Loops and Formatting Output

September 28, 2020

More on for loops

I clearly need to go through loops - especially nested loops - to illustrate what happens

We’re going to walk through a couple of simple loops to how variables change

```
states = #list of states
capitals = #list of state capitals
for state in states:
        print (“the next state is “, )
        input(“hit Enter to continue”)
for i in range(len(states)):
        print(“the capital of “, states[i], “ is “, capitals[i]
        input(“hit Enter to continue”)
```

```
for i in range(10):
        for j in range(5):
                print( i, “ * “, j, “ is “, i*j)
                Input (“hit Enter to continue”)
```

Formatting Output

Python has a whole *bunch* of different ways to print out results

- There’s always an “improvement” - I’ve taught different things every semester I’ve taught this course

We don’t make it a huge part of this course because it’s not fundamental “computer science” but it is a nice skill to have

This won’t be on the exam!!!

The basic “Print” statement

Variables and literals

Separate values with commas

Print statement ends with a newline (\n) unless you explicitly tell it otherwise

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(‘\”’)

String.format

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

f-strings

Introduced in Python version 3.6

- The “hot new toy” of Python programming
- Similar to str.format, but you can put any Python expression into your statement
- Start with ‘f’ or ‘F’
  - Then include anything you want in a string
  - Variable names get put in curly braces. They will be evaluated at runtime (when the statement is executed) and the actual value will be printed

Multiline f-strings are permitted

```
>>> profession = "comedian"
>>> affiliation = "Monty Python"
>>> message = (
...     f"Hi {name}. "
...     f"You are a {profession}. "
...     f"You were in {affiliation}."
... )
>>> message
 'Hi Eric. You are a comedian. You were in Monty Python.'
```

Setting field sizes

This works the same way for f-strings as str.format() works.