

---

# 연산자의 이해

2주차\_03

한 동 대 학 교  
김경미 교수

# 연산자(Operators)

---

- 연산자는 특수한 심볼(symbols)로 덧셈이나 곱셈과 같은 계산을 표현하는 기호이다
- 연산자에 적용되는 값(value)은 피연산자(operand)라고 한다

산술 연산자	관계 연산자	논리연산자
+ , - , * , ** , / , // , % , += , -= , *= , /=	< , > , <= , >= , = , !=	and , or , not , in

# 산술 연산자

---

연산자	의미	예제	결과
+	더하기	>>> 1 + 5	6
-	빼기	>>> 5 - 1	4
*	곱하기	>>> 12 * 12	144
/	나누기	>>> 15 / 2	7.5
//	나누기(몫만)	>>> 15 // 2	7
**	제곱	>>> 3 ** 2	9
%	나머지 값	>>> 9 % 2	1

# 산술연산자 예제

---

- BMI(Body Mass Index) 계산하기
- 체중(kg)/신장(m)<sup>2</sup>
- 자신의 체중은 변수 weight, 신장은 height에 저장한 후 BMI 값을 계산하여 출력한다
- 예를 들면
  - weight = 55      ## kg단위
  - Height = 1.70    ## m단위, cm가 아님

# 산술연산자 예제, 코드

---

```
weight = 75
```

```
height = 1.83
```

```
bmi = weight / (height * height)
```

```
print("나의 체중은 ", weight, "kg, 키는 ", height, "m 입니다.")
```

```
print("계산한 BMI 지수는 ", bmi)
```

# 관계 연산자

---

연산자	의미	예제	결과
<	작다	>>> 5 < 3	False
<=	작거나 같다	>>> 3 <= 5	True
>	크다	>>> 12 > 15	False
>=	크거나 같다	>>> 15 >= 12	True
==	같은가?	>>> 3 == 2	False
!=	같지 않은가?	>>> 3 != 2	True

# 관계 연산자 사용하는 경우

---

- 조건문이나 반복문을 활용 할 때, 주로 사용한다
  - BMI 값이 25보다 큰 경우
  - 키가 150보다 작거나 같은 경우
  - 한번도 실행되지 않은 경우
- 변수에 저장된 값이 100보다 작을 때 까지

# 논리 연산자

---

연산자	의미	예제	결과
Not	부정	>>> not (5 > 3)	False
And	~이고 ~이면 (모든 조건이 참일 경우만 결과가 참)	>>> (4%2 == 0) and ( 2 > 1 )	True
Or	~이거나 ~이면 (모든 조건이 거짓일 경우만 결과가 거짓)	>>> (3%2 ==0) or ( 2 > 1 )	True



# 논리연산자 사용하는 경우

---

- 관계연산자를 여러 개 묶어서 논리적인 구조를 만들 때 사용한다
  - 키가 150보다 작거나, 체중이 45kg보다 작은 경우
  - 키가 150과 180 사이에 있는 경우
  - 비밀번호가 맞지 않고, 비밀번호 입력한 횟수가 3번 이상인 경우

# 연산자 예제

---

```
>>> 1 + 1
```

```
2
```

```
>>> 2**3
```

```
8
```

```
>>> 2*3
```

```
6
```

```
>>> 8 % 3
```

```
2
```

```
>>> 8 > 5
```

```
True
```

```
>>> 8 < 5
```

```
False
```

# 연산자 우선 순위(Precedence)

---

- **문장 하나에 여러 종류의 연산자가 표현될 때**
  - 어떤 연산자를 먼저 처리할 지 정하는 기준
  - 1순위 그룹: 산술연산자
  - 2순위 그룹: 관계연산자
  - 3순위 그룹: 논리연산자
- **프로그래머가 원하는 연산 순서를 사용하고 싶다면 괄호를 사용해야 한다**

# 연산자 우선 순위 기준

연산자	설명
lambda	람다 표현식
if – else	
or	
and	
not x	
in, not in, is, is not, <, <=, >, >=, !=, ==	
<<, >>	자리수 이동(shift)
+, -	
*, /, //, %	
+x, -x, ~x	
**	제곱
(expressions...), [expressions...], {key: value...}, {expressions...}	튜플(tuple) 바인딩 또는 출력, 리스트(list) 출력, 사전 출력, 집합 출력

# 연산자 우선 순위 예제

---

- 다음 수식의 결과를 화면에 출력해 보자

```
>>> 5 + 10 * 15 / 5 - 10
```

```
>>> 25
```

```
>>> (5 + 10) * 15 / 5 - 10
```

```
>>> 35
```

```
>>> (5 + 10 * 15) / 5 - 10
```

```
>>> 21
```

# 이제부터...

---

- 입력 출력하는 문장을 배우고,
- 다양한 연산자를 마음껏 활용할 수 있게 해주는
  - 조건절
  - 반복절

# 요약

---

- 연산자와 피연산자 이해
- 산술연산자
- 관계연산자
- 논리연산자
- 연산자 우선순위

---

# 감사합니다

2주차\_03 연산자의 이해