


1주차 팀 미션: jwan코치_06팀

 **Q1.** 강의 내용에 따르면 '파이썬'과 같은 프로그래밍 언어로 컴퓨터에게 우리가 원하는 일을 시킬 수 있다고 배웠습니다. **파이썬으로 할 수 있는 일은 어떤 것들이 있고, 나는 그중에서 무엇을 해보고 싶은지 적어봅시다.** [난이도 : ★/5]

파이썬으로 할 수 있는 일:

웹 개발: 파이썬은 웹 애플리케이션을 개발하는 데 사용되며, Django와 Flask와 같은 웹 프레임워크를 활용하여 웹 사이트를 구축할 수 있다.

데이터 분석 및 인공지능: 파이썬은 데이터 분석과 머신 러닝, 딥 러닝 분야에서 많이 활용되고 있으며 pandas, NumPy, scikit-learn, TensorFlow, PyTorch 등의 라이브러리를 사용하여 데이터 분석, 예측 모델 개발, 이미지 처리, 자연어 처리 등 다양한 작업을 수행할 수 있다.

과학 및 공학 계산: 파이썬은 수치 계산, 시뮬레이션, 데이터 시각화 등 과학 및 공학 분야에서도 널리 사용되며, SciPy, matplotlib, NumPy 등의 라이브러리를 활용하여 과학적인 연구와 실험을 수행할 수 있다.


자동화 및 스크립팅: 파이썬은 반복적이고 자동화할 수 있는 작업들을 스크립트 형태로 작성하여 수행하고, 파일 조작, 데이터 처리, 시스템 관리 등 다양한 자동화 작업에 유용하게 사용된다.

게임 개발: 파이썬은 게임 개발에도 사용됩니다. Pygame과 같은 라이브러리를 사용하여 2D 게임을 개발하거나, Unity와 같은 게임 엔진과 연동하여 게임 개발에 파이썬을 활용할 수 있다.

파이썬으로 해보고 싶은 일:

Pandas 등 자료 분석에 특화된 모듈을 이용하여 데이터 분석을 해보고 싶습니다. 데이터 수집부터 시작해 데이터 전처리, 그리고 수많은 정보를 통해서 유용한 결론을 도출하는 과정까지를 하기 위해 파이썬을 배우고자 합니다.

저는 데이터 분석 공모전에 참여하고 싶어서 파이썬 스터디를 시작하고 싶어했습니다. 파이썬을 통해 데이터 분석이 가능하며 이를 토대로 빅데이터 분석을 하고, 인공지능까지 접목시키는 것이 목표입니다.

 **Q2.** 파이썬 코딩을 하기에 앞서 하드웨어를 이해하는 것이 중요하다고 배웠습니다. **하드웨어 아키텍처에서 CPU와 Main Memory 그리고 Secondary Memory의 역할을 간단하게 정리하여 봅시다.** [난이도 : ★/5]

CPU


- 중앙 처리 장치로 컴퓨터의 뇌 역할을 하는 핵심 부품
- 명령어를 해석하고 실행하는 데 필요한 연산을 수행함
- 프로그램의 명령어를 가져와서 해독하고, 필요한 데이터를 주기억장치(메인 메모리)에서 읽어와서 연산을 수행, 결과를 다시 주 기억장치에 저장하거나 출력장치로 전송하는 역할을 한다.

Main Memory

- 프로그램이 실행되는 동안 필요한 데이터와 명령어를 저장하는 공간
- 주 기억장치. CPU가 직접적으로 접근할 수 있는 고속의 저장장치이다.
- RAM으로 구성되며, 프로그램의 명령어와 데이터가 임시로 저장되어 CPU가 필요할 때 읽고 쓸 수 있도록 한다.
- 메인 메모리는 전원이 꺼지면 저장된 데이터가 사라지는 휘발성 메모리이다.

Secondary Memory

- Main Memory와 달리 비휘발성 메모리이다.
- 컴퓨터 시스템에 데이터를 영구적으로 저장하기 위해 사용되며, 하드디스크 드라이브, SSD, 광학 디스크 등이 대표적인 보조 메모리의 예시이다.
- 큰 용량을 가지고 있으며, 데이터를 저장하고 필요에 따라 읽고 쓸 수 있다.
- 프로그램과 데이터는 주기억장치로 복사되어 CPU에 의해 처리되기 전에 보조 메모리에서 주기억장치로 옮겨진다.
- 운영체제, 응용 프로그램, 사용자 파일 등이 보조 메모리에 저장되어 필요할 때 메인 메모리로 로드되어 사용된다.

 **Q3.** 파이썬은 우리의 명령을 이해하지 못했을 때, 친절하게 Error Message 로 어떤 명령을 이해하지 못했는지 알려줍니다. 이를 보고 코드의 버그를 수정해가는 과정을 Debugging 이라고 하는데요, 이것은 코딩에 있어서 매우 중요합니다. 따라서 이번 미션에서는 Error Message 에 대해 이해하는 시간을 가져봅시다. [난이도 : ★/5]

강의에서는 에러 메시지로 Syntax Error, ValueError, TypeError 3가지가 등장했습니다.

① 각각의 Error를 발생시키는 코드를 2가지씩 만들어보고

- ② Debugging한 코드도 만들어 봅시다.
- ③ 그리고 그 밖에 다른 Error도 3가지를 찾아 그 Error를 발생시키는 코드와
- ④ Debugging한 코드를 1가지씩 만들어 봅시다.

- Syntax Error: 코드 구문 자체가 잘못되었을 때 발생하는 에러

```
In [1]: # Syntax Error를 발생시키는 코드 예시 1
print("Hello, World!")

# Syntax Error를 발생시키는 코드 예시 2
for i in range(10)
    print(i)

File "C:\Users\kgw08\AppData\Local\Temp\ipykernel_29376\1000586076.py", line 6
    print(i)
    ^
SyntaxError: invalid syntax
```

```
In [2]: # Syntax Error를 디버깅코드 1
print("Hello, World!")

# Syntax Error를 디버깅코드 2
for i in range(10):
    print(i)

Hello, World!
0
1
2
3
4
5
6
7
8
9
```

- ValueError: 변수의 타입은 올바르지만 값이 적절하지 않을 때 발생하는 에러

```
In [3]: # ValueError를 발생 코드 1
        int("abc")

        # ValueError를 발생 코드 2
        float("Hello")
```

```
-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_29376\3500979586.py in <module>
      1 # ValueError를 발생시키는 코드 예시 1
----> 2 int("abc")
      3
      4 # ValueError를 발생시키는 코드 예시 2
      5 float("Hello")

ValueError: invalid literal for int() with base 10: 'abc'
```

```
In [4]: # ValueError를 디버깅 코드 1
        int("123")

        # ValueError를 디버깅 코드 2
        float("3.14")
```

Out [4]: 3.14

- TypeError: 변수의 타입 자체가 잘못되었을 때 발생하는 에러

```
In [5]: # TypeError를 발생 코드 1
        1 + "2"

        # TypeError를 발생 코드 2
        len(123)
```

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_29376\2732113657.py in <module>
      1 # TypeError를 발생 코드 1
----> 2 1 + "2"
      3
      4 # TypeError를 발생 코드 2
      5 len(123)

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
In [6]: # TypeError를 디버깅 코드 1
1 + 2

# TypeError를 디버깅 코드 2
len(str(123))
```

Out [6]: 3

- IndexError: 인덱스가 범위를 벗어날 때 발생하는 에러

```
In [7]: # IndexError를 발생 코드
my_list = [1, 2, 3]
print(my_list[3])
```

```
-----
IndexError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_29376\2798565608.py in <module>
      1 # IndexError를 발생시키는 코드 예시
      2 my_list = [1, 2, 3]
----> 3 print(my_list[3])

IndexError: list index out of range
```

```
In [8]: # IndexError를 디버깅 코드
my_list = [1, 2, 3]
print(my_list[2]) # 정상적인 인덱스 사용
```

3

- KeyError: 딕셔너리에서 존재하지 않는 키를 참조할 때 발생하는 에러

```
In [9]: # KeyError를 발생 코드
my_dict = {"apple": "사과", "banana": "바나나"}
print(my_dict["grape"])
```

```
-----
KeyError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_29376\477548601.py in <module>
      1 # KeyError를 발생 코드
      2 my_dict = {"apple": "사과", "banana": "바나나"}
----> 3 print(my_dict["grape"])
```

KeyError: 'grape'

```
In [10]: # KeyError를 디버깅 코드
my_dict = {"apple": "사과", "banana": "바나나"}
print(my_dict.get("grape", "해당 키가 없습니다.")) # 키가 없을때 기본값 반환
```

해당 키가 없습니다.

- ZeroDivisionError는 0으로 나누기 연산을 수행할 때 발생하는 에러

```
In [11]: # ZeroDivisionError를 발생 코드
result = 10 / 0
```

```
-----
ZeroDivisionError                       Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_29376\1943275962.py in <module>
      1 # ZeroDivisionError를 발생 코드
----> 2 result = 10 / 0
```

ZeroDivisionError: division by zero

```
In [12]: # ZeroDivisionError를 디버깅 코드
try:
    result = 10 / 0
except ZeroDivisionError:
    print("0으로 나눌 수 없습니다.")
```

0으로 나눌 수 없습니다.

🔴Q4. 강의에서 나온 미국과 유럽의 엘리베이터 층수를 변환하는 프로그램을 기억하시나요? 그와 비슷하게 법 개정 전 한국 나이를 미국 나이로 변환하는 프로그램을 코딩해 봅시다. [난이도 : ★★/5]

- hint: 미국 나이는 생일이 지났는지 안지났는지가 중요하죠!
- `birth = int(input("생일이 지났습니까? 맞으면 0 아니면 -1 : "))`

```

1 #법 개정 전 한국 나이를 미국 나이로
2 #바꿔주는 프로그램을 만들겠습니다.
3
4 korea_age= int(input('Please let me know your KOREAN AGE: '))
5 if not(korea_age>0): #한국 나이가 음수이거나 0인 경우 에러를 출력하고 탈출
6     print("error")
7     exit()
8 birthday= int(input('올해 생일이 지났다면 1을, 지나지 않았다면 0을 입력해주세요: '))
9 if birthday==1:
10     international_age=korea_age-1
11     print("제 미국 나이는", international_age, "살 입니다.")
12
13 if birthday==0:
14     international_age=korea_age-2
15     print("제 미국 나이는", international_age, "살 입니다.")
16 if not(birthday==1 or birthday==0):#사용자가 0이나 1을 입력하지 않은 경우 에러를 출력
17     print("error")
18     exit()
19
20

```

Microsoft Windows [Version 10.0.22621.1992]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ymy00\Documents\py4e>"C:/Program Files/Python311/python.exe" c:/Users/ymy00/Docum
ents/py4e/prac1.py
Please let me know your KOREAN AGE SYSTEM: 25
올해 생일이 지났다면 1을, 지나지 않았다면 0을 입력해주세요: 2
error

C:\Users\ymy00\Documents\py4e>

초안 코드

```

age = int(input("한국 나이를 입력하세요: "))
if age <= 0: # 한국 나이가 자연수가 아닌 경우
    print("error")

birthday = int(input("생일이 지났습니까? 맞으면 0, 아니면 -1을 입력하세요: "))
if not (birthday == 0 or birthday == -1): # 생일 여부가 0 또는 -1이 아닌 경우
    print("error")

us_age = age + birth -1 # 한국 나이와 생일 여부를 이용하여 미국 나이 계산
print("미국 나이: %s살" % us_age)

```

초안 코드에서 음수/ 혹은 0살의 한국 나이를 입력한 뒤에 error 출력 여부와 상관 없이 생일이 지났는지 여부를 묻는다는 점을 수정하기 위해 `exit()` 코드 작성하였습니다.