python

스마트인재개발원
Smart Human Resources Development

손 지 영 강사

python

학습목표

- 리스트와 튜플에 대해 이해 할 수 있다.
- 리스트 함수를 활용 할 수 있다.

리스트(list) 란?

- 파이썬의 자료구조 형태중 하나
- 순서가 있는 수정가능한 객체의 집합
- 대괄호([])로 작성되어지며, 리스트 내부의 값은 콤마(,)으로 구분
- 하나의 리스트에 다양한 자료형 포함
- 추가, 수정, 삭제 가능

튜플(tuple) 이란?

- 파이썬의 자료구조 형태중 하나
- 순서가 있는 집합
- 소괄호(())로 작성되어지며, 튜플의 내부 값은 콤마(,)으로 구분
- 하나의 튜플에 다양한 자료형 포함
- 추가, 수정, 삭제 불가능

python 리스트(list)와 튜플(tuple) 공통점과 차이점

Smart Media
스마트미디어인재개발원

공통점

- 타입과 상관 없이 일련의 요소(Element)를 갖을 수 있다.
- 요소의 순서를 관리한다.

차이점

- 리스트는 가변적(mutable)이며, 튜플은 불변적(immutable)
- 리스트는 요소가 몇 개 들어갈지 명확하지 않은 경우에 사용
- 튜플은 요소 개수를 사전에 정확히 알고 있을 경우에 사용

리스트명 = [요소1, 요소2, 요소3, ...]

```
a = [ ]
b = [1, 2, 3]
c = ['My', 'name', 'is', 'JY']
d = [1, 2, 'My', 'name']
e = [1, 2, ['My', 'name']]
```

python 리스트(list) 데이터 접근

인덱싱(indexing)

- 무엇인가를 '가리킨다'는 의미

슬라이싱(Slicing)

- 무엇인가를 '잘라낸다'는 의미

리스트[인덱스]

- 인덱스에 위치한 값 반환

```
list1 = [2, 5, 7, 9, 10]
print(list1[0])
print(list1[3])
print(list1[2]+list1[-1])
```

python 리스트(list) 인덱싱(indexing) 예제


```
list2 = [1, 2, 3, ['a', 'b', 'c']]
```

- 변수 temp에 [‘a’, ‘b’, ‘c’]를 저장하고 출력하시오.

- list2에서 문자 ‘b’만 뽑아서 출력하시오.

리스트[start 인덱스 : end 인덱스+1]

- start 인덱스부터 end 인덱스 바로 전까지 값 반환 (start <= x < end)

```
list3 = [0, 1, 2, 3, 4]
list3[1:3]
```

```
[1, 2]
```

```
list3[:2]
```

```
[0, 1]
```

```
list3[3:]
```

```
[3, 4]
```

```
list3[3:4]
```

```
[3]
```

python 리스트(list) 반복(Repetition)


```
list4 = [1,2,3]
list4 * 3
```

```
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

python 리스트(list) 연산


```
list4 = [1, 2, 3]
list5 = [3, 4, 5, 6]
list4 + list5
```

```
[1, 2, 3, 3, 4, 5, 6]
```

python 리스트(list) 값 추가1

리스트.append(값)

- 끝에 값 추가

```
list5 = [0, 1, 2, 3, 4]
list5.append(5)
list5
```

```
[0, 1, 2, 3, 4, 5]
```

```
list5.append(6)
list5
```

```
[0, 1, 2, 3, 4, 5, 6]
```

python 리스트(list) append() 실습


```
fruits = ['사과','포도',['수박','멜론'],'오렌지']
```

fruits에 담긴 '사과', '멜론', '오렌지'를 인덱싱하여 각 변수에 저장하고, 빈 리스트 choice_lst에 순서대로 추가하시오.

1

```
apple =  # apple 인덱싱
melon =  # melon 인덱싱
orange =  # orange 인덱싱
```

2

```
choice_lst =  # 빈 리스트
choice_lst.  # 사과 추가
choice_lst.  # 멜론 추가
choice_lst.  # 오렌지 추가
choice_lst
```

```
['사과', '멜론', '오렌지']
```

리스트.extend(리스트)

- 끝에 여러 개 값 추가

```
choice_lst = []
choice_lst.append(apple)
choice_lst.append(melon)
choice_lst.append(orange)
choice_lst
```

```
['사과', '멜론', '오렌지']
```

```
fruits = ['사과','포도',['수박','멜론'],'오렌지']
apple = fruits[0]
melon = fruits[2][1]
orange = fruits[3]

choice_lst = []
choice_lst.extend([apple,melon,orange])
choice_lst
```

```
['사과', '멜론', '오렌지']
```

python 리스트(list) 값 추가3

리스트.insert(인덱스위치, 값)

- 인덱스 위치에 값 추가

```
list5 = [0, 1, 2, 3, 4]
list5.insert(1, 5)
list5
```

```
[0, 5, 1, 2, 3, 4]
```

```
list5.insert(5, 6)
list5
```

```
[0, 5, 1, 2, 3, 6, 4]
```

python 리스트(list) insert() 실습

Smart Media
스마트미디어인재개발원

추가 위치 번째와 노래 제목, 가수명을 입력 받은 후 변수에 저장하고, music_lst리스트 목록에 곡정보를 추가하시오.

music_lst =[['사건의 지평선','윤하'],['After LIKE','IVE'],['새삥','지코']]

```
loc =                    ?
song =                    ?
singer =                   ?
music_lst.                   ?
```

```
추가 위치 번호 >> 3
노래 제목 >> Attention
가수 >> 뉴진스
```

```
music_lst
```

```
[['사건의 지평선', '윤하'], ['After LIKE', 'IVE'], ['Attention', '뉴진스'], ['새삥', '지코']]
```

python 리스트(list) 값 수정


```
list6 = [0, 1, 2, 3, 4]
list6
```

```
[0, 1, 2, 3, 4]
```

```
print("수정 전 :",list6[1])
list6[1] = 7
print("수정 후 :",list6[1])
```

```
수정 전 : 1
수정 후 : 7
```

```
list6
```

```
[0, 7, 2, 3, 4]
```

```
list6[2:4] = 7
```

```
TypeError
t)
<ipython-input-17-c640f2364ee6>
      1 print(list6[2:4], list6[
----> 2 list6[2:4] = 7

TypeError: can only assign an it
```

```
print(list6[2:4])
list6[2:4] = [7]
list6
```

```
[2, 3]

[0, 7, 7, 4]
```

python 리스트(list) 값 삭제

Smart Media
스마트미디어인재개발원

del 키워드 이용

```
list7 = [0, 1, 2, 3, 4, 5]
del list7[1]
list7
```

```
[0, 2, 3, 4, 5]
```

```
list7 = [0, 1, 2, 3, 4, 5]
del list7[1:5]
list7
```

```
[0, 5]
```

리스트.remove(값)

```
list7 = ['a','b','c','d','e']
list7.remove('b')
list7
```

```
['a', 'c', 'd', 'e']
```

```
list7.remove('b')
```

```
---------------------------
--
ValueError
t)
<ipython-input-42-ac74ba81fef3
----> 1 list7.remove('b')

ValueError: list.remove(x): x
```

python 리스트(list) 오름차순 정렬

리스트.sort()

- 리스트에 있는 값을 **오름차순**으로 **정렬**

```
list8 = [9, 77, 13, 51, 100, 3]
list8
```

```
[9, 77, 13, 51, 100, 3]
```

```
list8.sort()
list8
```

```
[3, 9, 13, 51, 77, 100]
```

python 리스트(list) 순서 거꾸로 뒤집기

리스트.reverse()

- 리스트에 있는 값을 역순으로 뒤집음

```
list9 = [9, 77, 13, 51, 100, 3]
list9
```

```
[9, 77, 13, 51, 100, 3]
```

```
list9.reverse()
list9
```

```
[3, 100, 51, 13, 77, 9]
```

python 리스트(list) 내림차순 정렬

리스트.sort() 와 리스트.reverse() 이용

- 리스트에 있는 값을 내림차순으로 정렬

```
list10 = [9, 77, 13, 51, 100, 3]
list10
```

```
[9, 77, 13, 51, 100, 3]
```

```
list10.sort()
list10
```

```
[3, 9, 13, 51, 77, 100]
```

```
list10.reverse()
list10
```

```
[100, 77, 51, 13, 9, 3]
```

python 리스트(list) 값 위치 반환 및 제거

리스트.index()

- 찾고자 하는 값의 **위치 반환**

```
list11 = ['a','b','c','d','e','f']
list11.index('c')
```

```
2
```

리스트.pop()

- 마지막 값을 **반환 후** 리스트에서 **제거**

```
list12 = ['a','b','c','d','e','f']
list12.pop()
```

```
'f'
```

```
list12
```

```
['a', 'b', 'c', 'd', 'e']
```

python 리스트(list) 값의 개수 확인

len(리스트)

- 리스트의 값 개수 반환

```
list13 = [0, 1, 2]
len(list13)
```

```
3
```

```
list14 = ['a','b','c','d','e','f']
len(list14)
```

```
6
```

튜플명 = (요소1, 요소2, 요소3, ...)

```
a = ( )
b = (1, 2, 3)
c = ('My', 'name', 'is', 'JY')
d = (1, 2, 'My', 'name')
e = (1, 2, ('My', 'name'))
```

튜플[인덱스]

- 인덱스에 위치한 값 반환

```
tuple1 = (0, 1, 2, 3, ('a','b','c'), 5)
```

```
tuple1[2]
```

```
2
```

```
tuple1[4]
```

```
('a', 'b', 'c')
```

튜플[start 인덱스 : end 인덱스+1]

- start 인덱스부터 end 인덱스 바로 전까지 값 반환 (start <= x < end)

```
tuple1 = (0, 1, 2, 3, ('a','b','c'), 5)
```

```
tuple1[1:3]
```

```
(1, 2)
```

```
tuple1[3:]
```

```
(3, ('a', 'b', 'c'), 5)
```

python 튜플(tuple) 값의 개수 확인

len(튜플)

- 튜플의 값 개수 반환

```
tuple1 = (0, 1, 2, 3, ('a','b','c'), 5)
len(tuple1)
```

```
6
```

```
tuple2 = ('a','b','c','d','e','f')
len(tuple2)
```

```
6
```

튜플은 추가, 수정, 삭제 불가능

```
tuple1 = (0, 1, 2, 3, ('a','b','c'), 5)
```

```
tuple1[0] = 3
```

```
---------------------------------------------------------------
--
TypeError                                 Traceback (most r
t)
<ipython-input-53-5e0f22de5ab3> in <module>
----> 1 tuple1[0] = 3

TypeError: 'tuple' object does not support item assignment
```

python 해당 데이터의 포함 여부 확인

Smart Media 스마트미디어인재개발원

조건문에 자주 활용하는 in, not in

in : 찾고자 하는 값(x)이 포함 되어 있으면 True

not in : 찾고자 하는 값(x)이 포함되어 있지 않으면 True

in	not in
x in 문자열	x not in 문자열
x in 리스트	x not in 리스트
x in 튜플	x not in 튜플

```
str1 = "파이썬 최고"

"파이썬" in str1
```

```
True
```

```
"파이썬" not in str1
```

```
False
```

```
list1 = [77, 38, 10]

33 in list1
```

```
False
```

```
33 not in list1
```

```
True
```

python in, not in 연산자 예제

다음과 같이 리스트에 사용자가 입력한 글자가 어디에 포함되어 있는지 출력하시오.

```
lst = ['딸기','바나나','수박','체리','포도']
```

```
**********************************************
검색할 문자를 입력하세요 >> 수박
수박는 리스트에 2번째 인덱스에 들어있습니다.
**********************************************
```

```
**********************************************
검색할 문자를 입력하세요 >> 오렌지
오렌지는 리스트에 들어있지 않습니다.
**********************************************
```

python in,not in 연산자 예제

다음과 같이 문자열에 사용자가 입력한 글자가 몇 개 포함되어 있는지 출력하시오.

```
s = "Hi, My name is Mihee"
```

```
**************************************
검색할 문자를 입력하세요 : hi
'hi'는 문자열에 들어있지 않습니다.
**************************************
```

```
**************************************
검색할 문자를 입력하세요 : i
'i'는 3 번 들어 있습니다.
**************************************
```

아래 결과 화면처럼 동작하는 프로그램을 작성하시오.

```
======================SJY Kitchen======================
요리 3가지 입력해보기
첫 번째 요리는? 탕수육
두 번째 요리는? 가지튀김
세 번째 요리는? 짬뽕
주문하신 메뉴는 탕수육,가지튀김,짬뽕입니다.
마지막 요리를 변경하시겠어요? 무엇으로 바꿀까요? 간짜장
다시 확인할게요. 주문하신 메뉴는 탕수육,가지튀김,간짜장입니다.
```

시퀀스 자료형에서 일반적으로 사용할 수 있는 개념

패킹: 한 변수에 여러 개의 데이터를 할당하는 것
언패킹 : 패킹된 데이터를 각각의 변수로 반환하는 것

```
t = [1,2,3]   packing
a,b,c = t     unpacking
print(t,a,b,c)
```

```
[1, 2, 3] 1 2 3
```

```
a,b = t
```

```
---------------------------------------------------------------------------
ValueError                                Tracebac
Input In [228], in <cell line: 1>()
----> 1 a,b = t

ValueError: too many values to unpack (expected 2)
```

```
a,b,c,d = t
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most r
Input In [227], in <cell line: 1>()
----> 1 a,b,c,d = t

ValueError: not enough values to unpack (expected 4, got 3)
```

다음시간에는?

반복문