용 알 수 없음
- 로컬저장소에서는 주기적으로 원격저장소의 변경 내용을 확인하여 업데이트
- 로컬저장소 index.html 파일 데이터 확인

```
<!DOCTYPE html>
<html>
<body>
index.html
</body>
</html>
```

- 데이터가 업데이트 되지 않음

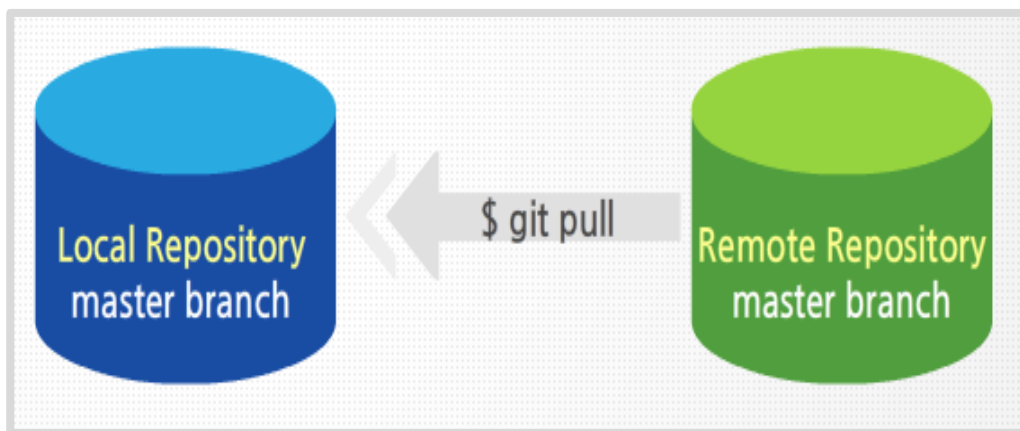


Git push 로 변경된 소스코드 적용

2. 원격저장소의 변경내용을 로컬저장소에 적용

2) 로컬저장소로 업데이트 내용 적용

- \$ git pull 명령어를 이용하여 원격저장소의 데이터를 로컬저장소로 반영



- index.html 파일의 내용 업데이트

- index.html의 업데이트 및 로컬저장소로 반영한다는 메시지 출력

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/GirrrMaster/myRepo
   6768cbf..313c276  master    -> origin/master
Updating 6768cbf..313c276
Fast-forward
 index.html | 1 +
 1 file changed, 1 insertion(+)

user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ vi index.html
```



Git push 로 변경된 소스코드 적용

2. 원격저장소의 변경내용을 로컬저장소에 적용

2) 로컬저장소로 업데이트 내용 적용

- 업데이트 된 index.html 파일 확인

```
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
</body>
</html>
~
~
```

- \$git log 명령어를 이용하여 GitHub 에서 commit 된 메시지 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git log
commit 515c2769b565173fc14d572e97f048c0ef6615b1
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Fri Nov 27 12:12:44 2015 +0900

    Update index.html (github)

    Add line (github)

commit 6768cbf3ef2e3be3dd34ea48b96baf1780fc8f9e
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Fri Nov 27 12:00:40 2015 +0900

    add index.html

commit 2b63771733b87bb30193df35b8b6c2a1b9a1cb33
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Fri Oct 23 11:44:36 2015 +0900

    Edit License

    Modify LICENSE

commit 885427a85c96af852e85fc8cca31a20f0313814a
Author: GirrrMaster <girrr.master@gmail.com>
Date:   Thu Oct 22 13:38:31 2015 +0900

    Initial commit
```

Initial commit



! 핵심정리



Git push로 변경된 소스코드 적용

1. 소스코드 적용 절차

- Clone 원격저장소 가져오기
- pull 원격저장소와 로컬저장소 합치기
- add 파일 추가
- commit 로컬저장소에 등록, log 확인
- pull 원격저장소와 로컬저장소 합치기
- push 원격저장소로 데이터 등록

2. 원격저장소의 변경 내용을 로컬저장소에 적용

- GitHub 데이터 수정
- 로컬저장소로 업데이트 내용 적



! 핵심정리



Tortoise git에서 소스코드 변경사항 반영

1. Diff 기능

- 파일 변경
- [Tortoisegit] - [Diff] 클릭
- 변경점 확인
- 이후 수행 동작 선택

2. Previous Diff 기능

- [Tortoisegit] - [Diff with previous version] 클릭
- Diff와 Previous Diff 비교
- Show log 기능