

PYTHON TUTORING #5

School of Computing, KAIST & 대덕고등학교

PYTHON OVERVIEW

- ① 변수와 출력
- ② 사칙연산
- ③ 함수
- ④ 조건문
- ⑤ 반복문

변수와 출력

변수는 값을 저장하는 공간

= 기호를 통해서 변수에 값을 저장!

변수 이름 = 변수에 저장할 값

예시

```
a = 3
print(a)

b = "Python3"
print(b)

c = b
print(c)
```

사칙연산 (연산자)

덧셈, 뺄셈, 곱셈, 나눗셈

//는 나눗기의 몫

%는 나눗기의 나머지

문자열에 대해서도 일부 사칙연산 적용!

예시

```
var = 1 + 2
```

```
var = 3 - 6
```

```
var = 2 * 7
```

```
var = 7 / 3
```

```
var = 7 // 3
```

```
var = 7 % 3
```

```
var = "Hello"+"KAIST"
```

```
Var = "Hello" * 3
```

함수 (Function)

$$f(x) = x^2 + x + 1$$

함수의 결과를 **return**을 통해 전달

return이 없는 함수도 가능

반복되는 코드를 줄이는 데 유용

예시

```
def f(x):  
    return x**2 + x + 1  
  
def g(x, y):  
    print(x * y)  
  
x = 3  
y = f(x)  
g(x, y)
```

조건문

if, elif, else + **condition:**
>, <, ==, != 와 같은 **비교 연산자**

위에 있는 if 또는 elif를 만족할 시,
밑에 있는 elif나 else는 **전부 무시**

예시

```
def positive(x):  
    if x > 0:  
        print("positive")  
    elif x < 0:  
        print("negative")  
    else:  
        print("zero")  
  
positive(-2)
```

조건문

if, elif, else + **condition:**
>, <, ==, != 와 같은 **비교 연산자**

위에 있는 if 또는 elif를 만족할 시,
밑에 있는 elif나 else는 **전부 무시**

예시

```
def positive(x):  
    if x > 0:  
        print("positive")  
    elif x >= 0:  
        print("zero")  
    elif True:  
        print("negative")  
  
positive(-2)
```

반복문

for 변수명 in range(횟수):

range(**n**) = [0, 1, 2, ..., **n-1**]

range(1, **n**) = [1, 2, ..., **n-1**]

range(5, **13**, 2) = [5, 7, 9, **11**]

range(**n**, **0**, -1) = [**n**, **n-1**, ..., 2, **1**]

예시

```
for i in range(1, 4):  
    for j in range(i):  
        print(i, j)
```

1 0

2 0

2 1

3 0

3 1

3 2

반복문

`while condition:`

`condition`이 참이면, 반복

예시

```
def gauss(n):  
    sum = 0  
    while n > 0:  
        sum = sum + n  
        n = n - 1  
    return sum  
  
mystery = gauss(100)  
print(mystery)
```

예제

`mult_pos = x*y*z`가 양수인지 반환

`print_prime = n` 이하의 소수를 출력

예시

```
def mult_pos(x, y, z):  
  
def print_prime(n):  
  
print(mult_pos(7,2,4))  
print(mult_pos(1,-3,2))  
print(mult_pos(0,5,11))  
True  
False  
False
```

예제

`mult_pos = x*y*z`가 양수인지 반환

`print_prime = n` 이하의 소수를 출력

예시

```
def mult_pos(x, y, z):
```

```
def print_prime(n):
```

```
print_prime(5)
```

2

3

5

예제

예시

```
def mult_pos(x, y, z):  
    isZero = (x == 0 or y == 0 or z == 0)  
    negativeCount = 0  
    if x < 0:  
        negativeCount += 1  
    if y < 0:  
        negativeCount += 1  
    if z < 0:  
        negativeCount += 1  
    if isZero:  
        return ???  
    return negativeCount % 2 == ?
```

예제

예시

```
def print_prime(n):  
    i = 2  
    while i <= n:  
        if is_prime(i):  
            print(i)  
  
def is_prime(n):  
    i = 2  
    while i < n:  
        if n % i == ?:  
            return ???  
        i = i + 1  
    return ???
```

list REVIEW

list는 하나 이상의 값을 가지는 자료형

대괄호로 감싸져 있으면 list

list 안의 요소는 숫자, string, list
관계 없이 전부 가능

list 사이의 연산도 따로 적용

예시

```
odd = [1, 3, 5, 7, 9]
```

```
empty = []
```

```
word = ["even", "odd"]
```

```
all = [1, 2, "number"]
```

```
print(odd + word)
```

```
print(word * 2)
```

list REVIEW

`list[n]`은 $n+1$ 번째 요소를 의미한다.

`all[0]: 1`

`all[1]: 2`

`all[2]: "number"`

예시

```
all = [1, 2, "number"]
```

```
print(all[0])
```

```
print(all[1])
```

```
print(all[2])
```

```
print(all[len(all)-1])
```

list REVIEW

`list[a:b]`은 $a+1$ 번째 요소부터 b 번째 요소까지를 의미한다.

`all[0]: 1`

`all[1]: 2`

`all[2]: "number"`

예시

```
all = [1, 2, "number"]
```

```
print(all[0])
```

```
print(all[1])
```

```
print(all[2])
```


예제

`input` 함수를 이용하여 문장을 입력 받은 뒤,
문장 중에 중복되는 단어가 있는지 검사하여라.

Hint: 각 단어를 하나의 `list` 안에 넣고,
`i != j and list[i] = list[j]`인 `i, j`를 찾는다.

예제

예시

```
string = input()

l = string.split()          # 띄어쓰기를 기준으로 각 단어를 list에 담는다.

for i in range(?):
    for j in range(?):
        if ??? and ???:
            print(l[i])
            exit()          # 중복되는 문자열을 출력 후 종료
```