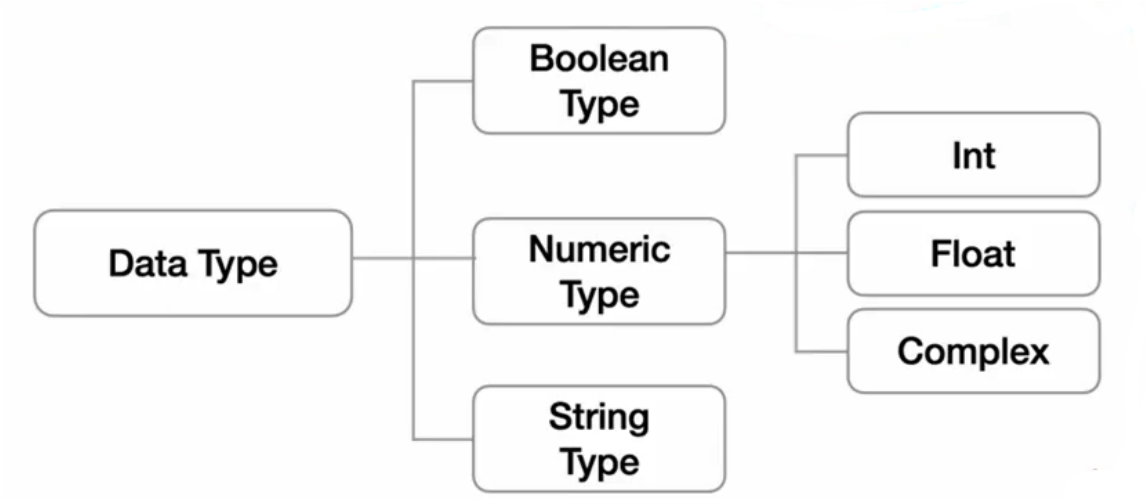
자료형(DataType)

작성일시	@2022년 7월 18일 오전 11:00
내용	문법 자료구조 파이썬
주차	1주차
자료	

자료형(Datatype)

- 프로그래밍에서 사용할 수 있는 데이터들의 종류



수치형 자료형

- 정수 자료형(int)
- 진수표현
  - 2진수(binary): 0b
  - 8진수(octal): 0o
  - 16진수(hexadecimal): 0x

```
print(0b10) # 2
print(0o30) # 24
print(0x10) # 10
```

- 실수 자료형(float)
  - 부동소수점
    - 컴퓨터는 2진수, 사람은 10진법을 사용
    - 변환하는 과정에서 10진법의 **근사값**만 표시
    - Floating point rounding error
  - 부동소수점 해결방법
    - 매우 작은 수인지 확인하거나 math 모듈 활용

```
a = 3.2 - 3.1 # 0.1000000000009
b = 1.2 - 1.1 # 0.0999999999987

# 1. 임의의 작은 수 활용
print(abs(a - b) <= (1e - 10) # True

# 2. python 3.5 이상
import math
print(math.isclose(a, b)) #True
```

문자열 자료형(String Datatype)

문자열 자료형의 정의

- 모든 문자열은 str 타입
- 문자열은 작은따옴표(')나 큰따옴표(")를 활용하여 표기
  - 문자열을 묶을 때는 동일한 문장부호 활용
  - PEP8에서는 소스코드 내에서 하나의 문장부호를 선택하여 유지

중첩 따옴표

- 따옴표 안에 따옴표를 표현하는 경우
  - 작은따옴표가 들어있는 경우 큰따옴표로 문자열 생성

- 큰따옴표가 들어있는 경우 작은따옴표로 문자열 생성

```
print("문자열 안에 '작은따옴표'를 사용하려면 큰따옴표로 묶는다")
print('문자열 안에 "큰따옴표"를 사용하려면 작은따옴표로 묶는다')
```

삼중 따옴표(Triple Quotes)

- 따옴표 안에 따옴표를 넣을 때
- 여러 줄을 나눠 입력할 때 편리

```
print('''문자열 안에 '작은 따옴표'나
"큰따옴표"를 사용할 수 있고
여러 줄을 사용할 때 편리하다''')
```

Escape Sequence

- 역슬래시(backslash)뒤에 특정 문자가 와서 특수한 기능을 하는 문자 조합

예약문자	내용(의미)
\n	줄바꿈
\t	탭
\r	캐리지 리턴
\0	널(null)
\\	\
\’	단일이용부호(’)
\”	다중이용부호(”)

```
print('철수 \'안녕\' ')
print('이 다음은 엔터. \n 그리고 탭\t탭')

#철수 '안녕'
#이 다음은 엔터.
# 그리고 탭 탭
```

문자열 연산

- 덧셈
  - String Concatenation

```
print('Hello' + 'World')
#HelloWorld
```

- 곱셈

```
print('Python' * 3)
#PythonPythonPython
```

String Interpolation(문자열을 활용해 변수를 만드는 방법)

- str.format()
- f-strings

```
name = 'Kim'
score = 4.5

print(f'Hello, {name}! 성적은 {score}')

#Hello, Kim! 성적은 4.5
```

None

- 값이 없음을 표현하게 위해 None 타입이 존재
- 일반적으로 반환 값이 없는 함수에서 사용하기도 함

Boolean

- 논리 자료형으로 참과 거짓을 표현하는 자료형
- **True** or **Flase**

비교 연산자

- 비교 / 논리 연산에서 활용

연산자	내용
<	미만

연산자	내용
<=	이하
>	이상
>=	초과
==	같음
!=	같지 않음
is	객체 아이덴티티(OOP)
is not	객체 아이덴티티가 아닌 경우

논리 연산자

- 여러 조건이 있을 때
  - 모든 조건을 만족하거나(AND), 여러 조건 중 하나만 만족해야 될 때(OR) 특정 코드를 실행하고 싶을 때 사용
  - 일반적으로 비교 연산자와 함께 사용

연산자	내용
A and B	A와 B 모두 True시 True
A or B	A와 B 모두 False시 False
NOT	True를 Flase로, Flase를 True로

- Falsy : False 는 아니지만 False로 취급되는 다양한 값
  - 0, 0.0, (), [], {}, NONE, “”(빈 문자열)
- 논리 연산자 우선순위
  - NOT → AND → OR

논리 연산자의 단축평가

- 결과가 확실한 경우 두번째 값은 확인하지 않고 첫번째 값 반환
- AND 연산에서 첫번째 값이 False인 경우 무조건 False → 첫번째 값 반환
- OR 연산에서 첫번째 값이 True인 경우 무조건 True → 첫번째 값 반환
- 0은 False, 1은 True

```
print(3 and 5) # 5
print(3 and 0) # 0
print(0 and 3) # 0
```

```
print(0 and 0) # 0

print(5 or 3) # 5
print(3 or 0) # 3
print(0 or 3) # 3
print(0 or 0) # 0
```

```
a = 5 and 4
print(a) # 4

b = 5 or 3
print(b) # 5

c = 0 and 5
print(c) # 0

d = 5 or 0
print(d) # 5
```