PRACTICE PYTHON

Beginner Python exercises

15 FEBRUARY 2014

List Less Than Ten

list numbers elements if conditional

Exercise 3 (and Solution)

Take a list, say for example this one:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

and write a program that prints out all the elements of the list that are less than 5.

Extras:

1. Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.
2. Write this in one line of Python.
3. Ask the user for a number and return a list that contains only elements from the original list `a` that are smaller than that number given by the user.

Discussion

This week's topics:

1. Lists
2. More conditionals (if statements)

Lists

This week's exercise hits on a topic critical for all types and styles of programming: **lists**. Lists are basically an ordered way of grouping things (called **elements**) - the cool thing about lists in Python is that you can have a list that contains objects of multiple types. Your list can mix between strings, integers, objects, other lists, what have you.

The way to construct an empty list is just to do

```
x = []
```

And your variable `x` now holds an empty list. To add things to this list, just "append" them to the list. Like so:

```
x = []
x.append(3)
```

Your list `x` now looks like `[3]`.

In Python, lists are also **iterables**, which means you can loop through them with a **for loop** in a convenient way. (If you come from other languages like C++ or Java you are most likely used to using a counter to loop through indices of a list - in Python you can actually loop through the elements.) I will let the code speak for itself:

```
my_list = [1, 3, "Michele", [5, 6, 7]]
for element in my_list:
  print(element)
```

Will yield the result:

```
1
3
"Michele"
[5, 6, 7]
```

There are many other properties of lists, but for the basic exercise all you should need is this for loop property. Future weeks will address other properties of lists.

For more information about lists in Python, check out these resources:

- The official Python documentation on lists
- Tutorialspoint on Python lists
- Someone else's blog post about lists

More Conditionals

The nice thing about conditionals is that they follow logical operations. They can also be used to test equality. Let's do a small example. Let's say I want to make a piece of code that converts from a numerical grade (1-100) to a letter grade (A, B, C, D, F). The code would look like this:

```
grade = input("Enter your grade: ")
if grade >= 90:
  print("A")
elif grade >= 80:
  print("B")
elif grade >= 70:
  print("C")
elif grade >= 65:
  print("D")
else:
  print("F")
```

What happens if `grade` is 50? All the conditions are false, so `"F"` gets printed on the screen. But what if `grade` is 95? Then all the conditions are true and everything gets printed, right? Nope! What happens is the program goes line by line. The first condition (grade >= 90) is satisfied, so the program enters into the code inside the `if` statement, executing `print("A")`. Once code inside a conditional has been executed, the rest of the conditions are skipped and none of the other conditionals are checked.

Happy coding!

Forgot how to submit exercises?

Share the fun!

« Previous exercise Next exercise »