

# [파이썬 기본 문법 정리]

## 변수

### 변수명의 규칙

- 변수명의 첫 글자는 반드시 영문자나 일부 특수문자(underscore, dollar sign)로 시작한다
- 변수명에는 숫자가 포함될 수 있다
- 변수명에는 공백이 포함될 수 없다
- 대소문자를 구분한다
- 예약어(프로그래밍언어 자체에서 사용되는 용어)를 변수명으로 사용할 수 없다

### 변수 선언

```
In [ ]: # 한번에 선언
a, b, c = 1, 2, 3
print(a, b, c)
```

```
In [ ]: # 값 교환
a, b = 10, 20
print(a, b)
a, b = b, a # 동시할당 가능
print(a, b)
```

### 변수 데이터 타입(6가지)

- Numeric Types → int / float / complex
- Sequence Types → str / list / tuple
- Mapping Type → dict
- Set Type → set
- Boolean Type → bool (True / False 데이터를 나타내는 자료형)
- Binary Types → bytes / bytearray / memoryview

```
In [ ]: # Numeric Types
int_ = 12234342524534634636682479
float_ = 12.33456789123456789 # 실수 타입은 8바이트 까지만 저장
complex_ = 1 + 2j

print(int_, float_, complex_)
print(type(int_), type(float_), type(complex_))
```

```
In [ ]: # Sequence Types - 리스트와 튜플은 여러가지 타입을 원소로 가질 수 있음. 튜플은 만든
str_ = "123"
list_ = [1, 2, 3]
tuple_ = (1, 2, 3)

print(str_, list_, tuple_)
print(type(str_))
```

```
print(type(list_))
print(type(tuple_))
```

```
In [ ]: # Mapping Type - 딕셔너리를 이용하여 키와 값으로 자료를 매핑

dict_ = {'key1': 'value1', 'key2': 2, 'key3': [1, 2, 3]}

print(dict_)
print(type(dict_))
```

```
In [ ]: # Set Type - set 데이터 타입은 원소의 중복이 불가능 함

set_ = set([1, 2, 3])

print(set_)
print(type(set_))
```

## 출력 - print() 함수

- sep 속성 : 각 인자 사이에 넣을 문자열을 지정
- end 속성 : 문자열 출력 이후에 나오는 문자열을 지정. "" NULL문자, " " 공백문자, "\n" 한 줄 띄우기

```
In [ ]: # 출력형태

a, b, c = 1, 2, 3

print(a, b, c)
print(a, b, c, sep=', ')
print(a, b, c, sep=' ')
print(a, b, c, sep='\n')
print(a, b, c, end=' ')
```

## 입력 - input() 함수

- 입력 받은 데이터는 문자열로 반환됨.

```
In [ ]: aa = input("문자를 입력하세요 : ")
print(aa)
```

```
In [ ]: # 분리해서 입력 받기

a, b = input("숫자를 입력하시오: ").split()
print(a, b)
print(a + b)
```

```
In [ ]: # int() 이용하여 숫자 한개 입력 받기

a = int(input("첫번째 숫자를 입력하시오: "))
b = int(input("두번째 숫자를 입력하시오: "))

print("두수의 합은 {}입니다.".format(a+b))
```

```
In [ ]: # 숫자 한꺼번에 여러개 입력받기 . map()함수 이용하기
# map()함수는 반복 가능한 객체(리스트 등)에 대해 각각의 요소들을 지정된 함수로 처리함
# split()함수의 괄호 안에 특정 값을 넣어주면 그 값을 기준으로 문자열을 나누고,
```

```
# 아무 값도 넣어주지 않으면 공백(스페이스, 탭, 엔터 등)을 기준으로 문자열로 나눠줌.

b, c = map(int, input("숫자를 입력하시오: ").split())
print(b, c)

print("b + c =", b+c)
print("b * c =", c * b)
print("b / c =", c / b)
print("b // c =", c // b) # 몫
print("b % c =", b % c) # 나머지
print("b ** c =", c ** b) # 거듭제곱
```

## 조건문 - if ~ elif ~ else

- 관계연산자 : ==(같다), !=(같지않다), >(초과), <(미만), >=(이상), <=(이하)
- 논리연산자 : and, or, not

```
In [ ]: # and 연산없이 사이값들 나타낼 수 있음. (파이썬만 가능)
x = int(input("숫자를 입력하시오 : "))
if 0 < x < 10:
    print('10 보다 작은 자연수입니다.')
else :
    print('10 보다 작은 자연수가 아닙니다.')
```

```
In [ ]: # 다중 조건문
x = int(input("점수를 입력하시오 : "))
if x >= 90:
    print('A')
elif x >= 80:
    print('B')
elif x >= 70:
    print('C')
elif x >= 60:
    print('D')
else:
    print('F')
```

## 반복문 - for() 함수

- range() 함수는 일정한 범위의 숫자로 구성된 리스트를 한번에 만들어 줌.
- 형식 : range(시작 숫자, 끝 숫자 + 1, 스텝). 시작숫자 생략하면 0, 스텝숫자 생략하면 1로 인식 됨.

```
In [ ]: for i in range(10): # i값 출력 10번 반복함.
        print(i)

for j in range(1, 11, 2): # 1,3,5,7,9 출력
    print(j)
```

```
In [ ]: sum_=0
for i in range(1, 11): # i는 1~10까지
    sum_ += i          # sum_ = sum_ + i
print(sum_)
```

```
In [ ]: for x in (1,3,2,60):
        print(x, "의 제곱수는 : ", x * x)
```

## 반복문 - while() 함수

- 특정조건이 만족하는 동안 반복 됨.
- 무한 루프 돌릴 수 있음.

```
In [ ]: i = 10
print("i 값은 : ", end = "")
while i > 0:
    print(i, end=" ")
    i = i - 1

k = 10
print("\nk 값은 : ", end = "")
while k > 0:
    print(k, end=" ")
    k -= 1
```

## 리스트(list)

- 여러 데이터를 하나의 변수에 저장하는 자료형. 각 항목의 데이터 타입이 달라도 됨. [] 대괄호 사용하여 생성.
- 중복 데이터 허용 가능.

```
In [ ]: # 리스트 합치기. "+"연산자 사용.
a = [1, 2, 3]
b = [3, 4, 5, 6]
c = a + b
print(c)
```

```
In [ ]: # 리스트 내장 함수 : append()
a = [1, 2, 3, 4]
print(a)

a.append(5) # 요소는 리스트의 끝에 추가
print(a)
```

```
In [ ]: # 리스트 내장 함수 : insert(인덱스, 넣을 값). 특정한 인덱스 위치에 요소를 추가
a.insert(1, 5)
print(a)
```

```
In [ ]: # 리스트 내장 함수 : pop(). 인자없으면 끝요소 삭제 또는 특정 인덱스의 값을 인자로 넣
a = ['a', 'b', 'c', 'd', 'e']
a.pop()
print(a)

a.pop(2)
print(a)
```

```
In [ ]: # 리스트 내장 함수 : remove(). 특정 요소의 값과 같은 요소를 삭제
a = ['a', 'b', 'c', 'd', 'a', 'e']
print(a)

a.remove('a')
print(a)
```

```
In [ ]: # 리스트 내장 함수 : index(). 특정 요소 값을 찾아 인덱스를 반환
a = ['a', 'b', 'c', 'd', 'e']
print(a)

tmp = a.index('b')
print(tmp)
```

```
In [ ]: # 리스트 내장 함수 : sum(), min(), max()
aa = [1, 5, 7, 2, 9]
print(aa)

print(sum(aa))
print(min(aa))
print(max(aa))
```

```
In [ ]: # 리스트 내장 함수 : sort() , 리스트 항목 정렬. 문자열에도 적용가능.
a = [1, 5, 7, 2, 9]
print(a)

a.sort() # 오름차순
print(a)

a.sort(reverse=True) # 내림차순
print(a)
```

```
In [ ]: # 리스트 내부 for
# 예) 변수 x를 0~4까지 사용하는 리스트를 만들어 각 인자의 10의 배수가 출력되도록 하
list2 = [x * 10 for x in range(5)]
print(list2)
```

## 튜플(tuple)

- 생성된 튜플은 내용 변경이 안되는 상수화된 리스트라고 볼 수 있다.
- 검색 속도가 빠름

```
In [ ]: t1 = (1,2,3)
t2 = 4,5,"hello" #튜플 생성시 () 사용하거나 괄호를 넣지 않아도 된다.
print(t1 + t2)
```

## 딕셔너리(dictionary) - {키 : 값}

- 모든 항목이 항상 키(key)와 값(value) 두가지로 짝을 이루어 구성되는 데이터 타입
- 저장된 순서는 중요치 않고, 항목의 값은 키 값을 인자로 하여 얻을 수 있다.

```
In [ ]: dic1 = {'이름':'철수', '국어':90, '영어':75, '수학':100 }
print(dic1['이름'])
print(dic1['수학'])

dic1['컴퓨터']=95 # 새 항목 추가
print(dic1)
```

```
In [ ]: # 딕셔너리 내장함수 : keys(), values(), items()
print(dic1.keys())      # 카만 추출
print(dic1.values())    # 값만 추출
print(dic1.items())     # 각 항목을 튜플로 만들어 출력.
```

## 함수 정의 - def 함수명 (함수인자) :

- 사용자가 임의의 기능을 함수로 정의 함.
- 함수는 main 스크립트 위에 존재해야 함.

```
In [ ]: # 함수명 double_, 인자로 받은 값의 두배를 리턴하는 함수만들기
def double_(x) :
    return x * 2

double_(489)    # 더블함수 호출함.
```

```
In [ ]: # 매개변수에 디폴트 값 지정가능
def double_2(x = 100) :
    return x * 2

double_2()
```

```
In [ ]: # 함수 실행 결과로 두개 이상의 값을리터 할 때에는 튜플로 처리하면 편리함.
def double_3(x) :
    return x * 2, x * 3    # 튜플 타입으로 반환됨

x, y = double_3(5)
print("5의 2배수는 {}이며, 3배수는 {}이다.".format(x, y))
```

```
In [ ]: def add(a, b):
    c = a + b
    d = a - b
    return c, d    # 튜플 자료구조로 반환

a, b = add(3, 2)
print(a, b)
print(add(3, 4))    # 튜플 자료구조
```

## 랜덤(random) 모듈

```
In [ ]: # random() 함수 : 0~1 미만의 숫자 중 하나 추출

import random as r
r.random()
```

```
In [ ]: # randrange()함수 : 주사위처럼 1에서 6까지의 정수 중 하나를 무작위로 추출하려면
r.randrange(1,7) # 1이상 7미만
```

```
In [ ]: # randint()함수 : 주사위처럼 1에서 6까지의 정수 중 하나를 무작위로 추출하려면
r.randint(1,6)    # 1이상 6이하
```

```
In [ ]: lst = [ 10, 20, 30, 40, 50]
r.shuffle(lst)    # lst 리스트의 순서를 무작위로 섞음
print(lst)
```

```
aa = r.choice(lst)      # lst 리스트의 원소 중 무작위로 하나를 추출
print("당첨된 숫자는 {}".format(aa))
```

## [실습 문제]

### 1. 제시하는 실행결과를 출력하기

### 2. 홀짝게임

컴퓨터가 1~99까지 정수 중 하나를 선택하면 사용자가 맞추는 게임

- 실행결과

컴퓨터가 1에서 100까지 수 중 하나를 선택했습니다.

홀수 일가요? 아니면 짝수일까요?: 짝

컴퓨터의 수는 57, 당신의 답은 짝으로, 틀렸습니다.

### 3. 5과목 '유효한' 성적 입력받아 총점과 평균 출력하기

- (실행결과)

1 번째 성적 입력 : 59

1 번째 성적 : 59

2 번째 성적 입력 : 11110

유효한 성적이 아닙니다

2 번째 성적 입력 : 89

2 번째 성적 : 89

3 번째 성적 입력 : 110

유효한 성적이 아닙니다

3 번째 성적 입력 : 98

3 번째 성적 : 98

4 번째 성적 입력 : 78

4 번째 성적 : 78

5 번째 성적 입력 : 11

5 번째 성적 : 11

총점 : 335

평균 : 67.0"

### 4. "3, 6, 9" 게임 - 숫자말하기

- 1~40까지만 하도록
- 10의 배수이면 "뽕송!" 우선 적용. 예) 10일때는 "뽕송!", 20일때는 "뽕뽕송!"

- 실행결과

1

2

박수

4

5

박수

7

8

박수

뽀송~!

11