

2025년 서울 민간기업 맞춤형 매력일자리 사업

# 파이썬 문법

- 2025.05.02 -

# Python 기초

# 파이썬 언어 - 출력문

1 출력문 : 문자열  
`print("안녕하세요.")`

2 출력문 : 숫자  
`print(5+4)`



The screenshot shows a code editor window titled 'helloworld.py' with a single line of Python code: `print("Hello world!")`. Below the editor is a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The terminal shows the command `/opt/homebrew/bin/python3 /Users/matacross.ai/Downloads/ai_exam/___IITP_광주교육/helloworld.py` being executed, resulting in the output 'Hello world!'. The terminal also shows the current directory as `~/Downloads/ai_exam/___IITP_광주교육`.

# 파이썬 언어 - 숫자 출력

**2** 출력문 : 숫자  
`print(5+4)`

Operation	Symbol	Example
Addition	+	$1 + 2 = 3$
Subtraction	-	$5 - 4 = 1$
Multiplication	*	$2 * 4 = 8$
Division	/	$6 / 3 = 2$
Exponent	**	$3 ** 2 = 9$

연산관련 명령어



# 파이썬 언어 – 숫자 출력

## 3 #은 주석문

# 5+4=9 를 위한 출력문입니다.

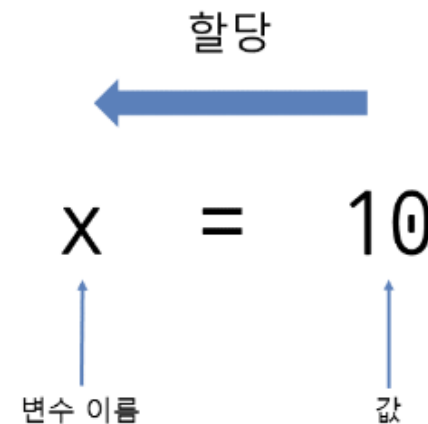
```
print(5+4)
```

# 파이썬 언어 – 변수(Variables)

① 변수 : 특정 위치(장소)에 어떤 값을 넣을 때 사용

```
test_var = "테스트"  
print(test_var)
```

```
testVar = "테스트2"  
print(testVar)
```



# 파이썬 언어 – 변수(Variables)

## 2 변수 선언과 할당

변수 이름 { age = 10  
sold\_out = False

변수 값

할당 연산자

<변수 선언과 할당>

age = 10

print(age)  
print(type(age))

# 파이썬 언어 – 변수(Variables)

## 3 변수에 새로운 값의 할당

```
age = 10  
print(age)
```

```
age = 2  
print(age)
```



# 파이썬 언어 – 데이터 타입

## 4 다양한 자료형태

파이썬의 5가지 자료 구조, 변수 유형 (Python's 5 Data/Variable Types)				
1 수 (Numbers)	2 문자열 (String)	3 리스트 (List)	4 튜플 (Tuple)	5 사전 (Dictionary)
정수(Int) 100	'I Love You'	['abc', 123]	('abc', 123)	{'name': 'R Friend', 'region': 'Seoul, Korea', 'phone': '02-123-3456'}
부동소수형(float) 12.345	- A sequence set of characters - Enclosed by '', or ""	- 1 dimensional sequence of different data type objects - <b>Can be updated</b> - Enclosed by []	- 1 dimensional sequence of different data type objects - <b>Can NOT be updated</b> - Enclosed by ()	- Hash table type - Associative array - Key-value pairs - Enclosed by {}
복소수(complex) 3.45j				

R분석과 프로그래밍 (<http://rfriend.tistory.com>)

# 파이썬 언어 – 데이터 타입

## 1 데이터의 형태 - 숫자

연산이 가능한 형태의 데이터 타입(정수) : integer

```
x = 14  
print(x)  
print(type(x))
```

# 파이썬 언어 – 데이터 타입

## 1 데이터의 형태 - 숫자

연산이 가능한 형태의 데이터 타입(실수) : float

```
pi = 3.14  
print(pi)  
print(type(pi))
```

# 파이썬 언어 – 데이터 타입

## 1 데이터의 형태 – 숫자(실수)

숫자 함수 사용하기

```
pi = 3.1415926534  
round_pi = round(pi, 5)  
print(round_pi)  
print(type(round_pi))
```

실수를 사용하는 방법

```
y_int = 1  
y_float = 1.  
print(y_int)  
print(type(y_int))  
print(y_float)  
print(type(y_float))
```

# 파이썬 언어 – 데이터 타입

## 1 데이터의 형태 - 문자열

문자열은 `'''`, `""` 둘 중 하나로 감싼다.

```
print('테스트')  
print('너비')  
print('abc')  
print('데이터' * 3)
```

# 파이썬 언어 – 데이터 타입

## 1 데이터의 형태 - 문자열

문자열의 길이를 구하는 함수 : len

```
w = "test, 1234"  
print(w)  
print(type(w))  
print(len(w))
```

# 파이썬 언어 – 데이터 타입

## 1 데이터의 형태 – 문자열을 숫자형으로 변환

문자열을 float으로 변환

```
f_str = "1.234"  
print(f_str)  
print(type(f_str))  
f_float = float(f_str)  
print(f_float)  
print(type(f_float))
```



# 파이썬 언어 – 데이터 타입

## 1 데이터의 형태 – 문자열 합치기

### 문자열 합치기

```
str_a = "안녕하세요."  
str_b = "파이썬 입니다."
```

```
print(str_a + str_b)
```

# 파이썬 언어 – 데이터 타입

## 1 데이터의 형태 – bool형 (true / false)

참(true) 또는 거짓(false)로 구분되는 데이터형태

```
print(5 > 10)
```

```
print(5 < 10)
```

```
print(True)
```

```
print(not True)
```

# 숫자처리함수(내장함수)

# 파이썬 언어 – 숫자처리함수

파이썬에서는 기본적으로 다음과 같은 여러 숫자 처리 함수를 제공합니다.

함수 이름	의미	예제
abs	절대값	abs(-5)
pow	제곱	pow(4, 2)
max	가장 큰 값	max(5, 12)
min	가장 작은 값	min(5, 12)
round	반올림	round(3.14)

## 1 숫자 처리 함수

```
print(abs(-5)) # -5 의 절대값 = 5
print(pow(4, 2)) # 4의 2제곱 = 4 * 4 = 16
print(max(5, 12)) # 5 와 12 중 큰 값 = 12
print(min(5, 12)) # 5 와 12 중 작은 값 = 5
print(round(3.14)) # 3.14 의 반올림 = 3
print(round(4.99)) # 4.99 의 반올림 = 5
```

# 입력함수 : input

# 파이썬 언어 – 입력함수

## 1 데이터 입력함수

```
X = input("숫자를 입력하세요 :")
```

```
Print(x)
```

# 랜덤함수(random 모듈) 많이 사용되는 함수



# 파이썬 언어 – random

## 1 랜덤함수 사용하기( 0 – 1 사이 )

```
# random 모듈의 모든 기능을 사용  
from random import *
```

```
print(random())  
print(random())  
print(random())
```

```
0.352106189305487  
0.8366168721289386  
0.27360520189531656  
(base) ~$ python3 random.py
```

# 파이썬 언어 – randrange, randint

② `randrange(시작, 끝)` 끝 숫자 미포함

```
print(randrange(1, 45))
```

③ `randint(시작, 끝)` 끝 숫자 포함

```
print(randint(1, 45))
```

# 기초 자료형, 연산자

# 파이썬 언어 – 자료형

## 1 데이터의 형태 – bool형 (true / false)

참(true) 또는 거짓(false)로 구분되는 데이터형태

```
print(5 > 10)
```

```
print(5 < 10)
```

```
print(True)
```

```
print(not True)
```

# 파이썬 언어 – 문자열 사용방법(len)

## 1 문자열의 길이를 구하는 방법

```
str = "hello world"  
strLen = len(str)
```

```
print(f'{str}의 길이는 {strLen}자 입니다.')
```

# 파이썬 언어 – 변수명 규칙

## 2 변수명 규칙(naming convention)

### 변수명 규칙

1. 변수명에는 소문자(a-z), 대문자(A-Z), 숫자(0-9) 언더바(\_)만 가능
2. 변수명은 대소문자를 구분한다.
3. 변수명은 숫자로 시작할 수 없다.
4. 키워드(예약어)는 사용할 수 없다.

# 파이썬 언어 - 주석문을 쓰는 2가지 방식

## 4 주석문

1. # : 한줄을 쓸때
2. """ ~ """ : 여러줄을 한꺼번에 쓸때



# 파이썬 언어 – 추가 연산자 알아보기

## 4 나누기 연산자

% 연산자

```
print(10 % 3)
```

## 5 나누기에서 몫의 값이 정수의 값이 필요한 경우

// 연산자

```
print(10 // 3)
```

```
Print(10 / 3)
```

# 파이썬 언어 – 형변환 연산자

## 3 형변환 연산자

1. 문자를 숫자로 `int()`, `float()`
2. 숫자를 문자로 `str()`

<주의사항>

```
print( int(3.5) )
```

# 결과 : 3

```
print(float("3.5"))
```

```
print(float(3))
```

# 파이썬 언어 – 비교 연산자

## 1 비교 연산자

비교연산자는 기존의 수학적 기호와 동일하다.

```
print(10 > 3)
```

```
print(4 >= 7)
```

```
print(10 < 3)
```

```
print(5 <= 5)
```

```
print(3 == 3)
```

```
print(4 == 2)
```

```
print(1 != 3)
```

Operator	Description	Example
==	값이 동일하다	(a == b) → false
!=	값이 동일하지 않다	(a != b) → true
>	왼쪽 값이 오른쪽 값보다 크다	(a > b) → false
<	왼쪽 값이 오른쪽 값보다 작다	(a < b) → true
>=	왼쪽 값이 오른쪽 값보다 크거나 동일하다	(a >= b) → false
<=	왼쪽 값이 오른쪽 값보다 작거나 동일하다	(a <= b) → true

# 파이썬 언어 – 논리 연산자

## 1 논리 연산자

논리연산자는 수식, 조건등이 참인지 거짓인지 판단

```
print((3 > 0) and (3 > 5)) # false
```

```
print((3 > 0) or (3 > 5)) # true
```

```
print(not(1 != 3)) # false
```

Operator	Description	Example
and	논리 AND 연산. 둘다 참일때만 참	(a and b) = False
or	논리 OR 연산. 둘 중 하나만 참이어도 참	(a or b) = True
not	논리 NOT 연산. 논리 상태를 반전	not(a and b) = True

# 파이썬 언어 – 함수 연산자

## 1 함수를 이용한 연산자

논리연산자는 수식, 조건등이 참인지 거짓인지 판단

```
print((3 > 0) and (3 > 5)) # false
```

```
print((3 > 0) or (3 > 5)) # true
```

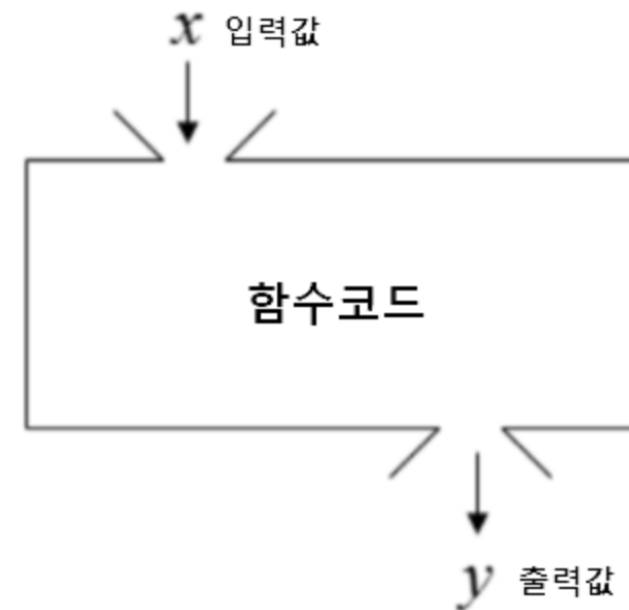
```
print(not(1 != 3)) # false
```

# 함수(Function)

# 파이썬 언어 – 함수(Function)

- 1 X(입력값)을 넣으면 계산에 의하여 y(출력값)을 처리  
똑같은 명령을 반복적으로 수행하는  
경우

```
def 함수이름 (매개변수):  
    <수행할 문장>  
    ...  
    return 리턴값
```





# 파이썬 언어 – 함수(Function)

## 1 함수 사용하기

```
def userCard(name, age, use_lang):  
    print("=" * 20)  
    print("이름 : ", name)  
    print("나이 : ", age)  
    print("언어 : ", use_lang)  
    print("=" * 20)
```

```
useCard("홍길동", 22, "파이썬")
```

```
=====
이름   :   홍길동
나이  :   22
언어   :   파이썬
=====
```

# 파이썬 언어 – 함수(Function)

## 2 함수 사용하기2

```
def userCard(name, age=20, use_lang="파이썬"):
    print("=" * 20)
    print("이름 : ", name)
    print("나이 : ", age)
    print("언어 : ", use_lang)
    print("=" * 20)
```

```
useCard("홍길동")
```

```
=====
이름   :   홍길동
나이  :   22
언어   :   파이썬
=====
```

# 파이썬 언어 – 입력(input)

## 2 키보드의 입력정보 입력 : input

```
def userCard(name, age, use_lang):  
    print("=" * 20)  
    print("이름 : ", name)  
    print("나이 : ", age)  
    print("언어 : ", use_lang)  
    print("=" * 20)
```

```
inputName = input("이름 : ")  
inputAge = input("나이 : ")  
inputLan = input("언어 : ")
```

```
userCard(inputName, inputAge, inputLan)
```

```
이름 : 홍길동  
나이 : 25  
언어 : 파이썬
```

```
=====
```

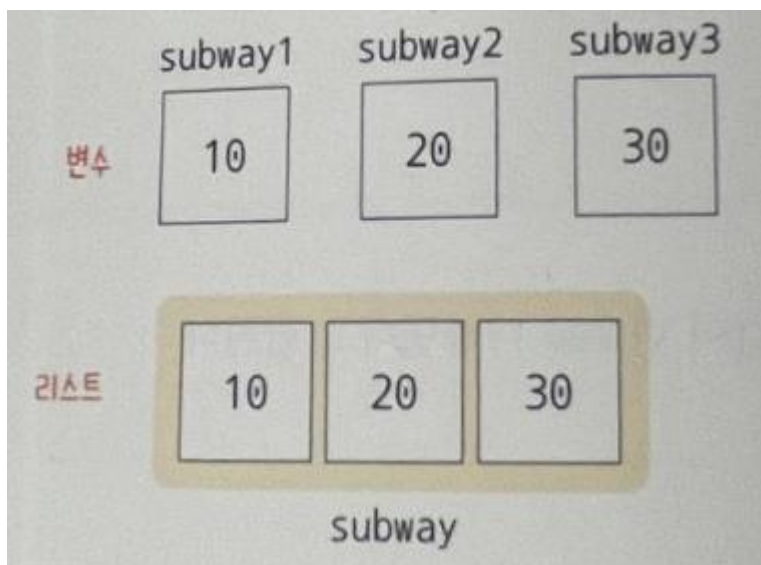
```
이름 : 홍길동  
나이 : 25  
언어 : 파이썬
```

```
=====
```

# 자료형(리스트)

# 파이썬 언어 – 자료형(리스트)

## 1 비슷한 형태의 변수를 하나로 묶어놓은 집합



Subway1 = 10

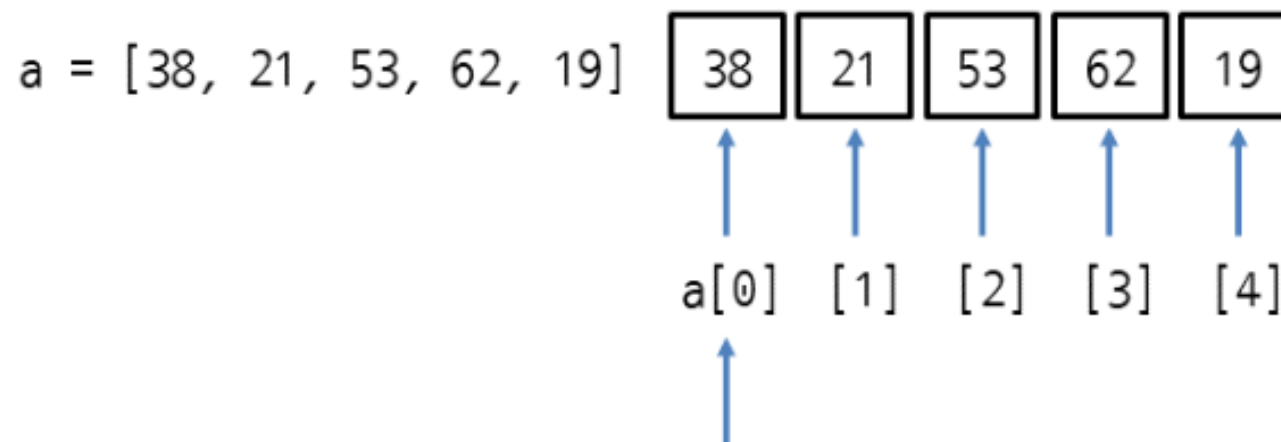
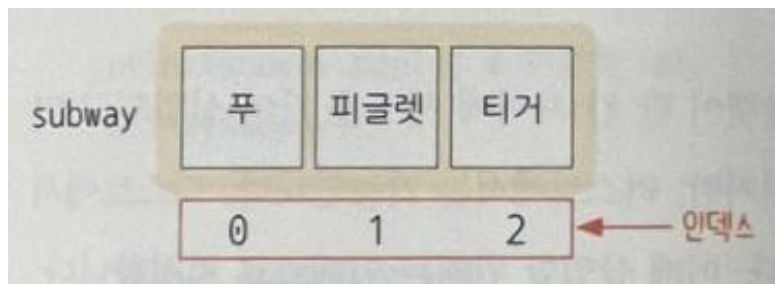
Subway2 = 20

Subway3 = 30

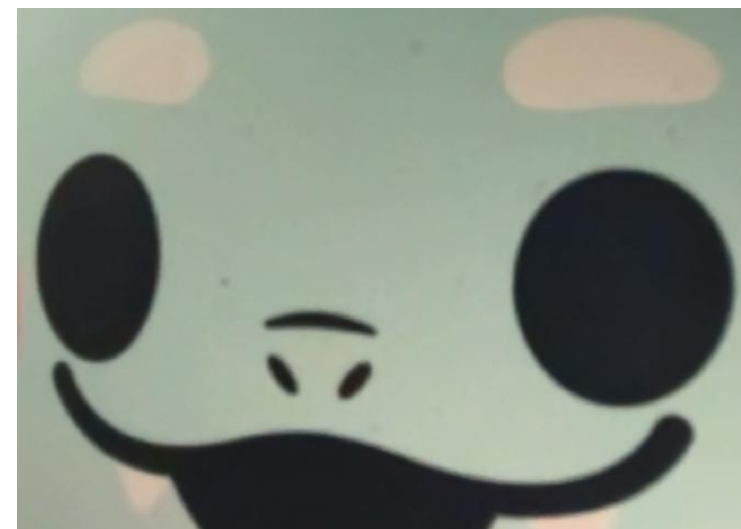
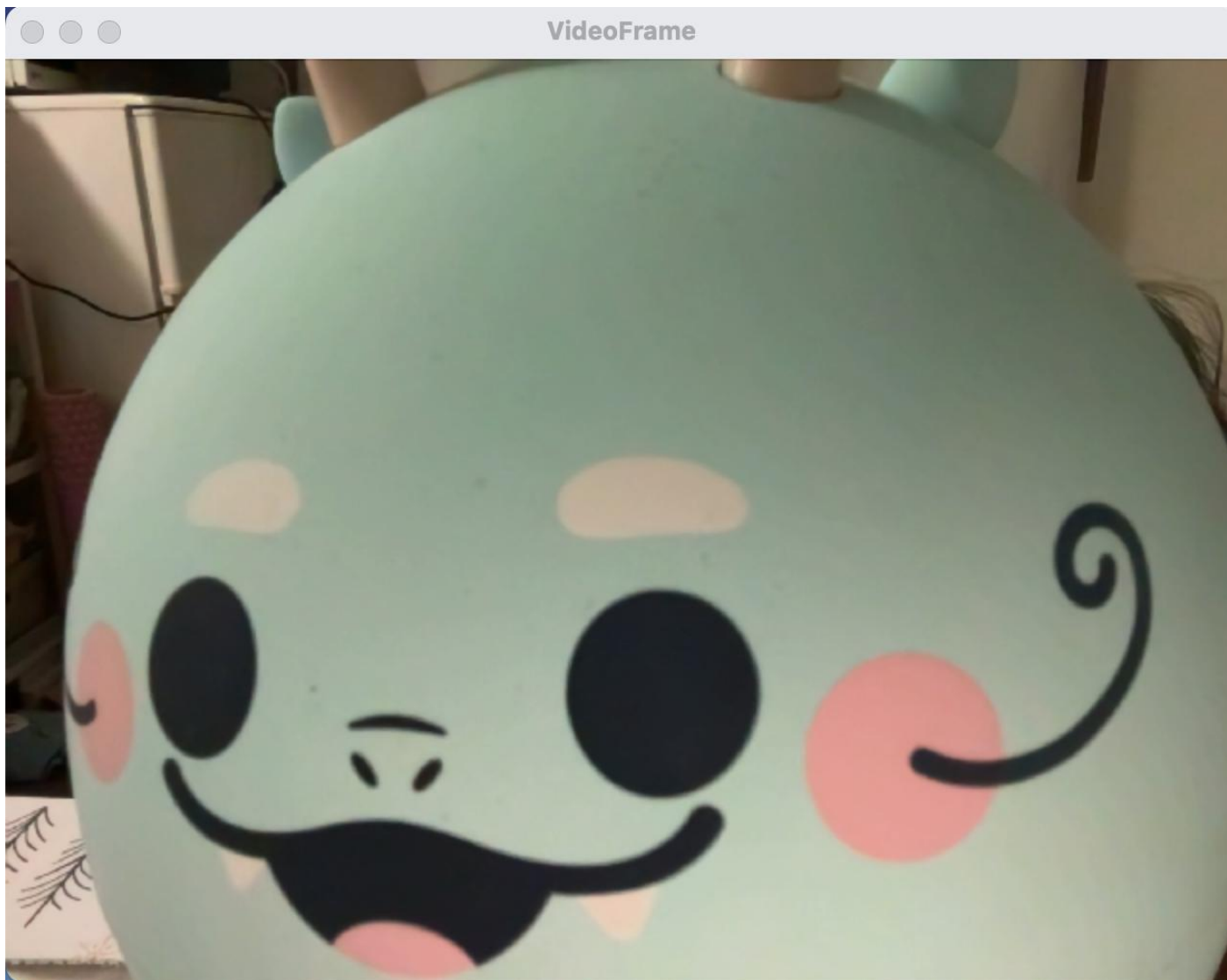
Subway = [10,20,30]

# 파이썬 언어 – 자료형(리스트)

## 1 비슷한 형태의 변수를 하나로 묶어놓은 집합



리스트(시퀀스 객체)의 인덱스는 0부터 시작



# 파이썬 언어 – 자료형(리스트)

## 1 리스트 인덱싱

```
a = [1,2,3]  
print(a[0])  
print(a[1])  
print(a[2])
```

#print 결과  
1  
2  
3



# 파이썬 언어 – 자료형(리스트)

## 2 리스트 이중 인덱싱

```
a = [1,2,3,['a', 'b', 'c']]  
print(a[-1])
```

```
#print결과  
['a', 'b', 'c']
```

```
print(a[-1][1])
```

```
#print 결과  
'b'
```

# 파이썬 언어 – 자료형(리스트)

## 3 리스트 연산

```
a = [1, 2, 3]  
print(a[0] + a[2])
```

#print 결과  
4

# 파이썬 언어 – 자료형(리스트:슬라이싱)

## 4 리스트[시작 index : 끝 index -1]

```
a = [1, 2, 3, 4, 5]  
print(a[1:4])
```

```
#print 결과  
[2, 3, 4]
```

# 파이썬 언어 – 자료형(리스트:슬라이싱)

## 5 리스트 중첩 슬라이싱

```
a = [1, 2, 3, ['a', 'b', 'c'], 4, 5]  
print(a[3][:2])
```

```
#print 결과  
['a', 'b']
```

# 파이썬 언어 – 자료형(리스트:슬라이싱)

## 5 슬라이싱 응용

```
a = [1, 2, 3, 4, 5, 6]
print(a[3:]) # 3번 index부터 끝까지
print(a[:3]) # 3번 index미만까지(2번 index까지)
print(a[1:5:2]) # 1번 index이상 5번 index미만 2개씩
print(a[::-1]) # 거꾸로
print(a[:]) # 처음부터 끝까지
```

*#print 결과*

[4, 5, 6]

[1, 2, 3]

[2, 4]

[6, 5, 4, 3, 2, 1]

[1, 2, 3, 4, 5, 6]

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 연산 +

```
a = [1, 2, 3]
```

```
b = [4, 5, 6]
```

```
print(a+b)
```

*#print 결과*

```
[1, 2, 3, 4, 5, 6]
```

```
a = [1, 2, 3]
```

```
print(a*3)
```

*#print 결과*

```
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 수정, 삭제

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

```
#print 결과  
[1,2,4]
```

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 수정, 삭제

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

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

```
del(a[1]) # 리스트 a의 1번 index 삭제
```

```
del(b[:4]) # 리스트 b의 0번 index부터 3번 index까지 삭제
```

```
print(a)
```

```
print(b)
```

```
#print 결과
```

```
[1, 3, 4, 5]
```

```
[5]
```



# 파이썬 언어 – 자료형(리스트)

## 5 리스트 추가 – append()

```
a = [1, 2, 3]  
a.append(4)  
print(a)
```

```
#print결과  
[1, 2, 3, 4]
```

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 추가 – insert()

```
a = [1, 2, 3]
a.insert(0, 0)
print(a)
```

```
#print 결과
[0, 1, 2, 3]
```

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 삭제 – remove()

```
a = [1, 2, 3, 3, 4, 5]  
a.remove(1)  
print(a)
```

```
#x가 리스트에 여러개 있을 경우  
a.remove(3)  
print(a)
```

```
#x가 리스트에 존재하지 않는 경우  
a.remove(6)  
print(a)
```

```
#print 결과  
[2, 3, 3, 4, 5] # 1 제거  
[2, 3, 4, 5] # 맨 처음 3 제거 (모든 3을 제거하지 않는다.)  
Error # 에러
```

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 맨 마지막 값 삭제 – pop()

```
a = [1, 2, 3, 4, 5]
b = a.pop()
print(a)
print(b)
```

```
#print 결과
[1, 2, 3, 4]
[5]
```

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 값 개수 확인 – count()

```
a = [1, 2, 3, 4, 4, 4, 5]  
print(a.count(4))
```

```
#print 결과  
3
```

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 값 개수 위치 – index()

```
a = [1, 2, 3, 3, 4, 5, 6]  
print(a.index(2))  
print(a.index(3))  
print(a.index(7))
```

#print 결과

1

2

Error

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 정렬 – sort()

```
a = [1, 4, 2, 3]
```

```
b = [1, 'a', 2]
```

```
c = ['b', 'a', 'c']
```

```
a.sort()
```

```
b.sort()
```

```
c.sort()
```

```
print(a)
```

```
print(b)
```

```
print(c)
```

*#print 결과*

```
[1, 2, 3, 4]
```

```
Error
```

```
['a', 'b', 'c']
```

# 파이썬 언어 – 자료형(리스트)

## 5 리스트 정렬 – reverse()

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

```
#print 결과  
[3, 2, 1]
```



# 파이썬 언어 - 자료형(리스트)

## 5 리스트 클리어 - 초기화

```
a = [1, 2, 3]  
a.clear()  
print(a)
```

```
#print 결과  
[]
```

# 파이썬 언어 – 자료형(리스트)

## 5 문자열을 리스트로 변환

```
a = list('hello')  
b = list((1,2,3))  
print(a)  
print(b)
```

```
#print 결과  
['h', 'e', 'l', 'l', 'o'][1, 2, 3]
```

# 자료형(튜플)

# 파이썬 언어 – 자료형(튜플)

**1** ()로 구분한다. (특징) 값 변경 불가

튜플명 = (값1, 값2, ...)

a = (1, 2, 3, 4)

Print(a[1])

Print(a[3])

# 자료형(딕셔너리)

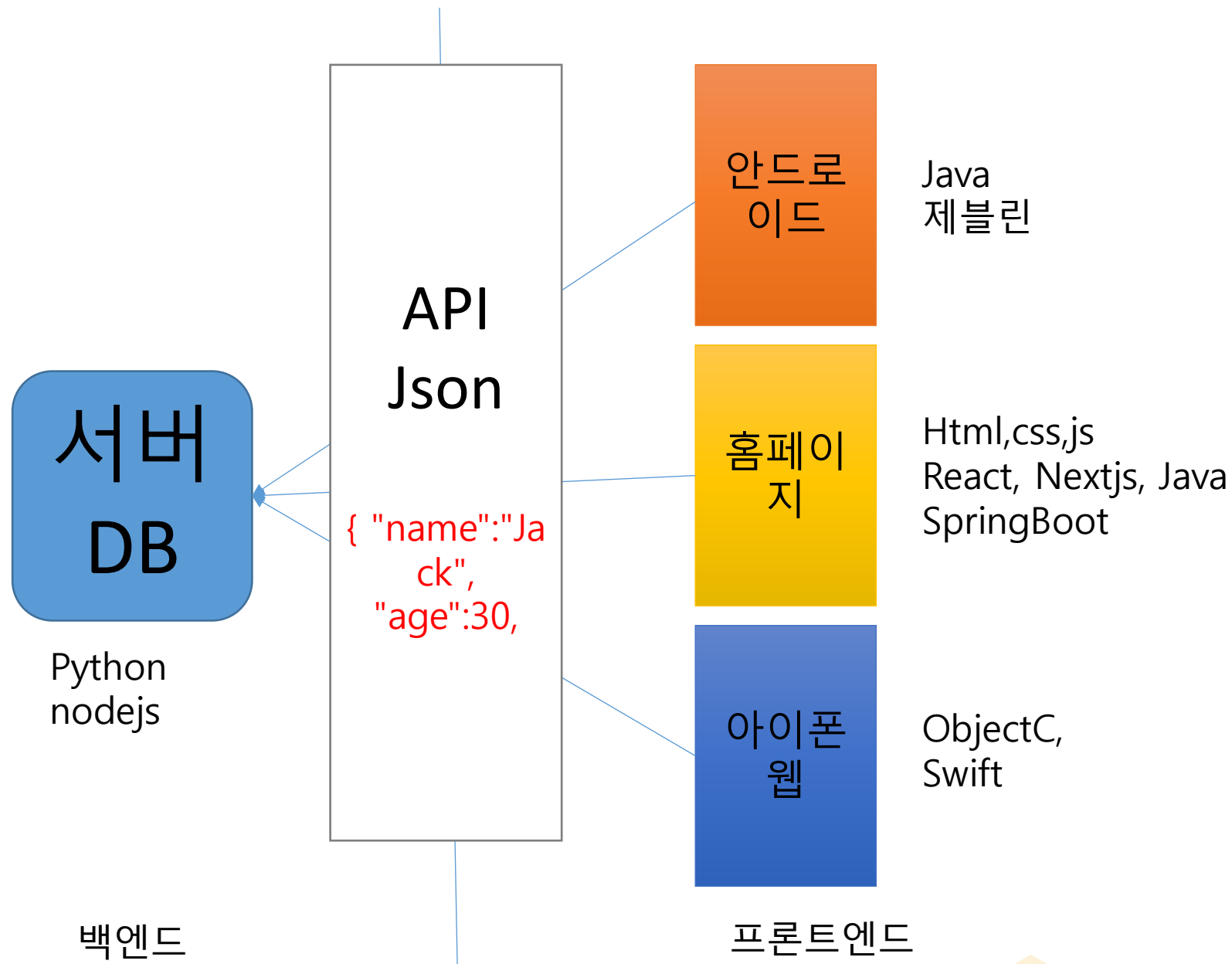
# 파이썬 언어 – 자료형(딕셔너리)

1 {}로 선언하고 key와 value를 한쌍으로 이룬다.

딕셔너리명 = {key1:value1, key2:value2, ...}

```
dic = {'name':'pey', 'phone':'010-9999-1234', 'birth': '1118'}
```

```
print(dic['name'])  
Print(dic['phone'])  
Print(dic['birth'])
```



# 파이썬 언어 – 자료형(딕셔너리)

## 1 Key 리스트 만들기

**a.keys()**

```
a = {'name': 'pey', 'phone': '010-9999-1234', 'birth': '1118'}
```

```
a.keys()
```

```
dict_keys(['name', 'phone', 'birth'])
```



# 파이썬 언어 – 자료형(딕셔너리)

## 1 Key 리스트 for문에서 불러오기

**a.keys()**

```
a = {'name': 'pey', 'phone': '010-9999-1234', 'birth': '1118'}
```

```
for k in a.keys():  
    print(k)
```

name phone birth

[형변환]

```
Print(list(a.keys()))
```

# 파이썬 언어 – 자료형(딕셔너리)

## 1 value 리스트

**a.value()**

a.values()

dict\_values(['pey', '010-9999-1234', '1118'])

# 파이썬 언어 – 자료형(딕셔너리)

## 1 Key, value 값얻기 – items()

```
a.items()
```

```
dict_items([('name', 'pey'), ('phone', '010-9999-1234'), ('birth', '1118')])
```

[초기화하기]

```
a.clear()
```

```
{}
```

# 파이썬 언어 – 자료형(딕셔너리)

## 1 Key로 value 값 얻기 – get()

```
a = {'name':'pey', 'phone':'010-9999-1234', 'birth': '1118'}
```

```
a.get('name')  
'pey'
```

```
a.get('phone')  
'010-9999-1234'
```

**Get(x, '디폴트')**  

```
a.get('nokey', 'foo')  
'foo'
```

# 파이썬 언어 – 자료형(딕셔너리)

## 1 딕셔너리 안에 값이 있는지 key로 비교

```
a = {'name':'pey', 'phone':'010-9999-1234', 'birth': '1118'}
```

```
'name' in a  
True
```

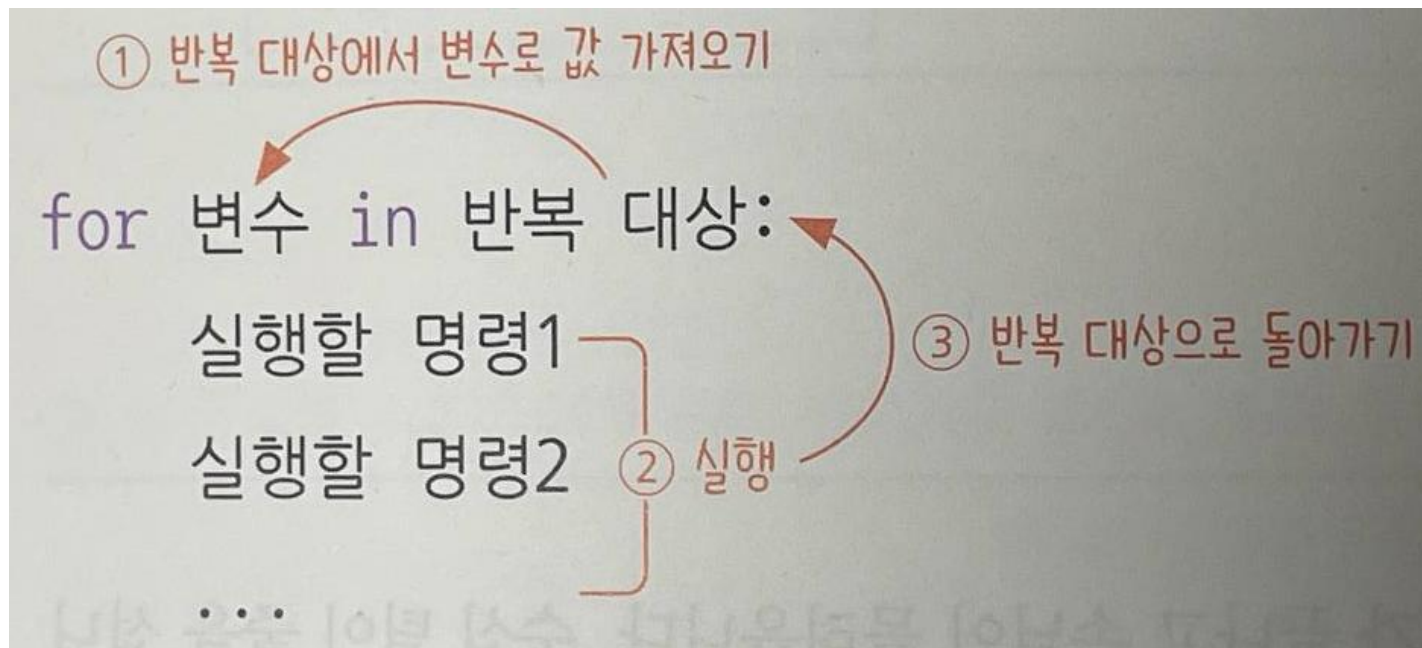
```
'email' in a  
False
```

# 제어문

# 제어문(if 조건문)

# 파이썬 언어 – 반복문

## 1 정해진 범위까지 반복하는 명령어





# 파이썬 언어 – 조건문

## 2 반복 대상 형식

**for 변수 in 반복 대상:**

실행할 명령1

실행할 명령2

...

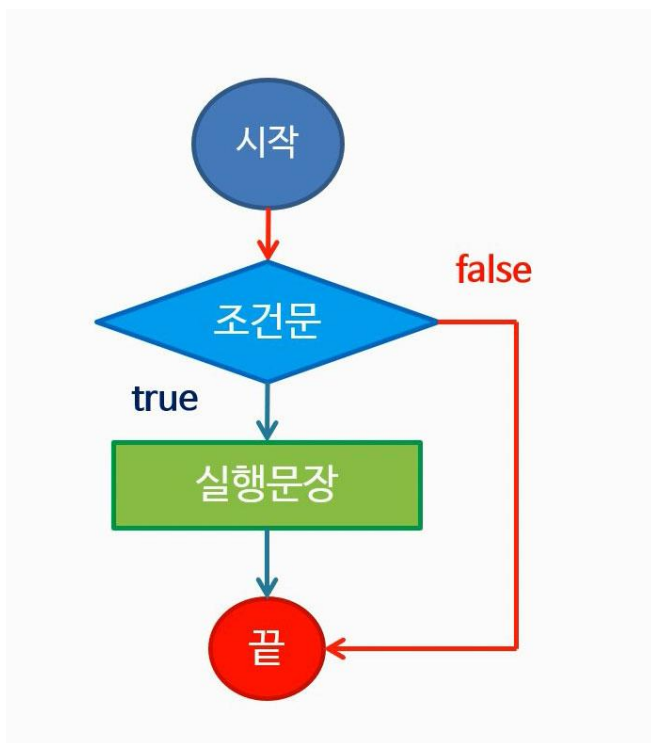
range(숫자)

range(시작 숫자, 끝 숫자)

range(시작 숫자, 끝 숫자, 간격)

# 파이썬 언어 – 조건문

## 1 조건문은 조건을 비교하여 처리하는 문장



**if 조건문:**

수행할 문장1  
수행할 문장2

**else:**

수행할 문장a  
수행할 문장b

# 파이썬 언어 – 조건문

## 2 들여쓰기(indentation)

if 조건문:

수행할 문장1
수행할 문장2

참일때 수행

else:

수행할 문장a
수행할 문장b

거짓일때 수행

# 파이썬 언어 - 조건문

## 3 비교 연산자

if **조건문:**  
수행할 문장1  
수행할 문장2

else:  
수행할 문장a  
수행할 문장b

비교연산자	설명
$x < y$	x가 y보다 작다
$x > y$	x가 y보다 크다
$x == y$	x와 y가 같다
$x != y$	x와 y가 같지 않다
$x \geq y$	x가 y보다 크거나 같다
$x \leq y$	x가 y보다 작거나 같다

# 파이썬 언어 – 조건문

## 3 and, or, not

if **조건문:**  
수행할 문장1  
수행할 문장2  
else:  
수행할 문장a  
수행할 문장b

연산자	설명
x or y	x와 y 둘중에 하나만 참이어도 참이다
x and y	x와 y 모두 참이어야 참이다
not x	x가 거짓이면 참이다

# 파이썬 언어 – 조건문

## 3 in, not in

if **조건문:**  
수행할 문장1  
수행할 문장2

else:  
수행할 문장a  
수행할 문장b

in	not in
x in 리스트	x not in 리스트
x in 튜플	x not in 튜플
x in 문자열	x not in 문자열

# 파이썬 언어 – 조건문

## 3 elif 여러 개의 조건 비교 조건문

if 조건문:

수행할 문장1

수행할 문장2

else:

if 조건문:

수행할 문장a

수행할 문장b

else:

수행할 문장c



if 조건문:

수행할 문장1

수행할 문장2

elif 조건문:

수행할 문장a

수행할 문장b

else:

수행할 문장c

# continue와 break문



# 제어문(while)

# 파이썬 언어 – while

## 1 조건문을 만족할 동안 반복

**while** 조건:

실행할 명령1

실행할 명령2

...

# 파이썬 언어 – while

## 1 조건문을 만족할 동안 반복

**cnt = 5**

**while cnt >= 1:**

실행할 명령1

실행할 명령2

...

**cnt -= 1**

비교연산자	설명
$x < y$	x가 y보다 작다
$x > y$	x가 y보다 크다
$x == y$	x와 y가 같다
$x != y$	x와 y가 같지 않다
$x \geq y$	x가 y보다 크거나 같다
$x \leq y$	x가 y보다 작거나 같다

# 파일 입출력

# 파이썬 언어 – open, close

## 1 Open 명령어

Read() : 파일 내용을 한꺼번에 읽어오기

Readline() : 파일 내용을 한 줄씩 읽어 오기

Readlines() : 파일 내용을 줄 단위로 리스트로 읽기

# 파이썬 언어 – open, close

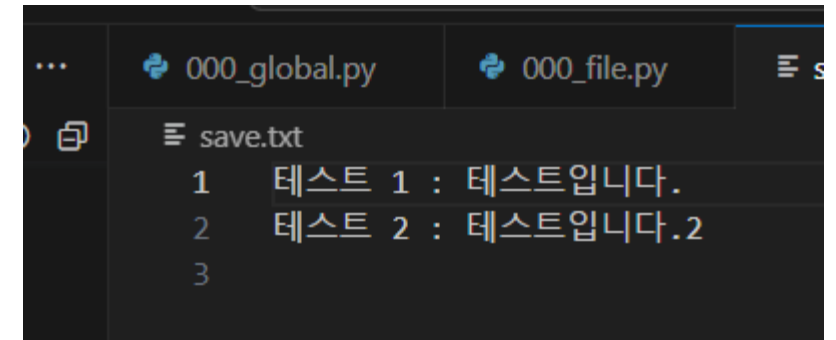
## 1 Open 명령어

`open('파일명','모드',encoding='인코딩 형식')`

```
save_file = open('save.txt', 'w', encoding='utf8')
```

```
print('테스트 1 : 테스트입니다.', file=save_file)  
print('테스트 2 : 테스트입니다.2', file=save_file)
```

```
save_file.close()
```



# 파이썬 언어 – open, close

## 1 Open 명령어

```
save_file = open('save11.txt', 'w', encoding='utf8')
```

```
save_file.write('리스트 파일 1\n')
```

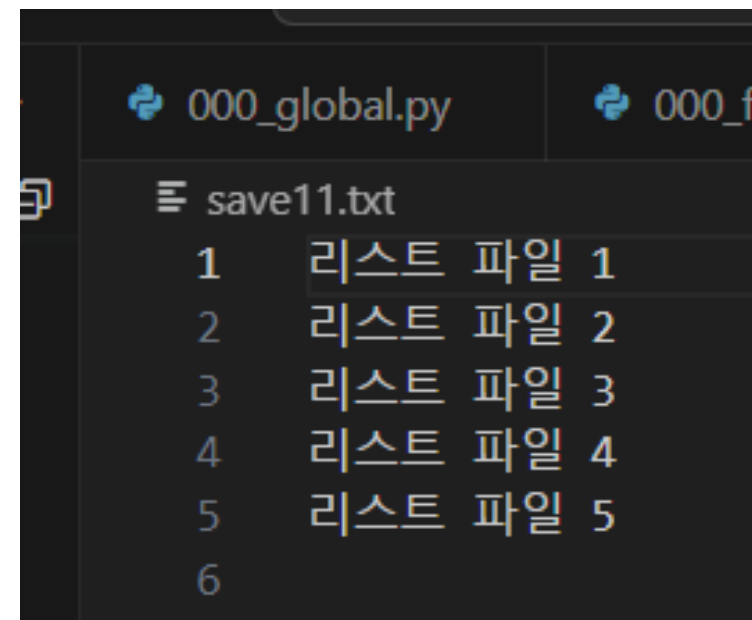
```
save_file.write('리스트 파일 2\n')
```

```
save_file.write('리스트 파일 3\n')
```

```
save_file.write('리스트 파일 4\n')
```

```
save_file.write('리스트 파일 5\n')
```

```
save_file.close()
```



The screenshot shows a code editor with two tabs: '000\_global.py' and '000\_fi'. The '000\_fi' tab is active, displaying the contents of 'save11.txt'. The file contains five lines of text, each consisting of a number followed by '리스트 파일' and another number, separated by a space. The lines are numbered 1 through 5 in the left margin.

Line	Content
1	리스트 파일 1
2	리스트 파일 2
3	리스트 파일 3
4	리스트 파일 4
5	리스트 파일 5

# 파이썬 언어 – read(), readline(), readlines()

## 1 read 명령어

```
score_file = open('save11.txt', 'r', encoding='utf8')
```

```
print(score_file.read())
```

```
score_file.close()
```

```
read.py  
리스트 파일 1  
리스트 파일 2  
리스트 파일 3  
리스트 파일 4  
리스트 파일 5
```

```
PS D:\20230427 광
```



# 파이썬 언어 – read(), readline(), readlines()

## 1 readline() 명령어

```
score_file = open('save11.txt', 'r', encoding='utf8')
```

```
while True:
```

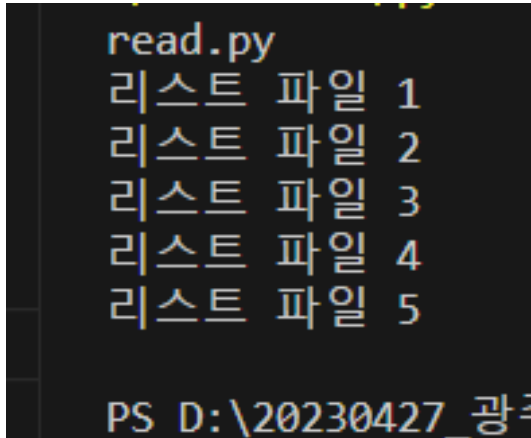
```
    line = score_file.readline()
```

```
    if not line:
```

```
        break
```

```
    print(line, end="")
```

```
score_file.close()
```



```
read.py  
리스트 파일 1  
리스트 파일 2  
리스트 파일 3  
리스트 파일 4  
리스트 파일 5
```

```
PS D:\20230427 광
```

# 파이썬 언어 – read(), readline(), readlines()

## 1 readlines() 명령어

```
score_file = open('save11.txt', 'r', encoding='utf8')
```

```
lines = score_file.readlines()
```

```
print(lines)
```

```
score_file.close()
```

```
readline.py
['리스트 파일 1\n', '리스트 파일 2\n', '리스트 파일 3\n', '리스트 파일 4\n', '리
스트 파일 5\n']
PS D:\20230427_광주인공지능교육_IITP\ai_exam\python_exam>
```

# 파이썬 언어 – read(), readline(), readlines()

## 1 readlines() 명령어

```
score_file = open('save11.txt', 'r', encoding='utf8')
```

```
lines = score_file.readlines()
```

```
for line in lines:  
    print(line, end='')
```

```
score_file.close()
```

```
c/anaconda3/python  
readline.py  
리스트 파일 1  
리스트 파일 2  
리스트 파일 3  
리스트 파일 4  
리스트 파일 5  
PS D:\20230427 공
```

# 피클(pickle) 모듈

# 파이썬 언어 – pickle

## 1 Pickle 로 저장하기

```
import pickle

profile_file = open('profile.pickle', 'wb')

profile = {"이름":"홍길동", "나이":30, "언어":["파이썬","자바"]}

print(profile)

pickle.dump(profile, profile_file)
profile_file.close()
```

# 파이썬 언어 – pickle

## 1 Pickle 로 불러오기

```
import pickle
```

```
profile_file = open('profile.pickle', 'rb')  
profile = pickle.load(profile_file)
```

```
print(profile)
```

```
profile_file.close()
```

```
pickle1.py  
{'이름': '홍길동', '나이': 30, '언어': ['파이썬', '자바']}  
PS D:\20230427_광주인공지능교육_IITP\ai_exam\python_exam> █
```

# 전역변수와 지역변수

# 파이썬 언어 – 전역변수와 지역변수

## 1 전역변수와 지역변수

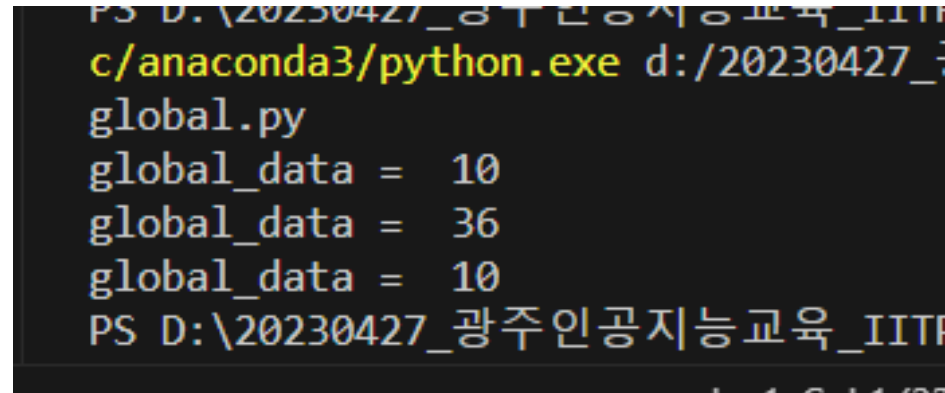
```
global_data = 10
```

```
def infos(infos):  
    global_data = 40  
    global_data = global_data - infos  
    print("global_data = ", global_data)
```

```
print("global_data = ", global_data)
```

```
infos(4)
```

```
print("global_data = ", global_data)
```



```
PS D:\20230427_광주인공지능교육_IITF  
c/anaconda3/python.exe d:/20230427_  
global.py  
global_data = 10  
global_data = 36  
global_data = 10  
PS D:\20230427_광주인공지능교육_IITF
```



# 출력문 관련 파일

# 파이썬 언어 - 출력문

## 1 출력문

```
print("파이썬", "자바", sep=",")  
print("파이썬", "자바", "자바스크립트",  
sep=" vs ")
```

# 파이썬 언어 – 공간 확보 정렬하기

## 1 딕셔너리 출력하기

```
exams = {"인공지능":90, "DB":80, "코딩":70}
```

```
for title, val in exams.items():  
    print(title, val)
```

# 파이썬 언어 – 공간 확보 정렬하기

## 1 딕셔너리 출력하기 : ljust(), rjust()

```
exams = {"인공지능":90, "DB":80, "코딩":70}
```

```
for title, val in exams.items():  
    print(title.ljust(10), str(val).rjust(5), sep=":")
```

```
print.py  
인공지능      :    90  
디비          :    80  
코딩          :    70  
PS D:\20230427_광주인공지능
```

# 파이썬 언어 - 0으로 채우기 zfill

## 1 0으로 채우기 zfill

```
exams = {"인공지능":90, "DB":80, "코딩":70}
```

```
for title, val in exams.items():  
    print(title.ljust(10), str(val).zfill(3).rjust(5),  
          sep=":")
```

```
print.py  
인공지능      :   090  
디비          :   080  
코딩          :   070  
PS D:\20230427_광주인공자
```

# 파이썬 언어 – format()

## 1 Print format문 사용하기

```
print('{0}'.format(500))
```

# 3자리마다 , 넣기

```
print('{0:,}'.format(100_000_000))
```

# 소수점 2자리까지만 출력

```
print('{0:.2f}'.format(5/3))
```

# 만들어봅시다.

# 파이썬 언어 – 프로젝트 실습

## 1 로또 번호 생성기(5개) – 중복금지, 2차원 리스트사용

---

A [ 39, 23, 1, 40, 8, 13 ]

B [ 27, 35, 28, 1, 40, 14 ]

C [ 43, 38, 25, 23, 28, 40 ]

D [ 25, 13, 15, 37, 12, 30 ]

E [ 16, 2, 43, 29, 36, 40 ]

---

C: 번호생성, s:저장, q:종료



# 예외처리

# 파이썬 언어 – 예외처리

## 1 Try ~ exception 처리

```
try:
```

```
    실행할 코드
```

```
except:
```

```
    예외가 발생했을 때 처리하는 코드
```

# 파이썬 언어 – 예외처리

## 1 Try ~ exception 처리

try\_except.py

```
try:
    x = int(input('나눌 숫자를 입력하세요: '))
    y = 10 / x
    print(y)
except:    # 예외가 발생했을 때 실행됨
    print('예외가 발생했습니다.')
```

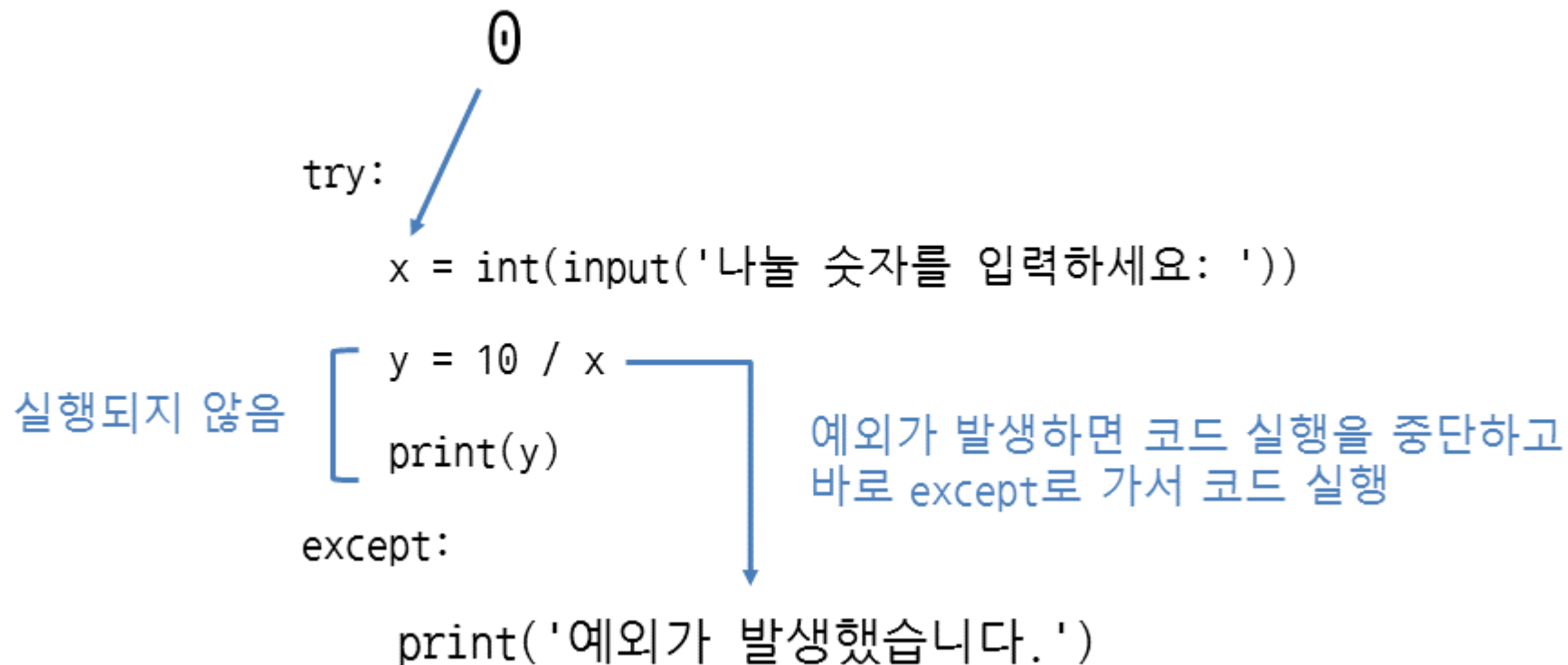
소스 코드를 실행한 뒤 0을 입력하고 엔터 키를 누르세요.

실행 결과

```
나눌 숫자를 입력하세요: 0 (입력)
예외가 발생했습니다.
```

# 파이썬 언어 – 예외처리

## 1 Try ~ exception 처리



# 파이썬 언어 – 예외처리

## 1 Try ~ exception 처리

try\_except\_exception.py

```
y = [10, 20, 30]

try:
    index, x = map(int, input('인덱스와 나눌 숫자를 입력하세요: ').split())
    print(y[index] / x)
except ZeroDivisionError:    # 숫자를 0으로 나눌 때 에러가 발생했을 때 실행됨
    print('숫자를 0으로 나눌 수 없습니다.')
except IndexError:          # 범위를 벗어난 인덱스에 접근하여 에러가 발생했을 때
    실행됨
    print('잘못된 인덱스입니다.')
```

# 파이썬 언어 – 예외처리

## 1 Try ~ exception 처리

try\_except\_as.py

```
y = [10, 20, 30]
```

```
try:
    index, x = map(int, input('인덱스와 나눌 숫자를 입력하세요: ').split())
    print(y[index] / x)
except ZeroDivisionError as e:
    # as 뒤에 변수를 지정하면 에러를
    받아들
    print('숫자를 0으로 나눌 수 없습니다.', e) # e에 저장된 에러 메시지 출력
except IndexError as e:
    print('잘못된 인덱스입니다.', e)
```

실행 결과

인덱스와 나눌 숫자를 입력하세요: 2 0 (입력)  
숫자를 0으로 나눌 수 없습니다. division by zero

실행 결과

인덱스와 나눌 숫자를 입력하세요: 3 5 (입력)  
잘못된 인덱스입니다. list index out of range

# 클래스(Class)

# 파이썬 언어 – 클래스

## 1 커피숍(창업)

Class Café:  
pass

Nobrand = Café()





# 파이썬 언어 - 클래스

**1** 커피숍(주문:명령)    Class Café:  
   def ordering(self):  
   print("주문받아요 !")



noBrand = Café()  
noBrand.ordering()

# 파이썬 언어 - 클래스

## 1 커피숍(변수:메뉴)



```
Class Café:  
    coffee = "아이스커피"  
  
    def ordering(self):  
        print("주문 : " + self.coffee)  
  
noBrand = Café()  
noBrand.ordering()
```

# 파이썬 언어 - 클래스:오브젝트 만들기

## 1 커피숍 장사가 잘됨



Class Café:  
coffee = "아이스커피"

```
def ordering(self):  
    print("주문 : " + self.coffee)
```

```
Starbucks = Café()  
Twosome = Café()  
Ediya = Café()
```

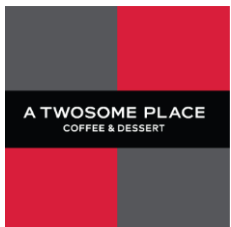
```
Starbucks.ordering()  
Twosome.ordering()  
Ediya.ordering()
```

# 파이썬 언어 - 클래스:변수할당

## 1 메뉴를 다르게



카페라떼



아메리카노



바닐라라떼

Class Café:

```
def __init__(self, menu):  
    self.menu = menu
```

```
def ordering(self):  
    print("주문 : " + self.menu)
```

Starbucks = Café("카페라떼")

Twosome = Café("아메리카노 ")

Ediya = Café("바닐라라떼 ")

Starbucks.ordering()

Twosome.ordering()

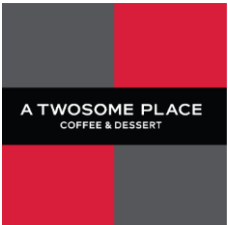
Ediya.ordering()

# 파이썬 언어 - 클래스:함수로 인자전달

## 1 크기 다르게



카페라떼  
(톨!)



아메리카노  
(레귤러!)



바닐라라떼  
(빅!)

```
class Cafe:
    def __init__(self, menu):
        self.menu = menu

    def ordering(self, cup_size):
        print('주문 : ' + self.menu + ', 사이즈 : ' + cup_size)
```

```
starbucks = Cafe('카페라떼')
twosome = Cafe('아메리카노')
ediya = Cafe('바닐라라떼')
```

```
starbucks.ordering("톨")
twosome.ordering("레귤러")
ediya.ordering("빅")
```

# 파이썬 언어 - 클래스: 함수에 변수 추가

## 1 주문시 싸이즈까지



카페라떼  
(톨!)



아메리카노  
(레귤러!)



바닐라라떼  
(빅!)

```
class Cafe:
```

```
    def __init__(self, menu, cup_size):
```

```
        self.menu = menu
```

```
        self.cup_size = cup_size
```

```
    def ordering(self):
```

```
        print('주문 : ' + self.menu + ', 싸이즈 : ' + self.cup_size)
```

```
starbucks = Cafe('카페라떼', '톨')
```

```
twosome = Cafe('아메리카노', '레귤러')
```

```
ediya = Cafe('바닐라라떼', '빅')
```

```
starbucks.ordering()
```

```
twosome.ordering()
```

```
ediya.ordering()
```



2025년 서울 민간기업 맞춤형 매력일자리 사업

# 파이썬 문법

- 질문과 답변 -