

파이썬

12강. 자료구조(리스트내포, 튜플)

1. 리스트 내포

- 리스트 내포(List comprehension)란 list 안에서 for와 if를 사용하는 문법을 의미한다. 이를 사용하면 매우 직관적인 코딩이 가능하다.
- 다음은 리스트 내포의 첫 번째 형식으로 리스트와 for문을 동시에 이용하여 리스트 안에서 반복을 수행하는 명령문을 처리할 경우 사용된다.

```
변수 = [실행문 for 변수 in 열거형객체 ] ↵
```

1. 리스트 내포

- 위 형식의 프로그래밍 처리 과정은 ① for문에서 열거형객체의 원소 하나를 변수로 넘겨 받는다. ② 변수에 할당된 값을 실행문으로 처리한다. ③ 처리된 결과를 변수에 순차적으로 추가(append)한다.
- 리스트 내포의 두 번째 형식으로 리스트 안에 for와 if문을 함께 사용하여 반복을 수행하는 과정에서 조건에 만족하는 내용을 선택적으로 처리할 경우 유용하게 사용할 수 있다.

```
변수 = [실행문 for 변수 in 열거형객체 if 조건식]
```

- 위 형식의 프로그래밍 처리 과정은 ① for문에서 열거형객체의 원소 하나를 변수로 넘겨 받는다. ② 변수에 할당된 값을 조건식으로 사용하여 비교 판단한다. ③ 조건이 참(True)이면 변수에 할당된 값을 실행문으로 처리한다. ④ 처리된 결과를 변수에 순차적으로 추가(append)한다.

1. 리스트 내포

chapter04.lecture.step02_list.py ↵

Python Console ↵

```
# 형식1) 변수 = [ 실행문 for ]  
x = [2, 4, 1, 5, 7] ↵  
# print(x ** 2) # error ↵
```

```
lst = [ i ** 2 for i in x ] # x변량에 제곱 계산 ↵  
print(lst) # [4, 16, 1, 25, 49] ↵
```

[4, 16, 1, 25, 49] ↵

```
# 형식2) 변수 = [ 실행문 for if ] ↵  
# 1~10 -> 2의 배수 추출 -> i*2 -> list 저장 ↵  
num = list(range(1,11)) ↵
```

```
lst2 = [ i*2 for i in num if i % 2 == 0 ] ↵  
print(lst2) ↵
```

[4, 8, 12, 16, 20] ↵

1. 리스트 내포

- 리스트 내포 예
- 형식1) 변수 = [실행문 for]
- 변수 x의 각 원소에 제곱을 계산하기 위해서 변수 x에 직접 제곱(**) 연산자를 적용하면 오류(Error)가 발생한다. 리스트는 선형대수 연산이 불가능하기 때문에 그렇다. 따라서 리스트 안에 for문을 이용해서 x의 원소를 하나씩 꺼내서 실행문($i ** 2$)으로 처리한 다음에 계산 결과를 lst 변수에 순차적으로 추가한다.
- 형식2) 변수 = [실행문 for if]
- 변수 num에는 range 클래스에 의해서 생성된 1에서 10까지 10개의 숫자를 객체로 갖는다. 리스트 안에 for문을 이용해서 1부터 하나씩 변수 i로 할당받은 다음 조건식에 의해서 i가 짝수이면 실행문 ($i*2$)에 의해서 2가 곱해진다. 따라서 lst2 변수에는 짝수에 2가 곱해진 5개 원소가 채워지는 것을 확인할 수 있다.

2. 튜플(tuple)

- 튜플은 순차 자료구조라는 점에서 리스트 자료 구조와 많은 부분에서 유사하다. 몇 가지 차이점은 읽기 전용으로 원소를 수정하거나 삭제할 수 없고, 리스트에 비해서 처리속도가 빠르다는 점이다. 다음은 튜플 클래스의 도움말을 확인한 결과이다.

```
>>>help(tuple) ↵
```

```
Help on class tuple in module builtins: ↵
```

```
class tuple(object) ↵
```

```
    | tuple(iterable=(), /) ↵
```

```
    | ↵
```

```
    | Built-in immutable sequence. ↵
```

```
    | ↵
```

```
    | If no argument is given, the constructor returns an empty tuple. ↵
```

```
    | If iterable is specified the tuple is initialized from iterable's items. ↵
```

2. 튜플(tuple)

- 가장 첫 번째 'class range in module builtins' 문장에서 tuple 클래스는 builtins 모듈에 포함 된 클래스임을 나타내고 있다. tuple 클래스는 object 인수를 이용하여 객체를 만든다. '가변 길이 순차 자료구조이다.'
- 만약 인수가 없으면 생성자는 빈 tuple을 만든다. 인수는 반복가능 해야 한다.'라 고 튜플 클래스를 설명하고 있다.

3. 튜플 객체 특징

- 순서 자료구조를 갖는 열거형객체를 생성할 수 있다.
- 다음 형식과 같이 소괄호(())안에 콤마(,)를 이용하여 순서대로 값을 나열한다

```
변수 = (값1, 값2, ..... 값n) ↵
```

- 값의 자료형은 숫자형, 문자형, 논리형 등을 함께 사용할 수 있다.
- 색인(index)을 이용하여 자료를 참조할 수 있고, 슬라이싱, 연결, 반복, 요소 검사 등이 가능하다.
- 읽기 전용이기 때문에 값을 추가, 삽입, 수정, 삭제가 불가능하다.
- 리스트 보다 처리속도가 빠르다

3. 튜플 객체 특징

chapter04.lecture.step03_tuple.py ↵

Python Console ↵

(1) 원소가 한 개인 경우 ↵

t = (10,) ↵

print(t) ↵

(10,) ↵

(2) 원소가 여러 개인 경우 ↵

t2 = (1, 2, 3, 4, 5, 3) ↵

print(t2) ↵

(1, 2, 3, 4, 5, 3) ↵

(3) 튜플 색인 ↵

print(t2[0], t2[1:4], t2[-1]) ↵

1 (2, 3, 4) 3 ↵

(4) 수정 불가 ↵

t2[0] = 10 # error ↵

(5) 요소 반복 ↵

for i in t2 : ↵

print(i, end=' ') ↵

1 2 3 4 5 3 ↵

(6) 요소 검사 ↵

if 6 in t2 : ↵

print("6 있음") ↵

else : ↵

print("6 없음") ↵

6 없음 ↵

3. 튜플 객체 특징

- 튜플 객체 예
- (1) 원소가 한 개인 경우
- 원소가 한 개인 경우 (10,) 형식으로 원소 뒤에 콤마(,)를 붙여야한다. 만약 (10) 형식으로 콤마를 생략한 경우에는 상수 10으로 출력된다.
- (2) 원소가 여러 개인 경우
- 6개의 원소를 갖는 변수 t2의 전체 원소가 출력되는 예문이다.
- (3) 튜플 색인
- 튜플 색인은 리스트와 동일하다.
- (4) 수정 불가
- 튜플의 원소는 수정, 삽입, 삭제 등이 불가능하다. 만약 수정이나 삽입 등을 위해서 리스트 자료형으로 변환해야 한다. (예: lst = list(t2))
- (5) 요소 반복
- 리스트와 동일하게 for문에서 요소 반복이 가능하다.
- (6) 요소 검사
- 리스트와 동일하게 if문을 이용하여 요소 검사가 가능하다.

4. 튜플 관련 함수

- 튜플 객체는 원소를 수정이나 삭제 등의 작업이 불가능하기 때문에 객체에서 지원하는 함수는 리스트에 비해서 적다.

4. 튜플 관련 함수

chapter04.lecture.step03_tuple.py ↵

```
# (1) 튜플 자료형 변환 ↵  
lst = list(range(1,6)) ↵  
t3 = tuple(lst) ↵  
print(t3) ↵
```

```
# (2) 튜플 관련 함수 ↵  
print( len(t3), type(t3) ) # 5 <class 'tuple'> ↵  
print(t3.count(3)) ↵  
print(t3.index(4)) ↵
```

Python Console ↵

```
↵  
↵  
↵  
(1, 2, 3, 4, 5) ↵  
↵  
↵  
5 <class 'tuple'> ↵  
1 ↵  
3 ↵
```

4. 튜플 관련 함수

- 해설 튜플 관련 함수 예
- (1) 튜플 자료형 변환
- 변수 lst는 range 객체를 리스트 객체로 자료형이 변환된 객체를 참조한다. 변수 t3 는 튜플 객체로 변환된 객체를 참조하고 있다. 이처럼 list()와 tuple() 함수는 특정 객체를 리스트나 튜플의 자료형 으로 변환하는데 이용한다.
- (2) 튜플 관련 함수
- 변수 t3의 전체 원소의 길이와 자료형을 출력하고 있다. 또한 튜플 객체에서 지원하는 count() 함수에 의해서 원소 3의 개수를 반환받아서 출력하고, index() 함수에 의해서 원소 4의 색인(위치)을 반환받아서 출력하는 예문이다.

THANK YOU