

파이썬 프로그래밍

---

# 파이썬 언어의 기본 문형



한국기술교육대학교  
온라인평생교육원

---

## ■ 파이썬 기초 문형

### 1. 주석문

```
#이것은 주석입니다.  
import sys #이것도 주석입니다.
```

- 주석문: #(샵) 뒤에 오는 문장
- 주석문은 첫 번째 줄에 올 수도 있고
- 일반적인 파이썬문 뒤에 주석문이 올 수도 있음
- 한 줄을 모두 차지하거나 명령문 뒤에 달렸거나 상관없이
- #(샵) 기호와 함께 사용되면 주석

## ■ 파이썬 기초 문형

### 2. 연속라인

```
a = 1
b = 3
if (a == 1) and \
(b == 3) :
    print 'connected lines'
```

```
connected lines
```

- a가 1이고 b가 3이면 'connected lines'라는 문구를 출력하는 예제
- 이 때 if부터 3까지는 한 줄이 되어야 함
- \ (백슬래시) → 나뉜 줄을 한 줄로 인식하도록 함
- \ (백슬래시)는 코딩이 길어져 한 화면에 나타나지 않을 때 사용

```
a = 1
b = 3
if (a == 1) and (b == 3) :
    print 'connected lines'
```

```
connected lines
```

## ■ 파이썬 기초 문형

### 3. 할당문

```
a = 1  
b = a
```

- 할당문 → 등호(=)를 사용한 연산
- $a = 1 \rightarrow a$ 에 1을 할당
- $b = a \rightarrow b$ 에  $a$ 를 할당  $\rightarrow a$ 가 1이므로  $b$ 도 1
- 그러므로  $a, b$  모두 1을 할당

```
1 + 3 = a
```

```
File "<ipython-input-15-3aa35fdab4b7>", line 1  
1 + 3 = a  
SyntaxError : can't assign to operator
```

- 주의! 등호 왼쪽에 표현식( $1+3$ )이 오면 안됨(변수가 와야 함)
- 표현식은 등호 오른쪽에 올 수 있음
- 이 때 오른쪽의 표현식이 평가됨

```
a = 1  
a = a + 1  
print a
```

```
2
```

- $a = a + 1 \rightarrow$  현재  $a$ 가 1이므로  $1 + 1$ , 즉,  $2 \rightarrow a = 2$
- 표현식은 등호 오른쪽에 올 수 있음

## ■ 파이썬 기초 문형

### 3. 할당문

```
c, d = 3, 4  
print c, d
```

```
x = y = z = 0  
print x, y, z
```

```
e = 3.5; f = 5.6  
print e, f
```

```
3 4  
0 0 0  
3.5 5.6
```

- `c, d = 3, 4` → `c`와 `d`에 각각 3과 4를 할당
- `x = y = z = 0` → 맨 오른쪽부터 이해해야 함
- `z = 0` → `z`에 0을 할당
- `y = z` → `z`의 값을 `y`에 할당 → `y = 0`
- `x = y` → `y`의 값을 `x`에 할당 → `x = 0`
- `print x, y, z` → 0 0 0
- 세미콜론(`;`): 하나의 문장이 끝났음을 의미
- 두 문장을 한 줄에 이어 쓸 때 세미콜론(`;`)을 사용
- 세미콜론(`;`)은 프로그램 가독성 문제로 자주 활용되지 않음

## ■ 파이썬 기초 문형

### 3. 할당문

- 두 변수의 값을 swap하는 방법

```
e = 3.5; f = 5.6  
e, f = f, e  
print e, f
```

5.6 3.5

- 스왑(swap): 값을 바꾸는 것을 의미
- $e, f = f, e \rightarrow e$ 에  $f$ 를 할당,  $f$ 에  $e$ 를 할당  $\rightarrow$  값을 스왑하는 문장
- print로 결과 확인하면 두 값이 바뀐 것을 알 수 있음

- 아래에서  $b = c + d$ 는 식(Expression)이 아니라 문(Statement)이기 때문에  $a$ 에 할당될 수 없다.

```
a = (b = c + d)
```

```
File "<ipython-input-29-79d71c957091>", line 1  
a = (b = c + d)
```

SyntaxError: invalid syntax

- $a = (b = c + d) \rightarrow$  잘못된 식
- $(b = c + d) \rightarrow$  식이 아니라 문
- 문 자체가  $a$ 에 할당될 수 없음

## ■ 파이썬 기초 문형

### 4. 확장 할당문

```
a = 1  
a += 4  
print a
```

5

- 확장 할당문: +=, -=, \*=, /=
- a += 4는 a = a + 4와 동일
- a += 4 → a에 4를 더한 후 다시 a에 할당

```
a = 10  
a -= 3  
print a
```

7

- a -= 3 → a에 3을 뺀 후 다시 a에 할당

```
a = 10  
a *= 2+3  
print a
```

50

- a \*= 2 + 3 → a에 2 + 3을 곱한 후 다시 a에 할당(2 + 3을 먼저 수행)
- 2 + 3 = 5, 그 다음 5 \* 10 = 50, a에 50을 할당

## ■ 파이썬 기초 문형

### 5. 객체와 할당

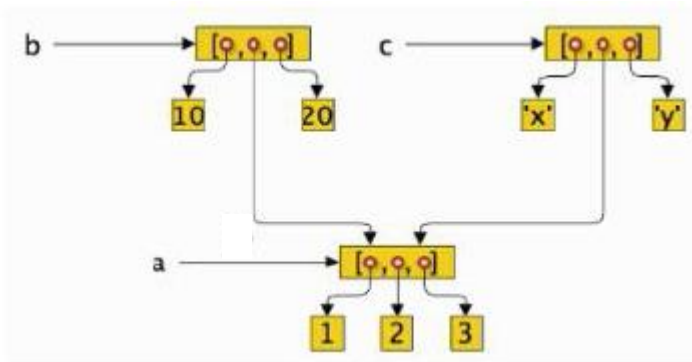
- 객체의 변수는 해당 객체의 레퍼런스를 지니고 있음
- `a = 1`이라는 Statement에서 `a`는 이름, `1`은 객체이며 `a` 변수는 `1`이라는 객체를 가리킨다.
  - 즉, `a` 변수는 `1` 객체의 레퍼런스를 지니고 있음

- `a = 1` → `a`는 변수 이름, `1`은 객체
- `a = 1` → `a`라는 변수가 어딘가에 있는 객체 `1`을 가리키는 것을 의미
- 가리킨다 → 레퍼런스를 지니고 있다

```
a = [1, 2, 3]
b = [10, a, 20]
c = ['x', a, 'y']
```

```
print a
print b
print c
```

```
[1, 2, 3]
[10, [1, 2, 3], 20]
['x', [1, 2, 3], 'y']
```



- `a = [1, 2, 3]` → 어딘가에 있는 `1, 2, 3`을 가리키는 형태
- 리스트 내부에는 `1, 2, 3`을 가리키는 레퍼런스 값이 존재
- `1, 2, 3`을 가리키는 레퍼런스가 존재하는 리스트를 `a`가 가리킴



## ■ 파이썬 기초 문형

### 5. 객체와 할당

- b도 어딘가에 있는 10, 20을 가리키는 레퍼런스 값이 존재
- 리스트 안에 있는 a는 [1, 2, 3]을 가리키는 형태
- c가 어떤 리스트를 가리키고 있음
- 리스트는 x, y라는 문자를 가리킴
- 리스트 안에 있는 a는 [1, 2, 3]을 가리키는 형태
- 파이썬에서는 모든 것이 다 객체
- 식별자 변수는 객체를 가리키는 형태
- `a = [1, 2, 3] → print a → [1, 2, 3]`
- `b = [10, a, 20] → a가 문자가 아닌 리스트[1, 2, 3]으로 나타남`
- `c = ['x', a, 'y'] → a가 문자가 아닌 리스트[1, 2, 3]으로 나타남`
- `a = [1, 2, 3]`에서 1은 인덱스 0, 2는 인덱스 1, 3은 인덱스 2

```
a[1] = 1000
```

```
print a  
print b  
print c
```

```
[1, 1000, 3]  
[10, [1, 1000, 3], 20]  
['x', [1, 1000, 3], 'y']
```

- `a[1] = 1000 → 인덱스 1에 1000을 넣음`
- 인덱스 1에 1000을 넣음 → 2 대신 1000을 넣음
- 리스트 a를 가리키고 있는 b, c 모두 변경됨
- b에 a가 연결되어 있기 때문에 a가 변하면 b도 변함
- 마찬가지로 c에 a가 연결되어 있기 때문에 a가 변하면 c도 변함