

CS101 – 연산자 및 튜플

Lecture 6

School of Computing
KAIST

학습 목표:

- 연산자를 통한 식을 이해하고 작성할 수 있다.
- 기초 자료 구조인 튜플을 이해하고 작성할 수 있다.

숫자의 연산에는 $+$, $-$, $*$, $/$, $//$, $\%$, $**$ 연산자를 사용할 수 있습니다.

$**$ 는 거듭제곱 연산입니다. ($a ** b = a^b$)

```
>>> 2**16
```

```
65536
```

$\%$ 는 나눗셈의 나머지 연산입니다.

```
>>> 7 % 3
```

```
1
```

$//$ 는 정수 나눗셈(소수 부분을 제외한 나눗셈) 연산입니다.

```
>>> 13.0 // 4.0
```

```
3.0
```

```
>>> 9 / 7
```

```
1.2857142857142858
```

식은 객체, 변수, 연산자, 함수 호출의 조합으로 이루어집니다.

`3.0 * (2 ** 15 - 12 / 4) + 4 ** 3`

수학에서 사용하는 것과 같이, 연산자는 우선순위에 따라 적용됩니다.

- ① 거듭제곱 (`**`)
- ② 곱셈, 나눗셈 (`*`, `/`, `//`, `%`)
- ③ 덧셈, 뺄셈 (`+`, `-`)

연산자의 계산 순서가 헷갈릴 때는 괄호를 쓰세요.

예시) $\frac{a}{2\pi}$ 는 `a/2*pi` 가 아닙니다.

`a/(2*pi)` 나, `a/2/pi` 와 같이 써야 합니다.

모든 연산자는 복소수에서도 사용할 수 있습니다.

문자열 연산

+ 와 * 연산자는 문자열에서도 사용할 수 있습니다.

```
>>> "Hello" + "CS101"
```

```
'HelloCS101'
```

```
>>> "CS101" * 8
```

```
'CS101 CS101 CS101 CS101 CS101 CS101 CS101 CS101 '
```

논리식

논리식은 계산 결과가 논리값인 식입니다.
이 식은 **if** 와 **while** 문에서 사용됩니다.

다음 연산자들의 연산 결과는 논리값입니다.

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

```
>>> 3 < 5
```

```
True
```

```
>>> 27 == 14
```

← '=='은 두 값이 같은지 비교하는 연산자입니다.
대입 연산자인 = 와 헷갈리지 마세요!

```
False
```

```
>>> 3.14 != 3.14
```

```
False
```

```
>>> 3.14 >= 3.14
```

```
True
```

```
>>> "Cheong" < "Choe"
```

```
True
```

```
>>> "3" == 3
```

```
False
```

논리 연산자

`not`, `and`, `or` 는 논리 연산자입니다.

```
(not True) == False
```

```
(not False) == True
```

```
(False and False) == False
```

```
(False and True) == False
```

```
(True and False) == False
```

```
(True and True) == True
```

```
(False or False) == False
```

```
(False or True) == True
```

```
(True or False) == True
```

```
(True or True) == True
```

주의: 만약 논리 연산자 왼쪽 값으로 연산의 결과가 결정된다면

(예. `False and ...` 또는 `True or ...`)

Python은 논리 연산자 오른쪽 값을 계산하지 않습니다.

튜플은 다른 객체들을 포함하는 객체입니다.

```
>>> position = (3.14, -5, 7.5)
>>> profs = ("In-Young Ko", "Sunghee Choi", "Lee
    YoungHee", "Duksan Ryu", "Key-Sun Choi")
```

튜플은 **tuple** 형태를 가진 하나의 객체입니다.

```
>>> print(position, type(position))
(3.14, -5, 7.5) <class 'tuple'>
```

튜플이 포함하는 객체들은 풀 수 있습니다.

```
>>> x, y, z = position
>>> print(x)
3.14
```

튜플의 값을 한 번에 풀고, 다시 묶으려면

```
>>> a, b = ("aa", "bb")
>>> a, b = b, a
>>> print(b)
aa
```

정리 및 연습

본 강의 학습 목표:

- 연산자를 통한 식을 이해하고 작성할 수 있다.
- 기초 자료 구조인 튜플을 이해하고 작성할 수 있다.

다음 강의 학습 목표:

- 튜플을 활용하여 디지털 사진을 표현하는 방식을 이해할 수 있다.
- 디지털 사진을 색 반전이나 흑백 모드로 변환할 수 있다.