데이터 처리를 위한 Python 프로그래밍 입문

5-1강. 데이터 구조 : List형

ERICA 2018-2

강의 내용

- List
- List index
- List 원소(element) 확인
- List 연산
- List 길이
- List 응용
- List 조작 함수

List(1/2)

- List형
  - 동일한 목적을 갖는 유사한 항목들을 묶어서 하나의 list형의 변수로 선언
  - List형 기호 : 변수 생성시 대괄호([ ]), 안에 들어가는 원소들은 쉼표(,)로 구분

```
>>> num_list = [1,2,3,4]
>>>
>>> num_list
[1, 2, 3, 4]
>>>
>>>
>>> string_list = ['spring','summer','fall','겨울']
>>>
>>> string_list
['spring', 'summer', 'fall', '겨울']
>>>
>>>
```

List(2/2)

- 다양한 list
  - List의 원소는 대개의 경우 같은 목적의 유사 항목을 위해
    동일형의 자료들로 구성되지만, 서로 다른 형인 경우도 허용함

```
>>> a = [10, 20, 30]
>>> b = ['Kim','Lee','Jung']
>>> c = [10,20,'Kim','Lee']
>>> d = [10,20,['spring','fall']]
>>> a
[10, 20, 30]
>>> b
['Kim', 'Lee', 'Jung']
>>> c
[10, 20, 'Kim', 'Lee']
>>> d
[10, 20, ['spring', 'fall']]
>>>
>>> e = []
>>> e
[]
>>>
>>> f = [      ]
>>> f
[]
>>>
```

* 아무것도 포함하지 않는, 원소가 없는 list도 허용함.

List index(1/3)

- Index
  - 원소가 배열된 순서를 나타내며, index로 색인 가능
  - 순서는 항상 '0'번째 요소부터 수서가 매겨짐

```
>>> season = ['spring','summer','fall','winter']
>>> season[0]
'spring'
>>> season[1]
'summer'
>>> season
['spring', 'summer', 'fall', 'winter']
>>>
```

List index(2/3)

- Index
  - 연산자 응용

표기 방법	내 용
list_name[int_a θ int_b]	int_a θ int_b 의 연산결과가 index가 됨 list형은 index에 해당하는 원소(element)값을 반환함 θ는 연산자를 의미함, θ = {+, -, *,%}

List index(3/3)

- Index
  - 연산자 응용

```
>>> list_ex =[1,3,4,6,9,11,13]
>>>
>>> list_ex[1+2]
6
>>> list_ex[10-4]
13
>>> list_ex[10-3]
Traceback (most recent call last):
  File "<pyshell#46>", line 1, in <module>
    list_ex[10-3]
IndexError: list index out of range
>>> a = 3
>>> b = 2
>>> list_ex[a+b]
11
>>>
```

* 대괄호 안의 연산 결과를 index합니다.

* 10-3=7로 list_ex는 6까지 index 가능함을 유의하세요.

* 대괄호 안에 변수 연산결과를 index 가능합니다.

List 원소 확인(1/2)

- List의 구성요소 확인

표기 방법	내 용
in	List의 원소(element)인가를 확인하는 연산자
not in	List의 원소(element)가 아닌 원소를 확인하는 연산자

```
>>> abc =[1,3,5,7,9,11]
>>> 3 in abc
True
>>> 2 in abc
False
>>>
```

List 원소 확인(2/2)

- List의 구성요소 확인

```
>>> name = ['Kim',"Lee",'''Jung''']
>>> 'Kim' in name
True
>>> 'Jeong' in name
False
>>> 'Jung' in name
True
>>> name
['Kim', 'Lee', 'Jung']
>>>
```

* String설정 기호는 다양합니다.

List 길이

- List 길이 계산
  - 많은 양의 원소를 가지고 있는 list형의 경우 len() 함수 사용하면 원소의 개수를 파악함

```
>>> numlist = [2,4,6,8]
>>> len(numlist)
4
>>> list_a = ['season',['Kim',3],['Jung','Lee'],6,9]
>>> len(list_a)
5
>>>
```

List 연산(1/2)

- List형의 연산
  - + : list형들을 합하여 새로운 list를 생성
  - * : 곱한 수만큼 list형이 반복해서 새로운 list를 생성

표기 방법	내 용
list_name + list_name	여러 개의 list형을 하나의 list형으로 병합하는 연산
list_name * number	주어진 숫자만큼 list형을 반복하는 연산

List 연산(2/2)

- List형의 연산

```
>>> int_num_a = [1,2,3,4]
>>> int_num_b = [5,6,7,8]
>>> int_num_c = int_num_a+int_num_b
>>> int_num_c
[1, 2, 3, 4, 5, 6, 7, 8]
>>>
>>>
>>> int_num = [3,6,9]
>>> int_num_m = int_num *3
>>> int_num_m
[3, 6, 9, 3, 6, 9, 3, 6, 9]
>>>
```

```
>>> s_re=['r','e']
>>> s_serve=['s','e','r','v','e']
>>> s_reserve = s_re + s_serve
>>>
>>> s_reserve
['r', 'e', 's', 'e', 'r', 'v', 'e']
>>>
>>> s_re
['r', 'e']
>>> s_serve
['s', 'e', 'r', 'v', 'e']
>>>
```

List 응용(1/3)

- List형의 일부분 추출 : slice

표기 방법	내 용
list_name[index1 : insex2]	index1이상, index2미만(index2-1)인 범위의 부분list를 생성하는 연산 index1값은 포함되고, index2값은 포함되지 않음
list_name[:]	모든 범위의 list를 생성하는 연산
list_name[: index2]	index2미만(index2-1)인 범위의 부분 list를 생성하는 연산
list_name[index1 :]	index1이상 범위의 부분list를 생성하는 연산

List 응용(2/3)

- List형의 일부분 추출 : slice

```
>>> spell=['h','a','n','d','i','c','r','a','f','t']
>>> spell
['h', 'a', 'n', 'd', 'i', 'c', 'r', 'a', 'f', 't']
>>> spell[1:6]
['a', 'n', 'd', 'i', 'c']
>>> spell[0:6]
['h', 'a', 'n', 'd', 'i', 'c']
>>> spell[:]
['h', 'a', 'n', 'd', 'i', 'c', 'r', 'a', 'f', 't']
>>> spell[:4]
['h', 'a', 'n', 'd']
>>> spell[5:]
['c', 'r', 'a', 'f', 't']
>>> spell[:2]+spell[9:]
['h', 'a', 't']
>>>
```

* Index는 0부터 시작합니다.

List 응용(3/3)

- List형의 원소 변경
  - List의 원소를 바꾸어야 하는 경우
  - Index를 통해 새로운 원소를 대입

```
>>> favorite_color =['Yellow','Black']
>>> favorite_color
['Yellow', 'Black']
>>> favorite_color[1]='White'
>>> favorite_color
['Yellow', 'White']
>>>
```

List 조작 함수

- List 조작 함수

함 수	내 용
append()	리스트 제일 뒤에 항목을 추가한다.
pop()	리스트 제일 뒤의 항목을 빼내고, 빼낸 항목은 삭제한다.
sort()	리스트의 항목을 정렬한다.
reverse()	리스트의 항목의 순서를 역순으로 만든다.
index()	지정한 값을 찾아서 그 위치를 반환한다.
insert()	지정된 위치에 값을 삽입한다.
remove()	리스트에서 지정한 값을 제어한다. 단 지정한 값이 여러 개일 경우 첫 번째 값만 지운다.
extend()	리스트 뒤에 리스트를 추가한다.
count()	리스트에서 찾을 값의 개수를 센다.

Thank you