



PYTHON

Day4

RECAP

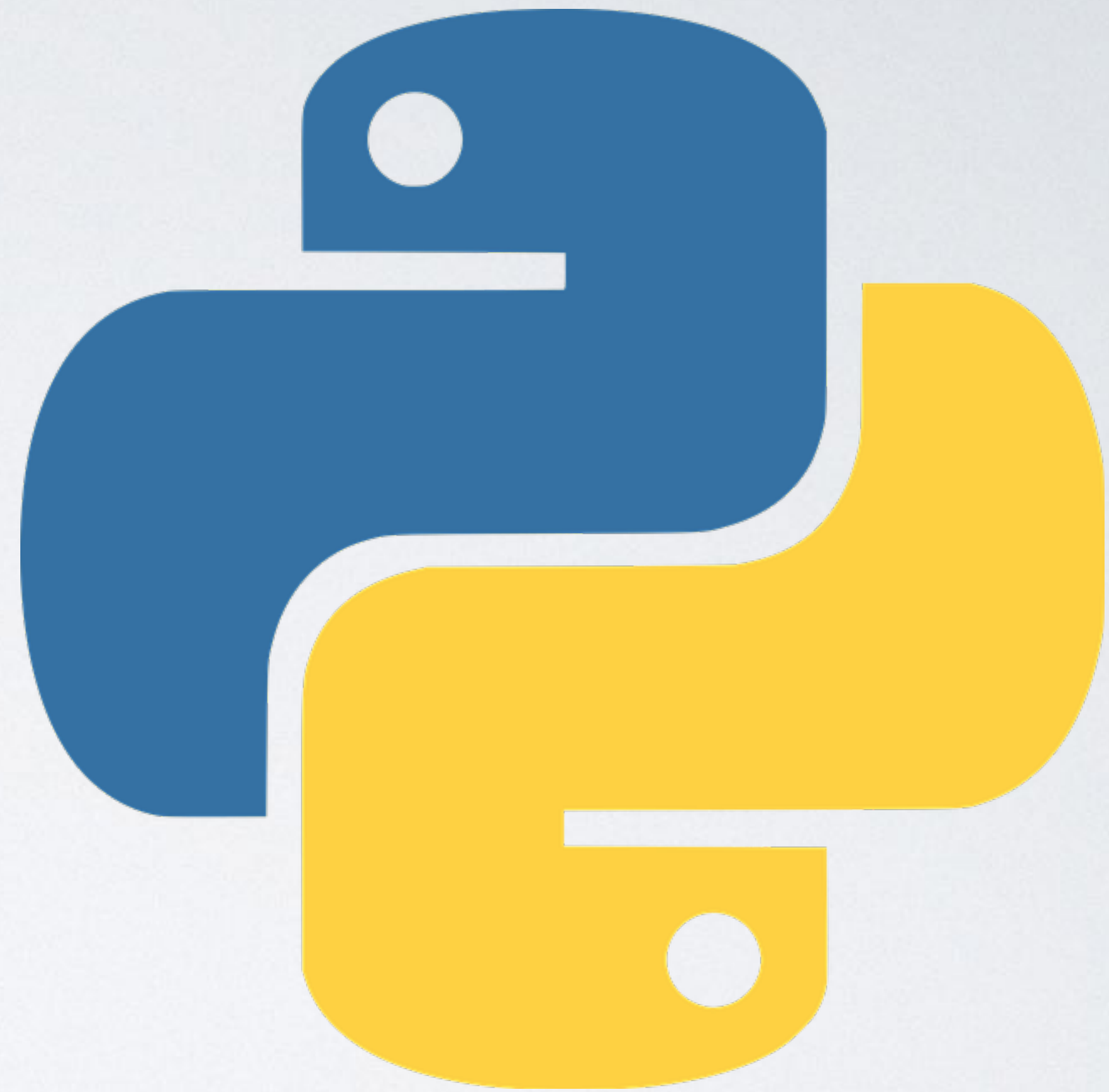
- List
- Conditionals
- Iterations

INDEX

- Conditionals
- Iterations
- Tuple
- Dictionary

PYTHON

Conditionals - if, else, elif



CONDITIONAL STATEMENTS

- 돈이 100000원 이상인가?
- YES: 레스토랑으로 간다
- NO: 편의점으로 간다

CONDITIONAL STATEMENTS

- 프로그래머가 명시한 조건이 참인지 거짓인지에 따라 다른 계산을 수행하는 것, 또는 그런 문법
- if - 만약에..

IF - PYTHON의 조건문

```
if 현금 > 100000:
```

```
    레스토랑으로 간다
```

그런데.. 거짓인 경우는 어떡하지?

ELSE

if 현금 > 100000:

레스토랑으로 간다

else:

편의점으로 간다

여러 조건을 두고 싶은데???

IF IN THE ELSE

if 현금 > 100000:

레스토랑으로 간다

else:

if 현금 > 50000:

근처 밥집으로 간다

else:

편의점으로 간다

조건에 조건이 반복되면
TAB 많이 쳐야 하네!!

ELIF

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

```
elif 현금 > 50000:  
    근처 밥집으로 간다
```

```
elif 현금 > 30000:  
    라멘모토로 간다
```

```
elif 현금 > 10000:  
    김밥천국을 간다
```

```
else:  
    편의점으로 간다
```

PRACTICE: IF

```
>>> money = 110000  
>>> if money > 100000:  
...     print("go to restaurant")  
...
```

go to restaurant

PRACTICE: IF ELSE

```
>>> money = 80000  
>>> if money > 100000:  
...     print("go to restaurant")  
... else:  
...     print("go to CVS")  
...
```

go to CVS

PRACTICE: IF IN ELSE

```
>>> if money > 100000:  
...     print("go to restaurant")  
... else:  
...     if money > 50000:  
...         print("go to bobjib")  
...     else:  
...         print("go to CVS")  
...  
go to bobjib
```


PRACTICE

- 어떤 숫자가 짝수인지 판단하는 조건문 작성하기

```
num = 3
```

```
if num % 2 == 0:
```

```
    print("%d is even" % num)
```

PRACTICE

- 어떤 숫자가 짝수인지 홀수인지 판단하는 조건문 작성하기

```
num = 3
```

```
if num % 2 == 0:
```

```
    print("%d is even" % num)
```

```
else:
```

```
    print("%d is odd" % num)
```


PRACTICE

- 어떤 숫자가 3의 배수인지 짝수인지 홀수인지 판단하는 조건문 작성하기

```
num = 3
```

```
if num % 3 == 0:
```

```
    print("%d is odd and multiples of 3" % num)
```

```
elif num % 2 == 0:
```

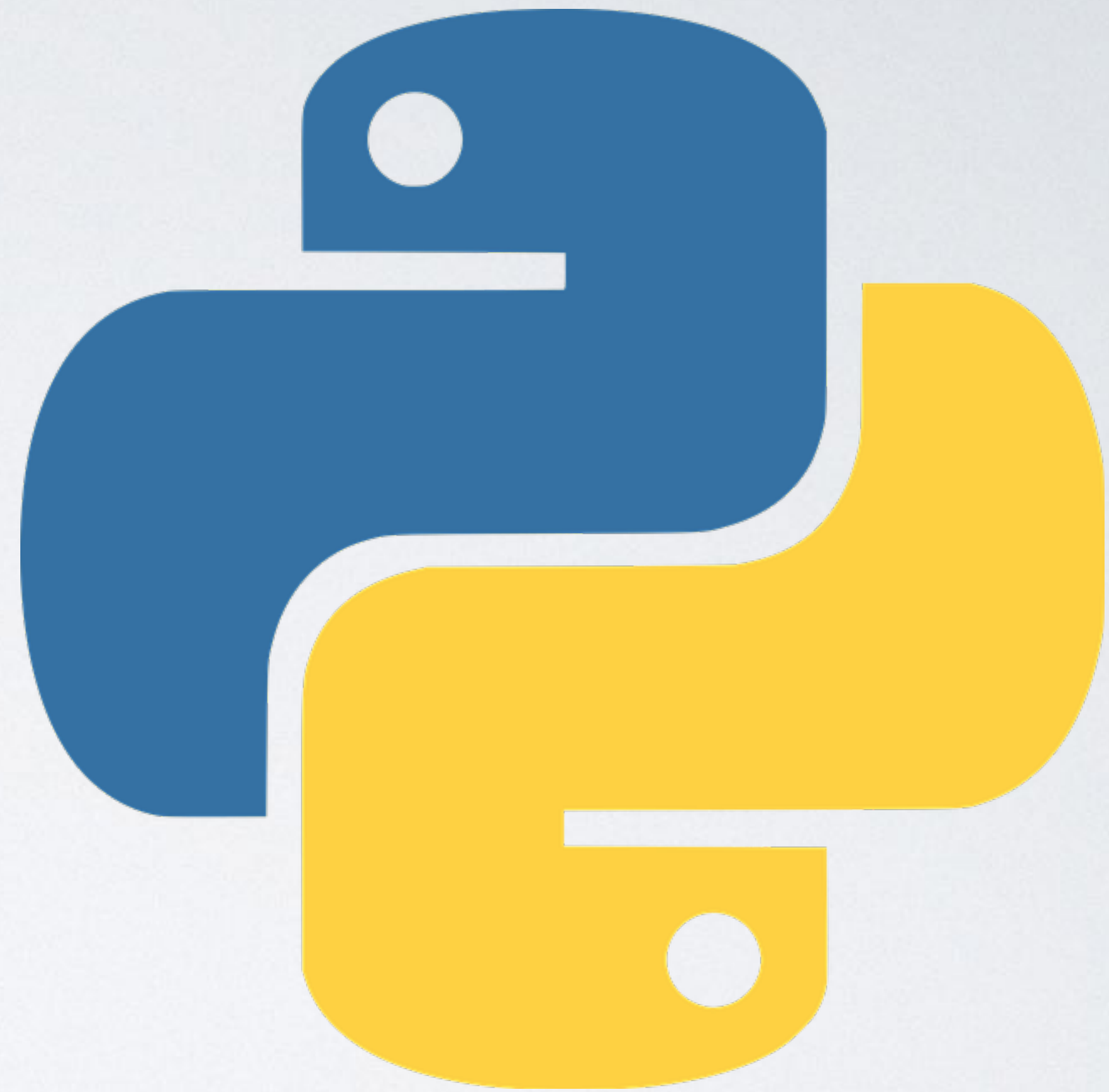
```
    print("%d is even" % num)
```

```
else:
```

```
    print("%d is odd" % num)
```

PYTHON

Iterations - for



ITERATIONS

- 어떤 문장을 반복하고 싶을 때 사용하는 구문
- for, while

ITERATIONS - FOR

문자열 혹은 리스트 요소 하나하나를 변수에 선언하여 반복

for 변수 in (문자열 또는 리스트):

실행문

ITERATIONS - FOR

```
>>> for i in range(1, 10+1):  
...     print(i)
```

```
...
```

```
>>> for i in ["python", "c", "java"]:  
...     print(i)
```

```
...
```

ITERATIONS - MAKE AWESUM

```
>>> result = 0
>>> for j in range(1, 10+1):
...     result = result + j
...     print(result)
...
>>> result
```


PRACTICE

- num이 1에서 10까지 증가하면서 출력문을 반복 수행하기

```
for num in range(1, 10+1):  
    print("hello for the %d time(s)" % num)
```

반복문과 조건문 함께 사용하기

- num이 1에서 10까지 증가하면서 앞서 만든 조건(3의 배수, 짝수, 홀수)을 적용해보기

```
for num in range(1, 10+1):
```

```
    if num % 3 == 0:
```

```
        print("%d is odd and multiples of 3" % num)
```

```
    elif num % 2 == 0:
```

```
        print("%d is even" % num)
```

```
    else:
```

```
        print("%d is odd" % num)
```


PYTHON

Iterations - while



ITERATIONS - WHILE

조건을 만족하지 않을 때 까지 반복

while 조건:

실행문

ITERATIONS - WHILE

```
>>> animal = ""  
>>> while animal != "dog":  
...     animal = input("What is your favorite animal? ")  
...     print("Oh, you like %s." % animal)  
...
```

What is your favorite animal? cat
Oh, you like cat.

What is your favorite animal? lion
Oh, you like lion.

What is your favorite animal? dog
Oh, you like dog.

ITERATIONS - INFINITE LOOP

```
>>> while 1:  
...     print("HAHA! Python is easy!")  
....
```


MINI PROJECT

with pair programming

MINI PROJECT

1. 1부터 100까지 반복을 합니다.

2. 조건문을 활용하여

3의 배수일 경우, “fizz”

5의 배수일 경우, “buzz”

15의 배수일 경우, “fizzbuzz”

나머지 경우, 그 숫자를 출력하는 프로그램을 작성하세요.

MINI PROJECT

* Sample output

1
2
fizz
4
buzz
...
fizzbuzz
...
100

MINI PROJECT - ANSWER

```
for i in range(1, 100+1):  
    if i % 15 == 0:  
        print("fizzbuzz")  
    elif i % 3 == 0:  
        print("fizz")  
    elif i % 5 == 0:  
        print("buzz")  
    else:  
        print(i)
```


FIZZBUZZ WITH WHILE

```
i = 1  
while i < 16:  
    print(i)  
    i = i + 1
```

FIZZBUZZ WITH WHILE

```
i = 1
while i < 16:
    if i % 15 == 0:
        print("fizzbuzz")
    elif i % 3 == 0:
        print("fizz")
    elif i % 5 == 0:
        print("buzz")
    else:
        print(i)
    i = i + 1
```


PYTHON

Tuple



TUPLE

- Tuple 은 소괄호를 이용해 선언합니다.

```
>>> tuple1 = (1,2,3,4)
```

- Tuple은 삭제나 추가가 불가능합니다.(값이 고정)

```
>>> del tuple1[1]
```

```
>>> tuple1[1] = "c"
```


TUPLE

- Tuple 끼리 더하거나 반복하는 것은 가능합니다.

```
>>> tuple2 = (5,6)
```

```
>>> print(tuple1 + tuple2)
```

```
>>> print(tuple1 * 5)
```

TUPLE

- Tuple은 변수끼리 값을 바꿀 때 사용할 수 있습니다.

w/o tuple

```
>>> x = 1
```

```
>>> y = 2
```

```
>>> x = y
```

```
>>> y = x ( impossible)
```


TUPLE

- Tuple은 변수끼리 값을 바꿀 때 사용할 수 있습니다.

w/o tuple

```
>>> x = 1
```

```
>>> y = 2
```

```
>>> temp = x
```

```
>>> x = y
```

```
>>> y = temp
```

TUPLE

- Tuple은 변수끼리 값을 바꿀 때 사용할 수 있습니다.

w/ tuple

```
>>> x = 1
```

```
>>> y = 2
```

```
>>> (x, y) = (y, x)
```


PYTHON

Dictionary



DICTIONARY

- dictionary는 어떤 값에 라벨이 필요할 때 사용합니다.
- 중괄호 { } 를 이용해 dictionary를 선언할 수 있습니다.

```
>>> dict1 = {}
```

```
>>> print(dict1)
```


DICTIONARY

- dictionary는 key 와 value로 이루어져 있으며, key는 그 값의 카테고리
를, value는 실제 값을 의미합니다.

```
>>> dict1 = {"name": "miraen"}
```

```
>>> print(dict1)
```

```
>>> dict1 = {"korean": 90, "math": 60, "science": [90, 80, 40, 70]}
```

```
>>> dict1["english"] = "pass"
```

DICTIONARY

- 값을 삭제하기 위해 del을 사용합니다.

```
>>> del dict1["math"]
```

```
>>> print(dict1)
```

- key를 이용해 value를 출력할 수 있습니다.

```
>>> print(dict1["korean"])
```


DICTIONARY

- dictionary가 가지고 있는 key를 출력할 땐

```
>>> print(dict1.keys())
```

- dictionary가 가지고 있는 value를 출력할 땐

```
>>> print(dict1.values())
```

- dictionary가 가지고 있는 key와 value 모두를 출력할 땐

```
>>> print(dict1.items())
```

NEXT CLASS

- function
- keyword arguments
- file I/O