course_1_assessment_6

Due: 2018-11-25 01:19:00

Description: Assessment for Way of Programmer Week 2 lesson. Score: 9.0 of 9 = 100.0%

Questions

Score: 1.0 / 1

Comment: autograded

Write one for loop to print out each character of the string `my_str` on a separate line.

Save & Run 6/21/2021, 2:55:08 PM - 8 of 8 Show in CodeLens

```python
my_str = "MICHIGAN"
for num in my_str:
    print(num)

```

```
M
I
C
H
I
G
A
N
```

ActiveCode (assess_ps_02_01)

Result	Actual Value	Expected Value	Notes

Pass	'for'	'my_st...num)\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	'M\nI\nC\nH\nI\nG\nA\nN'	'M\nI\nC...\nA\nN\n'	Testing output (Don't worry about actual and expected values).	Expand Differences

You passed: 100.0% of the tests

Score: 1.0 / 1

Comment: autograded

Write one for loop to print out each element of the list `several_things` . Then, write *another* for loop to print out the TYPE of each element of the list `several_things` . To complete this problem you should have written two different for loops, each of which iterates over the list `several_things` , but each of those 2 for loops should have a different result.

Save & Run 6/21/2021, 2:48:30 PM - 7 of 7 Show in CodeLens

```python
several_things = ["hello", 2, 4, 6.0, 7.5, 234352354, "the end", "", 99]
for i in range(len(several_things)):
    print(several_things[i])

for j in range(len(several_things)):
    print(type(several_things[j]))

```

```
hello
2
4
6.0
7.5
234352354
the end

99
<class 'str'>
<class 'int'>
<class 'int'>
<class 'float'>
<class 'float'>
<class 'int'>
<class 'str'>
<class 'str'>
<class 'int'>
```

ActiveCode (assess_ps_02_02)

Result	Actual Value	Expected Value	Notes
Pass	'for'	'sever...j]))\n'	Testing your code (Don't worry about actual and expected values).
Pass	True	True	Testing output (Don't worry about actual and expected values).

Expand Differences

You passed: 100.0% of the tests

Score: 1.0 / 1

Comment: autograded

Write code that uses iteration to print out **the length** of each element of the list stored in `str_list` .

Save & Run   6/21/2021, 2:47:43 PM - 4 of 4   Show in CodeLens

```python
str_list = ["hello", "", "goodbye", "wonderful", "I love Python"]

for i in range(len(str_list)):
    print(len(str_list[i]))

```

```
5
0
7
9
13
```

ActiveCode (assess_ps_02_03)

Result	Actual Value	Expected Value	Notes
Pass	'for'	'str_l...i]))\n'	Testing whether you used a for loop (Don't worry about actual and expected values).
Pass	'5\n0\n7\n9\n13'	'5\n0\n7\n9\n13\n'	Testing output (Don't worry about actual and expected values).

Expand Differences

You passed: 100.0% of the tests

Score: 1.0 / 1

Comment: autograded

Write code to count the number of characters in `original_str` using the accumulation pattern and assign the answer to a variable `num_chars`. Do NOT use the `len` function to solve the problem (if you use it while you are working on this problem, comment it out afterward!)

Save & Run 6/21/2021, 2:46:43 PM - 3 of 3 Show in CodeLens

```
original_str = "The quick brown rhino jumped over the extremely lazy fox."
num_chars = 0

for i in original_str:
    num_chars = num_chars + 1
    print(num_chars)

```

ActiveCode (assess_ps_02_05)

Result	Actual Value	Expected Value	Notes
Pass	57	57	Testing whether num_chars_sent has the correct value
Pass	'len'	'origi...\n '	Testing that you are not including the len function in your code. (Don't worry about Actual and Expected Values.)

Expand Differences

You passed: 100.0% of the tests

Score: 1.0 / 1

Comment: autograded

`addition_str` is a string with a list of numbers separated by the `+` sign. Write code that uses the accumulation pattern to take the sum of all of the numbers and assigns it to `sum_val` (an integer). (You should use the `.split("+")` function to split by `"+"` and `int()` to cast to an integer).

Save & Run 6/21/2021, 2:45:52 PM - 3 of 3 Show in CodeLens

```
addition_str = "2+5+10+20"
sum_val = 0

for i in addition_str:
    sum_val = sum(map(int,addition_str.split("+")))
    print(sum_val)


```

```
37
37
37
37
37
37
37
37
37
```

ActiveCode (assess_ps_02_07)

Result	Actual Value	Expected Value	Notes	
Pass	37	37	Testing whether sum_val has the correct value	
Pass	'split'	'addit...al)\n\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	'int'	'addit...al)\n\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences

You passed: 100.0% of the tests

Score: 1.0 / 1

Comment: autograded

`week_temps_f` is a string with a list of fahrenheit temperatures separated by the `,` sign. Write code that uses the accumulation pattern to compute the **average** (sum divided by number of items) and assigns it to `avg_temp`. Do not hard code your answer (i.e., make your code compute both the sum or the number of items in `week_temps_f`) (You should use the `.split(",")` function to split by `","` and `float()` to cast to a float).

Save & Run 6/21/2021, 10:01:34 AM - 2 of 2 Show in CodeLens

```
week_temps_f = "75.1,77.7,83.2,82.5,81.0,79.5,85.7"

avg_temp = 0.0

for i in week_temps_f:
    avg_temp = sum(map(float,week_temps_f.split(","))) / 7
    print(avg_temp)


```

```
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
80.6714285714
```

ActiveCode (assess_ps_02_08)

Result	Actual Value	Expected Value	Notes	
Pass	80.6714285714	80.6714285714	Testing that avg_temp has the correct value	
Pass	'split'	'week_...mp)\n\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	'float'	'week_...mp)\n\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences

You passed: 100.0% of the tests

Score: 1.0 / 1

Comment: autograded

Write code to create a list of numbers from 0 to 67 and assign that list to the variable `nums` . Do not hard code the list.

Save & Run 6/21/2021, 10:02:21 AM - 2 of 2 Show in CodeLens

```
nums = list(range(0, 68))
print(nums)

```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
```

ActiveCode (assess_ps_02_09)

Result	Actual Value	Expected Value	Notes
Pass	[0, 1..., 67]	[0, 1..., 67]	Testing that nums is a list that contains the correct elements.

Expand Differences

You passed: 100.0% of the tests

Score: 1.0 / 1

Comment: autograded

Write code to create a **list of word lengths** for the words in `original_str` using the accumulation pattern and assign the answer to a variable `num_words_list` . (You should use the `len` function).

Save & Run 6/21/2021, 10:03:22 AM - 2 of 2 Show in CodeLens

```
original_str = "The quick brown rhino jumped over the extremely lazy fox"
original_list = list(original_str.split())
num_words = len(original_list)
num_words_list = []
for i in original_list:
    num_words_list.append((len(i)))

print(num_words_list)


```

```
[3, 5, 5, 5, 6, 4, 3, 9, 4, 3]
```

ActiveCode (assess_ps_02_06)

Result	Actual Value	Expected Value	Notes	
Pass	[3, 5...4, 3]	[3, 5...4, 3]	Testing whether num_words_list has the correct value	Expand Differences
Pass	'for'	'origi...st)\n\n'	Testing that you are using a for loop in your code.	Expand Differences

You passed: 100.0% of the tests

Score: 1.0 / 1

Comment: autograded

Create an empty string and assign it to the variable `lett`. Then using range, write code such that when your code is run, `lett` has 7 b's (`"bbbbbbb"`).

Save & Run 6/21/2021, 10:04:14 AM - 2 of 2 Show in CodeLens

```
lett = ''
for i in range(7):
    lett += 'b'
    print(lett, end='')

```

```
bbbbbbbbbbbbbbbbbbbbbbbbbbbb
```

ActiveCode (assess_pc_02_10)

Result	Actual Value	Expected Value	Notes
Pass	'bbbbbbb'	'bbbbbbb'	Testing that lett has the correct value.
Pass	'bbbbbbb'	"lett ...='')\n"	Testing that you didn't hardcode the answer.

Expand Differences

You passed: 100.0% of the tests

Score: 0.0 / 0

Comment: autograded

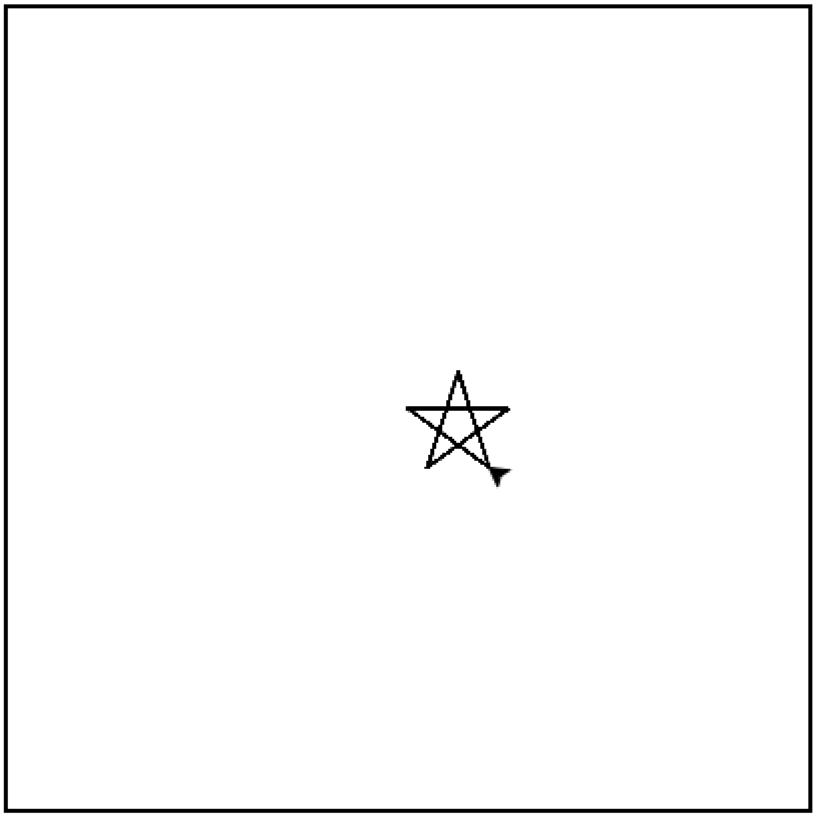
Write a program that uses the turtle module **and** a for loop to draw something. It doesn't have to be complicated, but draw something different than we have done in the past. (Hint: if you are drawing something complicated, it could get tedious to watch it draw over and over. Try setting `.speed(10)` for the turtle to draw fast, or `.speed(0)` for it to draw super fast with no animation.)

Save & Run 6/21/2021, 2:55:50 PM - 6 of 6 Show in CodeLens

```
import turtle
star = turtle.Turtle()

for i in range(50):
    star.forward(50)
    star.right(144)

turtle.done()
```



ActiveCode (assess_ps_02_04)

Error

```
TimeLimitError: Program exceeded run time limit. on line 6
```

Description

Your program is running too long. Most programs in this book should run in less than 10 seconds easily. This probably indicates your program is in an infinite loop.

To Fix

Add some print statements to figure out if your program is in an infinte loop. If it is not you can increase the run time with sys.setExecutionLimit(msecs)

Score Me

username: rishavkumarbks_ece18@its.edu.in | Back to top