

1. 리스트의 생성 및 조작법

- 리스트 : 대괄호[] 안에 서로 다른 자료형의 값을 콤마(,)로 구분해 하나 이상 저장할 수 있는 컬렉션 자료형
개별 항목은 0부터 시작하는 인덱스를 이용해 접근

저장된 항목 변경 가능

- 리스트 항목 접근

각 항목은 인덱스를 이용해서 접근 가능

상대적인 인덱스를 통해서도 리스트의 각 항목에 접근 가능(항목의 개수가 5이면 첫번째 항목 인덱스 -5로 접근 가능)

유효한 인덱스가 아닌 값을 사용하면 IndexError 예외 발생

인덱스 범위를 지정할 경우 콜론(:) 사용(data_list[0:3] => 0은 시작 인덱스, 3은 종료 인덱스 (0,1,2))

data_list.index(20) => 해당 항목의 값이 들어 있는 첫 번째 위치의 인덱스 반환

- 리스트 기본 연산

+ 연산자를 이용해 새로운 리스트 생성 가능

* 연산자를 이용해 리스트를 여러 번 연결하는 새로운 리스트 생성 가능

ex) data_list = [40, 50], data_list * 2 == [40, 50, 40, 50]

- 리스트 항목 추가

data_list.append(추가할 값)

data_list.insert(추가할 위치 인덱스, 추가할 값)

data_list.extend(추가할 리스트) : 대괄호 안에 있는 값들이 추가 됨

cf) data_list = [0, 1], data_list.extend([2,3]) == [0,1,2,3], data_list.append([2,3]) == [0,1,[2,3]]

- 리스트 항목 변경

data_list[변경할 값 위치의 인덱스] = 변경할 값

data_list[1:3] = [12,15] : 리스트 1번 값 12로 2번 값 15로 변경됨(범위 연산자의 마지막 값은 실제 범위에 포함되지 않음)

ex) data_list = [10,20,30,40], data_list[1:3] = [12,15,20], data_list == [10,12,15,20,40](전체 리스트의 크기가 변경됨)

- 리스트 항목 제거

del data_list[제거할 항목 인덱스]

del data_list[3:5] : 인덱스 3과 4의 값 제거됨

data_list.pop(제거할 항목 인덱스)

data_list.remove(100) : 100을 가진 첫 번째 항목 제거

data_list.clear() : 모든 항목을 제거해 빈 리스트 객체 생성

- 리스트 항목 확인

‘항목값 in 리스트객체’ 명령 : 리스트 객체에 해당 항목이 있는지 검사

‘항목값 not in 리스트 객체’ 명령 : 리스트 객체에 해당 항목이 없는지 검사

data_list.count(20) 함수 : 인자로 전달된 항목의 개수 확인 가능

- 리스트와 for 문

2. 리스트 내포의 특징

- 리스트 내포

[item for item in data_list if item % 2 == 1]

리스트 내포에서도 for 중첩 사용 가능

문자열도 리스트 내포 구조를 사용해 간단히 새로운 리스트 객체로 변환 가능

3. 튜플의 생성 및 조작법

- 튜플 : 중괄호() 안에 서로 다른 자료형의 값을 콤마(,)로 구분해 하나 이상 저장할 수 있는 컬렉션 자료형

개별 항목은 0부터 시작하는 인덱스를 이용해 접근

저장된 항목 변경 불가 -> 리스트와 다른 점

- 튜플의 생성

tuple(‘안녕하세요’) : 개별 문자를 원소로 하는 튜플 객체 생성

- 튜플 항목 접근

각 항목은 인덱스를 이용해 접근(data_tuple[0])

상대적인 인덱스를 통해서도 튜플의 각 항목에 접근가능

유효한 인덱스가 아닌 값을 사용하면 `IndexError` 예외 발생

인덱스 범위를 지정할 경우 콜론(:) 사용

`data_tuple.index(20)` : 해당 항목의 값이 들어 있는 첫 번째 위치의 인덱스 반환

- 튜플 기본 연산

- + 연산자를 이용해 두 항목을 연결한 새로운 튜플 생성 가능

- * 연산자를 이용해 튜플을 여러 번 연결하는 새로운 튜플 생성 가능

- 튜플 항목 확인

‘항목값 in 튜플 객체’ 명령 : 튜플 객체에 해당 항목이 있는지 검사

‘항목값 not in 튜플 객체’ 명령 : 튜플 객체에 해당 항목이 없는지 검사

`count(20)` 함수 : 인자로 전달된 항목의 개수 확인 가능

- 튜플과 for 문

4. 튜플의 내포의 특징

- 튜플 내포

(item **for** item **in** data **if** item % 2 == 1)

data = (1,2,3,4,5)

(x * y for x in data if x % 2 == 1

for y in data if y % 2 == 0) == (2,4,6,12,10,20)

- * for i in list :

print("{0".format(item), end=" ")

=> end = " " 로 인해 값들이 가로로 출력됨