Booleans,String Operations and Formatting Output

February 19, 2020

Administrative notes

Boolean Values

I said that the only values for Boolean expressions are True and False

That’s only partially true

- Any non-zero value will evaluate to “True” and any expression that evaluates to zero is “False.”

```
if 14:
    Print (“It’s True”)
```

What that means for Boolean expressions

Suppose I'm playing craps in a casino, and I win if I roll two dice and get total of 7 or a total of 11. I lose if I get a roll of 2, 3, or 12. If I get anything else, I roll again.

In code:

```
if roll == 7 or 11:
    print ("you win!")
elif roll == 2 or 3 or 12:
    print ("you lose")
else:
    print ("roll again")
```

Let's what through this code and understand why it doesn't work like we want it to.

```
if roll == 7 or roll == 11:
    print ("you win!")
elif roll == 2 or roll == 3 or roll == 12:
    print ("you lose")
else:
    print ("roll again")
```

Why is this the code we want?

String operations

Split - splits a string up into a list of its component parts - e.g., a sentence into words

“Hey diddle diddle the cat and the fiddle” becomes [“Hey”, “diddle”, “diddle”, “the”, “cat”, “and”, “the”, “fiddle”]

Strip - removes leading and/or trailing white space

“   Kansas City 31 San Francisco 20   “ becomes “Kansas City 31 San Francisco 20”

Splitting Strings

A method to break up a string into its component parts

- A method operates on the object to which it is linked

Can split on any character or position Python can recognize, but the default is to split on whitespace

“Whitespace” = spaces, tabs, newlines, formfeeds

Examples of splitting strings

```
str = “Hey diddle diddle the cat and the fiddle”
str_list = str.split() #this will split on whitespace, and the whitespace will be deleted
str_list = str.split(“e”) #what does this do? What happened to the “e” ?
int_str = “3 1 4 5 6 7 8 9 10”
int_list = int_str.split()
actual_int_list = []
for num in int_list:
        actual_int_list.append(int(num))
```

Strip

Removing whitespace from a string - get rid of leading and/or trailing tabs, spaces,...

str.lstrip() - gets rid of whitespace on the left side of the string; before any other characters

str.rstrip() - gets rid of whitespace on the right side of the string; after the last other character

str.strip() - gets rid of whitespace on both the left and the right sides

Note that this does not get rid of blank spaces between words in a string

Formatting output
Note: this will NOT be on Exam 1:

Formatting printed output

Default field lengths when printing:

String: Python takes exactly as many spaces as there are characters in the string

Int: Python takes exactly as many spaces as digits in the integer

Float: Python prints everything to the left of the decimal point, and up to 16 digits after the decimal point

Boolean: Python takes four spaces for True and five spaces for False

What if you want to change from the defaults?

See https://www.geeksforgeeks.org/python-format-function/

There are multiple ways to do it. The best is to use the “new method”, the str.format() method.

Treat whatever your printing as a long string

print(“ { } is the answer to the question”.format(42))

print(“ {:5d} is the answer to the question”.format(42))

Specifying the type of output

s – strings
d – decimal integers (base-10)
f – floating point display
c – character
b – binary
o – octal
x – hexadecimal with lowercase letters after 9
X – hexadecimal with uppercase letters after 9
e – exponent notation

Inserting line breaks

The Python “newline” character is \n. When Python encounters “\n” it prints a new line.

```
print(“This will result in one line”, “ being printed”)
print(“This will result in two lines”, “\n”,  “ being printed”)
```

By default, Python prints a new line at the end of every print statement

```
print(“This will result in two lines”)
print(“being printed”)
```

If you don’t want a new line, you can suppress it by using an “end” value

```
print(“This will result in one line”, end=” “)
print(“being printed”)
```

How do you print out a newline character?

Escaping - using \

print(“\\n”)

The tab character is \t. How do you print that out? print(“\\t”)

To print a single quote, print(‘\’’). A double quote is print(‘\”’)