

✓ **Congratulations! You passed!**
TO PASS 70% or higher

Keep Learning

GRADE
85.71%

Lists

LATEST SUBMISSION GRADE

85.71%

1. Which of the following expressions evaluates to the list `[0, 1, 2, 3, 4]`?

1 / 1 point

☒ `list(range(0, 5, 1))`

✓ Correct

This expression returns the list `[0, 1, 2, 3, 4]`. Note that the third parameter specifies that the integer entries vary by one.

☐ `list(range(0, 4, 1))`

☐ `range(0, 5)`

☒ `list(range(0, 5))`

✓ Correct

This expression returns the list `[0, 1, 2, 3, 4]`.

2. Let `my_list` be the list `["This", "course", "is", "great"]`.

1 / 1 point

- What is `len(my_list)`?
- What non-negative number is the index of `"great"`? I.e., how would you replace the question marks in `my_list[??]` so that the resulting value is `"great"`?

Submit two numbers, one for each of these two questions, separated by spaces.

4 3

✓ Correct

3. If we want to split a list `my_list` into two halves, which of the following uses slices to do so correctly?

1 / 1 point

More precisely, if the length of `my_list` is $2n$, i.e., even, then the two parts should each have length n . If its length is $2n+1$, i.e., odd, then the two parts should have lengths n and $n+1$.

☐ `my_list[0 : len(my_list) // 2]` and `my_list[len(my_list) // 2 + 1 : len(my_list)]`

☒ `my_list[0 : len(my_list) // 2]` and `my_list[len(my_list) // 2 : len(my_list)]`

✓ Correct

☒ `my_list[: len(my_list) // 2]` and `my_list[len(my_list) // 2 :]`

✓ Correct

☐ `my_list[0 : len(my_list) // 2 - 1]` and `my_list[len(my_list) // 2 : len(my_list)]`

4. If n and m are non-negative integers, consider the list `final_list` computed by the code snippet below.

1 / 1 point

```
1 init_list = list(range(1, n))
2 final_list = init_list * m
```

The length of this list depends on the particular values of n and m used in computation. Which option below correctly expresses the length of `final_list` in terms of n and m ?

☐ $n \times (m - 1)$

☐ $n + m$

- ☐ $n \times m$
- ☒ $(n - 1) \times m$

✓ Correct

5. If n is a non-negative integer, consider the list `split_list` computed by the code snippet below.

1 / 1 point

```
1 test_string = "xxx" + " " * n + "xxx"
2 split_list = test_string.split(" ")
```

The length of this list depends on the particular values of n used in computation. Which option below correctly expresses the length of `split_list` in terms of n ?

- ☒ $n + 1$
- ☐ 2
- ☐ n
- ☐ 3

✓ Correct

6. Select the code snippets below in which `list2` is a copy of list `list1` (as opposed to simply being another reference to the list `list1`).

0 / 1 point

☒

```
1 list1 = list(range(1, 10))
2 list2 = [] + list1
```

✓ Correct

This code snippet makes a copy. Try modifying `list2` and seeing if `list1` is mutated.

☒

```
1 list1 = list(range(1, 10))
2 list2 = list1[:]
```

✓ Correct

This code snippet makes a copy. Try modifying `list2` and seeing if `list1` is mutated.

☐

```
1 list1 = list(range(1, 10))
2 list2 = list1
```

☐

```
1 list1 = list(range(1, 10))
2 list2 = list(list1)
```

You didn't select all the correct answers

7. Write a function `strange_sum(numbers)` that takes a list of integers and returns the sum of those items in the list that are **not** divisible by 3. When you are done, test your function using the code snippet below.

1 / 1 point

```
1 print(strange_sum([1, 2, 3, 4, 5, 1, 2, 3, 4, 5]))
2 print(strange_sum(list(range(123)) + list(range(77))))
```

The first line in the test should print the number **24** in the console. Enter the second number printed in the console in the box below.

6994

✓ Correct