

04. Python **내장자료**

1. Python 내장 자료

1. List

1] List 개념

- 리스트 : 임의의 객체를 순차적으로 저장하는 집합적 자료형
- 리스트는 객체를 순차적으로 저장하므로 인덱스가 존재
- 리스트는 변경 가능,
- 문자열 아닌 대부분 연산들은 List도 지원

2] 적용 예시

```
- L = [1,2,3]
print (type(L))    # L의 타입은 무엇?
print()
print ("len(L)->" , len(L))  # len(L) → L의 길이(원소의 개수)
print()
print (" L[1]->" , L[1])      # 인덱스가 1인 값
print (" L[-1]->" , L[-1])    # 인덱스가 뒤에서 첫 번째인 값
print (" L[1:3]->" , L[1:3])  # 인덱스 1(2)부터 인덱스 3 앞(3)까지 슬라이싱
print()
print ("L + L ->" , L + L)    # 두 리스트의 원소를 합쳐 하나의 리스트
print ("L * 3 ->" , L * 3)    # 리스트를 숫자만큼 반복하여 하나의 리스트
```

3] Range

- **range()** 함수를 통한 인덱스 리스트 생성

```
L = range(10)  # 0부터 10-1(9) 까지의 숫자의 list 반환
print (L)
print (L[:2])  # 전체 리스트에서 인덱스가 2씩 차이나게 슬라이싱
print (L[::-1])  # 전체 리스트를 거꾸로 슬라이싱
print (5 in L)  # L안에 5가 있다(true)
```

1. Python 내장 자료

2. Tuple

1] Tuple 개념

- 리스트와 유사하지만 튜플 내의 값을 변경할 수 없음(상수와 비슷한 속성)
- 문자열 아닌 대부분 연산들도 Tuple 지원
- 튜플은 둥근 괄호()를 사용
- 튜플도 인덱스가 존재

2] 적용 예시

```
t = (1,2,3)
print ("len(t)->" , len(t))
print ()
print (t[0])      # t[0] = 100 → t의 인덱스 0인 1을 100으로 변경(불가)
print (t[-1])
print (" t[0:2] ->" , t[0:2]) # t[0:2] -> (1, 2)
print ("t[::2]->" , t[::2])   # t[::2]-> (1, 3)
print
print ("t + t + t ->" , t + t + t) # (1, 2, 3, 1, 2, 3, 1, 2, 3)
print (t * 3)                # (1, 2, 3, 1, 2, 3, 1, 2, 3)
t[0] = 100                   # 튜플은 내용 변경 불가, 오류
```

1. Python 내장 자료

3. 사전(dictionary)

1] dictionary 개념

- 정수형 인덱스가 아닌 키를 이용하여 값을 저장하는 자료 구조
- 저장된 각 자료에 대한 순서는 의미 없음
- 매핑(Mapping) 함수와 비슷한 역할(인덱스가 없음, 키를 이용해 값 저장)
- x라는 키값을 넣으면 값 y를 반환

coma(,)를 기준으로 아이템 구분

d = {key:value , key:value , key:value } → 사전 d의 원소 3개

사전의 원소 → key:value (쌍으로 이루어짐)

2] 적용 예시

```
d = {'one': '일', 'two': '이', 'three': '삼'}
```

```
d['four'] = '사' # 새 항목의 삽입
```

```
print d
```

```
d['one'] = 1 # 기존 항목의 값 변경
```

```
print d
```

```
print 'one' in d # 키에 대한 멤버십 테스트
```

```
d3 = {'one': 1, 'two': '이', 'three': '삼', 'four': '사'}
```

```
print "d3.keys()->" , d3.keys() # 키만 리스트로 추출함, 임의의 순서
```

```
print "d3.values()->" , d3.values() # 값만 리스트로 추출함, 임의의 순서
```

```
print "d3.items()->" , d3.items() # 키와 값의 튜플을 리스트로 반환
```