Solution Review: Yield Fibonacci Sequence From 1st to Nth Number

This lesson discusses how you can use generators to return the first n terms of the Fibonacci sequence.

WE'LL COVER THE FOLLOWING

Solution:

The first and second terms of the Fibonacci sequence are 0 and 1 respectively. Each successive term in the sequence is then calculated by adding the two preceding terms. That is, the nth term is calculated by adding the (n-2)th and (n-1)th terms. Thus, we need to store the already computed sequence in a list to be able to calculate each new element in the sequence. The code snippet below uses this logic to return the Fibonacci sequence. Lines 12 and 13 simply use a for loop to print the first 8 terms in the sequence.

```
def fibonacci(n):
  myArray = []
  for i in range(n):
    if i is 0 or i is 1:
      myArray.append(i)
      yield i
    else:
      x = myArray[i-2] + myArray[i-1]
      myArray.append(x)
      yield x

for i in fibonacci(8):
  print(i)
```

Now that you've learnt and practised generators, let's move on to asynchronous programming in the next chapter.