

# UNIT 5

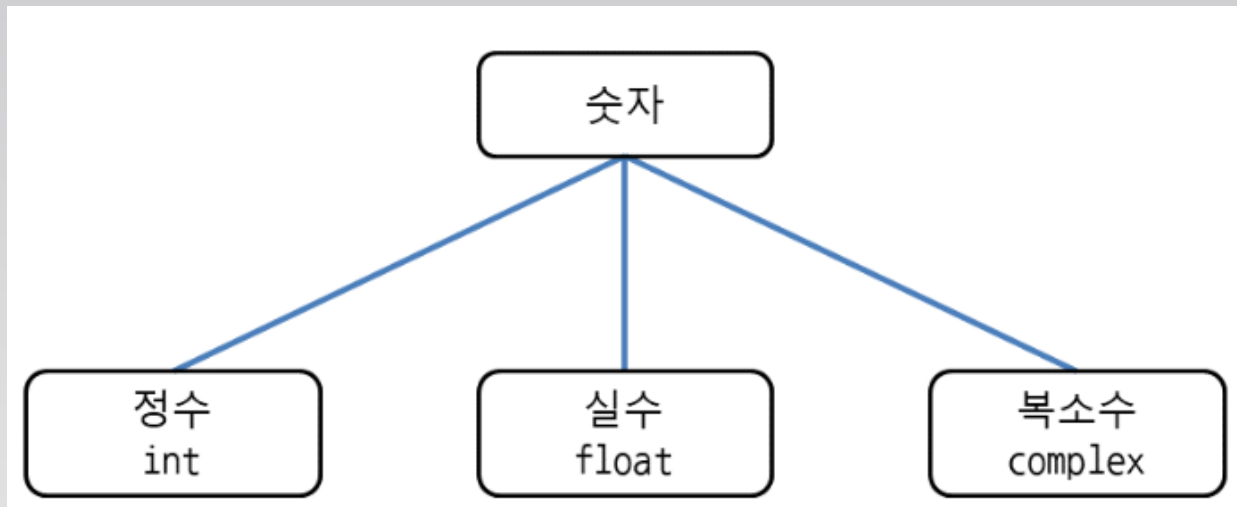
## 숫자 계산하기

# 5 숫자 계산하기

## » 숫자 계산하기

- 파이썬을 계산기처럼 사용해보기
- 숫자 계산을 하기 전에 먼저 숫자의 자료형(타입)부터 살펴보자
- 파이썬에서는 숫자의 자료형에 따라 결과가 달라질 수 있으므로 이 부분은 정확하게 구분할 필요가 있음
- 다음과 같이 파이썬에서는 숫자를 정수, 실수, 복소수로 구분

### ▼ 그림 파이썬의 숫자 자료형



# 5 숫자 계산하기

## » 숫자 계산하기

- 보통 프로그래밍에서는 정수와 실수를 주로 사용하며 복소수는 공학 분야에서 주로 쓰임
- 이 책에서는 정수와 실수 위주로 설명하겠음

# 5.1 정수 계산하기

## » 정수 계산하기

- 파이썬 IDLE를 실행하거나 콘솔(터미널, 명령 프롬프트)에서 파이썬을 실행  
(Windows는 Win+R > cmd 입력 > python 입력, 리눅스와 macOS는 터미널에서 python3을 입력)

```
C:\Users\dojang>python
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

# 5.1 정수 계산하기

## » 사칙연산

- 파이썬 프롬프트가 나오면 코드와 계산식을 입력받을 준비가 된 상태
- >>>에 1 + 1을 입력한 뒤 엔터 키를 누르면 결과값 2가 나옴

```
>>> 1 + 1  
2
```

- 덧셈뿐만 아니라 뺄셈, 곱셈도 가능

```
>>> 1 - 2  
-1  
>>> 2 * 2  
4
```

- 이번에는 나눗셈

```
>>> 5 / 2  
2.5
```

# 5.1 정수 계산하기

## » 사칙연산

- 파이썬 프롬프트가 나오면 코드와 계산식을 입력받을 준비가 된 상태
- >>>에 1 + 1을 입력한 뒤 엔터 키를 누르면 결과값 2가 나옴

```
>>> 1 + 1  
2
```

- 덧셈뿐만 아니라 뺄셈, 곱셈도 가능

```
>>> 1 - 2  
-1  
>>> 2 * 2  
4
```

# 5.1 정수 계산하기

## » 사칙연산

- 이번에는 나눗셈

```
>>> 5 / 2  
2.5
```

- 5 나누기 2는 2.5가 나옴
- 당연한 결과일 수도 있지만 이 부분이 파이썬 2와 파이썬 3의 차이점
- 파이썬 2에서  $5 / 2$ 는 2.5가 아닌 2가 나오는데, 정수끼리 나눗셈 결과는 정수가 나오도록 정했기 때문임
- 파이썬 3는 정수끼리 나눗셈을 해도 실수가 나옴
- 4 나누기 2를 계산해보면 정말 그런지 확인할 수 있음

```
>>> 4 / 2  
2.0
```

- 파이썬 3은 나눗셈이 완전히 나누어 떨어져도 실수가 나옴

# 5.1 정수 계산하기

## » 나눗셈 후 소수점 이하를 버리는 // 연산자

- 파이썬 3에서 정수끼리 나눗셈 결과가 정수로 나도록 만들어 보자
- //로 나눗셈을 하면 됨

```
>>> 5 // 2
2
>>> 4 // 2
2
```

- //은 버림 나눗셈(floor division)이라고 부르며 나눗셈의 결과에서 소수점 이하는 버림
- 실수에 // 연산자를 사용하면 결과는 실수가 나오며 소수점 이하는 버림
- 결과는 항상 .0으로 끝남

```
>>> 5.5 // 2
2.0
>>> 4 // 2.0
2.0
>>> 4.1 // 2.1
1.0
```



# 5.1 정수 계산하기

## » 나눗셈 후 나머지를 구하는 % 연산자

- 나눗셈 후 나누어 떨어지지 않을 때 나머지를 구해보자

```
>>> 5 % 2  
1
```

- 5를 2로 나누면 두 번 나눌 수 있고 1이 남음
- %는 두 수를 나누었을 때 나머지만 구하며 모듈로(modulo) 연산자라고 부름
- 몫은 버림 나눗셈(//)으로 구할 수 있음

## » 거듭제곱을 구하는 \*\* 연산자

- 파이썬에서는 거듭제곱도 쉽게 구할 수 있음

```
>>> 2 ** 10  
1024
```

- \*\*은 거듭제곱 연산자이며 숫자를 특정 횟수만큼 곱함
- $2^{**}10$ 은 1024이고  $2^{10}$ 을 뜻함

# 5.1 정수 계산하기

## » 값을 정수로 만들기

- 계산 결과가 실수로 나왔을 때 강제로 정수로 만들어보기
- int에 괄호를 붙이고 숫자 또는 계산식을 넣으면 됨
- int에 문자열을 넣어도 정수로 만들 수 있음
- 단, 정수로 된 문자열이어야 함

- int(숫자)
- int(계산식)
- int('문자열')

```
>>> int(3.3)
3
>>> int(5 / 2)
2
>>> int('10')
10
```

- int는 정수(integer)를 뜻하며 값을 정수로 만들어 줌(소수점 이하는 버림)

# 5.1 정수 계산하기

## » 객체의 자료형 알아내기

- 지금까지 사용한 숫자가 정수가 맞는지 확인해보자

• `type(값)`

```
>>> type(10)
<class 'int'>
```

- `type`은 말그대로 객체의 타입(자료형)을 알아내는 함수
- `type`에 괄호를 붙이고 10을 넣어보면 `<class 'int'>`라고 나오는데 정수(int) 클래스라는 뜻
- 파이썬에서는 숫자도 객체(object)이며, 객체는 클래스(class)로 표현함
- 참고로 앞에서 사용한 `int(3.3)`은 실수 3.3을 int 클래스로 된 객체로 만든다는 뜻

# 5.1 정수 계산하기

## » 참고 | 몫과 나머지를 함께 구하기

- 몫과 나머지를 함께 구하려면 divmod를 사용

```
>>> divmod(5, 2)
(2, 1)
```

- 지금까지 계산 결과가 1, 2, 1024처럼 값만 나왔지만 divmod는 결과가 (2, 1)처럼 나옴
- 파이썬에서 값을 괄호로 묶은 형태를 튜플(tuple)이라고 하며 값 여러 개를 모아서 표현할 때 사용
- 튜플은 변수 여러 개에 저장할 수 있는데 divmod의 결과가 튜플로 나오므로 몫과 나머지는 변수 두 개에 저장할 수 있음

```
>>> quotient, remainder = divmod(5, 2)
>>> print(quotient, remainder)
2 1
```

# 5.1 정수 계산하기

## » 참고 | 진수, 8진수, 16진수

- 정수는 10진수 이외에도 2진수, 8진수, 16진수로도 표현할 수 있음
- 2진수: 숫자 앞에 0b를 붙이며 0과 1을 사용
- 8진수: 숫자 앞에 0o(숫자 0과 소문자 o)를 붙이며 0부터 7까지 사용
- 16진수: 숫자 앞에 0x 또는 0X를 붙이며 0부터 9, A부터 F까지 사용  
(소문자 a부터 f도 가능)

```
>>> 0b110
6
>>> 0o10
8
>>> 0xF
15
```

## 5.2 실수 계산하기

### » 실수 계산하기

- 실수끼리 계산해보자
- >>>에  $3.5 + 2.1$ 를 입력해보자

```
>>> 3.5 + 2.1  
5.6
```

- 덧셈의 결과인 5.6이 잘 나왔음
- 이번에는 뺄셈, 곱셈, 나눗셈을 해보자

```
>>> 4.3 - 2.7  
1.5999999999999996  
>>> 1.5 * 3.1  
4.65  
>>> 5.5 / 3.1  
1.7741935483870968
```

## 5.2 실수 계산하기

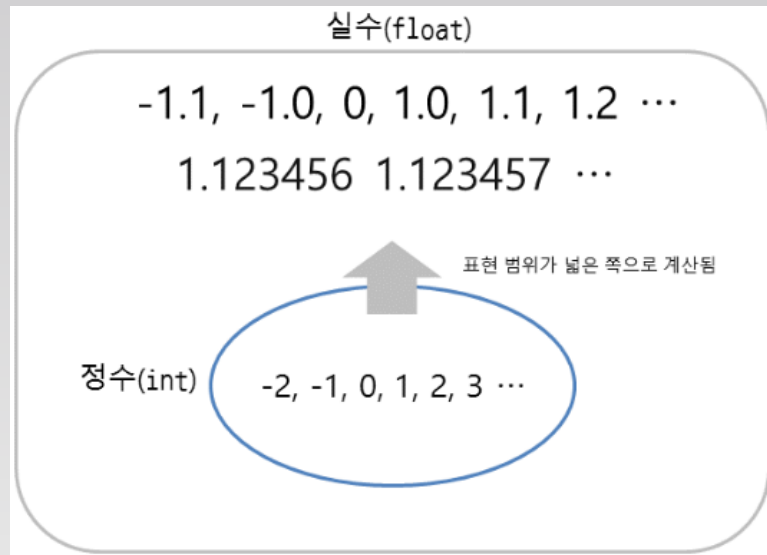
### » 실수와 정수를 함께 계산하면?

- 실수와 정수를 함께 계산해보자

```
>>> 4.2 + 5  
9.2
```

- 실수인 9.2가 나옴
- 실수와 정수를 함께 계산하면 표현 범위가 넓은 실수로 계산됨(실수가 정수보다 표현 범위가 넓음)

#### ▼ 그림 표현 범위가 넓은 쪽으로 계산됨



## 5.2 실수 계산하기

### » 값을 실수로 만들기

- 숫자 또는 계산 결과를 강제로 실수로 만들려면 어떻게 해야 할까요?
- float에 괄호를 붙이고 숫자 또는 계산식을 넣으면 됨
- float에 문자열을 넣어도 실수로 만들 수 있음
- 단, 실수 또는 정수로 된 문자열이어야 함

- float(숫자)
- float(계산식)
- float('문자열')

```
>>> float(5)
5.0
>>> float(1 + 2)
3.0
>>> float('5.3')
5.3
```



## 5.2 실수 계산하기

### » 값을 실수로 만들기

- float는 부동소수점(floating point)에서 따왔으며 값을 실수로 만들어줌
- 즉, 실수는 float 자료형이며 type에 실수를 넣어보면 <class 'float'>가 나옴

```
>>> type(3.5)
<class 'float'>
```

### » 참고 | 복소수

- 파이썬에서는 실수부와 허수부로 이루어진 복소수(complex number)도 사용할 수 있음
- 이때 허수부는 숫자 뒤에 j를 붙임(수학에서는 허수를 i로 표현하지만 공학에서는 j를 사용)

```
>>> 1.2+1.3j
(1.2+1.3j)
```

- 두 실수를 복소수로 만들 때는 complex를 사용하면 됨

```
>>> complex(1.2, 1.3)
(1.2+1.3j)
```

## 5.2 실수 계산하기

### » 참고 | 스크립트 파일에서 계산 결과가 출력되지 않아요.

- 스크립트 파일에서  $1 + 1$ 처럼 계산식만 넣으면 결과가 출력되지 않음

print\_add.py

```
1 + 1
```

실행 결과

(아무것도 출력되지 않음)

- 스크립트 파일에서 계산 결과를 출력하려면 print 함수를 사용해야 함

print\_add.py

```
print(1 + 1)
```

실행 결과

2

- 파이썬 셸은 결과를 즉시 보는 용도라서 값 또는 계산식만 넣어도 결과가 출력됨
- 스크립트 파일에서 값 또는 계산식의 결과를 출력하려면 반드시 print 함수를 사용해야 함

## 5.3 괄호 사용하기

### » 괄호 사용하기

- 그럼  $35 + 1 * 2$ 의 결과는 무엇일까요?
- 식에서는 덧셈, 곱셈 순으로 나와있지만 곱셈을 덧셈보다 먼저 계산하므로 72가 아닌 37이 정답임
- 파이썬에서도 마찬가지임

```
>>> 35 + 1 * 2  
37
```

- 곱셈보다 덧셈을 먼저 계산하고 싶다면 덧셈 부분을 괄호로 묶어주면 됨

```
>>> (35 + 1) * 2  
72
```

- 덧셈, 뺄셈, 곱셈, 나눗셈이 함께 있을 때는 곱셈과 나눗셈부터 계산
- 덧셈과 뺄셈을 먼저 계산하고 싶다면 괄호로 묶어줌

## 5.3 괄호 사용하기

### » 괄호 사용하기

- 실생활에서는 1과 1.0을 구분하지 않지만 컴퓨터는 1과 1.0을 정수와 실수로 구분함
- 파이썬에서도 정수와 실수는 구분해서 처리함
- 나눗셈 연산자 /의 결과는 실수, 버림 나눗셈 //의 결과는 정수