For loops in Python

February 12, 2020

Administrative Notes

“For” loops

Allow us to do some things with all the elements of a list

Two types of for loops:

- for each - does something with each element that’s in a list
    - Once for each element that’s in the list
- for i - does something that may differ
    - Can skip values and elements
    - This is is the general case; your action doesn’t have to differ.
    - for each is just a special case of this

For each loops - doing the same thing with each element in a list

So, now we have this list:

grocery_list = [“Milk”, “Eggs”, “Cereal”, “Coffee”, “Apples”, “Strawberries”, “Broccoli”, “Cucumber”, “Tomatoes”, “Green Onions”]

We want to print the list, one item on a line, so that we can send someone else to the store.

Use a “for” each loop.

Remember that reserved word “in”? We’ll use it here.

Example

```
For item in grocery_list:

    print(item)
```

Indent!!! Just like with if-else. White space is important in Python

“Item in Grocery_list” is a boolean conditional. The colon ends the conditional. This continues to execute as long as there are more items.

“For i” loops

Used when you may not want to iterate over every item in the object

Syntax:

```
for iterator in range(a, b, c): # see the next slide
    #do something
```

“Iterator” is just a variable that tracks where you are in the list or other object. It is most often an integer, although it doesn’t absolutely have to be. “I” is often used.

The value of “iterator” does NOT have to be pre-set.

Home on the range()

range() is a function that takes 3 parameters:

1. A starting integer
2. An ending integer
3. A hop size

Range will give you back all the integers from (1) to (2) hopping by (3) each time. Also, we have to wrap range in list() to see all the numbers at once.

Let’s try it!

Note: doesn’t have to be an integer, but it’s really confusing if it’s not.

Not all the arguments are needed

If you give one argument it’s the end of the range.

- What are the assumed start and hop size?

If you give two it’s the start and end of the range.

- What is the assumed hop size?

For loops

Here's how a "for i" loop iterates through the items of a list

```
lizards = [“gecko”, “iguana”, “komodo dragon”, “chameleon”]
for i in range(len(lizards)):
   print(lizards[i])
```

Modifying your list

You can modify the contents of a list you’re iterating through, while you’re iterating through it.

An example:

```
grade_list = [98, 92.5, 123, 199.8]
for i in range(len(grade_list)):
        grade_list[i] *= grade_list[i]
print(grade_list)
```

Let's try it!

Let's try to take a list of integers and increase each integer by one.

First we'll try it with a "for each" loop. What happens?

Next we'll try it with a "for i" loop. Any better?

Some questions:

1. How can we print the even numbers between two integers x and y with a for loop?
2. How can we use range to go backwards through a list?
3. Print all the elements of a list with their index!

For loops and strings

A string is zero or more characters treated as a single entity

Sounds kind of like a list, where each element of the list is one character long.

- You can treat it that way

```
word = “supercalifragilisticexpialidocious”:
for i in word:
        print(i)
```

Or:

```
word = “supercalifragilisticexpialidocious”:
for i in range(len(word)):
        print(word[i])
```