

Git 사용법

유호곤

ugonfor@gmail.com

[외부 유출 금지]



WhiteHat
School

목차

1. 개발환경 세팅 (Ubuntu)
2. Git 및 Github.com 환경 세팅
3. 코드 작성

개발 환경 세팅 (Ubuntu)

리눅스 환경 세팅

1. 앞으로 모든 개발은 Ubuntu 환경에서 진행합니다.
2. Window Application을 만드는 것이 아니라면, 모든 소프트웨어 개발의 Standard는 Linux입니다.
3. Linux 중에서도 Ubuntu, CentOS 등 여러가지 버전이 존재하지만, 가장 대중적인 것은 Ubuntu 입니다. (**MacOS**의 경우, Unix OS이기에 기본 terminal 사용해도 됩니다.)

개발 환경 세팅 (Ubuntu)

리눅스 환경 세팅

VMWare / VirtualBox와 같이 가상머신에 Ubuntu를 설치하여 사용하는 방법도 있습니다만,

WSL을 활용하는 것을 더 추천합니다. (Windows에서 Linux를 사용할 수 있게 되고, 훨씬 가볍기에)

개발 환경 세팅 (Ubuntu)

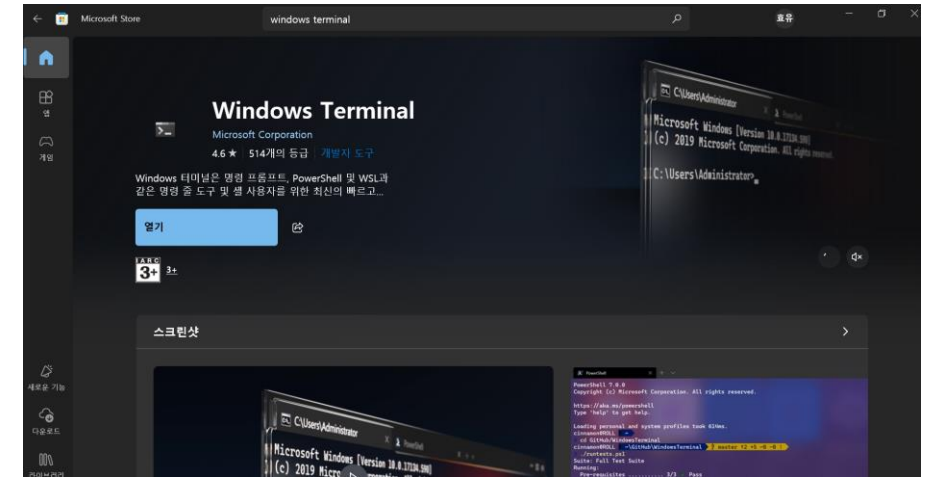
WSL 설치

1. Microsoft Store에서 Windows Terminal 설치 (우측 사진 참고)
2. Windows Terminal에서 아래 명령어 실행

```
$ wsl --install
```



```
$ wsl --list --onlinewsl --install -d Ubuntu
```
3. 재부팅 후 terminal에서 **wsl** 입력 시, WSL 사용 가능
4. 우측 화면과 같이 나오면 설치 완료.



```
PS C:\Users\ryuhyogon> wsl
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/ugonfor/.hushlogin file.
(base) ugonfor@gongon: /mnt/c/Users/ryuhyogon$ id
uid=1000(ugonfor) gid=1000(ugonfor) groups=1000(ugonfor),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),116(netdev)
(base) ugonfor@gongon: /mnt/c/Users/ryuhyogon$ |
```

개발 환경 세팅 (Ubuntu)

WSL 설치

1. (중요) 앞으로 개발할 때는 항상 WSL에서 합니다.

→ 앞으로 제가 작성하는 모든 명령어는 Linux에서 실행합니다.

2. (중요) VSCode를 사용할 때에도, WSL에서 실행합니다. (vscode 설치하는 구글링 통해서 알아보시다!)

```
ugonfor@gongon: /mnt/c/U × + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\ryuhyogon\Desktop\화이트햇\secure-coding> wsl
(base) ugonfor@gongon:/mnt/c/Users/ryuhyogon/Desktop/화이트햇/secure-coding$ code .
(base) ugonfor@gongon:/mnt/c/Users/ryuhyogon/Desktop/화이트햇/secure-coding$ |
```

Git 및 Github.com 환경 세팅

Github 세팅

1. 코드 보관, 공유, 버전 관리, 협업은 모두 git을 통해서 합니다.
2. Github는 git을 원활하게 사용하게 해주는 원격 저장소입니다.

Git 및 Github.com 환경 세팅

Git: 코드 버전 관리 시스템

→ 프로젝트에서 코드가 어떻게 바뀌었는지, 추적해주는 시스템

Github: git으로 관리하는 프로젝트를 올려두고 공유하는 플랫폼

→ 코드를 온라인으로 공유 가능하고, 사람들과 협업할 수 있게 해주는 플랫폼

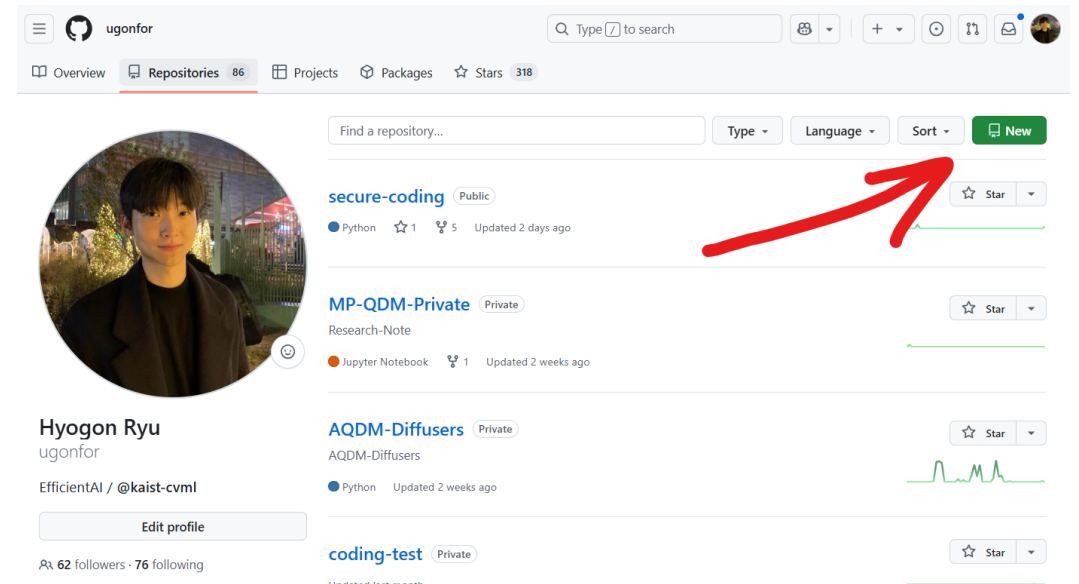
Git 및 Github.com 환경 세팅

Git 설치

```
$ sudo apt install git
```

Github 계정 생성 & Repository 생성

1. 계정 없으신 분들은 github.com에서 계정 생성
 2. Repository(저장소) 생성
- 오른쪽 사진과 같이, 본인 계정에 들어가면 Repository탭에, 초록색 New 버튼이 있습니다.



Git 및 Github.com 환경 세팅

Github 계정 생성 & Repository 생성

3. Repository 이름을 쓰고, Add a README file 체크

→ 이번 과제에 대한 repository이기에 “secure-coding”으로 명명하여 생성합니다.

4. Create Repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner *

ugonfor ▾

Repository name *

secure-coding

⚠ The repository secure-coding already exists on this account.

Great repository names are short and memorable. Need inspiration? How about [potential-broccoli](#) ?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

코드 작성

1. Local에 Repository 복제

\$ git clone <https://github.com/<your-id>/secore-coding>

2. 코드를 작성

helloworld.py → 해당 코드를 입력하세요.

\$ git add helloworld.py

\$ git commit -m "Add helloworld.py"

코드 작성

3. 코드 실행 방법 작성

README.md → markdown 포맷의 파일로, 본 프로젝트에 대한 설명을 기술, 어떻게 코드를 실행하는 지가 꼭 포함되어야 함.

```
$ git add README.md
```

```
$ git commit -m "Add README.md"
```

4. 원격 저장소(github)에 업로드

```
$ git push origin main
```

코드 작성

이제, <https://github.com/<your-id>/secore-coding> 에 가면,
본인이 작성한 코드와 README.md를 확인 가능합니다.

→ 앞으로 모든 프로젝트의 코드 공유는 웬만하면 github로 하는 것을 추천합니다.