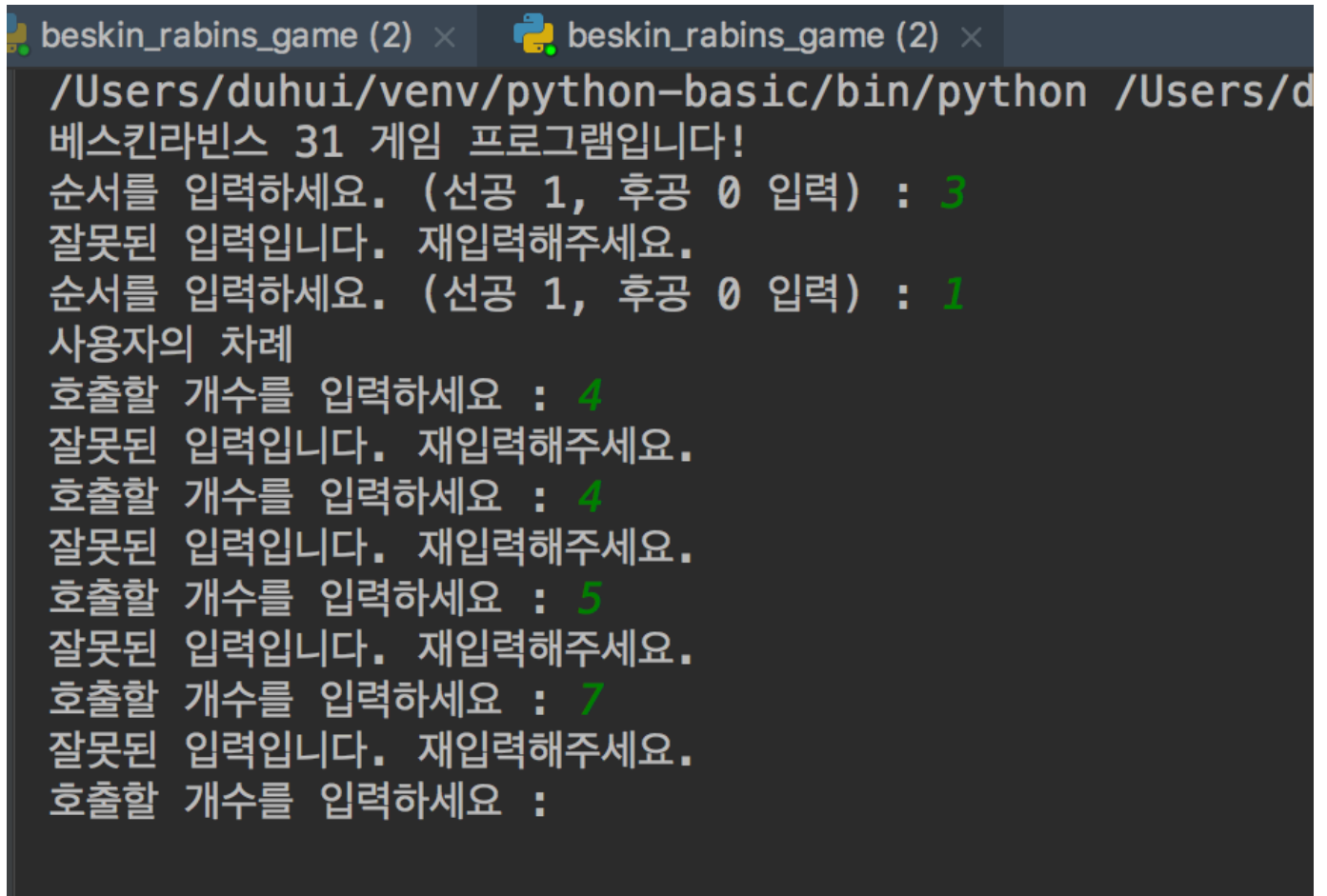


Stage 3 - 디테일과 예외처리

디테일하고 완벽한 게임

수고하셨습니다! Challenge 2를 여러분 스스로의 힘으로 해내셨다면, 이제 무엇이든 사용하는 방법만 익힌다면 만드실 수 있을 것이라고 생각합니다. 하지만 지금까지 만든 프로그램으로 일반 사람들이 프로그램을 사용하게 된다면 예러가 많다고 항의가 들어올 수 있습니다. 지금 시간에는 더욱 디테일하게 게임을 만들어보는 시간을 가져보겠습니다.



```
beskin_rabins_game (2) × beskin_rabins_game (2) ×
/Users/duhui/venv/python-basic/bin/python /Users/d
베스킨라빈스 31 게임 프로그램입니다!
순서를 입력하세요. (선공 1, 후공 0 입력) : 3
잘못된 입력입니다. 재입력해주세요.
순서를 입력하세요. (선공 1, 후공 0 입력) : 1
사용자의 차례
호출할 개수를 입력하세요 : 4
잘못된 입력입니다. 재입력해주세요.
호출할 개수를 입력하세요 : 4
잘못된 입력입니다. 재입력해주세요.
호출할 개수를 입력하세요 : 5
잘못된 입력입니다. 재입력해주세요.
호출할 개수를 입력하세요 : 7
잘못된 입력입니다. 재입력해주세요.
호출할 개수를 입력하세요 :
```

잘못된 입력을 예외 처리

31까지만 말하기

현재 상황

Challenge 1을 통하여 31까지 말하도록 처리해주신 분이 있다면, 정말 훌륭하십니다! 그래도 한 번 같이 보겠습니다. 모범 답안으로 짜여진 코드에는 반복문의 종료 조건이 31보다 크거나 같으면 종료가 됩니다. 따라서 마지막에 호출하는 사람이 31을 넘어서까지 말할 수 있습니다. 이를 방지하기 위해서 번호를 출력할 때, 31을 호출하는지 판단해서 31보다 큰 수가 호출된다면 반복문을 빠져나오는 코드를 추가해주도록 합니다.

break문

break문에 대해서 잠깐 짚고 넘어갔지만, 사용해본 적은 없습니다. break문은 반복을 중간에 중지해주는 역할을 합니다. 우리는 31이라는 수를 호출하고 반복을 강제로 중지해야하기 때문에 이번에 break문을 사용합니다.

아래의 코드를 사용자와 컴퓨터가 숫자를 출력하는 부분에 삽입해주면 정상적으로 작동합니다.

```
games/beskin_rabins_game.py
```

```
1  if call == 31:
2      break
```

잘못된 입력 방지하기

잘못된 입력

처음에 순서를 정할 때, 안내문으로 0 또는 1을 입력해달라고 요청했습니다. 그러나 사용자의 실수로 0 또는 1이 아닌 'abc'나 'ㄱ ㄴ ㅇ'와 같은 문자가 들어오거나 이외의 숫자가 들어오면 예러가 나거나 게임이 진행되지 않습니다. 그래서 이를 방지하기 위해 사용자로부터 입력받은 정보가 0 또는 1인지 판단하는 코드를 삽입합니다. 사용자가 숫자를 호출할 때도 마찬가지입니다. 이 때도 1, 2, 3 세 수 중 하나가 입력되어야 게임이 정상 진행됩니다.

잘못된 입력 개선하기

순서 입력받기 개선

```
game/beskin_rabins_game.py
```

```
1  while True:
```

```

2     order = input('순서를 입력하세요. (선공 1, 후공 0 입력) : ')
3     if order in ['0', '1']:
4         order = int(order)
5         break
6     else:
7         print("잘못된 입력입니다. 재입력해주세요.")

```

- while문을 사용하여 무한 반복합니다.
- 조건 검사 시 ['0', '1'] 리스트를 만들어 리스트 안에 문자가 포함되는지 판단합니다.
- break문을 사용하여 while문을 중지합니다.

사용자 호출 입력 개

game/beskin_rabins_game.py

```

1     print('사용자의 차례')
2     while True:
3         size_of_call = input("호출할 개수를 입력하세요 : ")
4         if size_of_call in ['1', '2', '3']:
5             size_of_call = int(size_of_call)
6             break
7         else:
8             print("잘못된 입력입니다. 재입력해주세요.")

```

- while문을 사용하여 무한 반복합니다.
- 조건 검사 시 ['1', '2', '3'] 리스트를 만들어 리스트 안에 문자가 포함되는지 판단합니다.
- break문을 사용하여 while문을 중지합니다.