

Wheel Of Fortune KATA

Wheel of Fortune – Bonus Game

게임 참가자가 뒤집혀 있는 알파벳을 맞추는 게임



이미지 출처 : https://www.youtube.com/watch?v=yAWHr-lJk_Q

기본 점수

알파벳을 맞추면, 한 글자당 \$100씩 획득

다음 턴에서도 알파벳을 맞추면, 한 글자당 \$200씩 획득

다음 턴에서도 알파벳을 맞추면, 한 글자당 \$300씩 획득

위와 같이 연속으로 N회 맞추면, 한 글자당 $N \times \$100$ 씩 획득

Let's First 점수

한 줄 중 가장 앞글자를 가장 먼저 맞추면 \$1000 추가 점수 획득 (Line 별 1회 chance)

Let's Second 점수

Let's First 발동 이후 바로 다음 턴에서,

해당 Line의 글자를 하나 이상 맞추면 \$2000 추가 점수 획득 (Line 별 1회 chance)

점수 획득 규칙 예시 1

L 시도, 1개 정답 → + \$100

E 시도, 2개 정답 → + \$200 x 2개

R 시도, 2개 정답 → + \$300 x 2개

현재까지 상금 = 100 + 400 + 600 = \$1,100



점수 획득 규칙 예시 2

게임이 시작하자마자 E를 시도 했다면?

- \$100 x 3개 = +\$300 획득
- 두 번째 Line 에서 Let's First 점수 획득 +\$1,000
- 총 \$1,300 획득



바로 다음 턴에서 R을 시도 했다면?

- \$200 x 2개 = +\$400 획득
- Let's Second 발동 = +\$2000 획득
- 총 상금 1300 + 2400 = \$3700 획득



입력 규칙

첫 줄에는 문자열의 수 입력 (LineCnt : 1 ~ 4)

다음 줄 에는 LineCnt 만큼 정답 문자열이 입력 됨

그 다음 줄에는 게임 참가자가 시도하는 A ~ Z, 26개 문자 입력 받음

예시

2

BUILDLEV
EATREALROBOT

정답문자열,
게임 참가자에게 공개가 안됨

ERABCD EFGHIJKLMNOPQRSTUVWXYZ

퀴즈 참가자의 시도
A ~ Z, 중복 없는 26개의 글자

세부 규칙

정답 문자열 세부 조건

최대 Line 수 : 4

각 줄 당 최대 글자수 : 15개

대문자만 취급



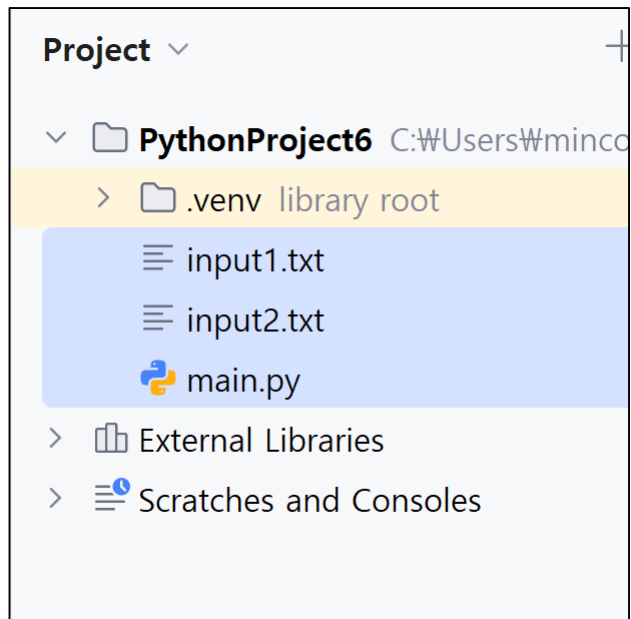
Legacy Code 링크

- 소스코드 링크

<https://github.com/mincoding1/WheelOfFortune>

KATA 세팅하기

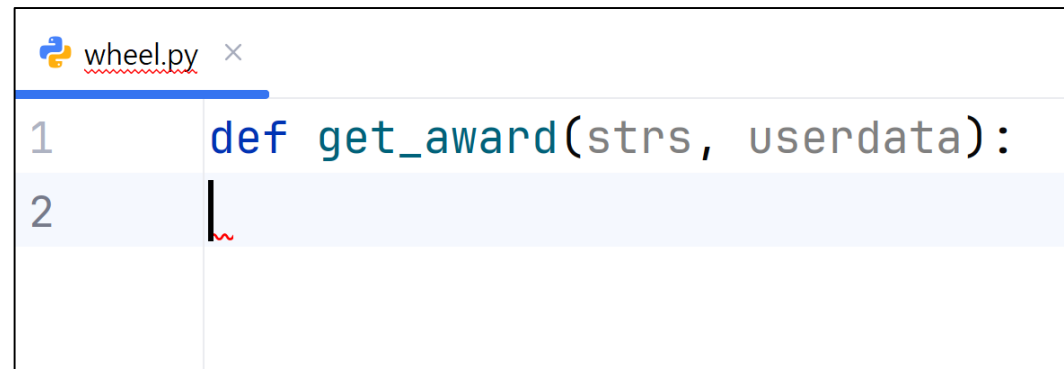
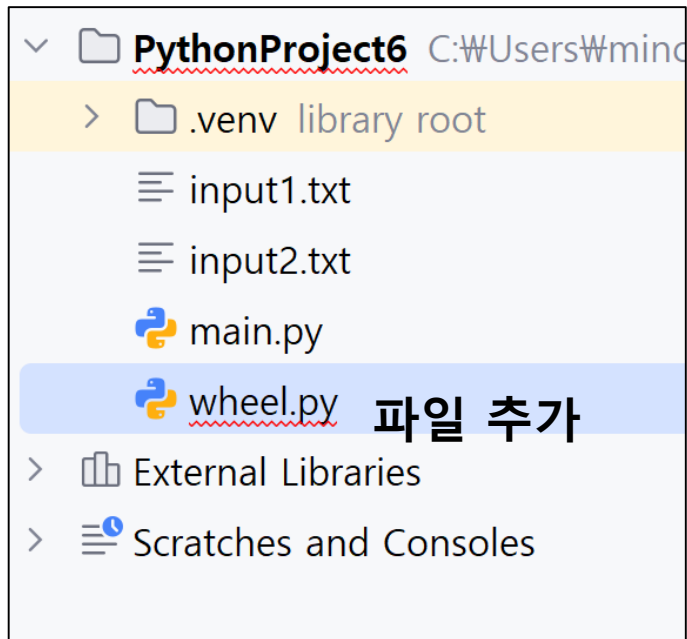
- input1.txt, input2.txt 파일을 추가한다.
github 파일 내용을 복사 붙여넣기로 내용을 채운다.
- Run을 했을 때 input1.txt의 상금 \$6500이 나오고 input2.txt는 \$9500 이 출력
되면 세팅 완료



\$6500

입력 값을 Unit Test로 변경하기 1

리팩토링할 때마다 매번 "input1.txt", "input2.txt"을 변경해가며 테스트할 수 없기에, Unit Test를 할수 있도록 준비과정을 거친다.



wheel.py 파일 신규 추가 후
코드 작성

입력 값을 Unit Test로 변경하기 2

코드 이동시키기

main.py

```
main.py x
1  # str에서 특정 index 문자만 변경
2  # 예시 : "____"에서 2번 index 문자를 Q로 변경
3  # replace_chat_at("____", 2, 'Q') # 결과 : "__Q_"
4  def replace_char_at(str, index, new_char):
5      return str[:index] + new_char + str[index+1:]
6
7  # 파일 입력
8  # input1.txt 파일을 읽어와 객체에 값 세팅
9  with open('input1.txt', 'r', encoding='utf-8') as file:
10     lines = [line.strip() for line in file]
11
12     n = int(lines[0])
13     strs = lines[1:1+n]
14     userdata = lines[1+n]
15
16     map = [[0 for _ in range(15)] for _ in range(4)]
17     conCnt = 0
18     ffirst = [0] * 4
19     sum = 0
20     chance = [-1] * 4
```

코드 이동시키기

wheel.py

```
wheel.py x
1  # str에서 특정 index 문자만 변경
2  # 예시 : "____"에서 2번 index 문자를 Q로 변경
3  # replace_chat_at("____", 2, 'Q') # 결과 : "__Q_"
4  def replace_char_at(str, index, new_char):
5      return str[:index] + new_char + str[index+1:]
6
7  def get_award(strs, userdata):
8
```

입력 값을 Unit Test로 변경하기 2

코드 이동시키기

```
main.py
1 # 파일 입력
2 # input1.txt 파일을 읽어와 객체에 값 세팅
3 with open('input1.txt', 'r', encoding='utf-8') as file:
4     lines = [line.strip() for line in file]
5
6 n = int(lines[0])
7 strs = lines[1:1+n]
8 userdata = lines[1+n]
9
10 map = [[0 for _ in range(15)] for _ in range(4)]
11 conCnt = 0
12 ffirst = [0] * 4
13 sum = 0
14 chance = [-1] * 4
15 # 하나씩 처리
16 # 26글자 for문 돌면서 퀴즈 참석자가 하나씩 시도를 하는 것
17 for i in range(26):
18     # 2000 달러 찬스를 얻었는지 검사
19     for y in range(len(strs)):
20         if chance[y] != -1:
21             for x in range(len(strs[chance[y]])):
22                 if map[chance[y]][x] == 0 and strs[chance[y]][x] == userdata[i]:
23                     # 획득 성공시 2000달러를 얻는다.
24                     sum += 2000
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45 # used 배열
46 map[y][x] = 1
47
48 # 해당 문자를 _ 로 바꿔버린다.
49 replace_char_at(strs[y], x, '_')
50 flag = 1
51 # 동일한 문자 개수를 counting 한다.
52 passCnt += 1
53
54 if flag == 1:
55     conCnt += 1
56     sum += (conCnt * 100) * passCnt
57 else:
58     conCnt = 0
59     chance = [-1] * 4
60
61 print("$" + str(sum))
62
63
```

main.py

코드 이동시키기

```
wheel.py
1 # str에서 특정 index 문자만 변경
2 # 예시 : "----"에서 2번 index 문자를 Q로 변경
3 # replace_chat_at("----", 2, 'Q') # 결과 : "--Q--"
4 def replace_char_at(str, index, new_char):
5     return str[:index] + new_char + str[index+1:]
6
7 def get_award(strs, userdata):
8     map = [[0 for _ in range(15)] for _ in range(4)]
9     conCnt = 0
10    ffirst = [0] * 4
11    sum = 0
12    chance = [-1] * 4
13    # 하나씩 처리
14    # 26글자 for문 돌면서 퀴즈 참석자가 하나씩 시도를 하는 것
15    for i in range(26):
16        # 2000 달러 찬스를 얻었는지 검사
17        for y in range(len(strs)):
18            if chance[y] != -1:
19                for x in range(len(strs[chance[y]])):
20                    if map[chance[y]][x] == 0 and strs[chance[y]][x] == userdata[i]:
21                        # 획득 성공시 2000달러를 얻는다.
22                        sum += 2000
23                        break
24                    chance[y] = -1
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
```

wheel.py

입력 값을 Unit Test로 변경하기 3

최종 결과값을

return 하는 함수로 변경하기

```
50         passCnt += 1
51     if flag == 1:
52         conCnt += 1
53         sum += (conCnt * 100) * passCnt
54     else:
55         conCnt = 0
56         chance = [-1] * 4
57
58     return sum
```

코드 추가

wheel.py

최종 결과를 return 코드를 추가한다.

입력 값을 Unit Test로 변경하기 4

기존 main.py 파일이 정상동작되는지 확인한다.

```
main.py ×
1  from wheel import get_award
2
3  # 파일 입력
4  # input1.txt 파일을 읽어와 객체에 값 세팅
5  with open('input1.txt', 'r', encoding='utf-8') as file:
6      lines = [line.strip() for line in file]
7
8  n = int(lines[0])
9  strs = lines[1:1+n]
10  userdata = lines[1+n]
11
12  sum = get_award(strs, userdata)
13  print("$" + str(sum))
14
```

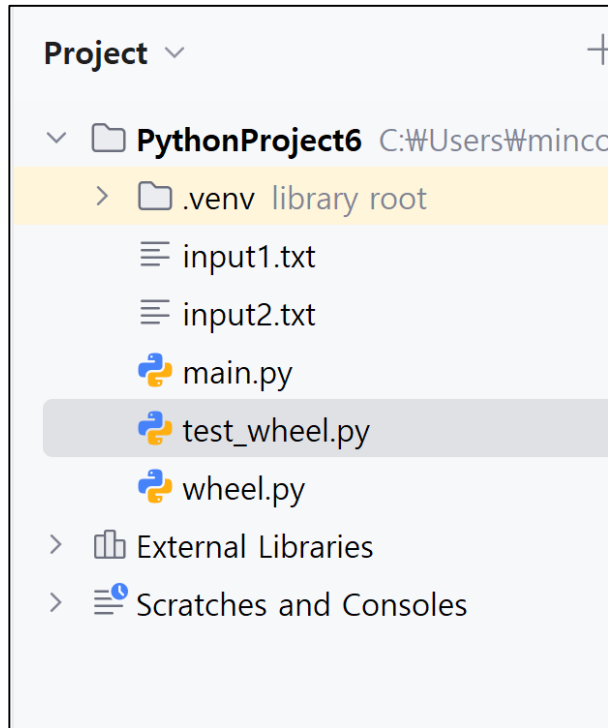
main.py

\$6500

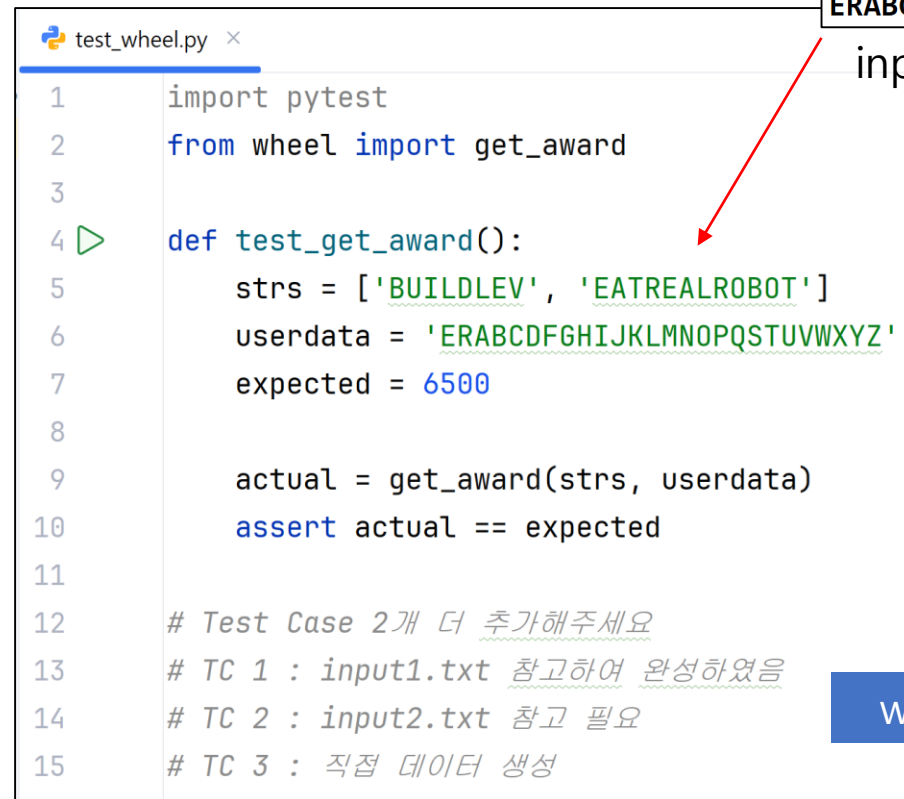
입력 값을 Unit Test로 변경하기 5

테스트 코드 작성

wheel_test.py 파일을 추가한다. 이후 Test Case 총 3개를 추가한다.
Test Case 1개는 아래와 같으며, 2개는 스스로 작성한다.



Test 파일 추가하기



2
BUILDLEV
EATREALROBOT
ERABCDGHIJKLMNOPQRSTUVWXYZ

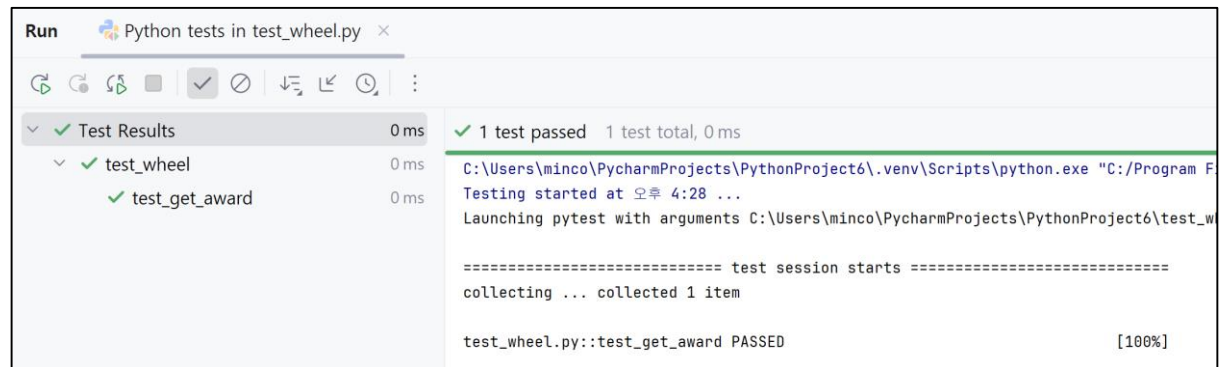
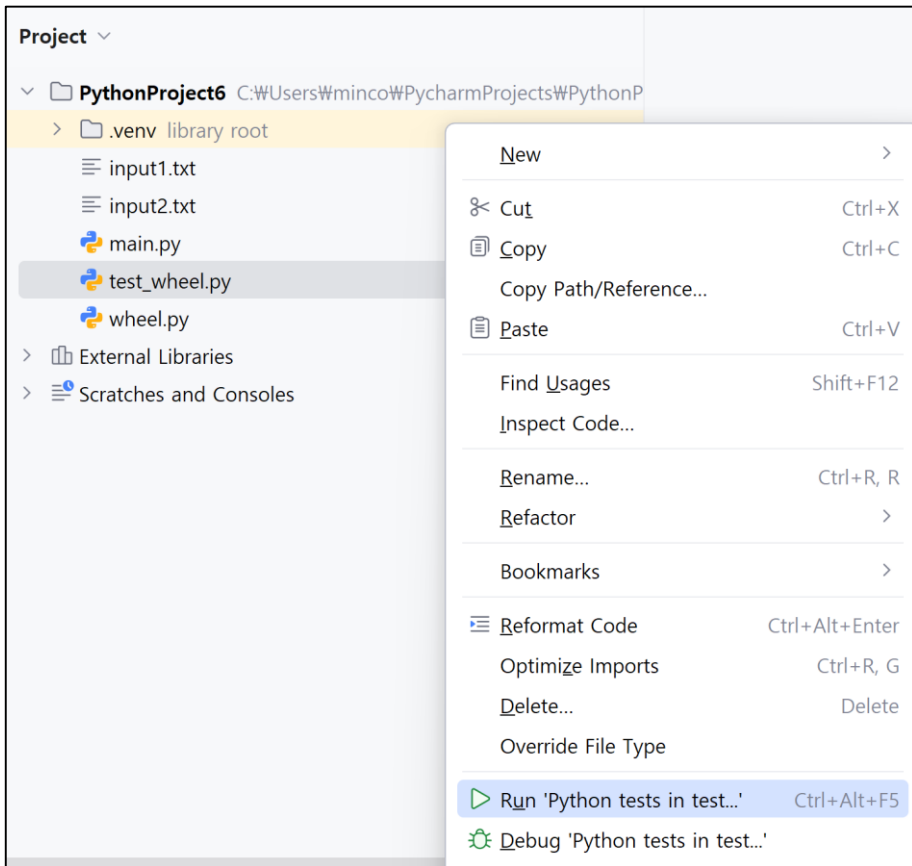
input1.txt 의 값 복사 붙여넣기

wheel_test.py

Test Case 예시

[참고] Unit Test 수행

test_wheel.py 파일 마우스 우측버튼 > Run Test



리팩토링 시작

1. 리팩토링 준비물 : Unit Test 준비하기
2. 복잡한 분기문 제거
3. 완전히 지우고 새로 코드를 작성하면 안된다.
코드는 늘 동작 코드이어야 함
4. Magic Number 제거
5. 주석이 필요 없을 정도로 변수, 함수 이름을 잘 짓기
6. **else 는 없어야 한다.**
7. **Indent Level을 최대한 줄이자.**

예시 : main 함수 내, for 내부에, if 내부에 함수호출로 끝



감사합니다.