Lab3. GitHub 접속 용 SSH 키 생성하기

※ Refer to https://www.lainyzine.com/ko/article/creating-ssh-key-for-github/
1. 안전하게 외부 Git 서버에서 코드를 Clone하거나 Push하려면 SSH 프로토콜을 사용한다.
2. GitHub 뿐만 아니라 원격 Git 저장소와 통신할 때 일반적으로 사용하는 방법이다.

3. SSH 공개키와 개인키 만들기
1)Windows에서는 Git을 설치하면 함께 설치되는 Git Bash를 사용하고, macOS나 Linux에서는 터미널을 실행해서 진행한다.
2)SSH 키를 만들기 전에 이미 키가 만들어져 있는지 확인해보는 것을 추천. 그냥 추가로 생성하면 기존 키가 덮어씌워질 수 있다.

```
$ cd ~/.ssh
$ ls
...
```

3)먼저 ~/.ssh 디렉터리로 이동해서 ls를 실행해 id_***와 id_***.pub 혹은 id_rsa와 id_rsa.pub 파일쌍이 있는지 확인한다.
4)이 파일이 있다면 이미 키를 생성했던 적이 있는 것이다.
5)다른 이름으로 여러개의 키를 만들어서 사용하는 것도 물론 가능하지만, 개인키의 위치를 따로 지정해줘야해서 불편하다.
6)이 파일들이 없다면 ssh-keygen으로 생성한다.

```
$ ssh-keygen -t ed25519 -C "your_email@example.com"
```

7)ed25519 방식으로 동작하지 않는 경우에는 아래와 같이 RSA로 옵션을 변경해 SSH 키를 생성한다.

```
$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

```
$ ssh-keygen -t ed25519 -C "javaexpert@nate.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/MZC01-HENRY/.ssh/id_ed25519):  <---현재 위치에 생성
Enter passphrase (empty for no passphrase):       <--비밀번호
Enter same passphrase again:                      <--비밀번호 확인
Your identification has been saved in /c/Users/MZC01-HENRY/.ssh/id_ed25519
Your public key has been saved in /c/Users/MZC01-HENRY/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:yOZB10W23VMfGsciXnG7xCeICHZrj8LUZ6x7hWmUnIY javaexpert@nate.com
The key's randomart image is:
+--[ED25519 256]--+
|      o .  .+oo+.|
|     . + B B **o=|
|      o E & =.+=+|
|     = + O +  ..+|
|      B S = .  . |
|     o o o .     |
|      . . .      |
|         .       |
|                 |
+----[SHA256]-----+
```

8)SSH 키가 생성되었다.
9)설명을 잘 읽어보면 개인키는 .ssh/id_ed25519에, 공개키는 .ssh/id_ed25519.pub에 저장되었다.
10)RSA 방식으로 키를 생성하는 경우 기본 파일 이름은 id_rsa, id_rsa.pub가 된다.

11)실제로 키가 잘 생성되었는지 개인키를 출력해보자.

```
$ cat id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABAGr/xjEB
LC9jwrnzO0H0E9AAAAEAAAAAEAAAAzAAAAC3NzaC1lZDI1NTE5AAAAIHhbrfY//xy/xJzk
1MYletTGNmiAXuVaioUPlWrbqKmXAAAAoO51CRiU6OY/IrKrGqgOqsPFbpopZsjV82edWV
ILK/jjY0jrobOIms54GxwT79BvKVnvDgFheecpM5TjT7XsnsVbQH3tQ6OoXDQR4PifeCCY
88lQ2TTXuqKK2le8NVAZmfj/KuuVyI+Ovn94E9YH9Mr9nCbQvMcL61IZZH9V5Ai0SfvjPW
ixz4Idv3j4hKs+NBpaG2ER2u+HsB3M/Y+kEpY=
-----END OPENSSH PRIVATE KEY-----
```

12)이번에는 공개키를 출력해보자.

```
$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHhbrfY//xy/xJzk1MYletTGNmiAXuVaioUPlWrbqKmX javaexpert@nate.com
```

13)이 내용을 전체를 직접 복사하거나, macOS에서는 pbcopy, 윈도에서는 clip 명령어를 사용해 클립보드에 복사한다.
14)cat으로 실행한 내용을 직접 복사하기보다는 텍스트가 깨지지 않도록 명령어를 통해 복사하는 걸 추천한다.

```
# macOS
$ pbcopy < ~/.ssh/id_ed25519.pub

# Windows
$ clip < ~/.ssh/id_ed25519.pub
```

4. 공개키를 GitHub 계정에 등록하기
1)로그인 후 오른쪽 상단의 프로필을 클릭하고, [Settings] 메뉴로 이동.
2)왼쪽 메뉴에서 [Access] > [SSH and GPG keys] 메뉴 선택

3)[SSH keys] > [New SSH key] 버튼 클릭
-[Title] : javaexpert.com SSH Key
-[Key type] : Authentication Key
-[Key] : 앞에서 복사한 공개키의 그대로 입력
-[Add SSH key] 버튼 클릭

4)GitHub에 키를 등록하면 github.com/[USERNAME].keys으로 접속하면 해당 사용자의 공개키를 확인할 수 있다.

5. 팁과 트러블 슈팅
1)GitHub에 SSH 키를 등록할 때 유용한 팁과 문제가 발생했을 때 대처하는 방법에 대해서 알아본다.
2)팁: SSH키 패스워드 입력을 생략하는 방법
-매번 SSH 키 패스워드 입력을 하려면 번거롭다.
-ssh-agent에 키를 등록해두면 좀 더 편리하게 사용할 수 있다.
-ssh-agnet는 백그라운드에서 SSH 인증 정보를 관리한다.
-ssh-agent가 실행되어있는지부터 확인해본다.

```
$ ssh-add -l
Error connecting to agent: Connection refused
```

-위와 같이 에러가 나오면 ssh-agent가 실행되어있지 않은 상태이다.
-SSH 키 목록이나 The agent has no identities.와 같은 메시지가 나오면 이미 ssh-agent가 실행된 상태다.
-실행되어있지 않다면, 먼저 ssh-agent를 실행한다.

```
$ eval "$(ssh-agent -s)"
```

-다음으로 ssh-add 명령어로 키를 등록한다.

```
$ ssh-add ~/.ssh/id_ed25519
Enter passphrase for /Users/lainyzine/.ssh/id_ed25519:
Identity added: /Users/lainyzine/.ssh/id_ed25519 (javaexpert@nate.com)
```

-ssh-add -l 명령어로 ssh-agent에 등록된 SSH 키들을 확인할 수 있다.

```
$ ssh-add -l
256 SHA256:JfXX05Pj352343pnuQiQ125Z6V88Q/F49cBViSumqVFeSp3LIQs javaexpert@nate.com (ED25519)
```

3)이 상태에서 git clone을 하면 더 이상 패스워드를 물어보지 않는다.

4)팁: macOS에서 키체인 사용하기
-macOS에서는 ssh-add 명령어를 사용할 때 -K 옵션을 붙여주면 패스워드를 키체인에 저장해서 사용할 수 있다.

```
$ ssh-add -K ~/.ssh/id_ed25519
Enter passphrase for /Users/lainyzine/.ssh/id_ed25519:
Identity added: /Users/lainyzine/.ssh/id_ed25519 (javaexpert@nate.com)
```

-처음 패스워드를 입력하면 키체인에 패스워드를 등록하고 패스워드를 다시 물어보지 않습니다.
-~/.ssh/config 파일에 AddKeysToAgent와 UseKeychain 옵션을 추가해두면 위의 과정들을 자동적으로 처리해줘서 편리하다.

5)트러블 슈팅: Permission denied
-ssh -T git@github.com로 SSH 접속 테스트를 할 때 에러가 발생할 수 있다.
-Permission denied 가 발생하는 경우 에러메시지를 잘 확인해보면 도움이 된다.

```
$ ssh -T git@github.com
no such identity: /Users/lainyzine/.ssh/id_ed25519: No such file or directory
git@github.com: Permission denied (publickey).
```

-No such file or directory 메시지에서 힌트를 얻을 수 있다.
-지정된 키 파일이 존재하지 않는 경우이다.
-i 옵션을 사용해 SSH 키 경로를 명시적으로 지정해보는 것도 좋다.

```
$ ssh -i .ssh/id_ed25519 -T git@github.com
```

-접속 과정을 좀 더 자세히 확인하고 싶다면 -v 옵션을 붙여본다.
-상세한 로그가 나와서 문제를 찾는데 도움이 됩니다.

```
$ ssh -v -T git@github.com
```