Exercises

1. Must

My Enumerate

Mark as done

Implement the enumerate Python function.

You can call your function  my\_enumerate

Read about it [here](https://www.programiz.com/python-programming/methods/built-in/enumerate)

Obviously, do not use the python built-in enumerate function itself.

**Tests**

Make sure the following tests pass:

for index, elem in my\_enumerate([10, 20, 30, 40]):

print(index, elem)

Result:

0 10

1 20

2 30

3 40

for index, elem in my\_enumerate([10, 20, 30, 40], 10):

print(index, elem)

Result:

10 10

11 20

12 30

13 40

1. Must

My Accumulate

Mark as done

Write a my\_accumulate generator that returns a series of accumulated sum.

For Example:

for elem in my\_accumulate([1,2,3,4,5]):

print(elem)

Will return:

1

3

6

10

15

1. Must

Prime Factors Generator

Mark as done

Write a generator that returns the [prime factors](https://www.mathsisfun.com/prime-factorization.html) of a number.

In short, factors are numbers which by multiplication give a number.

For example, 2,6 are factors of 12, but 6 is not a prime number.

The prime factors of 12 are: 2,2,3.

Now don't get lazy and check your solution.

For example:

for x in get\_prime\_factors\_generator(100):

print(x)

Would print:

2

2

5

5

1. Circle Iterator

Mark as done

Create an iterator that given a sequence and a <number of times>, will go through the elements in the sequence <number of times> times.

If the sequence is finished, start from the beginning.

Implement the iterator using a class that implements the iterator interface.

Note: You are not allowed to use the iterator interface in your implementation.

Usage Example:

for x in CircleIter([1,2],5):

print(x, end=" ")

The output should be:

1 2 1 2 1

Another Example:

for x in CircleIter([1,2,3],4):

print(x, end=" ")

for y in CircleIter([5,6],3):

print(y, end=" ")

print()

The output should be:

1 5 6 5

2 5 6 5

3 5 6 5

1 5 6 5

ראש הטופס