

모듈

모듈))클래스를 합친것(집합) 213p

import (가져오다 불러오다)→파이썬 파일을 가져오다

a.py(모듈(파이썬파일))파이썬 파일 하나에 담을 수 있는 것은 모두 모듈이라고 한다.

```
import matplotlib as plt  
(matplotlib(경로,환경변수))
```

which pip

Script

~~~

본인이 모듈을 직접 만드는 것도 가능하다.

home/lig/test.py

```
print.py    def print()
```

.py는 빼야 한다.(compile된 파일도 읽어올 수 있기 때문에)

환경 변수에 경로가 써져있기 때문에 (경로를 하나하나 다 지정해놓음 다 절대경로임)

본인이 직접 모듈을 만들수 있어??

## ChatGPT의 말:

ChatGPT

네, 파이썬에서는 직접 모듈을 만들 수 있습니다. 모듈은 파이썬 코드 파일(.py)로, 함수, 클래스, 변수 등을 포함할 수 있습니다. 이를 통해 재사용 가능한 코드를 작성하고, 다른 파이썬 프로그램에서 쉽게 사용할 수 있습니다.

## 모듈 만들기

### 1. 모듈 파일 생성:

- 원하는 이름으로 `.py` 파일을 생성합니다. 예를 들어, `mymodule.py` 라고 할 수 있습니다.

### 2. 모듈에 코드 작성:

- `mymodule.py` 파일 안에 함수나 클래스를 정의합니다. 예를 들어:

```
python
코드 복사
# mymodule.py

def add(a, b):
    return a + b

def subtract(a, b):
    return a - b
```

### 3. 모듈 사용하기:

- 다른 파이썬 스크립트에서 이 모듈을 가져와 사용할 수 있습니다. 예를 들어:

```
python
코드 복사
# main.py

import mymodule

result_add = mymodule.add(3, 5)
result_subtract = mymodule.subtract(10, 4)

print("덧셈 결과:", result_add)           # 덧셈 결과: 8
print("뺄셈 결과:", result_subtract)       # 뺄셈 결과: 6
```

## 모듈의 위치

- 모듈을 사용할 때는, 해당 모듈 파일이 현재 작업 중인 스크립트와 같은 디렉터리에 있거나, 파이썬 경로에 포함되어 있어야 합니다.

## 모듈의 장점

- 코드의 재사용성을 높이고, 코드의 구조를 더 깔끔하게 유지할 수 있습니다.
- 대규모 프로그램을 여러 개의 모듈로 나누어 관리하면 유지보수가 쉬워집니다.

이처럼, 본인이 원하는 기능을 가진 모듈을 직접 만들고, 이를 다른 프로그램에서 재사용하는 것이 가능합니다!

4o mini

## 나의 말:

cmd랑 파워셸이랑 차이가 뭐야?

## ChatGPT의 말:

ChatGPT

**CMD** (Command Prompt)와 **PowerShell** 은 모두 Windows 운영체제에서 사용할 수 있는 명령 줄 인터페이스이지만, 몇 가지 중요한 차이점이 있습니다.

## 1. 기본 개념

- **CMD (Command Prompt):**
  - 전통적인 Windows 명령 줄 인터페이스입니다.
  - MS-DOS 기반으로 만들어졌으며, Windows의 초기 버전부터 사용되었습니다.
  - 기본적인 파일 관리 및 시스템 관리 작업을 수행할 수 있습니다.
- **PowerShell:**
  - Microsoft에서 개발한 보다 강력한 명령 줄 인터페이스입니다.
  - .NET Framework 위에서 작동하며, 명령줄 환경에서 스크립트를 실행하고 시스템 관리 작업을 자동화하는 데 중점을 둡니다.
  - 객체 지향적이며, 파이프라인을 통해 데이터를 객체로 처리할 수 있습니다.

## 2. 기능

- **CMD:**

- 기본적인 파일 조작 (복사, 이동, 삭제 등)과 시스템 정보 조회를 지원합니다.
- 명령어는 주로 텍스트 기반입니다.

- **PowerShell:**

- 고급 스크립팅 언어와 명령어를 지원합니다.
- 다양한 시스템 관리 작업을 자동화할 수 있는 강력한 기능을 제공합니다.
- `cmdlets`이라는 명령어를 사용하여 복잡한 작업을 간단하게 수행할 수 있습니다.
- 다양한 데이터 형식과 객체를 처리할 수 있는 파이프라인을 지원합니다.

### 3. 명령어 예시

- **CMD:**

```
cmd
코드 복사
dir          # 현재 디렉터리의 파일 및 폴더 목록을 표시합니다.
copy file.txt backup.txt # 파일을 복사합니다.
```

- **PowerShell:**

```
powershell
코드 복사
Get-ChildItem          # 현재 디렉터리의 파일 및 폴더 목록을
                        표시합니다.
Copy-Item file.txt backup.txt # 파일을 복사합니다.
```

### 4. 스크립팅

- **CMD:**

- 배치 파일( `.bat` )을 사용하여 스크립트를 작성합니다.
- 기능이 제한적이며, 복잡한 로직 구현이 어렵습니다.

- **PowerShell:**

- PowerShell 스크립트 파일( `.ps1` )을 사용하여 더 복잡하고 강력한 스크립트를 작성할 수 있습니다.
- 조건문, 반복문 등 다양한 프로그래밍 기능을 제공합니다.

## 요약

- `CMD` 는 기본적인 명령어를 통해 시스템 작업을 수행하는 간단한 인터페이스이고, `PowerShell` 은 보다 고급 기능과 스크립팅 능력을 제공하는 강력한 도구입니다. PowerShell은 시스템 관리와 자동화 작업을 위한 현대적인 대안으로 점점 더 많이 사용되고 있습니다.

4o mini

환경변수가 뭐야?

## ChatGPT의 말:

ChatGPT

환경변수(Environment Variable)는 운영체제에서 사용하는 설정 값으로, 시스템의 동작을 조정하는 데 도움을 줍니다. 이러한 변수들은 시스템이나 사용자에게 대한 정보를 포함하고 있으며, 프로그래밍 언어나 스크립트에서도 참조할 수 있습니다. 환경변수는 운영체제나 애플리케이션의 작동 방식을 설정하는 데 사용됩니다.

## 환경변수의 주요 역할

### 1. 경로 설정:

- 프로그램이나 실행 파일이 저장된 디렉터리 경로를 지정합니다.
- 예를 들어, `PATH` 변수에 여러 디렉터리의 경로를 추가하면, 사용자가 명령 프롬프트나 터미널에서 프로그램을 실행할 때 해당 경로를 자동으로 참조합니다.

### 2. 시스템 정보 제공:

- 시스템의 정보나 사용자 정보를 저장합니다.
- 예를 들어, `HOME` 변수는 사용자의 홈 디렉터리를 나타내고, `USER` 변수는 현재 사용자 이름을 포함합니다.

### 3. 설정 및 옵션 조정:

- 애플리케이션의 설정이나 동작 방식을 제어합니다.
- 특정 프로그램이 어떤 설정 파일을 읽을지, 또는 어떤 환경에서 실행될지를 결정하는 데 사용할 수 있습니다.

## 예시

### • Windows 환경변수:

- `PATH`: 실행 가능한 파일이 있는 디렉터리 경로 목록을 저장합니다.
- `TEMP`: 임시 파일이 저장되는 경로를 지정합니다.

### • Linux/Unix 환경변수:

- `HOME`: 현재 사용자의 홈 디렉터리를 지정합니다.
- `PATH`: 실행 파일의 경로를 지정합니다.

## 환경변수 설정하기

- **Windows:**

1. 시작 메뉴에서 "환경 변수"를 검색합니다.
2. "시스템 속성"에서 "환경 변수"를 클릭합니다.
3. 여기서 새 변수 추가, 수정, 삭제를 할 수 있습니다.

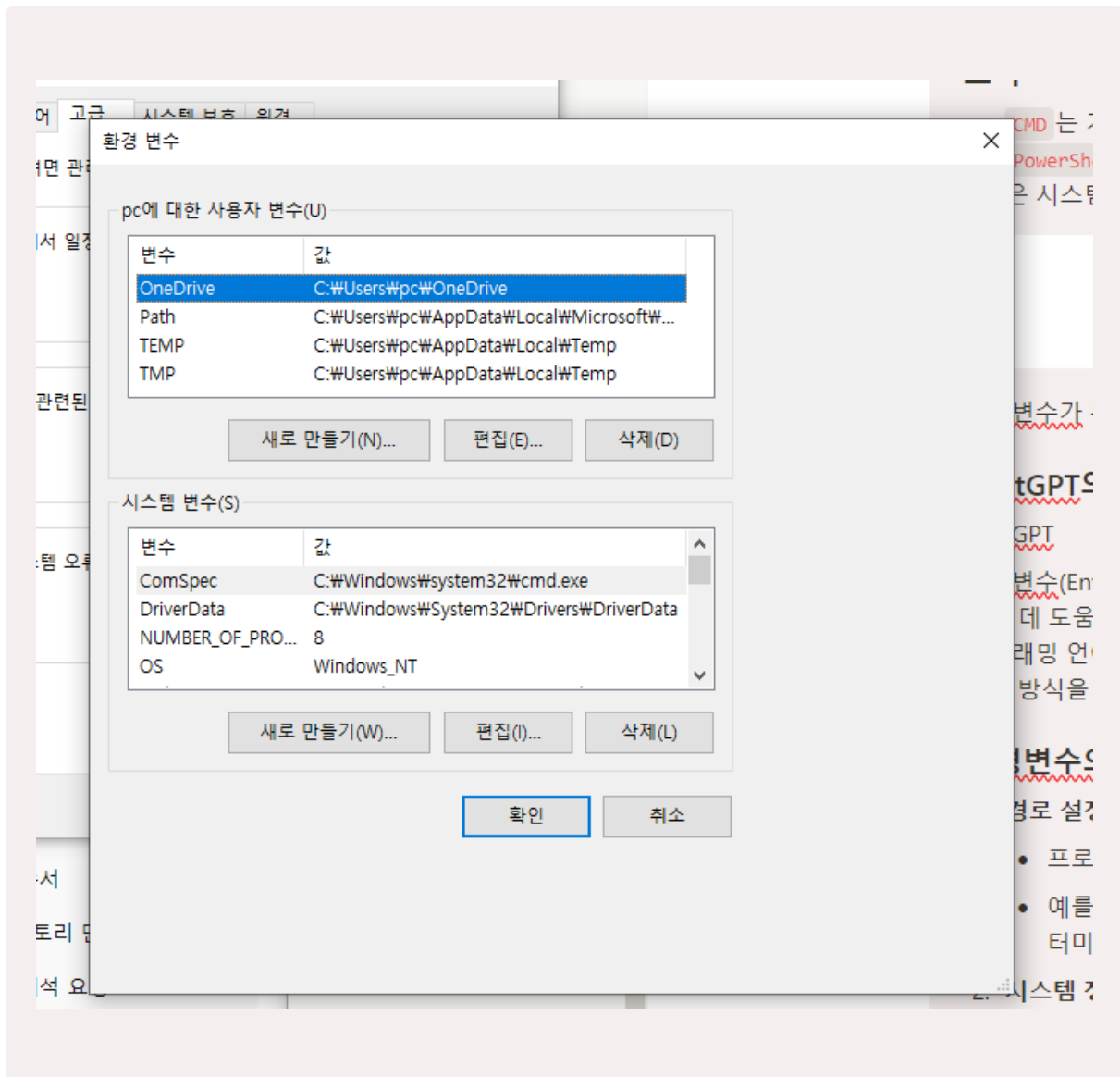
- **Linux/Unix:**

- 셸에서 `export` 명령을 사용하여 환경변수를 설정합니다.

```
bash
코드 복사
export VARIABLE_NAME=value
```

## 요약

환경변수는 운영체제와 애플리케이션의 동작을 제어하는 중요한 설정 값입니다. 이 변수를 통해 시스템에 대한 다양한 정보를 저장하고, 프로그램의 경로 및 설정을 조정할 수 있습니다. 환경변수를 잘 활용하면 시스템 관리와 애플리케이션 개발에 큰 도움이 됩니다.



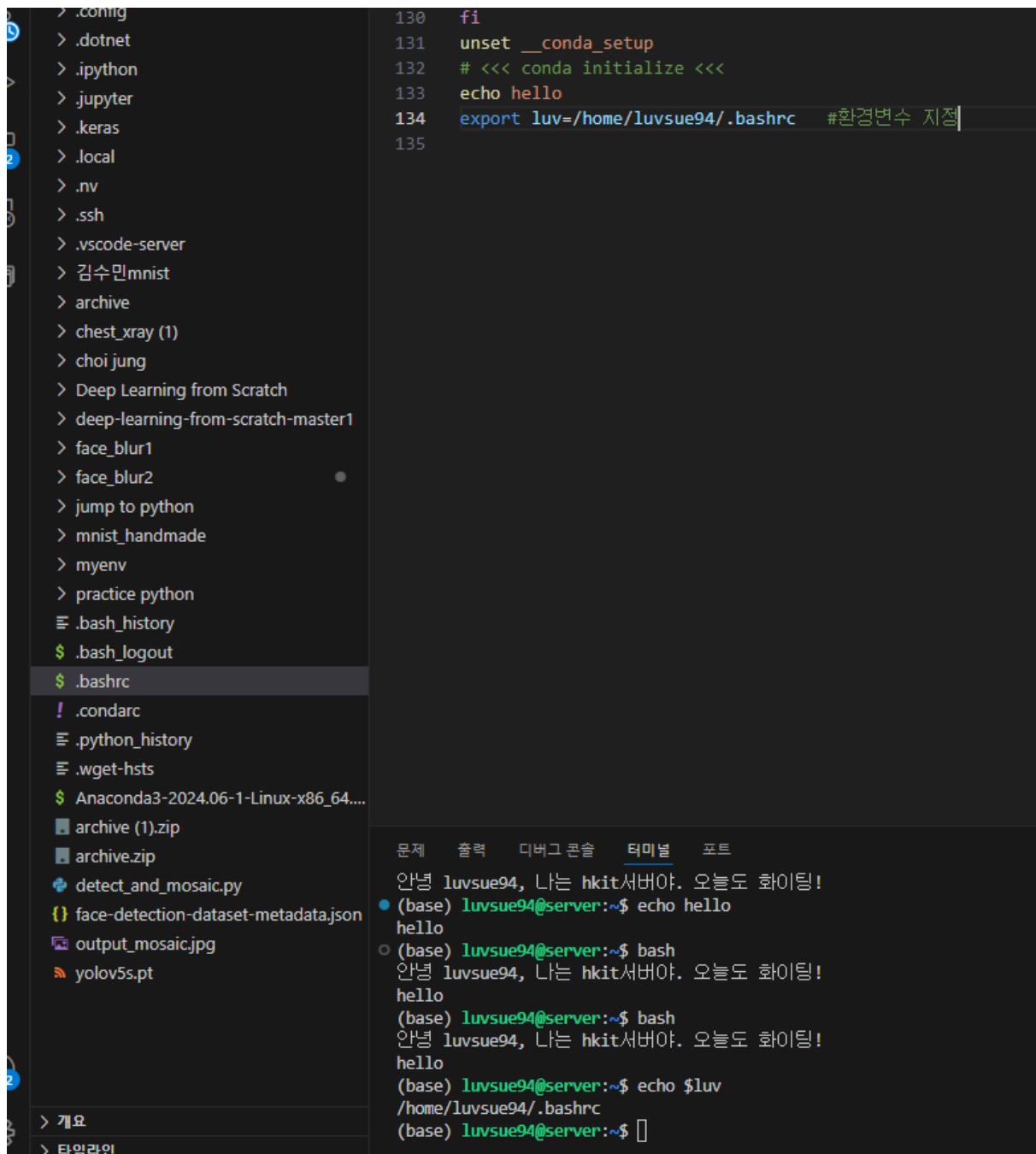
nano .zshrc

vim .bashrc(환경변수가 기록된 파일)-내파일에서 실행될때마다

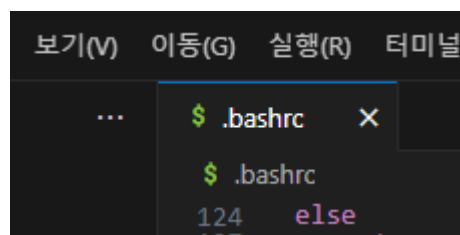
export 변수명=경로(리눅스서버에서 환경변수적용할때이렇게 씀)  
echo 안녕 ))







→ 환경 변수 설정 (달러 표시 붙이는 이유는 변수라서 그럼)



이부분에 ● 이 되면 안된다(x)로 되어야 저장되는 것

```
문제 출력 디버그 콘솔 터미널 포트
(base) luvsue94@server:~$ bash
안녕 luvsue94, 나는 hkit서버야. 오늘도 화이팅!
hello
(base) luvsue94@server:~$ echo $luv
/home/luvsue94/.bashrc
(base) luvsue94@server:~$ which list
(base) luvsue94@server:~$ which pip
/opt/anaconda3/bin/pip
(base) luvsue94@server:~$ cd /opt/anaconda3/bin/
(base) luvsue94@server:/opt/anaconda3/bin$ ll
합계 422356
drwxrwsrwx  3 root      anaconda  36864 Sep 24 14:09 ./
drwxrwsrwx 44 root      anaconda  4096 Sep 24 14:09 ../
```

Script어쩌구 하는거는 윈도우에서 하는 것

from 경로 import function or class.()(맨마지막에 함수랑 클래스()가 와야 한다.)  
from에 들어갈거랑 import에 들어갈거랑 서로이어져야하며 생략된 경로가 없어야 한다.

**from import**다음에는 경로가 온다.

- 1.from math import sin
- 2.import math → math. sin()

1. from torch import nn.module.losses.crossentropy.()
2. import torch  
nn.module.losses.cross

```
if __name__ == "__main__":
```

실행안하고

print.py

```
def print A():
```

```
    print("Hello")
```

```
    print("World")
```

```
print A()
```