



PYTHON

Day5

RECAP

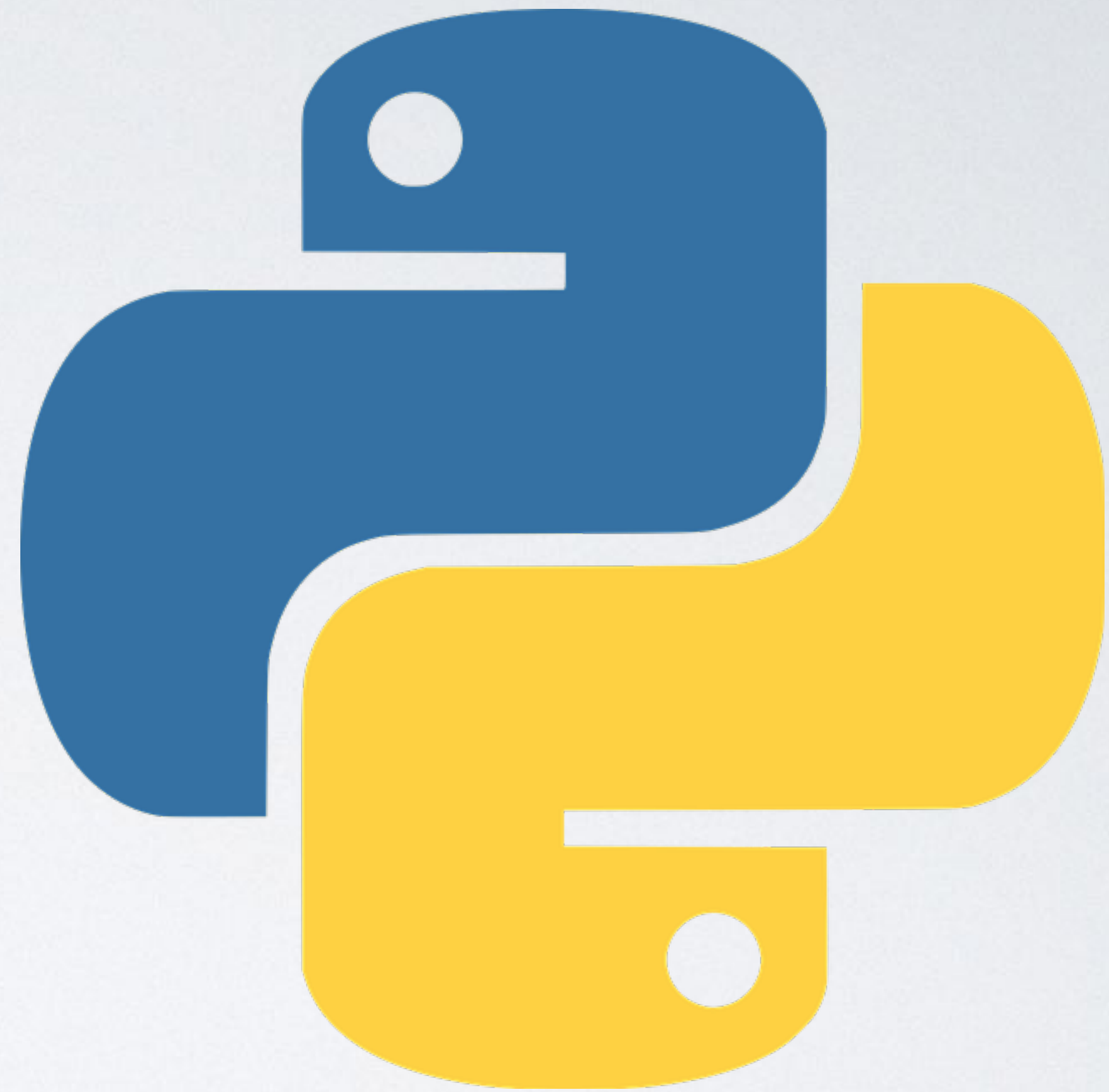
- Conditionals
- Iterations
- Tuple
- Dictionary

INDEX

- function
- File I/O
- List Comprehension

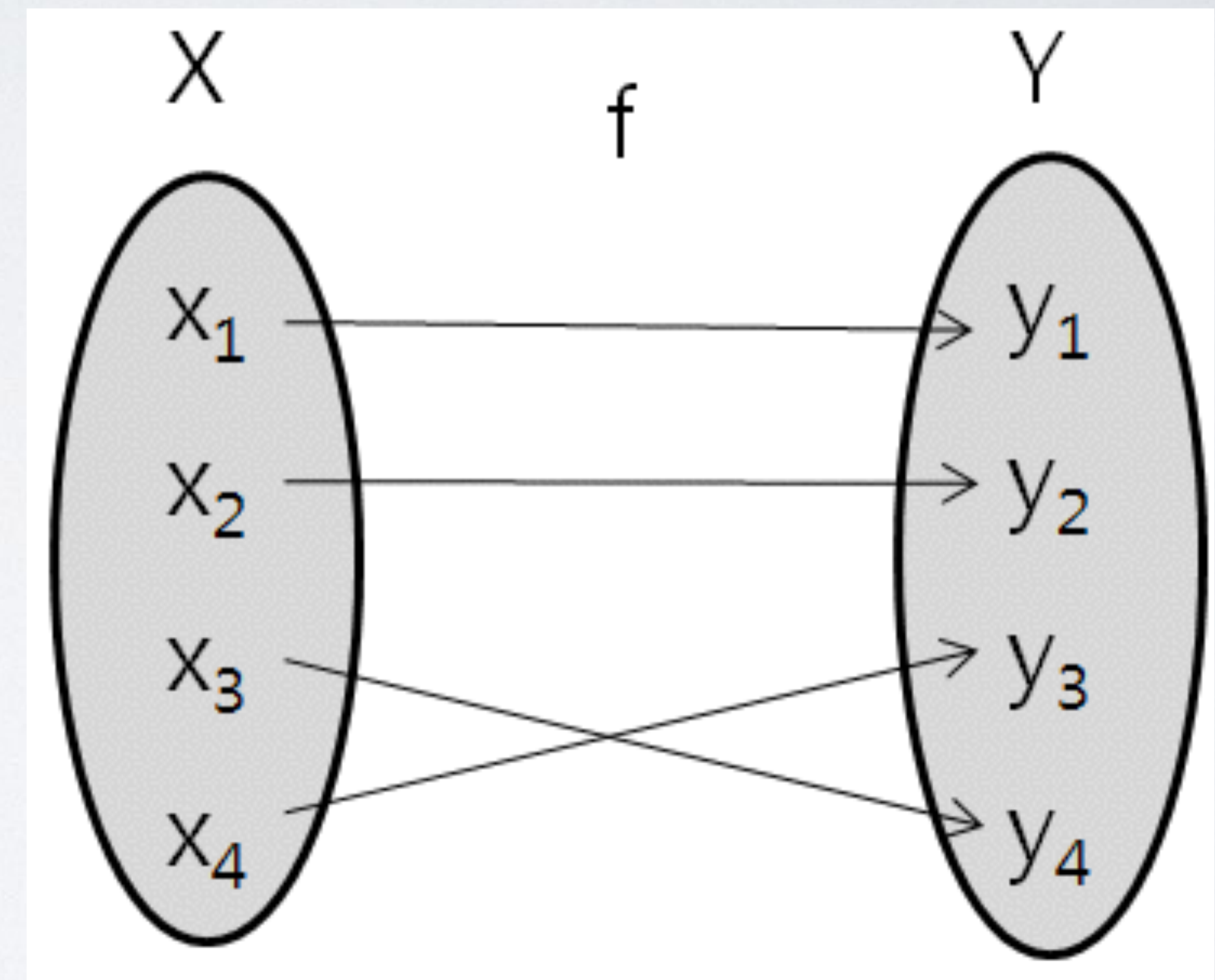
PYTHON

function



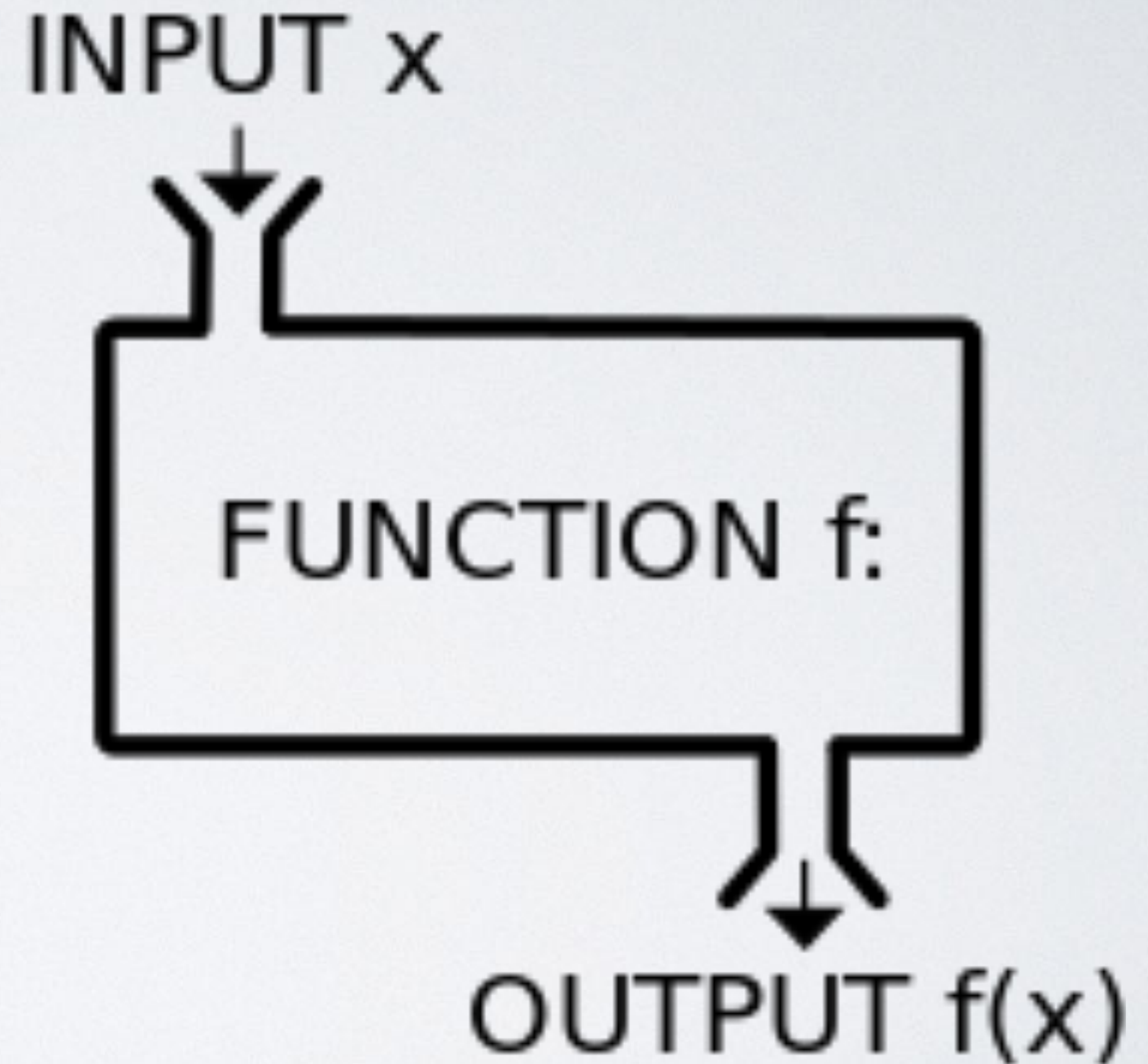
FUNCTION

- [수학] 첫 번째 집합의 임의의 한 원소를 두 번째 집합의 오직 한 원소에 대응시키는 대응 관계
- x : 정의역, y : 공역



FUNCTION

- [컴퓨터과학] 입력값을 내부에서 어떤 처리를 통해 결과값을 출력하는 것



FUNCTION IN PYTHON

```
def function_name(parameter):
```

```
    실행문1
```

```
    실행문2
```

```
    ..
```


POLITE FUNCTION

```
def hello(name):  
    return "Hello, %s." % name
```

```
>>> print(hello("John"))
```


MAKE AWESUM

```
def awe_sum(a,b):  
    result = a + b  
    return result
```

```
>>> x = 2
```

```
>>> y = 3
```

```
>>> awe_sum(x,y)
```

FUNCTION W/O INPUT

```
def print_hello():  
    return "hello"
```

```
>>> result = print_hello()
```

```
>>> print(result)
```


FUNCTION W/O OUTPUT

```
def no_return(a):  
    print("I can print without return whatever %s was." % str(a))
```

FUNCTION WITH MULTIPLE RETURN

```
def multi_return(a):
```

```
    b = a + 1
```

```
    return a,b
```

```
>>> multi_return(10)
```


함수 입력값 초기화하기

```
def say_hello(name="Fool", nick=True):  
    print("Hi, %s." % name)  
    if nick == True:  
        print("But, you are fool")  
    else:  
        print("Oh, you are not fool")
```

```
>>> say_hello()  
>>> say_hello("John")  
>>> say_hello("John", False)
```

함수를 사용할 때 주의할 점!!

함수에서 사용된 변수는 함수 내에서만 유효합니다.

```
a = "hello"  
def glob_test(a):  
    a = a + "world"  
    return a
```

```
a = "hello"  
def glob_test(x):  
    x = x + "world"  
    return x
```


함수의 결과값을 함수 밖에서도 쓰고 싶다면..

(1) return 값을 원래 변수에 다시 할당하기

```
a = "hello"
```

```
def glob_test(x):  
    x = x + "world"  
    return x
```

```
>>> a = glob_test(a)
```

함수의 결과값을 함수 밖에서도 쓰고 싶다면..

(2) global 사용하기

```
a = "hello"
```

```
def glob_test(a):  
    global a  
    a = a + "world"  
    return a
```

```
>>> glob_test(a)
```


함수의 결과값을 함수 밖에서도 쓰고 싶다면..

global(전역변수 선언)은 Python 함수의 본질(코드의 재사용)을 해칠 수 있으므로 정말 필요할 때만 사용하셔야 합니다.

MINI PROJECT

* Leap year

- 4로 나뉘어 떨어지면 윤년
- 100으로 나뉘어 떨어지면 평년
- 400으로 나뉘어 떨어지면 윤년

** 윤년을 계산하는 함수와 사용자에게 물어본 결과를 판단해
윤년과 평년을 출력하는 함수를 따로 만들어 주세요

ex) def is_leap(year):

...

def asker():

...

MINI PROJECT

* sample output

Is leap? 2012

Leap year.

Is leap? 2100

Common year.

Is leap? 2400

Leap year.

MINI PROJECT - ANSWER

```
def is_leap(year):  
    if year % 4 == 0:  
        if year % 400 == 0:  
            leap = True  
        elif year % 100 == 0:  
            leap = False  
        else:  
            leap = True  
    else:  
        leap = False  
    return leap
```

```
def asker():  
    yr = int(input("is leap? "))  
    if is_leap(yr) == True:  
        print("Leap year")  
    else:  
        print("Common year")  
  
asker()
```


PYTHON

File I/O



FILE I/O

- Python에서 파일을 생성, 수정, 삭제 할 수 있도록 도와주는 기능
- `open(파일이름, 모드)`

FILE I/O

```
f = open("exam.txt", "w")  
f.close()
```

* 모드

r - 읽기모드

w - 쓰기모드

a - 추가모드(파일의 마지막에 새로운 내용을 추가)

FILE I/O - 새 파일 만들기

```
f = open("newfile.txt", "w")  
f.close()
```


FILE I/O - 파일에 내용 추가하기

```
f = open("newfile.txt", "a")  
for i in range(1, 10+1):  
    text = "line %d. \n" % i  
    f.write(text)  
f.close()
```

FILE I/O - 파일 내용 불러오기

```
f = open("newfile.txt", "r")  
text = f.readline()  
print(text)  
f.close()
```


FILE I/O - 파일 내용 불러오기

```
f = open("newfile.txt", "r")
while True:
    text = f.readline()
    if not text:
        break
    print(text)
f.close()
```

FILE I/O - 파일 내용 “한번에” 불러오기

```
f = open("newfile.txt", "r")  
texts = f.readlines()  
for text in texts:  
    print(text)  
f.close()
```


FILE I/O - 파일 내용 추가하기

```
f = open("newfile.txt", "a")  
for i in range(11, 20+1):  
    text = "New line %d \n" % i  
    f.write(text)  
f.close()
```

FILE I/O - WITHOUT F.CLOSE()

```
with open("foo.txt", "w") as f:  
    f.write("foo is dummy text")
```


FILE FORMAT

.txt : 일반 텍스트 파일

.csv : Comma Separated Values

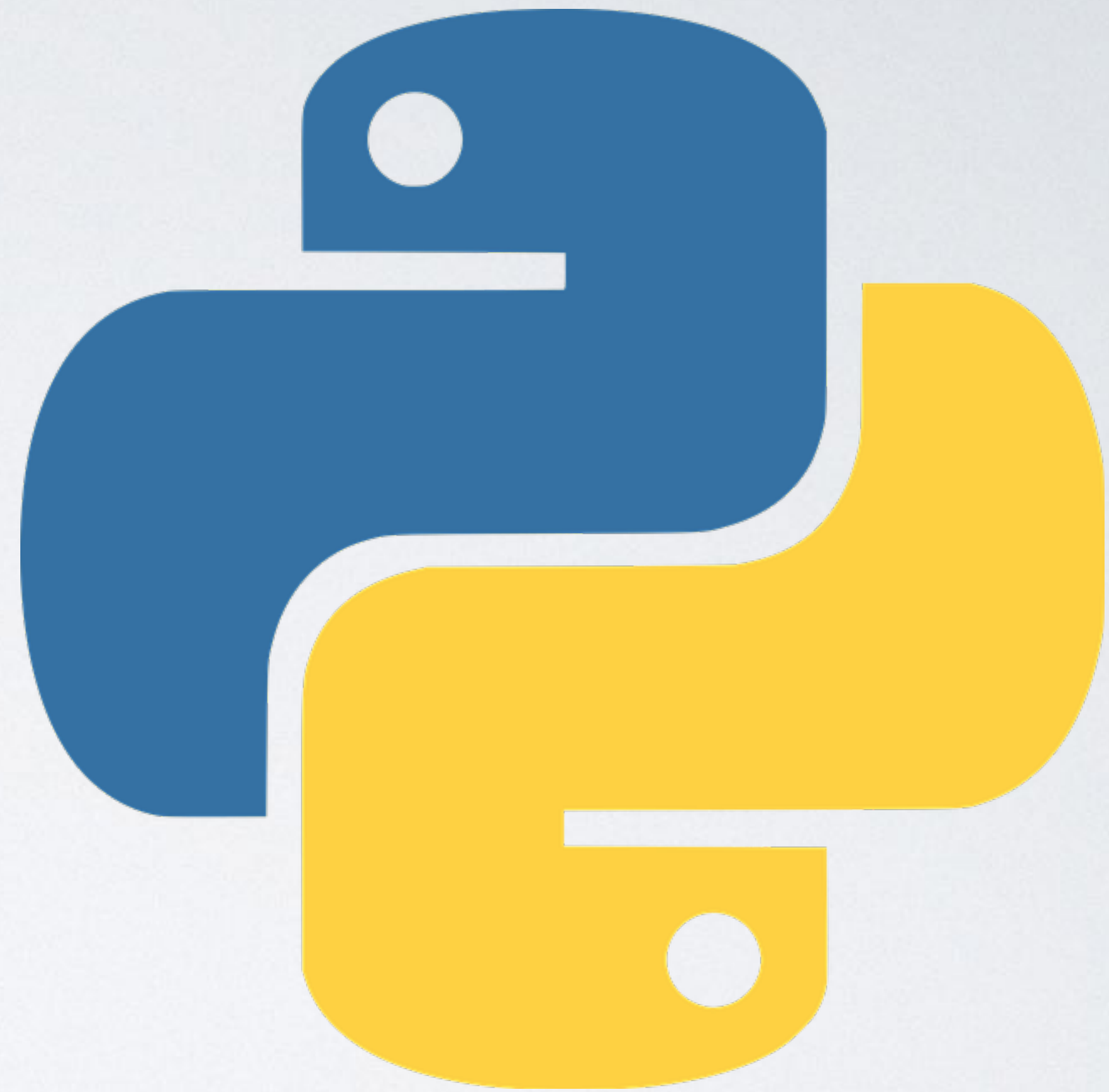
.tsv : Tab Separated Values

.xls : Microsoft Excel

.xlsx : Microsoft Excel(Extended)

PYTHON

List Comprehension



LIST COMPREHENSION

존재하는 리스트를 활용하여 새로운 리스트를 생성하는 방법

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []
```

```
for i in old_list:
```

```
    doubled_list.append(i * 2)
```


LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []
```

```
doubled_list = []
```

```
for i in old_list:
```

```
    doubled_list.append(i * 2)
```

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []
```

```
doubled_list = [i * 2]
```

```
for i in old_list:
```

```
    doubled_list.append(i * 2)
```


LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []
```

```
for i in old_list:
```

```
    doubled_list.append(i * 2)
```

```
doubled_list = [i * 2 for i in old_list]
```

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []
```

```
for i in old_list:
```

```
    if i % 2 == 0:
```

```
        doubled_list.append(i * 2)
```


LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []
```

```
for i in old_list:
```

```
    if i % 2 == 0:
```

```
        doubled_list.append(i * 2)
```

```
doubled_list = [i * 2]
```

LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []
```

```
for i in old_list:
```

```
    if i % 2 == 0:
```

```
        doubled_list.append(i * 2)
```

```
doubled_list = [i * 2 for i in old_list]
```


LIST COMPREHENSION

```
old_list = [1, 2, 3, 4, 5]
```

```
doubled_list = []
```

```
for i in old_list:
```

```
    if i % 2 == 0:
```

```
        doubled_list.append(i * 2)
```

```
doubled_list = [i * 2 for i in old_list if i % 2 == 0]
```

MINI PROJECT

with pair programming

MINI PROJECT

1. 1부터 100까지 리스트를 생성합니다.

2. 조건문을 활용하여

3의 배수일 경우, “fizz”

5의 배수일 경우, “buzz”

15의 배수일 경우, “fizzbuzz”

나머지 경우, 그 숫자를 저장하는 프로그램을 작성하세요.

MINI PROJECT

* Sample output

[1, 2, fizz, 4, buzz, ..., fizzbuzz, ..., buzz]

MINI PROJECT - ANSWER

```
num_list = []  
for i in range(1, 100+1):  
    num_list.append(i)
```

```
fizz_list = ["fizzbuzz" if i%15==0 else "fizz" if i%3==0 else "buzz" if  
i%5==0 else i for i in num_list]
```

NEXT CLASS

- Excel with Python
- Exception
- Class