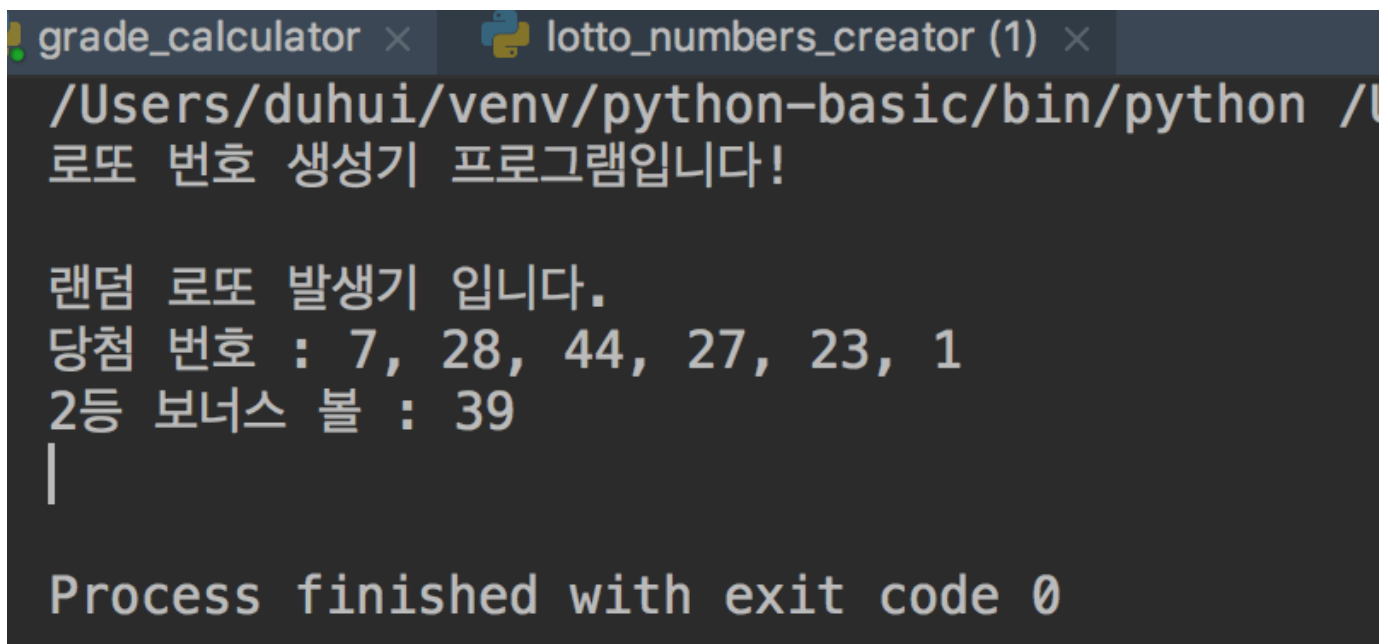


Stage 3 - 로또 발생기 중복 제거

앞선 Challenge를 통하여 로또 발생기를 만들어봤습니다. 하지만 중복된 숫자가 들어오는 문제가 있는데요. 이를 해결하기 위해서 중복이 났을때 다시 번호를 부여받을 수 있도록 하는 코드를 짜보겠습니다. 반복문을 이용합니다.

중복 제거

Challeng 1을 통하여 로또 번호를 생성해보았습니다. 그러나 현재의 로또 발생기는 중복을 허용합니다. 하지만 실제 로또 복권을 추첨할 때는 뽑혔던 숫자가 다시 뽑히는 일은 없습니다. 우리는 이 문제를 반복문을 사용해서 해결해보도록 하겠습니다.



```
grade_calculator x  python  /Users/duhui/venv/python-basic/bin/python /
로또 번호 생성기 프로그램입니다!

랜덤 로또 발생기 입니다.
당첨 번호 : 7, 28, 44, 27, 23, 1
2등 보너스 볼 : 39
|

Process finished with exit code 0
```

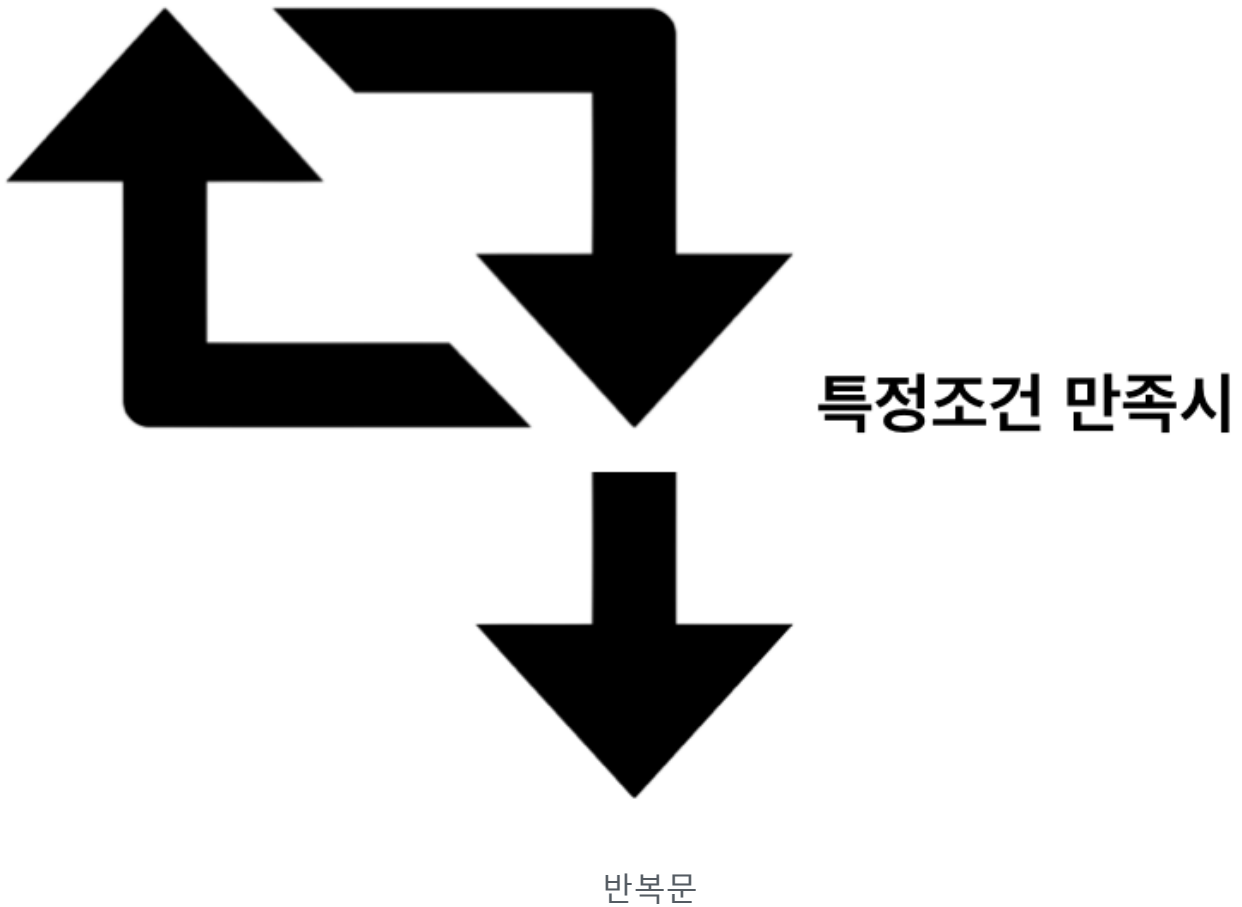
완성된 로또 발생기

반복문의 이해

반복문이란 무엇인가

반복문은 특정한 조건이 만족할 때까지 똑같은 문장을 반복적으로 수행하는 것을 의미합니다. 컴퓨터와 우리와의 차이는 바로 이런점이 아닐까 생각됩니다. 계산이 엄청나게 빠르다는 것이죠. 다

시 말해서, 반복적으로 무언가를 계산하는 속도가 우리와는 비교도 안 될 만큼 빠릅니다. 지금부터 반복문을 제대로 살펴보겠습니다.



while문 이해하기

while문의 사용법

while 뒤에 2주차때 배웠던 조건문에 사용되는 조건 관련된 문장이 쓰여집니다. 예를 들어 $a > 3$, $a == 1$, True, False 등이 올 수 있겠죠. 그 다음에 ':' 콜론을 찍고요. TAB으로 구분하여 반복할 문장을 작성합니다.

example

```
1 while 특정 조건:
2     수행할 문장
3     수행할 문장
4     수행할 문장
5     .
6     .
```

continue와 break

반복문을 사용할 때 꼭 알아두어야 하는 것이 'continue'와 'break'입니다. 'continue'는 '계속하다'라는 의미를 가지고 있고, 'break'는 '멈추다'라는 의미가 있죠. 반복문에서도 같은 의미로 사용됩니다. 반복문 안에서 'continue'가 나오면 '다음 반복 할 차례로 계속해라'라는 의미입니다. 'break'가 나오면 '이제 반복문을 멈추어라'라는 의미입니다. 아래의 예시를 통해 살펴봅시다.

continue

example

```
1  a = 0
2
3  while a != 10:
4      a = a+1
5      print("aaaaaa")
6      continue
7      print("bbbbbb")
8      print("cccccc")
```

- "aaaaaa"만 10번 출력됩니다.
- continue가 있으면 다음 반복으로 계속되기 때문에, 'bbbbbb'와 'cccccc'를 출력하는 문장은 수행되지 않습니다.

break

example

```
1  a = 0
2
3  while a != 10:
4      a = a+1
5      print("aaaaaa")
6      break
7      print("bbbbbb")
8      print("cccccc")
```

- "aaaaaa"가 1번 출력됩니다.
- "aaaaaa"를 출력하고 'break'를 만나서 반복문이 멈추었기 때문입니다.

로또 번호 중복 제거하기

만약에 중복으로 2개의 숫자가 같이 나와서, 번호가 제거되면 한번 더 난수를 발생시키는 수행을 실행해주어야 합니다. 그래서 로또 번호를 가진 리스트의 길이가 7이 될 때까지 중복을 모두 피하는 리스트를 만들어내야 합니다.

while문 사용하기

```
life/lotto_numbers_creator.py
```

```
1 while len(lotto_numbers) != 7:
2     new_number = random.randrange(1, 46)
3     lotto_numbers.append(new_number)
```


- while문 안에서 난수를 생성합니다.
- 난수를 리스트에 추가합니다.
- 리스트의 길이가 7이 될 때까지 반복합니다.

중복 검사하기

새로운 숫자가 로또 번호 리스트 안에 있으면 로또 번호 리스트 안에 추가해서는 안됩니다. 이를 위해 검사하는 방법은 아래와 같습니다.

```
life/lotto_numbers_creator.py
```

```
1 while len(lotto_numbers) != 7:
2     new_number = random.randrange(1, 46)
3     if new_number not in lotto_numbers:
4         lotto_numbers.append(new_number)
```

 'if new_number not in lotto_numbers' 는 말 그대로 lotto_numbers 리스트 안에 new_number가 없으면 아래의 문장을 실행하라는 것 입니다.