



제품소프트웨어 패키징 (Git과 GitHub를 활용한 소스코드 관리)

작업한 내용 되돌리기



학습내용

- Git 작업내용 되돌리기 - undo
- Tortoise Git에서 작업내용 되돌리기



학습목표

- 소스코드의 특정 변경점을 등록한 위치로 되돌릴 수 있다.
- 원격저장소에 반영된 소스코드를 이전 상태로 되돌릴 수 있다.



Git 작업내용 되돌리기 - undo

1. add 전에 되돌리기

1) 소스코드 변경 사항을 취소하거나 특정 지점의 소스코드로 되돌리기

- 실습진행
 - myRepo 로컬저장소의 index.html 파일 이용
 - ls 명령어를 통하여 저장소의 파일 확인
- 파일이 없을 경우 파일을 간단하게 생성하는 방법은?
 - touch index.html과 같이 빈 파일 생성
 - vi 에디터 등을 통해 간단하게 내용 입력
- ls 명령어로 저장소의 데이터 확인

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

- index.html 파일의 원본 데이터 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
</body>
</html>
```



Git 작업내용 되돌리기 - undo

1. add 전에 되돌리기

2) 생성되어 있는 파일을 간단히 수정하고 다시 되돌리기

- vi 에디터 이용, 파일 수정

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

- ls 명령어를 통하여 저장소의 파일 확인

- 'Add line2 (git bash)' 라인 입력 후 ESC → wq! 입력 저장

```
<!DOCTYPE html>  
<html>  
<body>  
index.html  
Add line (github)  
Add line2 (git bash)  
</body>  
</html>  
~  
~  
/c/gitFolder/myRepo/index.html [+]  
:wq!!
```

- 임의로 데이터 삽입 가능
- 입력모드 종료 - ESC키 클릭
- 저장 후 vi 에디터 종료

- 제대로 입력이 되었는지 내용 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)  
$ cat index.html  
<!DOCTYPE html>  
<html>  
<body>  
index.html  
Add line (github)  
Add line2 (git bash)  
</body>  
</html>
```

- 현재 파일의 내용 확인
- 수정한 내용이 포함된 파일인지 확인



Git 작업내용 되돌리기 - undo

1. add 전에 되돌리기

3) 단순히 파일 내용만 변경 하였을 때 이전으로 되돌리는 방법

- git checkout 명령어로 파일내용 복구
 - 수정한 내용을 git의 Stage area 추가 하기 전에 index.html 파일을 되돌리는 작업

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

- index.html 파일의 내용 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
</body>
</html>
```

- git 작업을 하지 않은 상태에서 소스코드가 수정된 것을 이전으로 되돌리기
 - “git checkout”



Git 작업내용 되돌리기 - undo

2. add 후에 되돌리기

1) Stage area에 수정된 파일 추가

- git add를 통해 수정한 파일을 반영하여 commit을 준비하는 상태

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
</body>
</html>
```

2) 새로운 변경내용 적용

- vi에디터 이용, 데이터 수정

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

- 에디터에서 키보드 버튼을 눌러서 입력모드로 변경

- 'Add line2 (git bash)'라인 입력 후 wq! 입력

```
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
Add line2 (git bash)
</body>
</html>
~
~ → 입력모드 종료 - ESC키 클릭
/c/gitFolder/myRepo/index.html [+] [dos]
:wq!
```

- 저장 후 vi 에디터 종료



Git 작업내용 되돌리기 - undo

2. add 후에 되돌리기

2) 새로운 변경내용 적용

- cat 명령어 이용, 수정 내용 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
Add line2 (git bash)
</body>
</html>
```

- git add명령어 이용, Staged

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

3) 작업한 내용 되돌리기

- git reset명령어 이용, Unstaged

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git reset HEAD index.html
Unstaged changes after reset:
M      index.html
```

- Stage area에 추가된 index.html 파일 제거



Git 작업내용 되돌리기 - undo

2. add 후에 되돌리기

3) 작업한 내용 되돌리기

- cat명령어 이용, 파일 내용 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
Add line2 (git bash)
</body>
</html>
```

- reset 명령어 사용 - 파일이 복원 되지 않음
- Stage area에서 제거 - 파일 내용은 복원 되지 않음
- git status명령 이용, git의 변경 상태 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will
                                   be committed)
  (use "git checkout -- <file>..." to discard
                                   changes in working directory)

modified:   index.html

no changes added to commit (use "git add"
                                   and/or "git commit -a")
```

Modified: index.html



Git 작업내용 되돌리기 - undo

2. add 후에 되돌리기

4) 파일 내용 복원하기

- git checkout명령어 이용, 원본으로 복원

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

- 파일의 변경된 사항을 이전 원본 상태로 돌려 놓음

- cat명령어로 복원된 index.html 파일 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)  
$ cat index.html  
<!DOCTYPE html>  
<html>  
<body>  
index.html  
Add line (github)  
</body>  
</html>
```

- git add가 수행되어 stage area에 적용된 파일을 되돌리기 위해서는 하나의 작업이 추가
 - stage area에 있는 파일을 unstaged
 - git add가 되기 전 상태
 - git add 전의 소스코드 되돌리는 작업



Git 작업내용 되돌리기 - undo

3. commit 후에 되돌리기

1) 파일 내용 복원하기 Stage area에 수정된 파일을 추가

- git checkout명령어 이용, 원본으로 복원

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
</body>
</html>
```

- t reset명령어 이용, Unstagedvi 에디터 이용, 파일 수정

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

- 에디터에서 키보드 버튼을 눌러서 입력모드로 변경
- 'Add line2(git bash)'라인 입력 후 wq! 입력

```
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
Add line2 (git bash)
</body>
</html>
~ → 입력모드 종료 - ESC키 클릭
/c/gitFolder/myRepo/index.html[+] [dos]
:wq!
```

- 저장 후 vi 에디터 종료



Git 작업내용 되돌리기 - undo

3. commit 후에 되돌리기

2) 로컬저장소에 수정된 파일을 등록한 후 되돌리기

- cat 명령어 이용, 수정 내용 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
Add line2 (git bash)
</body>
</html>
```

- git add 명령어 이용, Staged 추가

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

- git status 명령어로 git의 상태 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   index.html
```



Git 작업내용 되돌리기 - undo

3. commit 후에 되돌리기

3) 파일 내용 복원하기 Stage area에 수정된 파일을 추가

- commit 명령어로 로컬저장소에 등록

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git commit -m "undo test"
[master 3e44b54] undo test
1 file changed, 1 insertion(+)
```

- git log 명령어 실행

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git log
commit 3e44b544cd2282180a5f09931a111c3722b197e6
Author: GirrrMaster <girrr.master@gmail.com>
Date: Tue Dec 8 15:45:03 2015 +0900

    undo test

commit 313c2769b565173fc14d572e97f048c0ef6615b1
Author: GirrrMaster <girrr.master@gmail.com>
Date: Fri Nov 27 12:12:44 2015 +0900

    Update index.html (github)

    Add line (github)
```

- 변경점을 등록하고 원격저장소에 반영하기 직전의 상태



Git 작업내용 되돌리기 - undo

3. commit 후에 되돌리기

4) 로컬저장소에 수정된 파일을 등록한 후 되돌리기

- git reset 명령어 실행

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git reset --hard HEAD~1
HEAD is now at 313c276 update index.html (github)
```

- 최근 이전 commit 상태로 복구하는 명령어

- 313c276 상태로 업데이트 됨

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git log
commit 3e44b544cd2282180a5f09931a111c3722b197e6
Author: GirrrMaster <girrr.master@gmail.com>
Date: Tue Dec 8 15:45:03 2015 +0900

    undo test

commit 313c2769b565173fc14d572e97f048c0ef6615b1
Author: GirrrMaster <girrr.master@gmail.com>
Date: Fri Nov 27 12:12:44 2015 +0900

    Update index.html (github)

    Add line (github)
```

- 해당 시리얼 번호 : 모든 commit이 공유하게 가지고 있는 값. 소스코드를 복원하거나 다른 작업을 할 때 기준이 되는 고유 키

- index.html 파일 내용 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
</body>
</html>
```



Git 작업내용 되돌리기 - undo

3. commit 후에 되돌리기

4) 로컬저장소에 수정된 파일을 등록한 후 되돌리기

- git log를 이용, commit 기록 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git log
commit 313c2769b565173fc14d572e97f048c0ef6615b1
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
```



Git 작업내용 되돌리기 - undo

4. push 후에 되돌리기

1) 로컬저장소에 등록한 파일을 리모트 저장소에 등록

- 원본 index.html의 내용 확인

```
user@userpc: MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
</body>
</html>
```

- vi 에디터 이용, 파일 수정

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

- 에디터에서 키보드 |버튼을 눌러서 입력모드로 변경
- 'Add line2 (git bash)'라인 입력 후 wq! 입력

```
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
Add line2 (git bash)
</body>
</html>
~ → 입력모드 종료 - ESC키 클릭
/c/gitFolder/myRepo/index.html [+] [dos]
:wq!
```

- 저장 후 vi 에디터 종료



Git 작업내용 되돌리기 - undo

4. push 후에 되돌리기

1) 로컬저장소에 등록한 파일을 리모트 저장소에 등록

- cat 명령어 이용, 수정 내용 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
Add line2 (git bash)
</body>
</html>
```

- git status 명령어로 git의 상태 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git status
on branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will
                                   be committed)
  (use "git checkout -- <file>..." to discard
                                   changes in working directory)

modified:   index.html

no changes added to commit (use "git add"
                                   and/or "git commit -a")

Modified:   index.html
```




Git 작업내용 되돌리기 - undo

4. push 후에 되돌리기

1) 로컬저장소에 등록한 파일을 리모트 저장소에 등록

- git add 명령어 이용, Stage area 추가

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

- git commit 명령어 이용, 로컬 저장소에 등록

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git commit -m "undo test - push"
[master 67f5946] undo test - push
1 file changed, 1 insertion(+)
```

- 로컬저장소 등록내용을 리모트 저장소에 등록

```
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), 342 bytes |
0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/GirrrMaster/myRepo.git
313c276..67f5946 master -> master
```



Git 작업내용 되돌리기 - undo

4. push 후에 되돌리기

2) 다시 원본으로 복구하기

- git log 명령어로 commit history 조회

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git log
commit 67f5946f219ae0b9bbc5c53a90c49544b2b8f0e5
Author: GirrrMaster <girrr.master@gmail.com>
Date: Tue Dec 8 15:53:40 2015 +0900

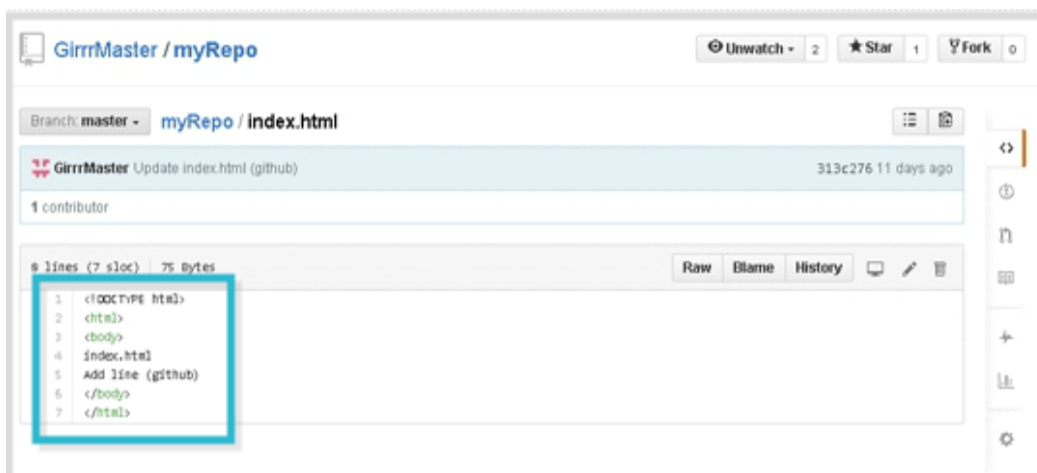
    undo test ~ push

commit 313c2769b565173fc14d572e97f048c0ef6615b1
Author: GirrrMaster <girrr.master@gmail.com>
Date: Fri Nov 27 12:12:44 2015 +0900

    Update index.html (github)

    Add line (github)
```

- 최근 등록된 commit 메시지 확인 가능
- git c파일 복구, 브랜치 이동, 파일 병합 등등의 작업
- github의 저장소에서 등록된 데이터 확인



- 원본 index.html

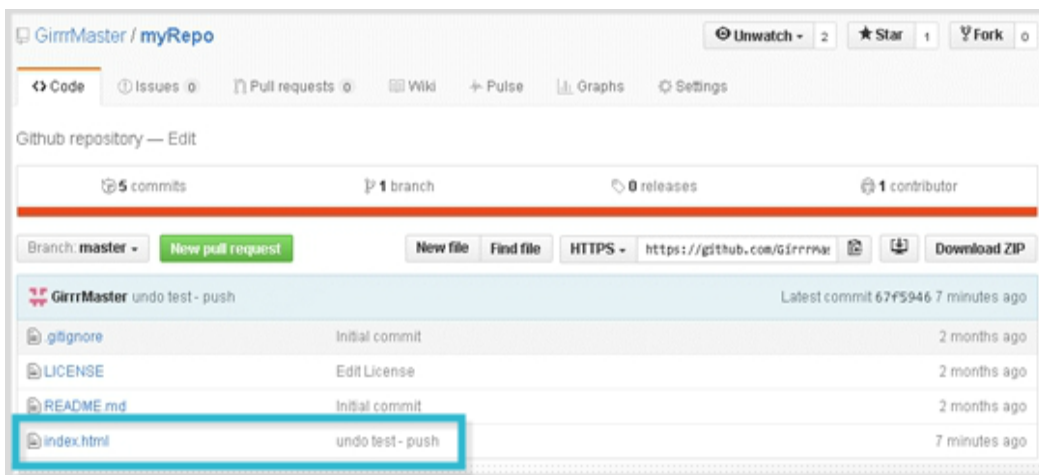


Git 작업내용 되돌리기 - undo

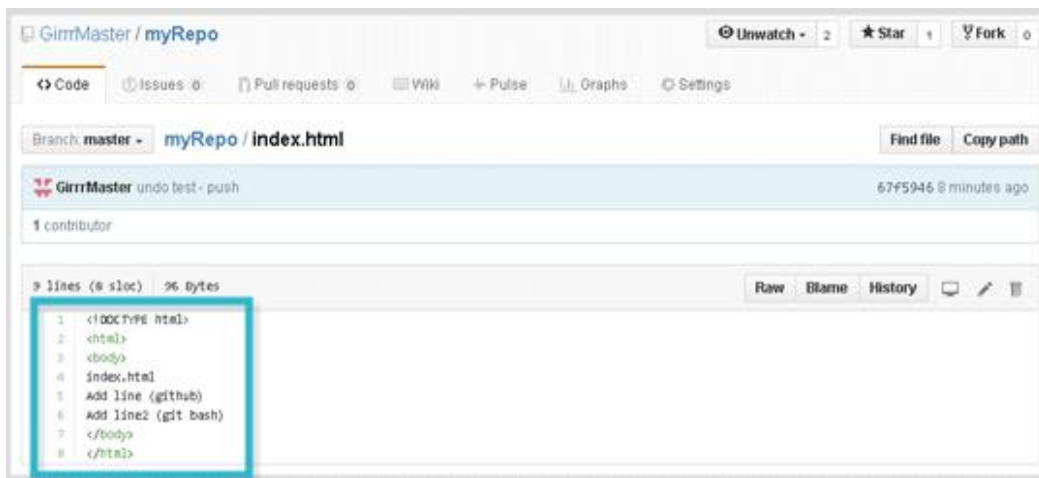
4. push 후에 되돌리기

2) 다시 원본으로 복구하기

- github의 저장소에서 등록된 데이터 확인



- 업데이트 된 index.html commit 메세지



- 수정된 index.html

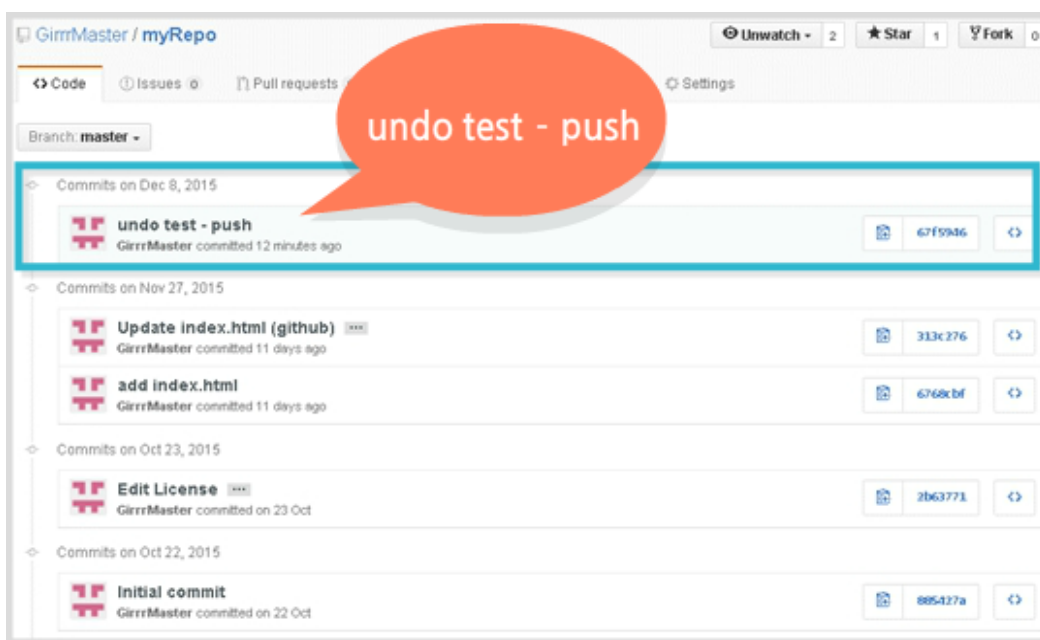


Git 작업내용 되돌리기 - undo

4. push 후에 되돌리기

2) 다시 원본으로 복구하기

- 원격저장소 commit 내용 확인
 - Commit 메뉴 : commit 내용 확인 git bash 창에서 git log 명령어와 같은 기능키



- 원격저장소까지 완벽하게 index.html 파일 등록완료



Git 작업내용 되돌리기 - undo

5. undo - push 후에 되돌리기

1) index.html 파일을 원본으로 수정하기

- 원본 파일 복원을 위해 git bash 창에서 git reset 명령어를 실행

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ git reset --hard HEAD~1
HEAD is now at 313c276 Update index.html (github)
```

Commits on Dec 8, 2015

- undo test - push
GirrrMaster committed 12 minutes ago

Commits on Nov 27, 2015

- Update index.html (github) [313c276](#)
- add index.html
GirrrMaster committed 11 days ago

Commits on Oct 23, 2015

- Edit License [2863771](#)

Commits on Oct 22, 2015

- Initial commit
GirrrMaster committed on 22 Oct

- 내 로컬에서 특정 변경점으로 되돌리기를 먼저 수행
- 이전 commit으로 복구
- 복구된 index.html 파일 내용 확인

```
user@userpc MINGW64 /c/gitFolder/myRepo (master)
$ cat index.html
<!DOCTYPE html>
<html>
<body>
index.html
Add line (github)
</body>
</html>
```



Git 작업내용 되돌리기 - undo

5. undo - push 후에 되돌리기

1) index.html 파일을 원본으로 수정하기

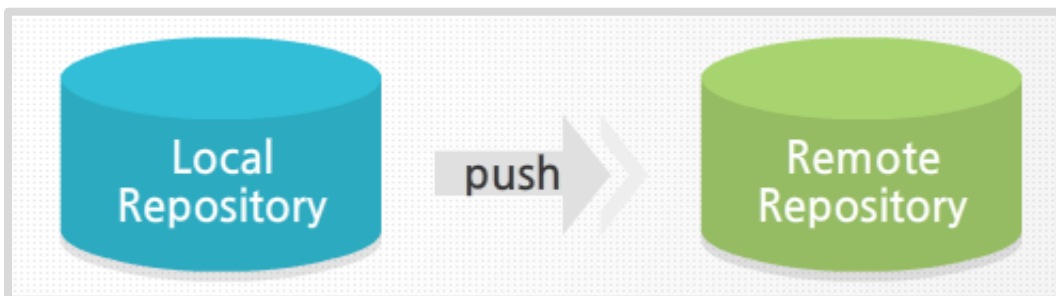
- git push 명령어를 사용하여 원격저장소에 복구된 파일 등록

```
user@userpc: ~/c/gitFolder/myRepo (master)
$ git push origin master
Username for 'https://github.com': girrrMaster
Password for 'https://girrrMaster@github.com':
To https://github.com/GirrrMaster/myRepo.git
! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/GirrrMaster/myRepo.git'
hint: updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

- 원격 저장소의 최신 파일보다 이전 작업으로 변경하였기 때문

```
user@userpc: ~/c/gitFolder/myRepo (master)
$ git push -f origin master
Username for 'https://github.com': girrrMaster
Password for 'https://girrrMaster@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/GirrrMaster/myRepo.git
+ 67f5946...313c276 master -> master (forced update)
```

- 강제로 push를 하는 명령
- 원격저장소에서 변경된 내용을 반영하지 않고 push하면 error가 발생함
- git pull 명령어를 수행한 후 push를 진행해야 하는 이유는?



- 업데이트된 다른 내용들을 지울 수 있기 때문
- push 명령어의 -f 옵션은 주의해서 사용



Git 작업내용 되돌리기 - undo

5. undo - push 후에 되돌리기

1) index.html 파일을 원본으로 수정하기

- 원격저장소의 index.html 내용 확인

GirrrMaster / myRepo

Unwatch 2 Star 1 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Branch: master - myRepo / index.html

Find file Copy path

GirrrMaster Update index.html (github) 313c276 11 days ago

1 contributor

8 lines (7 sloc) | 75 Bytes

Raw Blame History

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 index.html
5 Add line (github)
6 </body>
7 </html>
```

GirrrMaster / myRepo

Unwatch 2 Star 1 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Branch: master -

Commits on Nov 27, 2015

Update index.html (github) 313c276

GirrrMaster committed 11 days ago

add index.html 6768c9f

GirrrMaster committed 11 days ago

Commits on Oct 23, 2015

Edit License 2b63771

GirrrMaster committed on 23 Oct

Commits on Oct 22, 2015

Initial commit 085427a

GirrrMaster committed on 22 Oct



! 핵심정리



Git 작업내용 되돌리기 - undo

1. add전에 되돌리기

- `git checkout -- 파일명`

2. add후에 되돌리기

- `git reset`을 사용하여 파일을 stage area에서 제거
- `git reset HEAD 파일명`
- `git checkout` 명령어를 이용하여 되돌리기 수행

3. commit후에 되돌리기

- `git reset`을 활용하여 이전 commit 내용으로 되돌리기 수행
- `git reset --hard HEAD~1`

4. push 후에 되돌리기

- `git reset`을 사용하여 이전 commit으로 되돌리기 수행
- - `git push` 를 통해 서버에 반영
- `git push -f` 를 사용하여 반영
- `git push` 전 이전 서버의 최신 상태를 반영 후 재 반영을 위해 `git pull` 후에 `git push` 수행



! 핵심정리



Tortoise Git에서 작업내용 되돌리기

1. Git Reset을 사용하여 되돌리기

- 변경점에서 Reset 클릭
- Reset type - Hard 선택
- Reset 상태 확인

2. Git Revert를 사용하여 되돌리기

- 변경점에서 Revert 클릭
- Commit 실행
- Revert 상태 확인