

시스템 프로그래밍

Git와 Github

2015.9.24

허혜지

hyzzish@naver.com

1. 개요

- 실습명
 - Git와 Github 사용
- 목표
 - Git를 사용하여 소스를 관리할 수 있다.
- 실습내용
 - Git를 설치
 - Github와 연동하여 소스를 올리고 확인

2. Git, Github 란?

- Git
 - 프로젝트의 어떤 부분도 겹쳐쓰지 않게 프로젝트의 변경을 관리하는 소프트웨어
 - 동시에 프로젝트를 수정할 경우 데이터가 누락되는 것을 방지하기 위해서 사용
 - 버전으로 관리하여 이전에 수정한 부분으로 되돌아 갈수도 있음
- Github
 - Git에서 관리하는 프로젝트들이 거주하는 장소

3. Github 가입 및 repository 생성

- Github 가입

The image shows two side-by-side screenshots of the GitHub sign-up interface.

Left Screenshot (Sign-up for GitHub):

- Fields: "Pick a username", "Your email", "Create a password".
- Text below password field: "Use at least one lowercase letter, one numeral, and seven characters."
- Large green button: "Sign up for GitHub".
- Text below button: "By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally."

Right Screenshot (Create your personal account):

- Section title: "Create your personal account".
- Text in red box: "There were problems creating your account."
- Field: "Username" with error message "Login can't be blank".
- Text below field: "You will occasionally receive account related emails. We promise not to share your email with anyone."
- Field: "Password" with error message "Password can't be blank and is too short (minimum is 7 characters)".
- Text below field: "By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).
- Large green button: "Create an account".

- 아이디를 만들고 가입완료

3. Github 가입 및 repository 생성

- Repository 생성
- Repository를 만들기 위해서는 이메일 인증이 필요
- Repository 이름을 만들고 공개할 것인지 개인적으로 할것인지 선택
- 밑의 initial~~은 체크하지 말것

Owner Repository name

ruidant /

Great repository names are short and memorable. Need inspiration? How about [zealous-octo-giggle](#).

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

3. Github 가입 및 repository 생성

- 생성 후 생성된 repository에 들어가면 이러한 화면을 확인할 수 있음.

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

Set up in Desktop or [HTTPS](https://github.com/ruidant/public.git) [SSH](ssh://github.com/ruidant/public.git) <https://github.com/ruidant/public.git>

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

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

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

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

```
git remote add origin https://github.com/ruidant/public.git  
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

4. Git 설치

- 설치방법 1

소스코드로 설치

- 라이브러리(curl, zlib, openssl, expat, libiconv) 설치

```
$ yum install curl-devel expat-devel gettext-devel \
openssl-devel zlib-devel
```

```
$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \
libbz-dev libssl-dev
```

- 소스코드 다운: <http://git-scm.com/download>
- 컴파일

```
$ tar -zxf git-1.7.2.2.tar.gz
$ cd git-1.7.2.2
$ make prefix=/usr/local all
$ sudo make prefix=/usr/local install
```

4. Git 설치

- 설치방법 2

패키지 관리도구로 설치

```
$ yum install git-core
```

```
$ apt-get install git-core
```

- 여러분의 서버는 이미 설치되어 있으므로 건너뛰어도 됨.
- 설치를 시도해도 권한문제로 설치가 불가

5. Git 설정

- []안에는 github에 가입한 이메일과 user이름
- 에디터는 설정안해도 됨

기본 설정

```
$ git config --global user.name [user-name]      # 유저 이름 설정
```

```
$ git config --global user.email [user-email]      # 유저 이메일 설정
```

```
$ git config --global core.editor [editor-name]      # 사용할 편집기 종류 선택
```

5. Git 설정

- Github 연동할 폴더를 하나 생성후 이동

```
c201550326@ubuntu:~$ mkdir MyProject  
c201550326@ubuntu:~$ cd MyProject
```

- 현재 디렉토리를 로컬 깃 저장소로 지정

```
c201550326@ubuntu:~/MyProject$ git init  
Initialized empty Git repository in /home/sys01/c201550326/MyProject/.git/
```

- 명령어 touch를 사용하여 Readme.txt를 생성

```
c201550326@ubuntu:~/MyProject$ touch Readme.txt
```

- Touch 명령어는 어떠한 파일을 만드는 명령어

5. Git 설정

- 현재 업데이트된 파일 보기

```
c201550326@ubuntu:~/MyProject$ git status
On branch master

Initial commit

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

    README.txt

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

- 현재 업데이트된 내용을 Git가 주목하게 하기

```
c201550326@ubuntu:~/MyProject$ git add README.txt
```

- 업데이트 된 내용을 커밋

```
c201550326@ubuntu:~/MyProject$ git commit -m "Add README.txt"
[master (root-commit) 0ddce21] Add README.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.txt
```

6. Github와 로컬저장소 연결

- 로컬저장소의 변경사항을 보낼 원격 저장소 지정

```
c201550326@ubuntu:~/MyProject$ git remote add origin https://github.com/UserName  
/RepositoryName.git
```

- UserName : Github에 가입한 닉네임
 - RepositoryName : Github에 만든 Repository 이름
 - 일치하지 않으면 동작하지 않음
- 원격 저장소와 로컬 저장소의 변경사항 확인

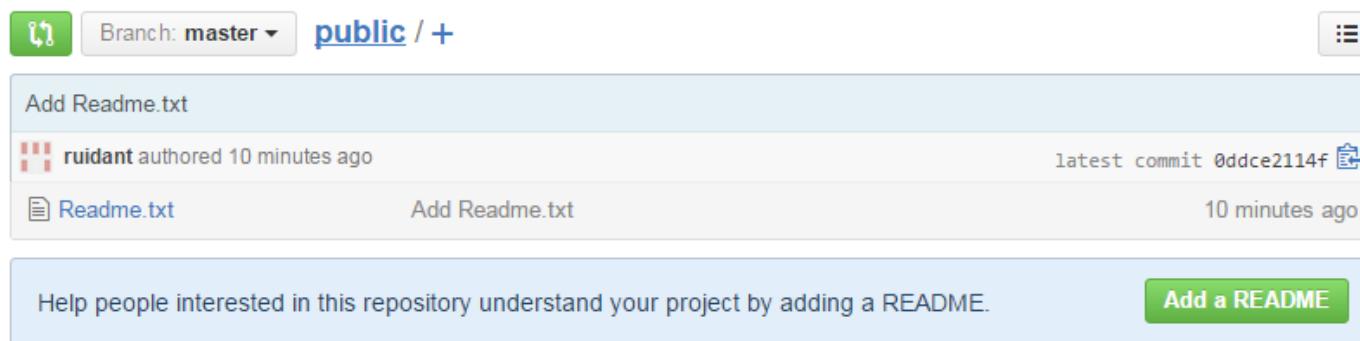
```
c201550326@ubuntu:~/MyProject$ git remote -v  
origin https://github.com/ruidant/public.git (fetch)  
origin https://github.com/ruidant/public.git (push)
```

6. Github와 로컬저장소 연결

- 원격 저장소로 변경사항 반영

```
c201550326@ubuntu:~/MyProject$ git push -u origin master
Username for 'https://github.com': ruidant
Password for 'https://ruidant@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 217 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ruidant/public.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

- Username : Github에 가입한 닉네임
- Password : Github에 가입한 비밀번호
- Github에 가면 다음과 같은 화면 확인 가능



7. Github에서 소스 가져오기

- 전체를 가져오기 위한 폴더생성후 이동

```
c201550326@ubuntu:~$ mkdir clone  
c201550326@ubuntu:~$ cd clone  
c201550326@ubuntu:~/clone$ ls
```

- 전체를 가져오기

```
c201550326@ubuntu:~/clone$ git clone https://github.com/Username/Repositoryname  
Cloning into 'public'...  
remote: Counting objects: 3, done.  
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.  
Checking connectivity... done.
```

- 가져온 내용 확인

```
c201550326@ubuntu:~/clone$ ls  
public  
c201550326@ubuntu:~/clone$ cd public  
c201550326@ubuntu:~/clone/public$ ls  
Readme.txt  
c201550326@ubuntu:~/clone/public$ cat Readme.txt
```

8. 파일 수정 및 수정된 파일 업데이트

- MyProject 폴더안의 Readme.txt를 마음대로 수정
- 수정후 커밋

```
c201550326@ubuntu:~/MyProject$ git add Readme.txt
c201550326@ubuntu:~/MyProject$ git commit -m "mod Readme.txt"
[master 3214bdd] mod Readme.txt
 1 file changed, 1 insertion(+)
c201550326@ubuntu:~/MyProject$ git push -u origin master
Username for 'https://github.com': ruidant
Password for 'https://ruidant@github.com':
Counting objects: 5, done.
Writing objects: 100% (3/3), 255 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ruidant/public.git
 0ddce21..3214bdd master -> master
Branch master set up to track remote branch master from origin.
```

8. 파일 수정 및 수정된 파일 업데이트

- 이전에 소스를 가져온 폴더로 이동
- 수정된 파일내용 반영 및 확인

```
c201550326@ubuntu:~/clone/public$ git pull https://github.com/Username/RepositoryName
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/ruidant/public
 * branch           HEAD      -> FETCH_HEAD
Updating 0ddce21..3214bdd
Fast-forward
 README.txt | 1 +
 1 file changed, 1 insertion(+)
c201550326@ubuntu:~/clone/public$ cat README.txt
modified
```