UNISTUDY

파이썬

5강. 변수와 자료형(연산자)

1. 연산자의 정의

- 파이썬에서 제공되는 연산자(Operator)는 산술연산자, 관계연산자, 논리연산자, 대입연산자 등을 제 공한다.
- 각 연산자에 대한 연산 기호와 기능 설명이다.

구분	연산자	기능 설명
산술연산자	+, -, *, /, %, //, **	사칙연산, 나머지 반환, 몫 반환, 지수 승
관계연산자	==, !=, >, >=, <, <=	동등비교, 크기비교
논리연산자	and, or, not	논리곱, 논리합, 부정
대입연산자	=, (a, b), *	할다, a와 b 교환, 패킹(묶어서 할당)

2. 산술연산자

- 산술연산자는 사칙연산자와 나머지를 계산하는 연산자 그리고 지수(거듭제곱)를 계산하는 연산자로 구성되어 있다.

chapter02.lecture.step02_operator.py

```
num1 = 100 # 피연산자1
num2 = 20  # 피연산자2

add = num1 + num2 # 덧셈
print('add=', add)
sub = num1 - num2 # 뺄셈
print(sub)
mul = num1 * num2 # 곱셈
print(mul)
div = num1 / num2 # 나눗셈
print(div)
div2 = num1 % num2 # 나머지 계산
print(div2)
square = num1**2 # 제곱 계산
print(square)
```

Python Console

```
add= 120
sub= 80
mul= 2000
div= 5.0
div2= 0
square= 10000
```

2. 산술연산자

- 피연산자에 해당하는 num1과 num2 변수를 이용하여 사칙연산을 수행하는 결과이다.
- 덧셈, 뺄셈, 곱셈, 나눗셈의 계산식과 100을 20으로 나누어서 나머지 0을 반환하는 % 연산자, 그리고 100에 제곱을 취하여 10,000을 반환하는 ** 연산자의 계산 결과를 나타내고 있다. 만약 아주 큰 값이나 아주 작은 값을 나타낼 때 지수형식으로 출력되는 경우도 있다.
- 예를 들면 '1e+40'의 출력결과는 1 * 10의 40승과 동일하다.

3. 관계연산자

- 관계연산자는 관계식의 결과가 참이면 True, 거짓이면 False 값을 반환하는 연산자이다.
- 크게 동등비 교와 크기비교로 분류할 수 있다.

chapter02.lecture.step02_operator.py

```
# (1) 동등비교
bool_result = num1 == num2 # 두 변수의 값이 같은지 비교
print(bool_result)
bool_result = num1 != num2 # 두 변수의 값이 다른지 비교
print(bool_result)

# (2) 크기비교
bool_result = num1 > num2 # num1값이 큰지 비교
print(bool_result)
bool_result = num1 >= num2 # num1값이 크거나 같은지 비교
print(bool_result)
bool_result = num1 < num2 # num2 값이 큰지 비교
print(bool_result)
bool_result = num1 <= num2 # num2 값이 크거나 같은지
비교
print(bool_result)
```

Python Console

```
False
True

True
True
False
False
```

3. 관계연산자

- 관계연산자의 결과는 조건이 참인 경우 True, 거짓인 경우에는 False의 상수 값으로 반환한다.
- 이렇게 반환된 값은 조건문이나 반복문에서 비교 판단할 경우 사용된다

4. 논리연산자

- 산술연산자와 관계연산자를 이용해서 작성된 논리식이 참이면 True, 거짓이면 False 값을 반환하는 연산자이다.

chapter02.lecture.step02_operator.py

```
# 두 관계식이 같은지 판단
log result = num1 >= 50 and num2 <=10
print(log_result)


# 두 관계식 중 하나라도 같은지 판단
log_result = num1 >= 50 or num2 <=10
print(log_result)

log_result = num1 >= 50 # 관계식 판단
print(log result)

# 괄호 안의 관계식 판단 결과에 대한 부정
log_result = not(num1 >= 50)
print(log_result)
```

Python Console

```
False


True

True

False
```

4. 논리연산자

- 논리연산자는 and, or, not 명령어로 제공된다. and와 or는 양변에 관계식을 가지고 있다.
- and는 두 관계식이 모두 참이면 True를 반환하고, or는 적어도 한쪽 관계식이 참이면 True를 반환한다. 한편 not는 논리형 자료를 대상으로 부정을 적용한 값을 반환한다.
- 예를 들면 True이면 False를 반환하고, False이면 True를 반환한다. 이러한 특성을 이용하여 두 가지 상황을 서로 반전시키는 스위칭 기법에 이용되기도 한다.

5. 대입연산자

- 대입연산자는 우변의 값을 좌변의 변수에 할당하는 연산자(=), 두 변수의 값을 교환하는 연산자, 그리 고 여러 개의 값을 묶어서 변수에 할당하는 연산자(*)로 구성되어 있다.

chapter02.lecture.step02_operator.py

```
# (1) 변수에 값 할당(=)
i = tot = 10 # i=10; tot = 10
i += 1 # i = i + 1
tot += i # tot = tot + i
print(i, tot)

# 같은 줄에 중복 출력
print('출력1', end=' , ') # end='구분자'
print('출력2')

v1, v2 = 100, 200

# (2) 변수 교체
```

Python Console

```
11 21
출력1 , 출력2
```

5. 대입연산자

- 대입연산자는 우변의 값을 좌변의 변수에 할당하는 연산자(=), 두 변수의 값을 교환하는 연산자, 그리 고 여러 개의 값을 묶어서 변수에 할당하는 연산자(*)로 구성되어 있다.

```
v2, v1 = v1, v2
print(v1, v2) # 200 100
```

```
100 200
```

```
# (3) 패킹(packing) 할당
lst = [1, 2, 3, 4, 5]
v1, *v2 = lst
print(v1, v2) # 1 [2, 3, 4, 5]
```

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

```
*v1, v2 = lst
print(v1, v2) # [1, 2, 3, 4] 5
```

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

5. 대입연산자

(1) 변수에 값 할당(=)
i와 tot 변수에 10을 동시에 할당하고, i에 1를 더하고, tot에 i를 더하여 11과 21이 출력된다. print() 함수에 두 개의 변수를 인자로 넣으면 같은 줄에 중복하여 출력된다. 또한 현재 출력 값과 다음에 출력되는 값을 같은 줄에 중복 출력하고, 두 값을 콤마(,)로 구분하기 위해서 print() 함수에 end = ‘,’ 형식으로 인자를 넣는다.

(2) 변수 교체
임시 변수 없이 v1과 v2 변수의 값을 서로 교체(swapping)하는 기능으로 ‘=’ 연산자를 기준으로 우변은 교체할 변수 또는 값, 좌변은 교체될 변수 또는 값을 작성한다.

(3) 패킹(packing) 할당
여러 개의 값을 묶어서 변수에 할당할 경우 패킹 연산자(*)를 이용한다. lst 변수는 5개의 원소를 갖는 배열 형식의 변수이다. 이 변수를 대상으로 v1에는 첫 번째 원소가 할당되고, 나머지 패킹 연산자로 지정한 v2 변수에는 나머지 4개 원소가 묶여서 할당된다. 반대로 v1 변수에 *기호를 붙이면 앞부분 4개 원소가 묶여서 v1 변수에 할당되고, 마지막 5번째 원소만 v2 변수에 할당된다.

THANK YOU