6.11. Chapter Assessment

Check your understanding

sequences-10-1: What will the output be for the following code?

```
let = "z"
let_two = "p"
c = let_two + let
m = c*5
print(m)
```

- A. zpzpzpzpzp
- B. zzzzzppppp
- [✓ C. pzpzpzpzpz]
- D. pppppzzzzz
- E. None of the above, an error will occur.

Check me Compare me

✓ Yes, because let_two was put before let, c has "pz" and then that is repeated five times.

Activity: 1 -- Multiple Choice (assess_question2_1_1_1)

Write a program that extracts the last three items in the list `sports` and assigns it to the variable `last`. Make sure to write your code so that it works no matter how many items are in the list.

Save & Run 6/9/2020, 2:00:00 PM - 2 of 3 Show in CodeLens

```
sports = ['cricket', 'football', 'volleyball', 'baseball', 'softball', 'track and f
last = sports[-3:]
print(last)
```

```
['curling', 'ping pong', 'hockey']
```

Activity: 2 -- ActiveCode (assess_ac_2_1_1_2)

Result	Actual Value	Expected Value	Notes	
Pass	['cur...key']	['cur...key']	Testing that the value of last is the last three items in sports.	Expand Differences
Pass	<__ma...ject>	True	Hardcode check	Expand Differences

You passed: 100.0% of the tests

6.10. Exercises">
7. Iteration">

Write code that combines the following variables so that the sentence "You are doing a great job, keep it up!" is assigned to the variable `message`. Do not edit the values assigned to `by`, `az`, `io`, or `qy`.

Save & Run 6/9/2020, 2:34:41 PM - 11 of 11 Show in CodeLens

```
by = "You are"
az = "doing a great "
io = "job"
qy = "keep it up!"
message = by+" "+az+io+", "+qy
print(message)
```

```
You are doing a great job, keep it up!
```

Activity: 3 -- ActiveCode (assess_ac_2_1_1_3)

Result	Actual Value	Expected Value	Notes	
Pass	'You are'	'You are'	Testing original variables.	
Pass	'doing a great '	'doing a great '	Testing original variables.	
Pass	'job'	'job'	Testing original variables.	
Pass	'keep it up!'	'keep it up!'	Testing original variables.	
Pass	'You a...t up!'	'You a...t up!'	Testing that the value of message is what was expected.	Expand Differences
Pass	'You a...t up!'	'by = ...e)\n\n\n'	Testing for hardcoding (Don't worry about actual and expected values).	Expand Differences

You passed: 100.0% of the tests

sequences-10-2: What will the output be for the following code?

```
ls = ['run', 'world', 'travel', 'lights', 'moon', 'baseball', 'sea']
new = ls[2:4]
print(new)
```

A. ['travel', 'lights', 'moon']
B. ['world', 'travel', 'lights']
[✓ C. ['travel', 'lights']]
D. ['world', 'travel']

Check me | Compare me

✓ Yes, python is a zero-index based language and slices are inclusive of the first index and exclusive of the second.

Activity: 4 -- Multiple Choice (assess_question2_1_1_4)

sequences-10-3: What is the type of `m`?

```
l = ['w', '7', 0, 9]
m = l[1:2]
```

A. string
B. integer
C. float
[✓ D. list]

Check me | Compare me

✓ Yes, a slice returns a list no matter how large the slice.

Activity: 5 -- Multiple Choice (assess_question2_1_1_5)

sequences-10-4: What is the type of `m`?

```
l = ['w', '7', 0, 9]
m = l[1]
```

[✓ A. string]
B. integer
C. float
D. list

Check me | Compare me

✓ Yes, the quotes around the number mean that this is a string.

Activity: 6 -- Multiple Choice (assess_question2_1_1_6) sequences-10-5: What is the type of `x` ?

```
b = "My, what a lovely day"
x = b.split(',')
```

A. string

B. integer

C. float

[✓ D. list]

Check me Compare me

✓ Yes, the .split() method returns a list.

Activity: 7 -- Multiple Choice (assess_question2_1_1_7)

sequences-10-6: What is the type of `a` ?

```
b = "My, what a lovely day"
x = b.split(',')
z = "".join(x)
y = z.split()
a = "".join(y)
```

[✓ A. string]

B. integer

C. float

D. list

Check me Compare me

✓ Yes, the string is split into a list, then joined back into a string, then split again, and finally joined back into a string.

Activity: 8 -- Multiple Choice (assess_question2_1_1_8)

Write code to determine how many 9's are in the list `nums` and assign that value to the variable `how_many`. Do not use a for loop to do this.

Save & Run 6/9/2020, 2:43:57 PM - 3 of 3 Show in CodeLens

```
nums = [4, 2, 23.4, 9, 545, 9, 1, 234.001, 5, 49, 8, 9 , 34, 52, 1, -2, 9.1, 4]
how_many = nums.count(9)
print(how_many)


```

```
3
```

Activity: 9 -- ActiveCode (assess_ac2_1_1_9)

Result	Actual Value	Expected Value	Notes
Pass	3	3	Testing that how_many is set correctly.
Pass	'for'	'nums ...ny)\n\n'	Testing that you didn't use a for loop.
Pass	None	False	Hardcode check

Expand Differences

You passed: 100.0% of the tests

Write code that uses slicing to get rid of the the second 8 so that here are only two 8's in the list bound to the variable nums.

Save & Run 6/9/2020, 2:54:31 PM - 5 of 5 Show in CodeLens

```
nums = [4, 2, 8, 23.4, 8, 9, 545, 9, 1, 234.001, 5, 49, 8, 9 , 34, 52, 1, -2, 9.1,
```

```python
nums = nums[:4]+nums[5:]
print(nums)
```

```
[4, 2, 8, 23.4, 9, 545, 9, 1, 234.001, 5, 49, 8, 9, 34, 52, 1, -2, 9.1, 4]
```

Activity: 10 -- ActiveCode (assess_ac2_1_1_10)

Result	Actual Value	Expected Value	Notes
Pass	[4, 2...1, 4]	[4, 2...1, 4]	Testing that nums is set correctly.
Pass	<__ma...ject>	True	Testing that you are using slices to remove the second 8 (Don't worry about actual and expected values)

Expand Differences

Expand Differences

You passed: 100.0% of the tests

Assign the last element of `lst` to the variable `end_elem`. Do this so that it works no matter how long lst is.

Save & Run 6/9/2020, 2:47:16 PM - 3 of 3 Show in CodeLens

```python
lst = ["hi", "goodbye", "python", "106", "506", 91, ['all', 'Paul', 'Jackie', "UMSI
end_elem = lst[-1]
print(end_elem)
```

```
26trombones
```

Activity: 11 -- ActiveCode (access_ac_2_1_1_11)

Result	Actual Value	Expected Value	Notes
Pass	'26trombones'	'26trombones'	Testing that end_elem has the correct element assigned.
Pass	None	False	Hardcoding Check (Don't worry about actual and expected values)

You passed: 100.0% of the tests

Assign the number of elements in `lst` to the variable `num_lst`.

Save & Run 6/9/2020, 2:48:14 PM - 2 of 2 Show in CodeLens

```python
lst = ["hi", "goodbye", "python", "106", "506", 91, ['all', 'Paul', 'Jackie', "UMSI
num_lst = len(lst)
print(num_lst)
```

```
30
```

Activity: 12 -- ActiveCode (access_ac_2_1_1_12)

Result	Actual Value	Expected Value	Notes
Pass	30	30	Testing that num_lst has the correct length assigned.
Pass	None	False	Hardcoding Check (Don't worry about actual and expected values)

You passed: 100.0% of the tests

Create a variable called `wrds` and assign to it a list whose elements are the words in the string `sent`. Do not worry about punctuation.

Save & Run   6/9/2020, 2:48:54 PM - 2 of 2   Show in CodeLens

```
sent = "The bicentennial for our university was in 2017"
wrds = sent.split()
print(wrds)

```

```
['The', 'bicentennial', 'for', 'our', 'university', 'was', 'in', '2017']
```

Activity: 13 -- ActiveCode (assess_ac_2_1_1_13)

Result	Actual Value	Expected Value	Notes
Pass	['The...017']	['The...017']	Testing that wrds has been correctly assigned.
Pass	None	False	Hardcoding Check (Don't worry about actual and expected values)

Expand Differences

You passed: 100.0% of the tests

You have attempted 14 of 13 activities on this page

6.10. Exercises">

7. Iteration">

6.10. Exercises">

✔ Completed. Well Done!

7. Iteration">Next Section - 7. Iteration

| Back to top