course_2_assessment_6

Due: 2018-11-25 01:32:00

Description: Assessment for More about Iteration lesson

Score: 0 of 5 = 0.0%

Questions

Not yet graded

Write a function, `sublist` , that takes in a list of numbers as the parameter. In the function, use a while loop to return a sublist of the input list. The sublist should contain the same values of the original list up until it reaches the number 5 (it should not contain the number 5).

Save & Run 17/09/2022, 21:43:42 - 2 of 2 Show CodeLens

```
1 def sublist(lst):
2     n = 0
3     newlst = []
4     while n < len(lst) and lst[n] != 5:
5         newlst.append(lst[n])
6         n += 1
7     return newlst
8
```

Activity: 14.8.1 ActiveCode (ac14_10_1)

Question in Context (/runestone/books/published/fopp/MoreAboutIteration/ChapterAssessment.html#ac14_10_1)

Not yet graded

Write a function called `check_nums` that takes a list as its parameter, and contains a while loop that only stops once the element of the list is the number 7. What is returned is a list of all of the numbers up until it reaches 7.

Save & Run 17/09/2022, 22:00:55 - 2 of 2 Show in CodeLens

```
1 def check_nums(lst):
2     n = 0
3     newlst = []
4     while n < len(lst) and lst[n] != 7:
5         newlst.append(lst[n])
6         n += 1
7     return newlst
8
```

Activity: 14.8.2 ActiveCode (ac14_10_2)

Result	Actual Value	Expected Value	Notes	
Pass	[0, 2..., 14]	[0, 2..., 14]	Testing that check_nums stops on the correct postion with input [0,2,4,9,2,3,6,8,12,14,7,9,10,8,3]	Expand Differences
Pass	[9, 3..., 10]	[9, 3..., 10]	Testing that check_nums stops on the correct position with input [9,302,4,62,78,97,10,7,8,23,53,1]	Expand Differences
Pass	[]	[]	Testing that check_nums stops on the correct position with input [7,8,3,2,4,51]	
Pass	[1, 6, 2, 3, 9]	[1, 6, 2, 3, 9]	Testing that check_nums([1, 6, 2, 3, 9]) returns ([1, 6, 2, 3, 9])	

You passed: 100.0% of the tests

Question in Context (/runestone/books/published/fopp/MoreAboutIteration/ChapterAssessment.html#ac14_10_2)

Not yet graded

Write a function, `sublist` , that takes in a list of strings as the parameter. In the function, use a while loop to return a sublist of the input list. The sublist should contain the same values of the original list up until it reaches the string “STOP” (it should not contain the string “STOP”).

Save & Run 17/09/2022, 21:44:14 - 3 of 3 Show in CodeLens

```
def sublist(lst):
    n = 0
    newlst = []
    while n < len(lst) and lst[n] != 'STOP':
        newlst.append(lst[n])
        n += 1
    return newlst
```

Activity: 14.8.3 ActiveCode (ac14_10_3)

Result	Actual Value	Expected Value	Notes	
Pass	['bob...ucy']	['bob...ucy']	Testing that sublist(['bob', 'joe', 'lucy', 'STOP', 'carol', 'james']) returns ['bob', 'joe', 'lucy']	Expand Differences
Pass	[]	[]	Testing that sublist(['STOP']) returns []	
Pass	['jac...aul']	['jac...aul']	Testing that sublist(['jackie', 'paul', 'STOP']) returns ['jackie', 'paul']	Expand Differences

You passed: 100.0% of the tests

Question in Context (/runestone/books/published/fopp/MoreAboutIteration/ChapterAssessment.html#ac14_10_3)

Not yet graded

Write a function called `stop_at_z` that iterates through a list of strings. Using a while loop, append each string to a new list until the string that appears is “z”. The function should return the new list.

Save & Run 17/09/2022, 21:44:17 - 2 of 2 Show in CodeLens

```
def stop_at_z(lst):
    n = 0
    newlst = []
    while n < len(lst) and lst[n] != 'z':
```

```
        newlst.append(lst[n])
        n += 1
    return newlst
```

Activity: 14.8.4 ActiveCode (ac14_10_4)

Result	Actual Value	Expected Value	Notes
Pass	['c',... 'r']	['c',... 'r']	Testing the function stop_at_z on the input ['c', 'b', 'd', 'zebra', 'h', 'r', 'z', 'm', 'a', 'k'].
Pass	['zoo...azz']	['zoo...azz']	Testing the function stop_at_z on the input ['zoo', 'zika', 'ozzie', 'pizzazz', 'z', 'pizza', 'zap', 'haze'].
Pass	[]	[]	Testing the function stop_at_z on the input ['z'].

Expand Differences

Expand Differences

You passed: 100.0% of the tests

Question in Context (/runestone/books/published/fopp/MoreAboutIteration/ChapterAssessment.html#ac14_10_4)

Not yet graded

Below is a for loop that works. Underneath the for loop, rewrite the problem so that it does the same thing, but using a while loop instead of a for loop. Assign the accumulated total in the while loop code to the variable `sum2` . Once complete, sum2 should equal sum1.

Save & Run 17/09/2022, 21:44:24 - 2 of 2 Show in CodeLens

```
sum1 = 0
sum2 = 0
lst = [65, 78, 21, 33]
n = 0
for x in lst:
    sum1 = sum1 + x
while n < len(lst):
    sum2 += lst[n]
    n += 1
```

Activity: 14.8.5 ActiveCode (ac14_10_5)

Result	Actual Value	Expected Value	Notes
Pass	197	197	Testing that sum2 is assigned to correct value.
Pass	'while'	'sum1 ... += 1'	Testing your code (Don't worry about actual and expected values).

Expand Differences

You passed: 100.0% of the tests

Question in Context (/runestone/books/published/fopp/MoreAboutIteration/ChapterAssessment.html#ac14_10_5)

Not yet graded

Challenge: Write a function called `beginning` that takes a list as input and contains a while loop that only stops once the element of the list is the string 'bye'. What is returned is a list that contains up to the first 10 strings, regardless of where the loop stops. (i.e., if it stops on the 32nd element, the first 10 are returned. If "bye" is the 5th element, the first 4 are returned.) *If you want to make this even more of a challenge, do this without slicing*

Save & Run 17/09/2022, 21:44:32 - 2 of 2 Show in CodeLens

```
1 def beginning(lst):
2     n = 0
3     newlst = []
4     while n < 10 and lst[n] != 'bye':
5         newlst.append(lst[n])
6         n += 1
7     return newlst
8
```

Activity: 14.8.6 ActiveCode (ac14_10_6)

Result	Actual Value	Expected Value	Notes

Pass	['wat...nes']	['wat...nes']	Testing that beginning returns the correct list on input ['water', 'phone', 'home', 'chapstick', 'market', 'headphones', 'bye', 'stickie notes', 'snapchat', 'facebook', 'social media']	Expand Differences
Pass	[]	[]	Testing that beginning returns the correct list on input ['bye', 'no', 'yes', 'maybe', 'sorta']	
Pass	['hel...sup']	['hel...sup']	Testing that beginning returns the correct list on input ['hello', 'hi', 'hiyah', 'howdy', 'what up', 'whats good', 'holla', 'good afternoon', 'good morning', 'sup', 'see yah', 'toodel loo', 'night', 'until later', 'peace', 'bye', 'good-bye', 'g night']	Expand Differences

You passed: 100.0% of the tests

Question in Context (/runestone/books/published/fopp/MoreAboutIteration/ChapterAssessment.html#ac14_10_6)

Score Me