

1. 파이썬(www.python.org)

2. 에디터툴

1)파이참

2)이클립스 (pydev)

3)atom

4)visual studio code

5)jupyter notebook

6)anaconda(spyder), miniconda

3. 파일질라



학습순서:데이터타입- 연산자-제어문 -함수-클래스

파이썬은 모두 객체(속성 , 메소드):

- 객체.속성,
- 객체.메소드(..)

1. 특정값
2. 약속된 기호
3. 객체 = 클래스(초기값)

(C언어 에서 객체생성)

```
Test* p = malloc(sizeof(Test))
struct Test
{
    int a; //멤버데이터(속성)
    int b;
};
p->a = 10;
p->b = 20;
```

(C++ 객체 생성)

```
class Test
```

```
{
```

```
public:
```

```
    int a; // 멤버데이터
```

```
    int b;
```

```
// 멤버함수
```

```
    //void setData(Test*this, int x, int y)
```

```
void setData( int x, int y)
```

```
{
```

```
    this->a = x;
```

```
    this->b = y;
```

```
}
```

```
};
```

```
Test* p = new Test;
```

```
p->setData(100,200); // p->setData(p,100,200);
```

(java 객체생성)

```
class Test
```

```
{
```

```
    int a;
```

```
    int b;
```

```
    void setData( int x, int y)
```

```
    {
```

```
        this.a = x;
```

```
        this.b = y;
```

```
    }
```

```
}
```

```
Test p = new Test(); //Test* p 동일 (참조변수)
```

```
p.setData(100,200);
```



(python 객체생성)

class Test:

def __init__(self):

self.a = 0

self.b = 0

def setData(self, x, y):

self.a = x

self.b = y

p = Test()

p.a = 100

p.b = 200

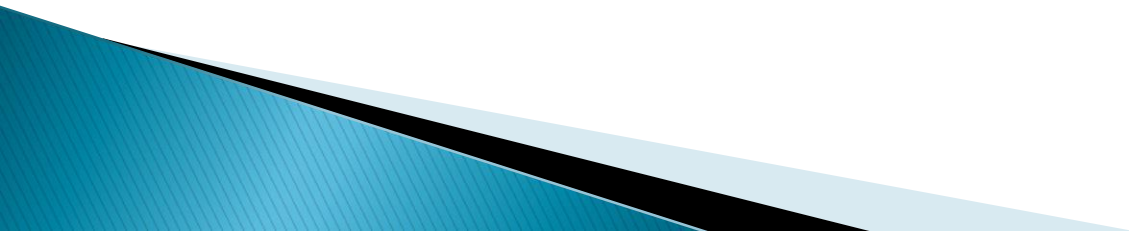
p.setData(10,20) #p.setData(p,10,20)

기본데이터

- 정수, 실수, bool, 복소수

복합데이터(시퀀스, iterable)

- str, bytes, 리스트, 튜플, 세트, 딕셔너리



복합데이터 타입

1) 순서있는(str, bytes, 리스트, 튜플)
인덱스, 슬라이싱

2) 순서없는(세트, 딕셔너리):
인덱스X, 슬라이싱X

3) mutable(변경가능(추가,삭제,수정)): 리스트, 딕셔너리, bytearray, set

4) immutable(X): str, bytes, 튜플 , frozen set

❖ 연산자의 우선순위

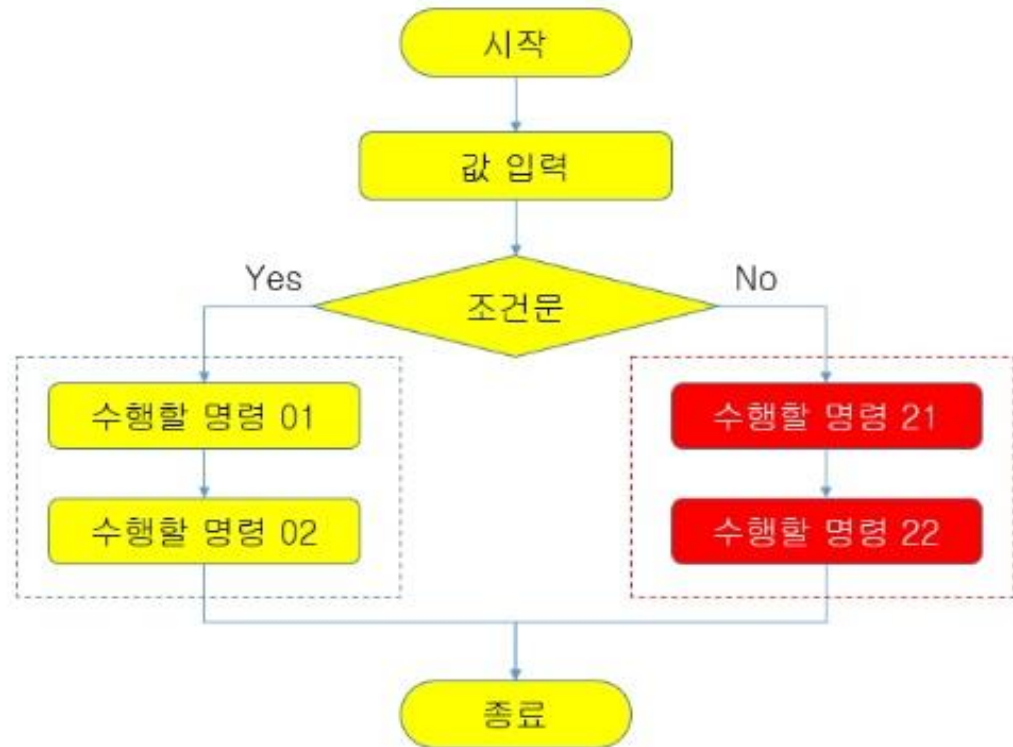
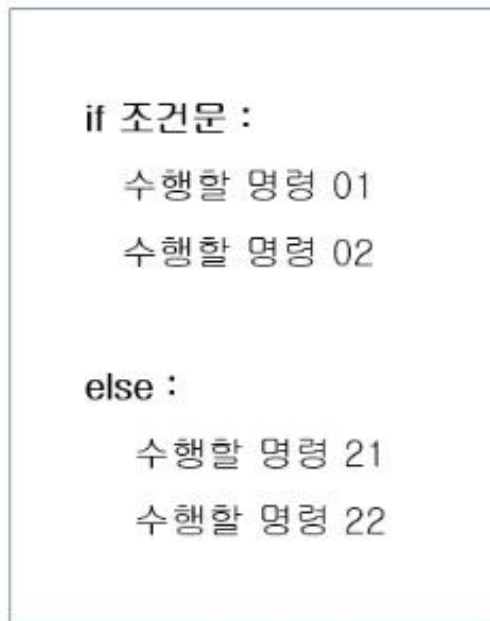
순위	연산자	의미
1	() [] { }	괄호, 리스트, 딕셔너리, 세트 등
2	**	지수 연산 (제곱)
3	+ - ~	단항 연산자 (양수, 음수, 보수)
4	* / % //	산술 연산자 (곱셈, 나눗셈, 나머지 연산)
5	+ -	산술 연산자 (덧셈, 뺄셈)
6	<< >>	비트 시프트 연산자 (왼쪽, 오른쪽 시프트 연산)
7	&	비트 논리곱 (2진수 논리곱)
8	^	비트 배타적 논리합 (2진수 배타적 논리합)
9		비트 논리합 (2진수 논리합)
10	< <= > >=	관계 연산자 (작음, 작거나 같음, 큼, 크거나 같음)
11	== !=	동등 연산자 (같음, 같지 않음)
12	= %= /= //= += -= *= **=	대입 연산자 (오른쪽을 왼쪽에 대입)
13	not , in, not in	논리 연산자 (True, False)
14	and	논리 연산자 (True, False)
15	or	논리 연산자 (True, False)
16	if ~ else	비교식 (주어진 조건이 참과 거짓으로 구분 실행)

●●● 대입 연산자의 종류

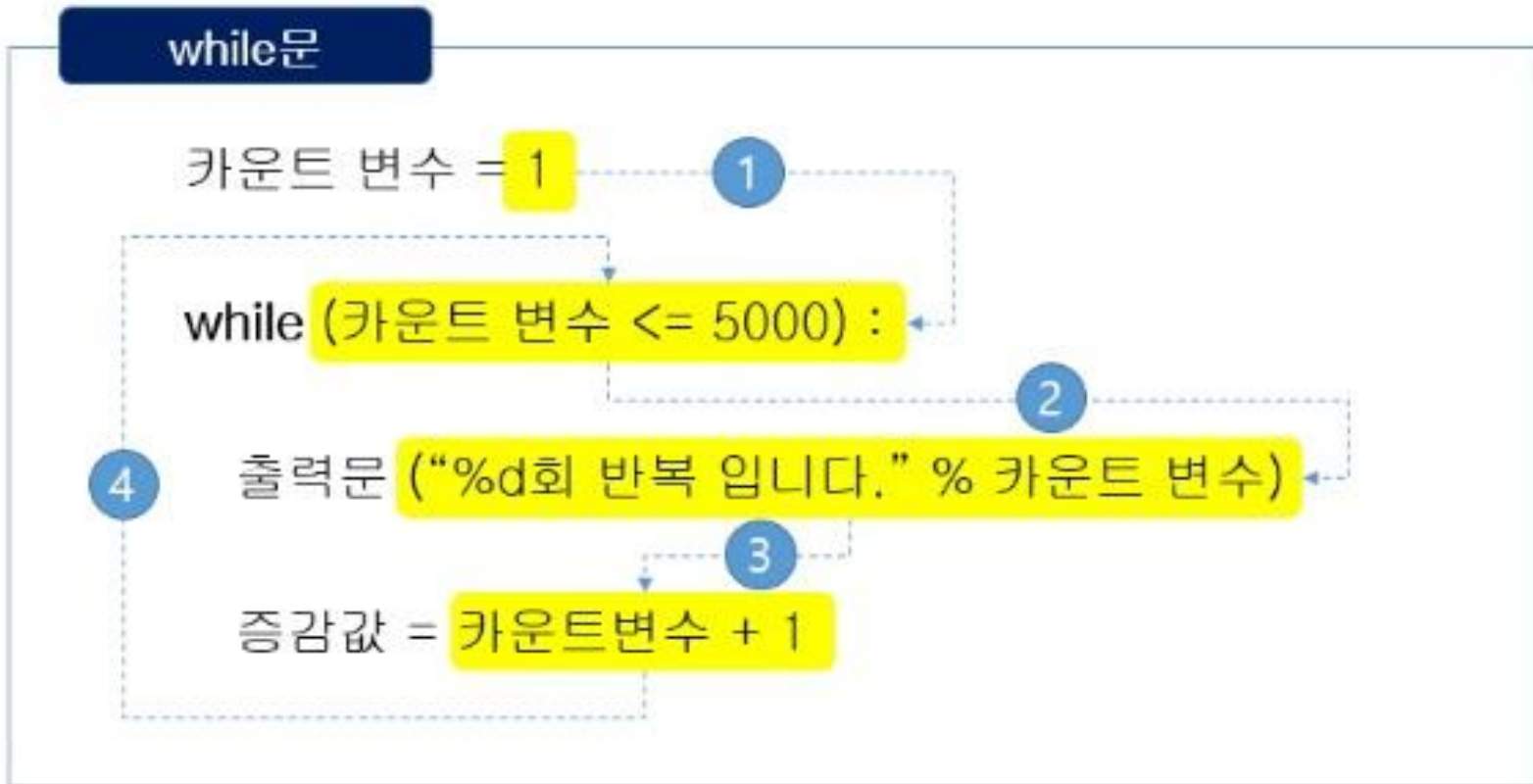
- 파이썬에서 사용되는 대입 연산자의 종류

연산자	의미	사용 예	설명
=	대입	<code>a = 5</code>	변수 a에 정수 5의 값을 대입
+=	덧셈 후 대입	<code>a += 5</code>	<code>a = a + 5</code> 와 동일한 연산식
-=	뺄셈 후 대입	<code>a -= 5</code>	<code>a = a - 5</code> 와 동일한 연산식
*=	곱셈 후 대입	<code>a *= 5</code>	<code>a = a * 5</code> 와 동일한 연산식
/=	나눗셈 후 대입	<code>a /= 5</code>	<code>a = a / 5</code> 와 동일한 연산식
//=	몫을 구한 후 대입	<code>a //= 5</code>	<code>a = a // 5</code> 와 동일한 연산식
%=	나머지를 구한 후 대입	<code>a %= 5</code>	<code>a = a % 5</code> 와 동일한 연산식
**=	제곱을 구한 후 대입	<code>a **= 3</code>	<code>a = a ** 5</code> 와 동일한 연산식

- if ~ else문의 문법 구조와 순서도



- while문의 문법 구조



- for문의 문법 구조

