

변수의 종류:

불형(Boolean, True or False), 정수형(int)

실수형(float), 문자열(string)

변수명 지정:

대소문자 구분

문자, 숫자, _ 포함 but 첫글자≠숫자,

예약어는 사용금지 - True, False, or, and, import,...

eg) 2x=1.5, _2x=1.5, &x=3, x=4, X=40

type() 함수사용 ~ 변수형 확인

eg) x=4.12567, y=12

```
print(x,y), print('x=',x,'y=',y), print('%7.5f, %d'%(x,y)),  
print('%7.3f'%(x)), print('x=%7.3f'%(x))
```

eg) 문자열에 작은(큰)따옴표 포함

```
z1="It is 'You'. "; print(z1); z2='It is "You". '; print(z2)
```

숫자형 연산:

사칙연산(+,-,*,/), 제곱(**), 나눗셈의 몫과 나머지(//,%)

정수와 실수의 연산 결과는 실수

```
x=14.0; y=4; print(x+y, x*y, y**2, x//4, x*y)
```

문자열 연산:

문자열 더하기, 문자열 곱하기, 문자열 길이구하기

```
eg) x1="Python "; x2="is fun!"; x12=x1+x2; print(x12)
```

```
x111=x1*3; print(x111); print('x111 길이는',len(x111))
```

대입연산자: =, +=, -=, *=, /=, %=, **=

eg) a+=3 ≡ a=a+3, a-=3 ≡ a=a-3

a*=3 ≡ a=a*3, a/=3 ≡ a=a/3

a//=3 ≡ a=a//3, a%=3 ≡ a=a%3, a**=3 ≡ a=a**3

a=10;

관계(비교)연산자: 비교할 때 사용. 결과는 참(True) or 거짓(False)

==, !=, >, <, >=, <=

eg) a,b=100,200 ≡ a=100; b=200

```
print(a==b, a!=b, a>b, a<b, a>=b, a<=b)
```

논리연산자: 여러조건을 복합해서 사용. and, or, not

eg) a=99

```
(a>100) and (a<200), (a>100) or(a<200), not(a==100)
```

eg) x=True; y=False

```
type(x), x, y, x or y, x and y, not x, x>y, y<x
```

이진수 10 → 십진수로는 2 : int('10',2),

이진수 1111 → 십진수로는 15 : int('1111',2)

리스트(list)

여러 원소(요소)를 묶어 하나의 변수로 활용할 때 사용,

eg) a1=[3, 2, 1, 4]; a2=['I', 'am', 'fine'];

0~(k-1)번째 원소(요소)의 위치=0~(k-1)

type(a1); type(a2)

(1) 인덱싱(indexing)

(2) 슬라이싱(slicing)

a1[0]	a2[0]	a1[0:2]	a2[0:2]
a1[1]	a2[1]	a1[1:3]	a2[1:3]
a1[2]	a2[2]	a1[1:]	a2[1:]
a1[-1]	a2[-1]	a1[:3]	a2[:3]

(3) 연산

a1[1]=3

a1

a1[0]+a1[2]

a2[0]+a2[2]

a3[1]+a3[2][0]

*a1[0]+a2[0]

*a3[1]+a3[2]

a1*2

(4) 리스트 관련함수: len(), sort(), append(), index(), insert(), remove(), del

x=[7, 3, 2, 1, 3, 4, 5, 6];

len(x)

x.sort(); x

x.reverse(); x

x.append(8); x

x.index(4) # x내 4의 위치

x.insert(0,5) #x의 0번째 원소에 5 추가

x

x.insert(1,4) #x의 1번째 원소에 9 추가

x

x.pop() #x의 마지막 원소 제거

x

x.count(3) # x안에 3이 몇 개?

c1=x.copy() # x과 내용이 같은 리스트 c1 생성, c1=x[:]

x.remove(4) # x에서 첫번째로 나오는 4 제거

x

x[1:3]=[] # x의 1~(3-1)번째 원소 제거

x

del x # x 제거

input() 함수: 사용자가 어떤 값을 입력하면 그 값을 변수에 저장

```
x=input() # 문자열 입력, x=int(input()) :정수 입력, float(input()) : 실수 입력
Sam
x
```

튜플(tuple)

여러 원소(요소)를 묶어 하나의 변수로 활용할 때 사용~ 리스트와 비슷
but 소괄호() 이용, 튜플은 값바꾸기 불가능.

```
eg) t1=(3,2,1,3,4)
type(t1); t1[1]=22; t1
```

딕셔너리(dictionary)

쌍(key & 값(value))으로 이루어진 구조의 자료형,

```
eg) d1={1:'a', 2:'b', 3:'c'}; d1
type(d1)
d1[1] # key를 사용해서 값구하기
d2={1:'a', 3:'b', 2:'c'}
d2 # key를 기준으로 자동 정렬
# key는 중복되면 안됨
d11={1:'a', 1:'b'}; d11
d1={'name':'Kim', 'ht':178, 'wt': 72}; d1
d1['name']
쌍추가: eg) d1['phone']='010-321-4567'; d1
쌍제거: eg) del d1['wt']; d1
기존값 변경: eg) d1['phone']='010-789-1234' ; d1
```

집합(set)

집합을 쉽게 처리하기 위해 만든 자료형.

```
eg) s1=set(['a','b','c','d']); s2=set(['c','d','e','f','g'])
type(s2)
s1|s2 # 합집합
s1&s2 # 공통집합
s1-s2 # 차집합
```

eg) 두 수를 입력 받아 첫번째 입력한 수가 두번째 입력한 수보다 크면 '첫번째가 두번째이상', 아니면 '첫번째가 두번째 미만'을 출력하는 프로그램.

```
x,y=float(input('x 입력 : Wn')), float(input('y 입력 : Wn'))
```

```
if x>=y:
    print('첫번째가 두번째이상')
else:
    print('첫번째가 두번째미만')
```

eg) x=17; y=20

```
if x>y:  
    print('x>y')  
else:  
    print('x<=y')
```

```
if x>y:  
    z=0  
else:  
    z=1  
print(z)
```

```
z=x*y  
if z>300:  
    z=z*0.1  
else:  
    z=z+10  
print(z)
```

```
if x>=17 and y>=20:  
    print('x>=17 and y>=20')  
else:  
    print('x<17 or y<20')
```

```
if x<17 or y<20:  
    print('x<17 or y<20')  
else:  
    print('x>=17 and y>=20')
```