Intro. to Programming: Outline for Week 4

Liam J. Cattell

June 9, 2017

1 For Loops

In the previous class we wrote a program to draw an n-gon with turtle graphics. The main technique we used was a *for loop* . In Python, for loops are a little different than other languages because they use a list or range object. The syntax of the for loop is as follows:

for i in range(0,10):
    first line of code
    ⋮
    last line of code

Notice that the *block* of code after the : symbol is indented and every line should be indented the same amount. This tells the Python interpreter to loop all the lines that are indented. Python uses indentation instead of begin block and end block symbols, which can save alot of programming time. The above example will loop the block 10 times.

The *range(0,10)* statement produces a range object. In earlier versions of Python this used to give a list object. Try typing $list(range(0, 10))$ into the Python shell and you will get a list, $[0, 1, \ldots, 9]$. It does not go up to 10 because the range statement is equivalent to $0 \leq i < 10$.

2 Lists

Python uses *lists* extensively. It is an important part of the language. To create an empty list just use an assignment: $x = [\,]$. If you want a list of 10 zeros you could write $x = [0] * 10$, which is an unusual statement for a programming language. This statement is sometimes called "Pythonic" because it does things in the easiest way for Python.

You can add to a list in different ways. You could use the *append* method of a list object. For example,

$x = [10]$
$x.append(3)$
$print(x)$

prints the list $[10, 3]$.

You can add two lists x and y, this operation is called concatenation, just by writing $x + y$. For example,

$x = [1, 2, 3]$
$x = [6, 7, 8]$
$z = x + y$
$print(z)$

prints the list $[1, 2, 3, 6, 7, 8]$.

If you wanted to change the $3rd$ item of the list z you could write $z[2] =$ "$green$" and now $z = [1, 2, "green", 6, 7, 8]$. Python doesn't care what data type you put into a list, the data types can be mixed. To delete an item from a list you need to know it's index. For example,

$i = z.index(7)$
$del\ \ z[i]$

will delete 7 from the list. Lists in Python are called *mutable* data types because they can be changed.

3 Strings

Most of the operations on strings are like the ones on lists, but strings are *immutable* data types in Python, and characters in strings cannot be changed like lists. This has to do with how the object is stored in memory. If you wanted to change a string you would need to make a copy of it. For example,

$s =$ "$green$"

$s2 = "G" + s\,[1:]$
$print(s2)$

would print "*Green*". The statement $s[1:]$ is called a *slice*. You can take slices of lists or strings. If you want to build a string out of other strings then you would use slices. For example, if you wanted the substring "*big*" of the string "*ambiguous*", then you could use a slice

$myString = "ambiguous"$
$newString = myString[2:5]$
$print(newString)$

prints the string "*big*".

For example, if we don't specify the second part of the slice then is goes all the way to the end of the string,

$myString = "indecisive"$
$newString = myString[2:]$
$print(newString)$

prints the string "*decisive*".

4 More turtle graphics

Use turtle graphics to place an X or an O at a position on the screen given by a list of two coordinates $[x, y]$.