

파이썬 프로그래밍

---

# 리스트, 튜플, 사전 및 내장 자료형 특성



한국기술교육대학교  
온라인평생교육원

## ■ 내장 자료형의 정리와 객체 신원 파악

### 1. 내장 자료형의 특성 정리

자료형	저장/접근 방법	변경 가능성	저장 모델
수치형	직접(Direct)	변경불가능(Immutable)	리터럴 (Literal)
문자열	시퀀스 (Sequence)	변경불가능(Immutable)	리터럴 (Literal)
리스트	시퀀스 (Sequence)	변경가능(Mutable)	컨테이너 (Container)
튜플	시퀀스 (Sequence)	변경불가능(Immutable)	컨테이너 (Container)
사전	매핑 (Mapping)	변경가능(Mutable)	컨테이너 (Container)

- 내장 자료형 → 수치형, 문자열, 리스트, 튜플, 사전
- 문자열, 리스트, 튜플 → 시퀀스(인덱스 존재)
- 사전 → 매핑(인덱스 없음)
- 리스트, 사전 → 변경 가능 / 나머지는 불가능
- 수치형, 문자열 → 리터럴
- 리터럴은 정수나 숫자, 표기법 등이 하나씩 저장됨
- 0x→ 16진법을 나타내는 리터럴(표기법)
- 표기법을 사용하여 숫자(수치형 자료)를 나타냄
- -0.2e-4도 하나의 표기법(리터럴)
- 단일/이중 따옴표, 연속된 단일/이중 따옴표 세개도 표기법(리터럴)
- 리스트, 튜플, 사전 → 컨테이너(집합체 형태)

## ■ 내장 자료형의 정리와 객체 신원 파악

### 2. 내장 자료형 알아보기

```
print type(3)      #정수  
print type(3.3)    #실수  
print type('abc') #문자열
```

```
<type 'int'>  
<type 'float'>  
<type 'str'>
```

- type(리터럴) → 리터럴의 자료형

```
print type([])     #리스트  
print type(())      #튜플  
print type({})     #사전(dict)
```

```
<type 'list'>  
<type 'tuple'>  
<type 'dict'>
```

- 내용이 없는 리스트, 튜플, 사전도 타입 조사 가능

## ■ 내장 자료형의 정리와 객체 신원 파악

### 2. 내장 자료형 알아보기

#### • 자료형의 비교

```
a = 0
L = [1,2,3]
print type(a) == type(0)
print type(L) == type([])
print type(L[0]) == type(0)
```

```
True
True
True
```

- `type(A)==type(B)` → A와 B의 타입이 같은지 확인
- `type(L)==type([])` → L과 리스트의 타입이 같은지 확인
- `type(L[0])` → L의 0번 인덱스(1)의 타입 → 정수(int)
- `type(0)` → 0의 타입 → 정수(int)
- `type(L[0]) == type(0)` → True

```
print type(None)  #None 객체, 아무 값도 없다(혹은 아니다)를 나타내는 객체
print
a = None
print a
print type(a)
```

```
<type 'NoneType'>

None
<type 'NoneType'>
```

- none 객체는 none 타입
- `a = none` → a는 아무 것도 아닌 객체
- `print a` → none 출력됨
- `print type(a)` → a의 타입을 출력 → none타입
- none 객체는 none 타입, none 타입은 none 객체

## ■ 내장 자료형의 정리와 객체 신원 파악

### 3. 객체의 신원 식별하기

- id(): 객체의 식별자를 반환한다.

```
a = 500
b = a
print id(a)
print id(b)
print
x = 1
y = 1
print id(x)
print id(y)
```

```
4478569376
4478569376

4298182776
4298182776
```

- id(a) → a 객체의 식별자 반환
- print id(a) → 파이썬 인터프리터가 관리하는 식별자 출력
- print id(b) → b의 식별자 출력
- b = a 문장으로 인해 b도 500을 가리킴
- a = 500 → 500을 가리키는 레퍼런스가 a에 할당
- b = a → b에도 동일한 레퍼런스가 할당됨
- 따라서 a, b가 동일한 객체(500)를 가리키게 됨
- 때문에 a와 b의 식별자가 동일하게 나타남
- 파이썬에서는 숫자를 따로 객체로 만들지 않음(이미 존재)
- x, y에 1을 가리키는 레퍼런스 할당
- 따라서 x, y는 동일한 객체(1)를 가리킴

## ■ 내장 자료형의 정리와 객체 신원 파악

### 3. 객체의 신원 식별하기

- is 키워드: 두 객체의 식별자가 동일한지 테스트한다.

```
c = [1,2,3]
d = [1,2,3]
print c is d
```

```
a = 500
b = a
print a is b
```

```
x = 1
y = 1
print x is y
```

```
e = f = [4,5,6]
print e is f
```

```
False
True
True
True
```

- c와 d가 동일한 리스트를 가질 경우
- c is d → c, d가 가리키는 객체의 식별자가 동일한지 확인
- 결과는 False(식별자가 다름)
- id 함수로 c, d의 식별자를 확인해보면
- 식별자가 다른 c, d → is 함수(id가 동일한가?) → 결과 False 출력
- 내용이 같은 리스트라도 서로 다른 객체
- 수치는 새로 만들어지지 않고 이미 존재
- 이미 존재하는 수치를 c, d가 같이 참조
- 때문에 c, d 식별자가 동일함
- 이미 존재하는 500을 a가 가리킴

## ■ 내장 자료형의 정리와 객체 신원 파악

### 3. 객체의 신원 식별하기

- b가 a와 같은 레퍼런스를 가짐 → b도 동일한 500을 가리킴
- f=[4, 5, 6] → [4, 5, 6]을 가리키는 레퍼런스 값이 f에 할당
- e = f → f에 할당된 레퍼런스 값이 e에도 할당
- 따라서 e와 f는 동일한 식별자를 가짐

• == 연산자: 두 객체의 값이 동일한지를 테스트한다.

```
c = [1,2,3]
d = [1,2,3]
c == d
```

True

- c==d → c와 d의 내용이 같은가를 확인
- c와 d는 서로 다른 객체를 가지지만 내용이 같으므로 True
- ==는 두 객체의 내용이 같은지 확인, is는 식별자가 같은지 확인