

Stage 4 - 출력을 보기 좋게

메뉴판에서는 메뉴들이 반복되어 출력되고, 로또 발생기에서는 로또 번호들이 반복되어 출력됩니다. for문을 사용하여 반복되는 문장을 보기 좋게 바꿔봅시다.

반복코드 합치기

이전 메뉴판 그리기와 로또 발생기에서 출력하는 부분의 코드를 보겠습니다. 보기 좋게 나열되어 있지만, 만약에 메뉴가 1000000개가 넘거나 로또 번호가 몇 개인지 모른다면 저렇게 쓸 수 있을까요? 아마 힘드실 겁니다. 그걸 줄이기 위해서 이번에도 반복문을 사용해볼텐데요. 이번에는 while문이 아닌 for문을 사용해서 해결하겠습니다.

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 '''
13     .format(menus[0], prices[0],
14             menus[1], prices[1],
15             menus[2], prices[2],
16             menus[3], prices[3],
17             menus[4], prices[4],
18             menus[5], prices[5],
19             menus[6], prices[6], won="원"))
```

life/lotto_numbers_creator.py

```
1 print('''
2 랜덤 로또 발생기 입니다. (중복이 가능합니다)
3 당첨 번호 : {0}, {1}, {2}, {3}, {4}, {5}
4 2등 보너스 볼 : {6}
5 '''
6     .format(lotto_numbers[0],
```

```
6         lotto_numbers[1],
7         lotto_numbers[2],
8         lotto_numbers[3],
9         lotto_numbers[4],
10        lotto_numbers[5],
11        lotto_numbers[6]))
```

for문이란

for문이란 무엇인가

앞서 반복문으로 while문에 대해 배웠지만, 사실 for문을 더 많이 사용합니다. 둘은 분명한 차이가 존재합니다. while문은 뒤에 조건이 왔다면, for문은 for문만의 반복에 대한 형식이 존재합니다. 아래와 같습니다.

example

```
1  for 변수 in 리스트(또는 튜플, 문자열):
2      수행할 문장
3      수행할 문장
4      수행할 문장
5      .
6      .
```

- 리스트나 튜플, 문자열의 길이만큼 반복됩니다.
- 반복이 되면서 요소 하나하나가 변수에 담겨집니다.
- for문 내에서 'for'와 'in' 사이의 수를 사용할 수 있습니다.

enumerate문의 사용

우리는 for문을 사용할 때 enumerate문을 함께 사용하여 더욱 편리하게 사용합니다.

'enumerate'는 '열거하다'라는 뜻입니다. enumerate문 안에 'in' 뒤에 오는 리스트나 튜플, 문자열을 넣어주면, 그 안에 요소만 전달해주는 것이 아니라 그 요소에 해당하는 인덱스까지 전달해줍니다. 예를 통해 살펴보겠습니다.

example

```

1 my_list = ['a', 'b', 'c', 'd', 'e']
2 for index, value in enumerate(my_list):
3     print("index : {0}, value : {1}".format(index, value))
4
5 # index : 0, value : a
6 # index : 1, value : b
7 # index : 2, value : c
8 # index : 3, value : d
9 # index : 4, value : e

```

메뉴 출력 바꾸어보기

메뉴판 메뉴 반복 해결

life/lunch_recommender.py

```

1 print("메뉴판")
2 print("=====")
3 for index, value in enumerate(menus):
4     print("{0:<10}{1:>10}{won}".format(value, prices[index], won='원'))
5 print("=====")

```

로또 번호 반복 해결


life/lotto_numbers_creator.py

```

1 print("랜덤 로또 발생기 입니다.")
2 print("당첨 번호 : ", end="")
3
4 for index, value in enumerate(lotto_numbers):
5     if index == len(lotto_numbers)-1:
6         print("\n2등 보너스볼 : {0}".format(value))
7     else:
8         print(value, end=' ')

```

- index가 마지막 인덱스(len(lotto_numbers)-1)이면 2등 보너스볼에 대한 내용을 출력합니다.

 print문 뒤에 `end=""` 를 넣어주면 기본적으로 print문이 개행되는 것을 방지하고 원하는 끝 문자를 설정할 수 있습니다.