

Stage 1 - 점심 메뉴 관리하기

1주차 때 메뉴판을 작성했던 것이 기억나나요? 그때 메뉴에 대해서 어떤 변수에 담지않고, 그대로 사용하였습니다. 이번엔 리스트라는 자료형을 가지고 점심 메뉴를 변수로 관리하겠습니다.

다시보는 메뉴판

1주차 때 format문을 사용하여 메뉴판을 출력했던 것이 기억나시죠? 그 때 format문 안에 메뉴 이름과 가격을 적어보았는데요. 그곳에 적었던 내용들은 변수에 담아두지 않아서 가지고 있지 못합니다. 그래서 이번 시간에는 리스트라는 자료형을 사용해서 하나의 군집을 이루는 자료들을 정리하는 법을 배워보도록 하겠습니다.

life/lunch_recommender.py

```
1 print("점심 추천기 프로그램입니다!")
2
3 print(''
4 메뉴판
5 =====
6 {0:<10}{1:>10}{won}
7 {2:<10}{3:>10}{won}
8 {4:<10}{5:>10}{won}
9 {6:<10}{7:>10}{won}
10 {8:<10}{9:>10}{won}
11 {10:<10}{11:>10}{won}
12 {12:<10}{13:>10}{won}
13 =====
14 ''.format('갈비탕', 10000,
15           '떡볶이', 7000,
16           '오뎅', 5000,
17           '감자탕', 8000,
18           '김치찌개', 8000,
19           '제육볶음', 7000,
20           '김치볶음밥', 5000, won="원"))
```

리스트의 사용

2주차 마지막 시간에 문자열 인덱싱과 슬라이싱을 배웠습니다. 리스트의 원리도 이와 같습니다. 문자열은 문자 하나하나를 가져올 수 있었다면, 리스트에서는 더 큰 단위의 자료들을 가져올 수 있습니다. 깊이 살펴보도록 하겠습니다.

리스트란 무엇인가?

첫째로, 순서가 존재합니다. 각 하나하나의 요소마다 자기 번호가 있습니다. 이 번호를 우리는 인덱스라고 부릅니다. 우리는 인덱스를 통해 리스트를 쉽게 사용할 수 있습니다. 둘째로, 중복을 허용합니다. 집합안에 같은 값이 들어와도 다른 인덱스를 가져서 같은 값이지만 다른 값으로 생각할 수 있습니다. 셋째로, 요소안에는 모든 자료형이 가능합니다. 리스트 또한 자료형입니다. 그래서 리스트 안에 리스트가 들어올 수도 있고요. 숫자, 문자는 물론, 불, 딕셔너리, 집합 같은 자료형들도 요소로서 들어올 수 있습니다.

0	1	2	3	4	5
'I love you'	2	34.244	[a, b, c, 1123]	2	'갈비탕'

리스트 예시

- 각각의 요소들은 0부터 시작하여 인덱스를 가집니다.
- 1번과 4번에는 같은 값으로 2가 들어가 있습니다.
- 각 요소들은 0번 부터 문자열, 정수, 실수, 리스트, 정수, 문자열 형태로 들어가 있습니다.

리스트 사용 방법

리스트의 생성

리스트를 사용하기 위해서는 변수를 리스트로 만들어주어야 합니다. 변수를 리스트로 만드는 방법은 두가지 방법이 있는데, 하나는 생성자라는 개념입니다. 리스트를 생성하는 특정 명령어이다. 아래와 같이 작성하면 됩니다.

example

```
1 my_list = list()
```

두번째로는 리스트 형태를 그대로 선언하면 됩니다. 이 때 값을 넣어주면 초기화까지 가능합니다. 아래를 살펴봅시다.

example

```
1 my_list = []
2 # 아무것도 들어있지 않은 리스트로 선언
3
4 my_list = ['I love you', 2, 34.244, [a, b, c, 1123], 2, '갈비탕']
5 # 위의 예시와 같이 리스트를 초기
```

리스트 인덱싱

인덱스를 통해 리스트를 가져오는 방법을 알아봅시다.

example

```
1 my_list = ['I love you', 2, 34.244, [a, b, c, 1123], 2, '갈비탕']
2 # 위의 예시와 같이 리스트를 초기
3
4 print(my_list[0])
5 # I love you
6
7 print(my_list[4])
8 # 2
9
10 print(my_list[3])
11 # [a, b, c, 1123]
```

문자열과 똑같아요! 어렵지 않죠?

요소 추가 생성

이제는 요소를 추가하는 방법에 대해 알아보도록 합니다. 초기 생성과정이 아닌 코드 중간에 요소가 추가되는 과정을 동적인 생성 과정이라 불러요! 반대로 초기화 과정에서 생성하는 과정은 정적인 생성이라고 부르고요. 동적인 생성 과정을 알아봅시다.

example

```

1 my_list = ['I love you', 2, 34.244, [a, b, c, 1123], 2, '갈비탕']
2 # 위의 예시와 같이 리스트를 초기
3
4 my_list.append(123)
5 # my_list = ['I love you', 2, 34.244, [a, b, c, 1123], 2, '갈비탕', 123]
6
7 my_list.append('okay good')
8 # my_list = ['I love you', 2, 34.244, [a, b, c, 1123], 2, '갈비탕', 123, 'okay good']

```

위와 같이 점을 찍고 'append' 하면 리스트에 요소를 추가적으로 생성이 가능합니다.

리스트 응용

리스트는 다양한 방법으로 응용이 가능합니다. 요소들을 순서대로 정리할 수 도 있고요. 뒤집을 수 도 있고요. 위치를 알아올 수도 있죠. 그것을 다 다루기에는 여러분도 지치고 힘드시기 때문에, 이 시간에는 주의해서 알아야 할 것만 알려드리고요. 필요한 리스트의 응용 방법들은 구글링을 통해 직접 문제를 해결해나가기기를 바랍니다.

리스트의 덧셈과 반복

리스트와 리스트를 더하거나 리스트에 정수를 곱하면 어떻게 될까요? 아마 여러분이 생각한 것이 맞으실 겁니다. 이러한 면들 때문에 파이썬을 굉장히 직관적인 언어다라고 많이들 말하는 것 같아요. 리스트의 덧셈에 대해 봅시다.

example

```

1 my_list1 = [1, 2, 3, 4, 5]
2 my_list2 = [1, 2, 3, 4, 5]
3
4 my_list3 = my_list1 + my_list2
5 print(my_list3)
6 # [1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
7
8 my_list4 = my_list1 * 3
9 print(my_list4)
10 # [1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5]

```

리스트의 길이

리스트의 길이는 무엇을 뜻할까요? 맞습니다. 리스트의 요소들이 몇개인지를 물어봅니다. 코딩을 하면서 제일 많이 사용하게 되는 것이 아닌가 싶은데요. 예를 들면, 메뉴판을 작성할 때도 메뉴가

몇개인지 알아야 그 중에서 추천해 줄 수가 있잖아요. 그래서 리스트의 길이를 아는 것은 정말 중요합니다.

example

```
1 my_list1 = [1, 2, 3, 4, 5]
2
3 len_of_list1 = len(my_list1)
4 print(len_of_list1)
5 # 5
```

위의 코드와 같이 리스트의 길이는 'len'으로 알아냅니다. 'len'은 길이를 뜻하는 'length'의 줄임말입니다. 잘 기억하세요!

메뉴리스트와 가격리스트

가격리스트를 정적으로 생성하기

리스트의 생성 방법을 알아보면서 정적인 생성 방법에 대해서도 보여드렸습니다. 리스트를 초기화하면서 값을 선언하는 과정을 정적 생성이라고 부르는데요. 메뉴의 가격리스트를 정적으로 생성해 봅시다.

life/lunch_recommender.py

```
1 prices = [10000, 7000, 5000, 8000, 8000, 7000, 5000]
```

메뉴리스트를 동적으로 생성하기

이번에는 메뉴리스트를 생성합니다. 위의 가격리스트처럼 정적으로 생성해도 무방합니다. 그러나 우리는 연습하는 과정이므로 'append'를 사용해서 동적으로 생성해보겠습니다.

life/lunch_recommender.py

```
1 menus = list()
2 menus.append("갈비탕")
3 menus.append("떡볶이")
```

```
4 menus.append("오뎅")
5 menus.append("감자탕")
6 menus.append("김치찌개")
7 menus.append("제육볶음")
8 menus.append("김치볶음밥")
```

리스트를 이용하여 메뉴판 그리기기

생성한 메뉴리스트, 가격리스트를 인덱싱을 통하여 값을 가져와 출력하는 코드를 수정해봅시다.

life/lunch_recommender.py

```
1 print('''
2 메뉴판
3 =====
4 {0:<10}{1:>10}{won}
5 {2:<10}{3:>10}{won}
6 {4:<10}{5:>10}{won}
7 {6:<10}{7:>10}{won}
8 {8:<10}{9:>10}{won}
9 {10:<10}{11:>10}{won}
10 {12:<10}{13:>10}{won}
11 =====
12 ''' .format(menus[0], prices[0],
13             menus[1], prices[1],
14             menus[2], prices[2],
15             menus[3], prices[3],
16             menus[4], prices[4],
17             menus[5], prices[5],
18             menus[6], prices[6], won="원"))
```