



Python Study Week2 Solve

문제

? 입력값이 **팰린드롬**인지를 판별하는 프로그램을 만들어보세요.

※ **팰린드롬**: 거꾸로 읽어도 제대로 읽는 것과 같은 문장이나 낱말, 숫자, 문자열 등

조건

1. 사용자로부터 입력값을 받습니다. (단, 공백은 포함하지 않습니다.)
2. 입력값이 팰린드롬인지 확인합니다. (단, 대소문자는 구분하지 않습니다.)
3. 결과 값을 출력합니다.
 - a. 맞을 시, `입력값 is a Palindrome!`
 - b. 틀릴 시, `입력값 is not a Palindrome..`

학습 목표

1. 입출력문
2. 문자열 거꾸로 뒤집기
3. if문 사용



문자열을 거꾸로 뒤집는 방법은 다양합니다! 여러 풀이를 생각해보세요 😊

해설

```
word = input()          # 입력값 받기
re_word = word[::-1]    # 문자열 뒤집기

if word.lower() == re_word.lower(): # 맞을 시
    print(word + ' is a Palindrome!')
else:                      # 틀릴 시
    print(word + ' is not a Palindrome..')
```

1. `[시작:끝:규칙]` 이기 때문에, `[::-1]` 은 역순이 됩니다.
2. 대소문자를 구분하지 않기 위해, `lower()` 메소드를 사용해줍니다.
3. 맞지 않으면 틀린 것이기에, `else` 로 간단하게 코드를 작성합니다.



`[::-1]` 은 유용하게 쓰이니 잘 알아두시면 좋아요!

다른 방법

`for` 문 사용

```

word = input()          # 입력값 받기
re_word = ''           # 뒤집을 문자열 생성

for x in word:
    re_word = x + re_word # 앞에다가 하나씩 문자를 추가

if word.lower() == re_word.lower(): # 맞을 시
    print(word + ' is a Palindrome!')
else:                                # 틀릴 시
    print(word + ' is not a Palindrome..')

```

1. `for` 문을 사용하여 `word` 의 문자열을 각각 `re_word` 에 앞에서부터 넣어줍니다.
2. 나머지는 동일합니다.



앞으로 `for` 문은 자주 쓰이니 잘 알아두시면 좋아요!
이 문제에서는 위의 방법이 더 좋아보이네요 😞

`list()` , `reverse()` , `join()` 메소드 사용

```

word = input()          # 입력값 받기
word_lst = list(word)   # 문자열을 리스트로 변환
word_lst.reverse()      # 리스트 뒤집기
re_word = ''.join(word_lst) # join 메소드를 사용하여 다시 문자열로 변환

if word.lower() == re_word.lower(): # 맞을 시
    print(word + ' is a Palindrome!')
else:                                # 틀릴 시
    print(word + ' is not a Palindrome..')

```

1. `list()` 를 사용하여 문자열을 리스트로 변환해 `word_lst` 에 넣어줍니다.
2. 리스트 내장 메소드인 `reverse()` 를 사용하여 `word_lst` 를 역순으로 만들어줍니다.
3. `join()` 메소드를 사용하여 리스트 안에 각각 들어있는 문자들을 하나로 합쳐 문자열로 만들어줍니다.
 - a. `join()` 앞을 공백 `' '` 으로 두었기 때문에 그대로 합쳐집니다.
 - b. 만약 `'#'.join(word_lst)` 으로 코드를 작성하면, 각 문자 사이가 `#`으로 채워집니다.
4. 나머지는 동일합니다.



이 문제에 한해서는 불필요한 부분이 많은 코드인 듯하지만!
리스트와 관련된 메소드는 알면 알수록 코드를 간결하게 작성하는데 도움이 됩니다 😊