

R 교육 세미나

ToBig's 8기 이현경

Git 사용법

contents

Unit 01 | Git 이란?

Unit 02 | Git 기초

Unit 03 | Github 이란?

Unit 04 | Github 기초

Unit 01 | Git 이란?

버전관리

- 내가 파일을 수정한 기록들
- 여러 branch로 나누어 작업하기

Unit 01 | Git 이란?

버전관리의 필요성

토요일



과제 끝!!!



월요일 11시 50분



코드 좀만
고쳐서 내볼까?



```
> 1:length(q1)+(q1>60)
Error: object 'q1' not found
> q1[q1>60]
Error: object 'q1' not found
> fqda <- qda(c1a~.,data=train)
Error: could not find function "qda"
> fts2 <- predict(fqda,newdata=test)
Error in predict(fqda, newdata = test) : object 'fqda' not found
> tab2 <- table(fts2$class,fts2$class)
Error in table(fts2$class, fts2$class) : object 'fts2' not found
> mcqda <- 1-sum(diag(tab2))/sum(tab2)
Error in diag(tab2) : object 'tab2' not found
```

월요일 11시 59분



Unit 01 | Git 이란?

내가 버전관리를 해야한다면?



과제.R



과제_최종.R



과제_진짜_최종.R

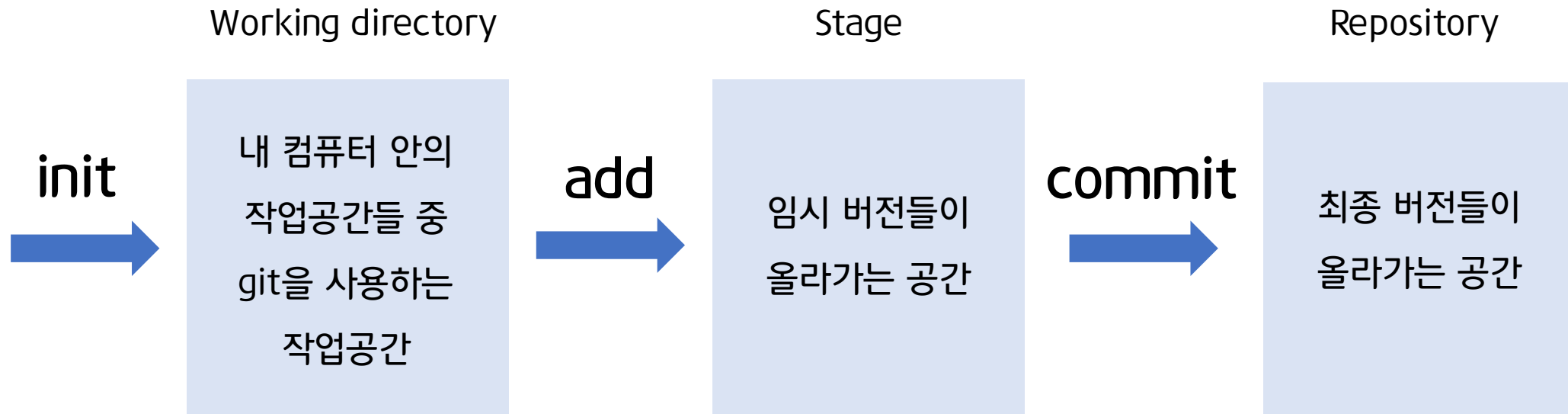


과제_진짜_진짜_최종.R

귀찮아..



Unit 01 | Git 이란?



Unit 02 | Git 기초

```
USER@LG-PC ~/Desktop/tobigs/git (master)
$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

The most commonly used git commands are:

add          Add file contents to the index
bisect       Find by binary search the change that introduced a bug
branch       List, create, or delete branches
checkout     Checkout a branch or paths to the working tree
clone        Clone a repository into a new directory
commit       Record changes to the repository
diff         Show changes between commits, commit and working tree, etc
fetch        Download objects and refs from another repository
grep         Print lines matching a pattern
init         Create an empty Git repository or reinitialize an existing one
log          Show commit logs
merge        Join two or more development histories together
mv           Move or rename a file, a directory, or a symlink
pull         Fetch from and integrate with another repository or a local branch

push         Update remote refs along with associated objects
rebase       Forward-port local commits to the updated upstream head
reset        Reset current HEAD to the specified state
rm           Remove files from the working tree and from the index
show         Show various types of objects
status       Show the working tree status
tag          Create, list, delete or verify a tag object signed with GPG

'git help -a' and 'git help -g' lists available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
```

Unit 02 | Git 기초

참고) 간단한 bash 명령어 정리

mkdir : 디렉토리를 만듦

ls : 현재 디렉토리의 list

ls : 파일명만 보여줌

ls -a : 디렉토리 내 모든 것 (ex. 폴더, ..)

ls -al : 디렉토리 내 모든 것 + 접근권한 + 생성날짜 등

pwd : 현재 위치

cd : 디렉토리 변경

cd : 최상위 폴더로 이동

cd dir_name : dir 로 이동

vim file_name : 파일 생성, 편집

file_name 이 현재 dir에 없는 경우 생성

file_name 이 현재 dir에 있는 경우 편집

(편집 방법들

i - 삽입, :w - 저장,

:q - 나가기, :wq - 저장 후 나가기)

cat file_name : 파일의 내용 출력

Unit 02 | Git 기초

1. git init

Init 이란 ?

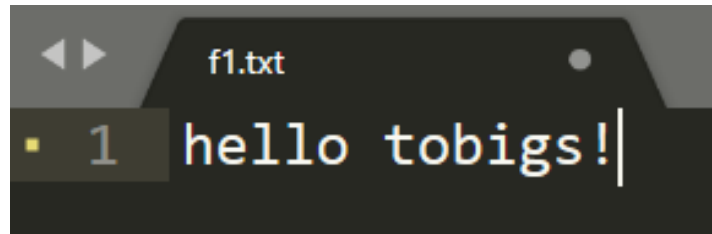
➤ 내 컴퓨터의 어떤 장소에 git 을 사용하겠다고 말해주는 것

```
USER@LG-PC ~/Desktop/tobigs  
$
```

```
$ git init  
Initialized empty Git repository in c:/Users/USER/Desktop/tobigs/.git/  
USER@LG-PC ~/Desktop/tobigs (master)  
$
```

Unit 02 | Git 기초

```
USER@LG-PC ~/Desktop/tobigs/git (master)
$ cat f1.txt
hello tobigs!
USER@LG-PC ~/Desktop/tobigs/git (master)
$
```



```
USER@LG-PC ~/Desktop/tobigs/git (master)
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

  f1.txt

nothing added to commit but untracked files present (use "git add" to track)
```

git 이라는 폴더에 f1.txt가 저장되어 있지만, git은 현재 무시하고 있는 상태

Unit 02 | Git 기초

2. git add

add 란 ?

➤ Init 한 working directory 내 파일의 생성/수정 상황을 임시 저장소에 저장

```
USER@LG-PC ~/Desktop/tobigs/git (master)
$ git add f1.txt

USER@LG-PC ~/Desktop/tobigs/git (master)
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   f1.txt
```

git add file_name : file 의 생성/수정 상황을 저장

git add . : 이전 commit 이후의 모든 file의 생성/수정 상황을 저장

git add -p : 파일의 변경된 내용을 hunk단위로 일부만 선택해 add 가능

git add -i : p (patch) 이외의 옵션들을 대화형으로 사용 가능

Unit 02 | Git 기초

3. git commit

commit 이란 ?

➤ Add로 임시저장소에 저장한 생성/수정 내용을 하나의 최종 버전으로 저장

1) 누가 commit 한건지 알려주기 – git config

```
USER@LG-PC ~/Desktop/tobigs/git (master)
$ git config --global user.name hyunkyoung12

USER@LG-PC ~/Desktop/tobigs/git (master)
$ git config --global user.email khc0275@naver.com
```

Unit 02 | Git 기초

2) Commit 하기 - git commit

```
USER@LG-PC ~/Desktop/tobigs/git (master)
$ git commit
[master (root-commit) b460094] version1
1 file changed, 1 insertion(+)
create mode 100644 f1.txt
```

git commit -m "commit message" : commit 할 메시지 작성

git commit file_name : add 한 파일들 중 일부만 commit

git commit -am "commit message"

: add했던 파일을 수정한 경우 add와 commit을 동시에

3) Commit 한 기록들 보기 - git log

```
USER@LG-PC ~/Desktop/tobigs/git (master)
$ git log
commit b4600948ee8e72ce3546f1e96aa8feeddb2f9c7f
Author: hyunkyung12 <khc0275@naver.com>
Date: Tue Jan 16 22:42:31 2018 +0900

    version1
```

git log commit1_id .. commit2_id

: commit1과 commit2 사이의 log만 보여줌

git log -p : commit 한 버전들 간의 차이를 보여줌

Unit 02 | Git 기초

+) 버전간의 차이 확인하기

`git log -p` - commit 한 버전들 간의 차이

`git diff version1_id .. Version2_id` - 두 commit 버전간의 차이

`git diff` - add 전과 add 후의 차이

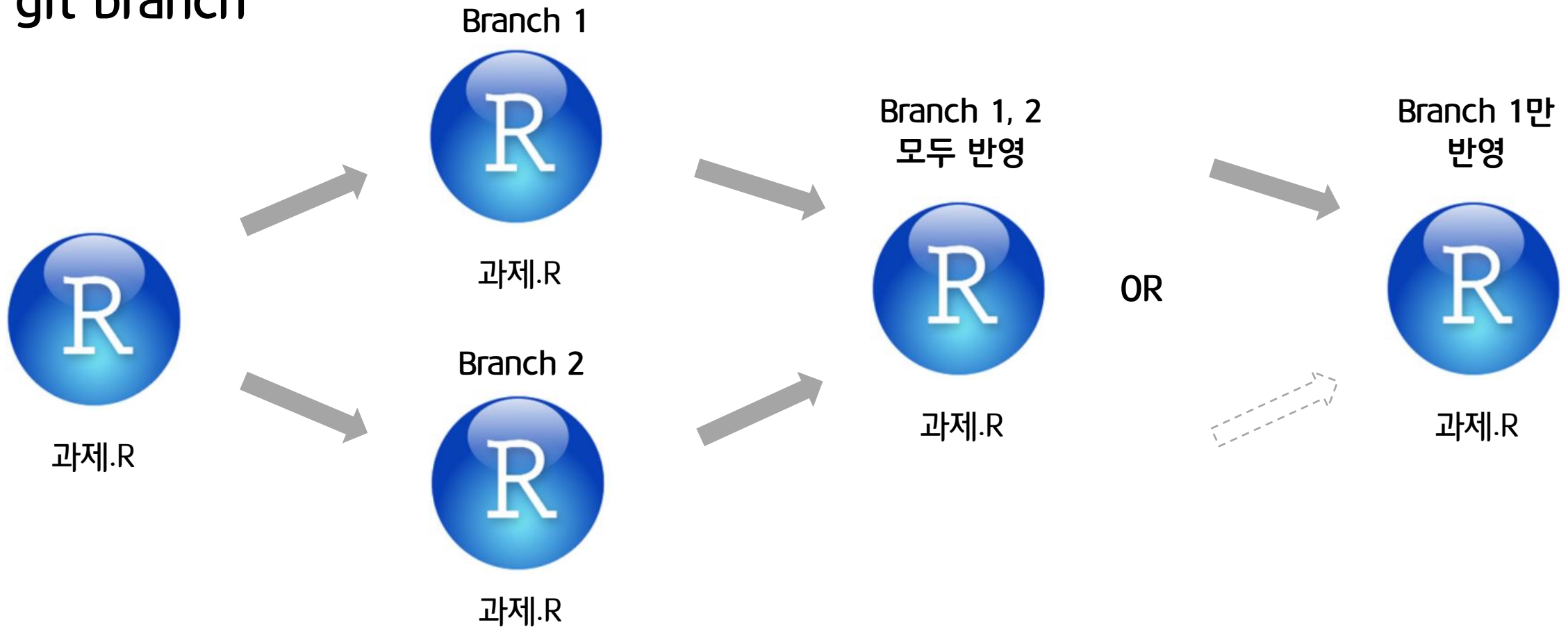
+) 이전 버전으로 돌아가기

`git reset --hard version_id` - 이전 commit 버전으로 돌아감

`git revert version_id` - 이전 commit 버전으로 돌아가는 상황 자체를 하나의 commit으로 저장

Unit 02 | Git 기초

4. git branch



Unit 02 | Git 기초

1) 내 디렉토리의 branch 상황 확인 - git branch

```
USER@LG-PC ~/Desktop/tobigs/branch (master)
$ git branch
* master
```

2) 새 branch 만들기 - git branch branch_name

```
USER@LG-PC ~/Desktop/tobigs/branch (master)
$ git branch exp
```

3) 다른 branch로 변경 - git checkout branch_name

```
USER@LG-PC ~/Desktop/tobigs/branch (master)
$ git checkout exp
Switched to branch 'exp'
```

```
USER@LG-PC ~/Desktop/tobigs/branch (exp)
```


Unit 02 | Git 기초

4) branch 합치기 - git merge

master에 exp를 합치고 싶다?

1. master로 checkout 한 후
2. git merge exp

Conflict (충돌) 이 일어난 경우 : 파일의 같은 부분을 여러 branch가 수정했을때

1. git status로 해당 파일을 확인 후 수동으로 고쳐줌
2. git add , git commit 해줌

```
first branch^M
b^M
<<<<<<< HEAD
c^M
=====
^M
second branch^M
>>>>>> exp
```

현재 branch의 내용

exp branch의 내용

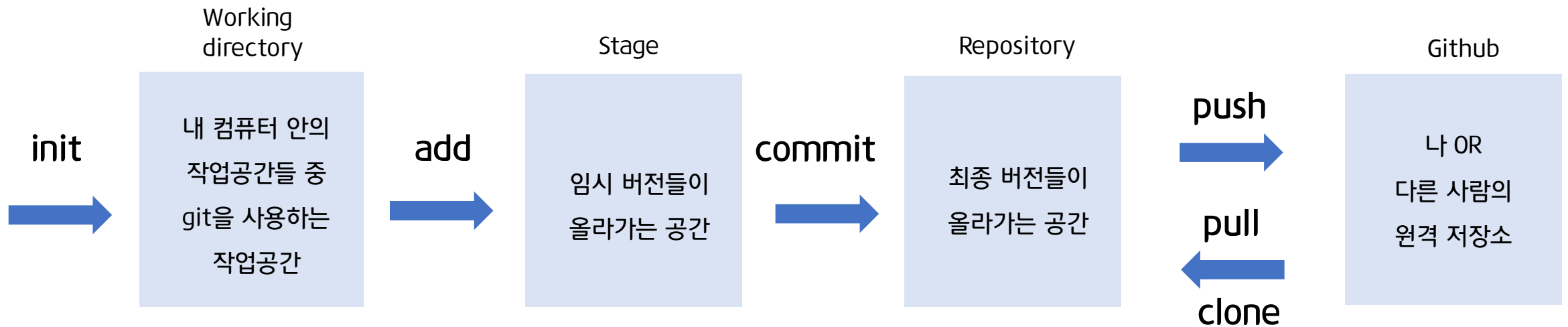
Unit 03 | Github 란



GitHub

- 원격 저장소 관리
- 오픈소스들의 모임

Unit 03 | Github 란?



Unit 04 | Github 기초

1) Github 홈페이지에서 repository 만들기

The screenshot shows the GitHub repository creation interface for a user named 'hyunkyung12' with the repository name 'w1_git'. At the top, there are buttons for 'Watch', 'Star', and 'Fork', each with a count of 0. Below these are tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. The main content area is divided into three sections: 'Quick setup', '...or create a new repository on the command line', and '...or push an existing repository from the command line'. The 'Quick setup' section offers three options: 'Set up in Desktop', 'HTTPS', and 'SSH', with the 'HTTPS' option selected. The 'SSH' option shows the URL 'https://github.com/hyunkyung12/w1_git.git'. Below the 'Quick setup' section, there is a note: 'We recommend every repository include a README, LICENSE, and .gitignore.' The '...or create a new repository on the command line' section shows a series of commands to create a new repository and push it to GitHub. The '...or push an existing repository from the command line' section shows commands to add a remote repository and push it to GitHub.

hyunkyung12 / w1_git

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/hyunkyung12/w1_git.git

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# w1_git" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/hyunkyung12/w1_git.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/hyunkyung12/w1_git.git
git push -u origin master
```

Unit 04 | Github 기초

2) 내 로컬 저장소를 github에 연결하기 – git remote add

```
$ git remote add origin https://github.com/hyunkyung12/tobigs_w1_github
```

3) 내 로컬 저장소의 master branch 와 remote의 master를 연결 후 push – git push

```
$ git push -u origin master
Username for 'https://github.com': hyunkyung12
Password for 'https://hyunkyung12@github.com':
Counting objects: 9, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (9/9), 658 bytes | 0 bytes/s, done.
Total 9 (delta 0), reused 0 (delta 0)
```

4) git pull (일반적으로 master branch의 것을 가져옴)

git pull origin master 처럼 옵션을 줄 수도 있음

Unit 04 | Github 기초

5) git clone (이미 존재하는 원격 저장소의 내용을 복사해옴)

This repository Search Pull requests Issues Marketplace Explore

KleinYuan / easy-yolo Watch 4 Star 25 Fork 9

Code Issues 3 Pull requests 0 Projects 0 Wiki Insights

Yolo (Real time object detection) model training tutorial with deep learning neural networks

deep-learning yolo machine-learning real-time object-detection python deep-learning-tutorial darknet multiple-gpu training-yolo

labeling camera tutorial

12 commits 1 branch 0 releases 2 contributors

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

KleinYuan Fix one more bug Latest commit f06dc4c on 7 Nov 2017

cfg	Fix one more bug	2 months ago
scripts	Fix some bugs	2 months ago
src	Fix some bugs	2 months ago
.gitignore	Initial commit	9 months ago
LICENSE	Adding everything	9 months ago

Create new file Upload files Find file Clone or download

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/KleinYuan/easy-yolo.git Copy to clipboard

Open in Desktop Download ZIP

9 months ago

Unit 05 | 실습

로컬 저장소 실습

Init > add > commit

원격 저장소 실습

1. Clone 해서 사용하기
2. 내 로컬 저장소 올리기

Unit 06 | 과제

https://github.com/hyunkyung12/w1_git_hw

📖 README.md

투빅스 9기 1주차 Git 기초 과제

1주차 과제 git으로 올려보기

★ 조건

1. commit 5개 이상
2. 자신의 이름으로 branch 생성
3. 반드시 이 repository의 branch로 올리기

😓 master 말고 자신의 branch!!!

★ 기한

1/22 저녁 6시 까지

Unit 00 | 출처

동영상 강의 : <https://opentutorials.org/course/2708>

깃 명령어 정리

: <https://blog.outsider.ne.kr/572>

: <http://noritersand.tistory.com/86>

Q & A

들어주셔서 감사합니다.