Back to Week 2 Lessons Prev Next

Loops are a powerful programming concept. It is very common to want to repeat the same action(s) for every element of a list. This is exactly what a loop does. Python allows you to use a loop to *iterate* over lists. In this way, you do not need to know the length of the list ahead of time, rather you just tell Python to repeat some action as many times as there are items in the list. Iteration is not tied to lists in any way, however. Python allows iteration over a wide variety of types that contain other elements.

Anatomy of a For Loop in Python

Let's consider a simple loop that prints every item of a list:

```
1   pets = ["cat", "dog", "ferret"]
2
3   for animal in pets:
4       print(animal)
```

Let's break down the elements of the for loop:

1. **for**: All for loops begin with the keyword `for`.
2. **loop variable**: After the keyword for, you must provide a loop variable name. This variable will hold each of the values in the list as the for loop is executed. This is a variable like any other in Python.
3. **in**: The keyword `in` must appear directly after the loop variable and indicates that the following is the object that will be iterated over. This makes the for loop read somewhat like an English sentence.
4. **iterable object**: After the keyword in, you must provide an iterable object. This is any type of Python object that allows iteration. Python lists are iterable. So are range objects and strings. There are also many other iterable types that we have not seen yet.
5. **colon**: After the iterable object, there *must* be a colon. Python requires this colon to indicate that what follows will be the body of the loop.
6. **indentation**: Note that the remainder of the loop *must* be indented. In some languages, indentation is optional and is used simply to help the person reading the code to understand what is and what is not a part of the loop. In Python, this indentation is **required**. By convention, the body of the loop should all be indented exactly 4 spaces.
7. **body**: The line immediately following the colon starts the body of the loop. This is the actual code that will be executed for each value in the iterable object. During the execution of the loop body, the loop variable (in this case `animal`) will contain one of the values within the iterable object. For lists, the loop variable will take on the values of the list in order. So, in the example code, the loop body will execute three times. The first time, `animal` will have the value `"cat"`, the second time it will have the value `"dog"`, and the last time, it will have the value `"ferret"`. The body will be executed once, and only once, for each item in the list. The body of this loop is a single line, line 4, but the loop body can have one or more lines. A loop body can contain any valid Python code. Note that you can call functions or even have loops within the loop body.

All for loops in Python require these seven elements.

Another Example

Consider this more complicated loop:

```
1   numbers = [3, 8, -2, 4, 13]
2   sumsq = 0
3
4   for num in numbers:
5       square = num * num
6       sumsq += square
7
8   print(sumsq)
```

This loop will compute the sum of the squares of the values in a list. Consider the elements of this program:

1. Initialization: On lines 1 and 2, the variables `numbers` and `sumsq` are assigned their initial values.
2. For loop: Lines 4 through 6 are the for loop. Line 4 begins the loop, indicating that the `numbers` list will be iterated over and the loop variable `num` will take on each value from `numbers` in turn during the repeated execution of the loop body.
3. Loop body: LInes 5 and 6 are the body of the loop. These lines will be executed 5 times, once for each item in `numbers`. Each time the body executes, the loop variable `num` will have a different value from `numbers`. This allows the program to square and accumulate the values in the `numbers` list.
4. Print: On line 8 the sum of squares of the values in the list `numbers` is printed. This should print 262. (You should convince yourself you understand why that is correct.)

Mark as completed