

Basic Practice with Python

Sarah Middleton

Last Updated: September 1, 2015

1 Guess the output: print statement practice

For each of the following statements, first guess what the output will be, and then run the code yourself. You can use the interpreter if you want.

Code	Predicted output	Actual output
<code>print "what's", "up"</code>		
<code>print "what's" + "up"</code>		
<code>print "I have", 5, "cats"</code>		
<code>print "I have" + 5 + "cats"</code>		
<code>print 9 - 6 * 2</code>		
<code>print (9 - 6) * 2</code>		
<code>print 24 % 6</code>		
<code>print 24 % 7</code>		
<code>print -3 ** 2</code>		
<code>print (-3) ** 2</code>		
<code>print 9 / 2</code>		
<code>print 9.0 / 2</code>		
<code>print 9 / float(2)</code>		

2 Guess the output: variables practice

For each of the following statements, first guess what the output will be, and then run the code yourself. You can use the interpreter if you want.

Code	Predicted output	Actual output
<pre>x = 5 print x * 3</pre>		
<pre>x = "5" print x * 3</pre>		
<pre>x = "5" print int(x) * 3</pre>		
<pre>x = "cat" y = x print y</pre>		
<pre>x = 5 x = 1 print x</pre>		
<pre>x = 5 x + 1 print x</pre>		
<pre>x = 5 x = x + 1 print x</pre>		
<pre>x = 2 y = 4 print (x * y) ** x</pre>		
<pre>x = "cat" y = "dog" print x + y</pre>		

3 Math practice

Write a script to calculate the quadratic formula. Your program should output both possible values for x.

The program should start by defining the values of a, b, and c. For example, the first three lines of your script might look like this:

```
a = -2
b = 2
c = 1
```

If you use these values, the answers should be -0.366 and 1.366 (check and make sure you get this). Try a few different values of these variables. Note, you will get an error message if $b^2 - 4ac$ is negative, since you can't take the square root of a negative number.

Quadratic formula, for reference:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Hint: There are a few different ways to find the square root in Python. Try googling it.

4 Guess the output: conditionals practice

For each of the following statements, first guess what the output will be, and then run the code yourself. You can use the interpreter if you want. If you get a result you didn't expect, make sure you understand why.

The following variables will be used (**add these to your program before running**):

```
a = True
b = False
x = 2
y = -2
cat = "Mittens"
```

Code	Predicted output	Actual output
<code>print a</code>		
<code>print (not a)</code>		
<code>print (a == b)</code>		
<code>print (a != b)</code>		

<code>print (x == y)</code>		
<code>print (x > y)</code>		
<code>print (x = 2)</code>		
<code>print (a and b)</code>		
<code>print (a and not b)</code>		
<code>print (a or b)</code>		
<code>print (not b or a)</code>		
<code>print not (b or a)</code>		
<code>print (not b) or a</code>		
<code>print (not b and a)</code>		
<code>print not (b and a)</code>		
<code>print (not b) and a</code>		
<code>print (x == abs(y))</code>		
<code>print len(cat)</code>		
<code>print cat + x</code>		
<code>print cat + str(x)</code>		
<code>print float(x)</code>		
<code>print ("i" in cat)</code>		
<code>print ("g" in cat)</code>		
<code>print ("Mit" in cat)</code>		

5 Guess the output: if statement practice

For each of the following code snippets, first guess what the output will be, and then run the code yourself. You can use the interpreter if you want. If you get a result you didn't expect, make sure you understand why.

The following variables will be used (**add these to your program before running**):

```
a = True
b = False
x = 2
y = -2
cat = "Mittens"
```

Code	Predicted output	Actual output
<pre> if (x % 2) == 0: print "x is even" else: print "x is odd" </pre>		
<pre> if (x - 4*y) < 0: print "Invalid!" else: print "Valid" </pre>		
<pre> if cat == "Mittens": print "Awww" else: print "Get lost, cat" </pre>		
<pre> if "Mit" in cat: print "Hey Mits!" else: print "Where's Mits?" </pre>		

6 Quadratic formula: checking for negative roots

Recall that when calculating the quadratic formula, you will get an error if $b^2 - 4ac$ is negative, since you can't take the square root of a negative number. Edit your quadratic formula program from question 3 so that it checks for this potential error before it occurs. If the error is going to occur, print an informative message saying "non-real answer" or similar, and do not calculate the values of x. If the error is not going to occur, continue on to calculating and printing the values of x.

7 Guess the output: loop practice

For each of the following statements, first guess what the output will be, and then run the code yourself. You can use the interpreter if you want. If you get a result you didn't expect, make sure you understand why. **Watch out for endless loops, and remember you can terminate any terminal program with Ctrl+C.**

Code	Predicted output	Actual output
<pre>for i in range(1,10,2): print i</pre>		
<pre>for i in range(5,1,-1): print i</pre>		
<pre>count = 0 while (count < 5): print count</pre>		
<pre>count = 0 while (count < 5): print count count = count + 1</pre>		
<pre>total = 0 for i in range(4): total = total + i print total</pre>		
<pre>name = "Mits" for letter in name: print letter</pre>		
<pre>name = "Wilfred" newName = "" for letter in name: newName = newName + letter print newName</pre>		

<pre> name = "Wilfred" newName = "" for letter in name: newName = letter + newName print newName </pre>		
<pre> x = "C" if x == "A" or "B": print "yes" else: print "no" </pre>		
<pre> x = "C" if (x == "A") or (x == "B"): print "yes" else: print "no" </pre>		

Surprised by the last two? It's important to note that when you compare a variable against multiple things, you only compare it to one thing at a time. Although it makes sense in English to say, is x equal to A or B?, in Python you must write: `((x == "A") or (x == "B"))` to accomplish this. The same goes for e.g. `((x > 5) and (x < 10))` and anything along those lines.

8 Writing simple loops

Write code to accomplish each of the following tasks using a for loop or a while loop. Choose whichever type of loop you want for each problem (or both for extra practice).

(a) **(1 pt)** Print the integers between 3 and 35, inclusive.

(b) **(1 pt)** Print the positive integers less than 100 that are multiples of 7.

- (c) **(1 pt)** Starting with $x = 1$, double x until it's greater than 1000. Print each value of x as you go along.
- (d) **(1 pt)** Print each character of the string "supercalifragilisticexpialidocious" on a separate line.

9 Guess the output: list practice

You know the drill. Remember that if you do these problems in the interpreter, you must hit 'enter' a second time after an indented block (basically, if you still see the `...` prompt, the interpreter thinks you're still adding to the block. Once you see `>>>`, you can continue to write code.

The following variables will be used. **Remember to reset lists if they get changed!!!**

```
names = ["Wilfred", "Manfred", "Wadsworth", "Jeeves"]
ages = [65, 34, 96, 47]
cat = "Mitsworth"
str1 = "Good morning, Mr. Mitsworth."
```

Code	Predicted output	Actual output
<pre>print len(ages)</pre>		
<pre>print len(ages) == len(names)</pre>		
<pre>for age in ages: print age</pre>		
<pre>print ages[1:3]</pre>		

<pre> if "Willard" not in names: names.append("Willard") print names </pre>		
<pre> for i in range(len(names)): print names[i],"is",ages[i] </pre>		
<pre> ages.sort() print ages </pre>		
<pre> ages = ages.sort() print ages </pre>		
<pre> print max(ages) </pre>		
<pre> for i in range(len(cat)): print cat[i] </pre>		
<pre> print cat[:4] </pre>		
<pre> parts = str1.split() print parts print str1 </pre>		
<pre> parts = str1.split(",") print parts </pre>		
<pre> print names[-1] </pre>		
<pre> oldList = [2, 2, 6, 1, 2, 6] newList = [] for item in oldList: if item not in newList: newList.append(item) print newList </pre>		

10 Guess the function

The following code uses several new functions you may not have seen before. For each, try to guess what the function does by running the code and doing your own experiments. Then look up the function in the Python Docs (or wherever) and write down a short explanation of the actual purpose.

`.replace()`

```
str1 = "ATCGATCGATTTCGGGATCGATCGGCGTGTGAT"
str2 = str1.replace("T", "U")
print str2
print str1

str3 = str1.replace("ATC", "*")
print str3
```

Your guess:

Actual answer:

`.count()`

```
str1 = "ATCGATCGATTTCGGGATCGATCGGCGTGTGAT"
print str1.count("T")
print str1.count("GGG")
print str1.count("TCG")
print str1.count("ABC")
```

Your guess:

Actual answer:

`.find()`

```
str1 = "ATCGATCGATTTCGGGATCGATCGGCGTGTGAT"
print str1.find("T")
print str1.find("GGG")
print str1.find("GATCG")
print str1.find("ABC")
```

Your guess:

Actual answer:

`random.choice()`

```
import random

fruits = ["apple", "banana", "grape"]
print random.choice(fruits)
print random.choice(fruits)
print random.choice(fruits)

fruitColors = {"apple":"red", "banana":"yellow", "grape":"purple"}
print random.choice(fruitColors)
print random.choice(fruitColors.keys())
print random.choice(fruitColors.values())
```

Your guess:

Actual answer:

`sum()`

```
nums = [1,5,4,10]
print sum(nums)
```

Your guess:

Actual answer:

`sorted()`

```
nums1 = [1,5,5,7,9,10,1,2,5]
print sorted(nums1)
print nums1

# in contrast to:
nums1 = [1,5,5,7,9,10,1,2,5]
print nums1.sort()
print nums1
```

Your guess:

Actual answer: