

Stage 3 - 스트라이크! 볼! 아웃!

게임에 대한 이해가 완료된 상태에서 스트라이크와 볼을 판별하고, 결과가 어떻게 나오는지 출력해봅시다.

스트라이크! 볼! 아웃!

악어 게임의 가장 핵심적인 부분을 코딩해보도록 하겠습니다! 스트라이크, 볼, 아웃 판별법에 대해 다시 말씀드리겠습니다.

- 스트라이크 : 자리, 숫자 모두 일치
- 볼 : 자리는 불일치, 숫자 일치
- 아웃 : 4개의 수 모두가 자리 숫자 불일

```
baseball_game (3) × /Users/duhui/venv/python-basic/bin/python /Users/duh
야구 게임 프로그램입니다!
[2, 7, 1, 9]
답을 맞추어 보세요 : 2345
index : 0, value : 2
index : 1, value : 3
index : 2, value : 4
index : 3, value : 5
1 스트라이크 0 볼입니다.

Process finished with exit code 0
```

스트라이크 볼 판별

스트라이크와 볼 검사하기

변수 생성하기

먼저는 스트라이크와 볼을 카운트 할 수 있는 변수를 만들어 줍니다. 최초에 스트라이크와 볼은 0으로 초기화해주도록 하겠습니다.

games/baseball_game.py

```
1  # 유저에게 입력 받음
2  your_answer = input("답을 맞추어 보세요 : ")
3
4  strike = 0
5  ball = 0
6
7  # 스트라이크, 볼 검사
8  for index, value in enumerate(your_answer):
9      print("index : {0}, value : {1}".format(index, value))
```

값 비교하기

이제 스트라이크와 볼을 판별하는 코드를 넣어주겠습니다. 앞서 만든 'answer'리스트의 같은 인덱스의 숫자와 사용자가 입력한 숫자가 일치하면 스트라이크를 1 더해주고, 그게 아닌 상태에서 'answer'에 포함되어 있다면 볼을 1 더해줍니다.

games/baseball_game.py

```
1  # 스트라이크, 볼 검사
2  for index, value in enumerate(your_answer):
3      print("index : {0}, value : {1}".format(index, value))
4      if int(value) == answer[index]:
5          strike += 1
6      elif int(value) in answer:
7          ball += 1
```

결과 출력하기

조건에 따라 출력하기

이제 스트라이크와 볼 값을 가지고 사용자에게 알려주는 일만 남았습니다. strike와 ball이 모두 0이면 'Out' 상태를 출력하고요. 그게 아니라면 스트라이크와 볼 정보를 출력하여 보여줍니다.

games/baseball_game.py

```
1  if strike == 0 and ball == 0:
2      print("Out!!!!!!!!!!!!")
3  else:
4      print("{0} 스트라이크 {1} 볼입니다.".format(strike, ball))
```