Python, Day 4: Conditional Statements

Andrew Bydlon

January 11, 2019

If Statements

One of the cornerstones of Python is performing a given action if a particular statement is satisfied.

Syntax

```
if some_boolean_statement:
    some_command_1  # run if some_boolean_statement==True
    some_command_2  # run if some_boolean_statement==True

some_command_3  # Always run.
```

Note the importance of the tabs and line breaks in distinguish conditional code.

An Example: Spell checking

Example

```
Word = input("Please type the word 'Apostrophe'")

if Word == 'Apostrophe':
    print("Great job")

if Word != 'Apostrophe':
    print("That's not quite right")
```

Let's try running this.

Using else statements

if statements are often paired with else statements if you want to trigger a command in all circumstances but one.
The syntax is as follows:

Syntax

```
if some_boolean_statement:
    some_command_1  # run if some_boolean_statement==True
    some_command_2  # run if some_boolean_statement==True
else:
    some_command_3  # run if some_boolean_statement==False
    some_command_4  # run if some_boolean_statement==False

some_command_3  # Always run.
```

Alternative to spell checking

Example (More standard spell checker)

```
Word = input("Please type the word 'Apostrophe'")

if Word == 'Apostrophe':
    print("Great job")
else:
    print("That's not quite right")
```

Spherical Math

Example (Spheres)

```
Radius = float(input("Please enter the radius of the Sphere."))

if Radius >= 0:
    import math
    print("The surface area of the sphere is", 4*math.pi*Radius**2, ".")
    print("The volume of the sphere is", 4/3*math.pi*Radius**3, ".")
else:
    print("The radius of a sphere needs to be a non-negative \
    real number.")
```

Quick password checker

Example (Password)

```
password = float(input("Please enter the secret code word."))
if password == "Math18Python":
    print("Welcome to the exclusive club!")
else:
    print("You are not welcome here! ;)")
```

Multiple levels: if else

In other languages, such as C++, else if is used to to divide a problem into more than 2 cases.
In Python, there is an analogue. However, else and if can combine to serve the same purpose.

Syntax (if and else statements)

```
if some_boolean1:
    some_command_1
    some_command_2
else:
    if some_boolean2:
        some_command_3
    else:
        some_command_4
```

Example: Grading

Example

```
score = int(input("Enter your percentage grade: "))

if score >= 90:
    print("Excellent! Your grade is an A")
else:
    if score >= 80:
        print("Great! Your grade is a B")
    else:
        if score >= 70:
            print("Good! Your grade is a C")
        else:
            if score >= 60:
                print("Your grade is a D. You should work harder.")
            else:
                print("You failed.")
```

elif

This number of tabs and lines is somewhat impractical.

The elif command, shortened from else-if, allows one to subsequently check conditions until the desired one is met.

Syntax

```
if some_bool1:
    some_command_1 # run if bool1==True
    some_command_2 # run if bool1==True
elif some_bool2:
    some_command_3 # run if bool1==False and bool2==True
else:
    some_command_4 # run if bool1==False and bool2==False
```

Rewrite of grader

Example (Grader)

```
score = int(input("Enter your percentage grade: "))

if score >= 90:
    print("Excellent! Your grade is an A")
elif score >= 80:
    print("Great! Your grade is a B")
elif score >= 70:
    print("Good! Your grade is a C")
elif score >= 60:
    print("Your grade is a D. You should work harder.")
else:
    print("You failed.")
```

Basic error checking example

You can use if-else statements to do basic data type checking.

Example

```
FavNum = input("Type your favorite number: ")

if FavNum.isnumeric():
    FavNum = int(FavNum)
    print(FavNum, "is a great integer.")
elif FavNum.count(".")==1 and FavNum.replace(".","").isnumeric():
    FavNum = float(FavNum)
    print(FavNum, "is a great rational number.")
else:
    print("That is not an number!")
```

We will use a more robust method to accomplish this goal later on in the course.

Assignment 6

Write a program to accept a password string from the user. Do it twice and verify the following:

1) The 2 inputs are identical!
2) The password is at least 8 characters.
3) The password begins with a letter.
4) The password contains either @ or # (This can be done with .find(). Check what happens if the character is not in the string!)

If the user failed a given step, let them know what they did wrong.

Submit the .py file when you finish.