

복습

- Anaconda, Jupyter Notebook
- `+` `-` `*` `/` `//` `%` `**`
- `>` `<` `>=` `<=` `==` `!=`
- True or False
- Variables
- $y = f(x)$

파이썬 기본

1. 데이터 타입과 변수, 그리고 출력
2. 문자열 다루기 기본과 리스트 데이터 구조
3. 조건문
4. 반복문
5. 함수
6. 다양한 데이터 구조: 튜플, 딕셔너리, 집합

이외에 필요한 부분은 실제 프로그래밍을 하며 필요할 때 익힙니다.!

- 객체와 클래스, 라이브러리, 문자열등등

문자열 (string)

```
some_string = "python"  
len(some_string)
```

문자열 특수 기능

- 인덱스 (index)
 - 인덱스는 특정 데이터를 가리키는 번호를 나타냄

p	y	t	h	o	n
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

```
some_string[0]  
some_string[1]
```

- 슬라이싱 (slicing)

```
some_string[3:5]  
some_string[1:5]  
some_string[1:2]
```

문자열 특수 함수

특이하게 변수.함수 형태를 가짐
추후 강의에서 좀더 깊게 다루며 익숙해지도록!

```
func = "python is easy programming language"
```

```
func.count('p')
```

```
func.find('p')
```

```
python_is_easy = "python is easy"  
python_is_easy.replace("python", "golang")
```

문자열 특수 함수

```
some_string = "    computer    "  
some_string.strip()
```

```
some_string = ",,,DaveLee..."  
some_string.strip(",")  
some_string.strip(".")
```

다양한 출력

```
print("I have a {}, I have an {}".format("pen", "apple"))
```

```
print("I have a {1}, I have an {0}".format("pen", "apple"))
```

```
interest = 0.087  
print (format(interest, ".2f"))
```

문자열 출력 (참고)

```
print("I have a %s, I have an %s." % ("pen", "apple"))
```

%s - string
%c - character
%d - int
%f - float

데이터 구조 - 리스트 (list)

데이터 구조: 연관있는 데이터를 어떻게 컴퓨터로 효율적으로 다룰 수 있을까?

- 리스트
 - 여러가지 관계가 있는 데이터를 하나의 변수로 다루는 방법
 - [] 안에 , 를 사용해서 데이터를 써넣으면 됨
 - 예: mydata = [1, 2, 'hello', 1.2]
- 각 데이터는 인덱스를 사용해서 다룰 수 있음

SmallProject1

- 처음 리스트 변수 데이터
 - `lang = ["java", "c", "c++"]`
- 리스트 함수 활용해서 다음과 같이 리스트 만들기
 - `lang = ["java", "c", "c++", "golang"]`

어떻게 SmallProject1을 풀 수 있을지, 리스트 함수 알아보기

- list 선언 방법: 선언과 동시에 값을 채워넣을 수 있습니다.

```
lang = []
```

```
lang = list()
```

```
lang = ["python", "c", "java", "golang"]
```

- 리스트 변수에서 특정 데이터 읽고 바꾸기

```
lang[0]
```

```
lang[1]
```

```
lang[2] = "c++"
```

- 리스트 변수 끝에 데이터를 추가하기

```
lang.append("python")
```

```
lang.append("java")
```

```
lang.append("go lang")
```

- 특정한 위치에 원하는 값을 추가하기

```
lang.insert(1, "c")
```

- 특정한 요소 삭제하기

```
lang.remove("go lang")
```

```
del lang[2]
```

- 리스트 데이터 정렬하기

```
numbers = [2, 1, 4, 3]
print(numbers)

numbers.sort()
print(numbers)
```

- 리스트 데이터 역순으로 정렬하기

```
numbers = [2, 1, 4, 3]
numbers.reverse()
print(numbers)
```

문자열 특수 함수와 리스트

```
python_is_easy = "python is easy"  
python_is_easy.split()
```

SmallProject2

나이를 입력받아서, 나이가 19 이상이면 당신은 성인입니다. 출력하기

```
age = input("나이는?")
```

조건문이 필요합니다.

```
if 조건:  
    실행문1  
    실행문2  
  
if 조건1 and 조건2:  
    실행문  
  
if 조건1 or 조건2:  
    실행문  
  
if not 조건:  
    실행문
```


Comparison Operators

`x == n`

`x != n`

`x < n`

`x > n`

`x <= n`

`x >= n`

if

```
if 현금 > 100000:  
    레스토랑으로 간다
```

```
cash = 120000  
if cash > 100000:  
    print("go to restaurant")
```

SmallProject3

나이를 입력받아서, 나이가 19 이상이면 당신은 성인입니다. 출력하기, 19 이하이면 당신은 청소년입니다. 출력하기

```
age = input("나이는?")
```

else

```
if 조건:
    실행문1
else:
    실행문2
```

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
else:
    print("go to home")
```

SmallProject5

나이를 입력받아서, 나이가 19 이상이면
당신은 성인입니다.

13 이상 19 이하 이면
당신은 청소년입니다.

13 이하 이면
당신은 아동입니다.

출력하기

```
age = input("나이는?")
```

elif

```
if 조건1:
    실행문1
elif 조건2:
    실행문2
elif 조건3:
    실행문3
...
else:
    실행문n
```

elif

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
elif cash > 50000:
    print("go to bobjib")
elif cash > 30000:
    print("go to buffet")
elif cash > 20000:
    print("go to ramen store")
elif cash > 10000:
    print("go to chinese restaurant")
else:
    print("go to home")
```

SmallProject6

나이를 입력받아서, 나이가 19 이상이면
당신은 성인입니다.

13 이상 19 이하 이면
당신은 청소년입니다.

7 이상 13 이하 이면
당신은 아동입니다.

0 이상 7 이하 이면
당신은 유아입니다.

조건문 안에 조건문

```
if 조건1:
    실행문1
else:
    if 조건2:
        실행문2
    else:
        실행문3
```

```
cash = 120000
if cash > 100000:
    print("go to restaurant")
else:
    if cash > 50000:
        print("go to bobjib")
    else:
        print("go to home")
```