📖 **microsoft** / **aed-wonder-woman**

generated from microsoft/aed-learn-template

<> **Code**   ⊙ Issues   ⇄ Pull requests   ▷ Actions   ⊞ Projects   📖 Wiki   ⊘ Security

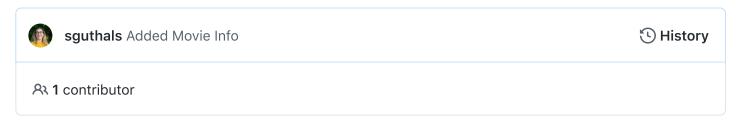⑂ main ▾   ···

aed-wonder-woman / 0-wonderwomanlp / 3-superquiz / includes / **4-quiz-making-part-2.md**

sguthals Added Movie Info   ⟳ **History**

🗠 **1 contributor**

Raw   Blame   🖥 ✎ 🗑

80 lines (56 sloc)   3.5 KB

The Making of a Wonder Woman Quiz Part 2

So far your code asks one question to the user, stores their response, and if their response was 'A' it prints out a message. In this unit you will explore how to handle the user choosing 'B' and create the rest of the questions for your quiz.

To account for the user choosing 'B' insetad of 'A' you can choose from a few ways to do this; multiple if-statements, an else-statement, or an elif-statement.

Multiple If-Statements

The first way you can handle the user putting 'B' is to have a second if-statement after the one that checks for 'A':

```
# ask the candidate a question
weapon = input( "Which weapon?\n(A) Lasso\n(B) No weapon, thank you\n" )

# print out which weapon they chose
print( f"You chose {weapon}.")

if weapon == "A":
```

```python
    print( "Nice choice!" )
if weapon == "B":
    print( "You're brave!" )
```

What is happening here is the code will store the response to the input asking which weapon the user wants in the variable called `weapon`. Then, the following will happen:

- The code will check if `weapon` is "A"
- If it is, it will print "Nice choice!"
- The code will check if `weapon` is "B"
- If it is, it will print "You're brave!"

That sounds right, and does work, but it also does some unnecessary work. For example, if the user did choose "A", then the code really doesn't have to check if they chose "B" because we already know that they chose "A"!

That's where the other two methods come in.

Else-Statement

You've already seen an else-statement in the Python Basics unit in this module, it was just written in English and not in code. You can write else-statements in code too:

```python
# ask the candidate a question
weapon = input( "Which weapon?\n(A) Lasso\n(B) No weapon, thank you\n" )

# print out which weapon they chose
print( f"You chose {weapon}.")

if weapon == "A":
    print( "Nice choice!" )
else:
    print( "You're brave!" )
```

This will work the same as the two if-statements above, if the user types either "A" or "B". The only difference if that *now* if the user chooses "A" then the code won't even check if they chose "B". So code doesn't get run unnecissarily. This is an *optimization*.

The only issue here is that if the user actually types "C", the program would still print "You're brave!" because the code doesn't check what value `weapon` has, it just prints "You're brave!" if `weapon` is anything but "A".

Elif-Statement

The last way to handle this situation is to use an elif-statement. An elif-statement, also called "else-if-statement" is one where you get the benefit of stopping the code from running if a choice was already identified, but also ensures that the user did in fact type either "A" or "B":

```python
# ask the candidate a question
weapon = input( "Which weapon?\n(A) Lasso\n(B) No weapon, thank you\n" )

# print out which weapon they chose
print( f"You chose {weapon}.")

if weapon == "A":
    print( "Nice choice!" )
elif weapon =="B":
    print( "You're brave!" )
else:
    print("You must type A or B, let's just say you wanted the Lasso")
```

A few things to note with elif-statements:

- You must have an if at the beginning and an else at the end
- You can have as many elif's in the middle
- If you wanted to give the user a chance to choose a correct answer, it is a much more complicated solution which is beyond the scope of this module

Now you're ready to ask all of the questions!

WONDER WOMAN 1984 TM & © DC and WBEI. RATED PG-13