
PYTHON

Python
Programming

9

- 담당교수 : 최희식
- eMail :choihs3054@seoultech.ac.kr

강의 내용

■ 학습목표

- 파이썬 리스트 구조에 대해 학습한다.
- 파이썬 리스트 활용에 대해 학습한다.

■ 학습목차

- 리스트 구조
- 리스트 추가/삭제하기
- 리스트 슬라이싱
- 리스트 정렬하기

1

이번
차시에서는

- 리스트 구조

리스트 살펴보기

■리스트(list)

- 리스트는 여러 개의 자료를 하나의 변수로 관리할 때 사용되는 집합
- 리스트는 데이터 항목 값이 삽입, 삭제, 수정, 정렬, 데이터 추출과 같은 데이터 처리 및 데이터 항목 변경이 가능

리스트 변수=[항목1, 항목2, 항목3...]

num1=1

num2=2

num3=3

num4=4

num5=5

num=[1,2,3,4,5]

리스트 살펴보기

■리스트 변수

- 리스트 변수에는 숫자형 데이터, 문자열 데이터, 숫자와 문자열 혼합형 데이터 항목과 같은 여러 형태의 다양한 자료 항목도 저장할 수 있다.

```
① numList = []                                  #빈 리스트
② numList = [10, 20, 30, 40, 50]                #숫자 리스트 변수
③ strList=["SBS", "KBS", "MBC", "TVN", "ICON"]  #문자열 데이터 리스트 변수
④ milkiest = [10, "KBS", 120.15, True, False]   #혼합형 데이터 리스트 변수
```

리스트 살펴보기

■파이썬 자료형

▪type() 함수를 이용하여 변수의 데이터 유형을 알 수 있다.

1. 단일 데이터 유형
   ① integer : 정수형
   ② float : 실수형
   ③ string : 문자형
   ④ boolean : 부울(Boolean)

2. 여러 항목의 값을 저장하는 데이터 유형
   ① list : [ ]
   ② tuple : ( )
   ③ dictionary : { }
   ④ set : { }

리스트 살펴보기

■리스트 항목 출력

- 리스트 항목 값 출력하기

```
num=[10,15,20,25,30]
for i in num:
    print(i)
```

```
10
15
20
25
30
```

[실습]

■리스트 변수 num=[10,15,20,25,30]에 숫자 항목이 저장되어 있다.

[실행 결과]와 리스트 항목에 대한 합계를 출력하시오

```
num=[10,15,20,25,30]
```

```
리스트 합계=100
```

리스트 살펴보기

■리스트와 문자열 길이(개수)

- 문자열 길이 출력하기 : len()함수 이용

```
str=["Korea","Seoul","Incheon","Pusan","Jeju"]
for i in str:
    if len(i)==5 :
        print(i)
```

```
Korea
Seoul
Pusan
```

[실습]

■ 리스트 변수 num=[1,2,3,3,4,5,6,7,2,3,8,9,10]에 숫자 항목이 저장되어 있다.

[실행 결과]와 3의 개수를 구하여 출력하시오. Hint : count()함수를 사용하여 개수 구하기

```
3의 개수=3개
```

[실습]

■ 리스트 변수 num=[15,23,18,47,23]에 숫자 항목이 저장되어 있다.

[실행 결과]와 같이 각 항목 데이터에 따른 홀수 개수, 짝수 개수를 출력하시오.

홀수 개수 = 4개

짝수 개수 = 1개

감사합니다.

2

이번
차시에서는

- 리스트 활용

리스트 활용

■빈 리스트 만들기

- 빈 리스트를 만들기 위해서는 항목이 없는 [] 빈 대괄호 기호를 사용

```
>>> numList = []
>>> numList                    #빈 리스트 확인
[]


>>> strList=list()             #내장함수 list()함수를 이용하여 strList라는 이름으로 생성
>>> strList                    #빈 리스트 확인
[]
```

리스트 활용

■리스트 추가하기 : append()

- append() 함수를 사용하게 되면 리스트 맨 끝에 새로운 항목을 추가할 수가 있다.
- 빈 numList 리스트 변수에 숫자 5와 3을 추가해 보자.

```
>>> numList=[]                  #빈 리스트
>>> numList.append(5)           #항목 5추가
>>> numList.append(3)           #항목 3추가
>>> numList                     #리스트 변수 확인
[5, 3]                          #2개의 항목이 추가된 것을 확인할 수 있음
```

리스트 활용

■저장된 리스트 변수 가져와서 합치기 : extend()

▪[실습] numA리스트 변수에 numB 리스트변수를 가져와서 numA리스트 변수 확장하기

```
>>> numA=[1,2,3]
>>> numB=[10,20,30]

>>> numA.extend(numB)

>>> numA

[1, 2, 3, 10, 20, 30]
```

리스트 활용

■숫자 자료형

- 정수형(진법 데이터) : 정수형 데이터로 16진수, 8진수, 2진수도 컴퓨터에서 취급되는 정수형 숫자이다.
- 16진수를 나타낼 때는 앞에 0x를 붙이고, 8진수를 나타낼 때는 0o, 2진수를 나타낼 때는 앞에 0b를 붙인다.

```
>>> a=0xF
>>> type(a)
<class 'int'>

>>> b=0o25
>>> type(b)
<class 'int'>

>>> c=0b0010
>>> type(c)
<class 'int'>
```

리스트 활용

■문자열 자료형

▪문자열 자료형은 큰 따옴표(“”)나, 작은따옴표(‘’)를 사용한다

```
>>> a="Python"
>>> type(a)
<class 'str'>
>>> b='Programming'
>>> type(b)
<class 'str
```

리스트 활용

■리스트 자료형

- 리스트 자료형은 여러 자료들을 목록 형태로 묶어서 관리하는 자료형이다.
- 리스트는 자료가 저장되는 순서를 인덱스로 지정하고 있으며, 지정된 항목은 언제든지 추가, 삽입, 삭제, 내용 변경 등이 가능하다.

리스트변수 = [항목1, 항목2, 항목3....]

■리스트 자료 생성하기

```
>> grade_list=[90,85,70,85,90]
>>> print(grade_list)
[90, 85, 70, 85, 90]
```

리스트 슬라이싱

■리스트 슬라이싱(Slicing)

▪파이썬에서 슬라이싱(slicing)은 리스트에서 한 번에 여러 개의 항목을 범위를 지정하여 추출하는 기법으로 문자열, 리스트, 튜플과 같은 객체로부터 데이터 값을 가져올 수 있다. 리스트를 추출하기 위해서는 [] 대괄호 안에 숫자와 콜론 기호(:)를 이용한다.

```
>>> alphaList = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
```

인덱스 \ 항목	a	b	c	d	e	f	g	h
양수(좌→우)	0	1	2	3	4	5	6	7
음수(우←좌)	-8	-7	-6	-5	-4	-3	-2	-1

리스트 슬라이싱

■리스트 슬라이싱 : 특정 위치 값 인덱스로 추출

- 기억된 문자열 리스트 변수에서 특정 위치 값을 인덱스로 추출하기

```
>>> alphaList=['a','b','c','d','e','f','g','h']        #문자열 항목을 alphaList 리스트변수에 저장
>>> alphaList
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
>>> alphaList[0]                                        #0번째 항목 출력
'a'
>>> alphaList[1]                                        #1번째 항목 출력
'b'
>>> alphaList[-1]                                       #오른쪽에서 1번째 항목 출력
'h'
>>> alphaList[-2]                                       #오른쪽에서 2번째 항목 출력
'g'
```

리스트 슬라이싱

■리스트 슬라이싱 : 범위를 지정하여 추출

▪파이썬 리스트에서 원하는 범위를 지정하여 추출하기 위해서는 인덱스 번호를 사용한다.

```
리스트변수명[:]          #처음부터 끝까지
리스트변수명[시작:]      #시작 위치부터 끝까지
리스트변수명[:종료]      #처음부터 종료-1 인덱스까지
리스트변수명[시작 : 종료]  #가장 많이 사용하는 방법으로 시작부터 종료-1 인덱스까지
```

리스트 슬라이싱

■리스트 슬라이싱

- 슬라이싱을 통해 특정 범위에 해당하는 데이터 값 추출

```
>>> alphaList=['a','b','c','d','e','f','g','h']
```

① 시작 위치부터 끝까지 가져오기

```
>>> alphaList[0:]                    #인덱스 0, 즉 시작 위치부터 끝까지
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
```

② 시작 위치부터 특정 위치 까지 가져오기

```
>>> alphaList[0:5]                   #인덱스 0, 즉 시작 위치부터 시작해서 5개 항목 까지
['a', 'b', 'c', 'd', 'e']
```

③ 사용자가 지정한 범위

```
>>> alphaList[3:7]                   #인덱스 3, 즉 3번 인덱스부터 시작해서 7번 인덱스 까지
['d', 'e', 'f', 'g']
```

[실습]

■alphaList=['a','b','c','d','e','f','g','h']에 문자열을 기억시킨 후, 왼쪽에서 4번째 항목 값과 오른쪽에서 5번째 항목 값을 접근해 보세요.

```
>>> alphaList
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
>>>  _____
'e'
>>>  _____
'd'
```

[실습]

■alphaList=['a','b','c','d','e','f','g','h'] 에 알파벳 문자열을 기억시킨 후 왼쪽→오른쪽 방향,
오른쪽 방향←왼쪽 방향으로 슬라이싱 방법을 사용하여 [‘c’,‘d’, ‘e’, ‘f] 문자 범위를 추출하시오.

```
>> alphaList
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
>>> _____ #왼쪽 2번째 인덱스에서 시작 위치 1부터 시작해서 6개 항목 까지
['c', 'd', 'e', 'f']
>>> _____ #오른쪽에서 왼쪽 방향 –6번째 인덱스에서, 왼쪽 시작 위치 1부터
['c', 'd', 'e', ‘f’ ]          시작해서 6개 항목 까지
```

감사합니다.

3

이번
차시에서는

- 리스트 정렬

리스트 데이터 정렬하기

■리스트 데이터 정렬

▪파이썬에서 리스트, 튜플, 딕셔너리 등에 저장된 데이터를 순서에 의해 정렬해주는 함수가 소트(sort)이다. 소트는 크게 오름차순과 내림차순이 있으며 리스트변수에 저장된 데이터 항목의 위치를 원하는 순서로 정렬할 수가 있다.

- 오름차순(Ascending) : 작은 순에서 큰 순으로 정렬
- 내림차순(Descending) : 큰 순에서 작은 순으로 정렬

① 리스트변수.sort()

② sorted(리스트변수)

리스트 데이터 정렬하기

■sort()와 sorted() 비교

① sort()

- 리스트 자료형에서만 사용할 수 있다.
- 리스트변수.sort() 정렬 시 리스트 항목이 정렬된 내용으로 새롭게 저장된다.
- sorted() 정렬보다 빠르다.
- reverse=True 옵션으로 매개변수를 전달하면 내림차순 정렬이 가능하다.

② sorted()

- 리스트, 문자열, 튜플, 딕셔너리 자료형에서 폭 넓게 사용할 수 있다.
- sorted() 명령 적용 시 정렬은 표시되지만 리스트 원래 항목에 대한 정렬 변화는 저장되지 않는다.
- 정렬된 새로운 리스트를 반환한다.
- sort()에 비해 매개변수가 하나 더 추가되면 reverse 매개변수로 True를 전달하면 내림차순 정렬이 가능하다.

리스트 데이터 정렬하기

■리스트 오름차순 정렬 : 작은 값에서 큰 값으로 정렬하기

```
>>> numList=[43,12, 3, 7, 72, 25]
>>> numList. sort()
>>> numList
[3, 7, 12, 25, 43, 72]
```

[실습]

■strLIst 문자열 변수에 알파벳 항목을 기억시킨 후, 오름차순으로 정렬하시오.

```
>>> strList=['K','C', 't', 'E', 'b', 'a', 'M']
>>>_____________________________________
>>> strList
['C', 'E', 'K', 'M', 'a', 'b', 't']
```

리스트 데이터 정렬하기

■리스트 내림차순 정렬

▪내림차순으로 정렬하기 위해서는 리스트변수.sort 명령을 입력한 후, 추가 옵션으로 (reverse=True)로 주게 되면 리스트에 저장된 항목을 내림차순으로 정렬할 수 있다.

① 리스트변수.sort(reverse=True) #리스트만 정렬 가능

② sorted(리스트변수, reverse=True) #리스트, 문자열 모두 정렬 가능

리스트 데이터 정렬하기

■리스트 내림차순 정렬 : 큰 값에서 작은 값

```
>>> num = [15, 11, 9, 6, 3]
>>> num.sort(reverse=True)        #sort() 함수에 옵션 reverse=True를 추가, 내림차순 정렬
>>> num                           #숫자 항목이 내림차순으로 정렬된 것을 확인할 수 있음
[15, 11, 9, 6, 3]
```

[실습]

■리스트 변수 numList = [11,23,5,7,15,9,8]에 숫자 정수 항목이 기억되어 있다.

■[실행 결과]와 같이 출력되도록 프로그램을 작성하시오.

```
정렬 전: [11, 23, 5, 7, 15, 9, 8]
---------------------------------
정렬 후: [23, 15, 11, 9, 8, 7, 5]
```

감사합니다.