



# 1주차 미션\_이정은

📌 속성	1주차
🕒 작성일시	@2022년 7월 14일 오후 3:11
📅 날짜	@2022년 7월 13일

## 🔔🔔 1주차 미션 내용을 알려드립니다 🔔🔔

👍👍 1주차 미션 목적 - 파이썬 프로그래밍에 대한 동기부여

부스터 여러분들, 1주차 강의는 잘 들으셨나요?!

학습한 내용을 토대로 풀이하여야 할 1주차 미션 내용을 아래와 같이 공개합니다!

미션 내용을 팀원들과 함께 풀이해주세요!(적극적인 토론이 필요합니다!)

📢매주 화요일 23:59까지 리드부스터가 제출해주세요! 모두들 화이팅입니다!

---

### 🔔🔔 1주차 미션 내용을 알려드립니다 🔔🔔

미션 1.

미션 2.

미션 3.

미션 4.

---

## 미션 1.



**Q1. 파이썬으로 컴퓨터에게 우리가 원하는 일을 시킬 수 있다고 하였습니다. 파이썬으로 할 수 있는 일은 어떤 것들이 있고, 나는 그중에서 무엇을 하고 싶은지 적어봅시다.**

- 시스템 유틸리티 제작, GUI 프로그래밍, 웹 프로그래밍, 수치 연산 프로그래밍, 데이터베이스 프로그래밍, 데이터 분석, 사물 인터넷 등 .

#### 시스템 유틸리티 제작

: 파이썬은 운영체제 시스템 명령어를 사용할 수 있는 각종 도구를 갖추고 있기 때문에 , 이를 바탕으로 여러 시스템 유틸리티를 만들 수 있다.

#### GUI 프로그래밍 (Graphic User Interface)

: 화면에 새로운 윈도우 창을 만든다거나 , 프로그램을 동작시킬 메뉴, 버튼, 그림 등을 추가하는 것이다. 파이썬은 GUI 프로그래밍을 위한 도구들이 잘 갖추어져있다.

- 대표적인 예로 Tkinter

#### C/C++ 과의 결합

: 파이썬은 접착(glue)언어이다. 다른 언어와 결합해서 사용 가능하기 때문이다.

C나 C++로 만든 프로그램을 파이썬에서 사용할 수 있고 C나 C++에서 파이썬으로 만든 프로그램을 사용할 수 있다.

#### 웹 프로그래밍

: 파이썬은 웹 프로그램을 만들기에 매우 적합한 도구, 웹 페이지를 제작할 수 있다.

#### 수치 연산 프로그래밍

: 파이썬은 수치 연산 프로그래밍에 적합한 언어는 아니지만 C언어로 작성한 Numpy라는 수치 연산 모듈을 사용하면 파이썬에서도 빠른 수치 연산이 가능하다.

#### 데이터베이스 프로그래밍

: 파이썬은 사이베이스(Sybase), 인포믹스(Infomix), 오라클(Oracle), MySQL, PostgreSQL 등 굵직한 데이터베이스 접근 도구를 제공하고 있다.

- pickle 이라는 모듈을 통해 파이썬에서 사용하는 자료를 변형없이 파일에 그대로 저장하고 불러오는 일을 수행할 수도 있다.

#### 데이터분석과 사물인터넷

: 파이썬 Pandas 모듈은 데이터 분석을 더 쉽고 효과적으로 할 수 있도록 도와준다.

R을 사용해왔지만 Pandas가 등장한 이후 파이썬의 사용도가 점점 증가하는 추세이다.

: 라즈베리파이의 경우 리눅스기반의 아주 작은 컴퓨터이다. 파이썬은 이 라즈베리파이를 제어하는 도구로 사용된다. 라즈베리파이의 기능을 파이썬으로 작동시킬 수 있다.

파이썬 Django 등의 프레임워크를 사용하여 웹 어플리케이션을 제작해보고 싶습니다!

## 미션 2.



**Q2. 파이썬 코딩을 하기에 앞서 하드웨어를 이해하는 것이 중요하다고 했습니다. 하드웨어 아키텍처에서 CPU와 Main Memory 그리고 Secondary Memory의 역할을 간단하게 정리하여 봅시다.**

CPU란?

: 중앙 처리 장치, 프로그램의 명령어를 해석하여 데이터를 연산/처리를 하고 그렇게 돌아가도록 제어해주는 부분, 혹은 그 기능을 내장한 칩  
컴퓨터에서 기억, 해석, 연산, 제어라는 4대 주요 기능을 관할한다.

Main Memory란?

: 주기억장치, 프로그램이나 데이터류를 기억시키는 장치  
프로그램 실행동안 자료나 프로그램 설명에 빠른 속도로 접근하게 해준다.

Secondary Memory란?

: 보조기억장치, 컴퓨터의 CPU에 직접 연결된 주기억장치가 아닌 디스크, 테이프 등 외부의 기억장치를 말한다.

주기억장치에 기억시킬 수 없는 대량의 데이터를 기억하기 위한 장비로 자주 사용되지 않는 프로그램이나 데이터를 기억시켜 두었다가 필요할 때마다 사용할 수 있다.

### 미션 3.



**Q3. 파이썬은 우리의 명령을 이해하지 못했을 때, 친절하게 Error Message를 통해 어떤 명령을 이해하지 못했는지 알려줍니다. 그것을 보고 코드의 버그를 수정해가는 과정을**

**Debugging** 이라고 합니다. 이것은 코딩에 있어서 매우 중요합니다. 따라서, Error Message에 대해서 이해하는 시간을 가져봅시다.

강의에서는 SyntaxError, 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.\*\*

### [답안]

## 1번\_ Error를 발생시키는 코드 2가지

```
# SyntaxError : 문법이 잘못된 경우 발생하는 오류
a=1
if a==1
    print('hi') // IF 뒤에 콜론이 붙어있지 않기때문에 ERROR

> File , line 2
      if a==1
        ^
SyntaxError: invalid syntax

-----

# ValueError : 부적절한 값을 입력받은 경우 발생하는 에러
arr = ['a', 'b', 'c', 'd', 'e']
arr.index('f')

> Traceback (most recent call last):
  File , line 2, in <module>
    arr.index('f')
ValueError: 'f' is not in list

-----

# TypeError : 잘못된 타입을 전달했을 때 발생하는 에러
a = 1 + "abc"

> Traceback (most recent call last):
  File, line 1, in <module>
    a = 1 + "abc"
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

## 2번\_ Debugging

```
# SyntaxError
a=1
if a==1:
    print('hi')

> hi

-----

# ValueError
arr = ['a', 'b', 'c', 'd', 'e']
print(arr.index('a'))

> 0

-----
```

```
# TypeError
a = str(1) + "abc"
print(a)

> 1abc
```

### 3번\_그 외 다른 Error 코드 3가지

```
# IndexError : 범위를 벗어나는 경우 발생하는 에러

arr = ['a', 'b', 'c', 'd', 'e']
arr[10]

> Traceback (most recent call last):
  File , line 2, in <module>
    arr[10]
IndexError: list index out of range

-----

# AttributeError : 클래스(모듈)의 객체에 해당하는 메서드나 속성을 잘못 호출하거나 대입할 경우 발생하는
에러

import math
a = math.ceilnumber(1.2)
print(a)

> Traceback (most recent call last):
  File , line 8, in <module>
    a = math.ceilnumber(1.2)
AttributeError: module 'math' has no attribute 'ceilnumber'

-----

# KeyError : 딕셔너리에서 접근하려는 키 값이 없을 경우 발생하는 에러

dic = {"a": 12, "b": 33}
print(dic["z"])

> Traceback (most recent call last):
  File, line 2, in <module>
    print(dic["z"])
KeyError: 'z'
```

### 4번\_Debugging

```
# IndexError
arr = ['a', 'b', 'c', 'd', 'e']
print(arr[0])

> a
```

```
-----  
  
# AttributeError  
import math  
a = math.ceil(1.2)  
print(a)
```

```
> 2
```

```
-----  
  
# KeyError  
dic = {"a": 12, "b": 33}  
print(dic["a"])
```

```
> 12
```

## 미션 4.



**Q4. 강의에서 미국과 유럽의 엘리베이터 층수를 변환하는 프로그램이 나왔습니다.**

**그와 비슷하게 우리는 한국 나이를 미국 나이로 변환하는 프로그램을 코딩 해봅시다.**

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

[python]

```
age = int(input("한국 나이를 입력해주세요 : "))  
birth = int(input("생일이 지났습니까? 맞으면 0 아니면 -1 : "))  
my_usa_age = age - 1 + birth  
print(f"미국 나이는 {my_usa_age}살 입니다")
```

```
>
```

```
한국 나이를 입력해주세요 : 40  
생일이 지났습니까? 맞으면 0 아니면 -1 : 0  
미국 나이는 39살 입니다
```