Solution Review: Return Sequence of Fibonacci Numbers

This lesson discusses how you can use iterators to return return a Fibonacci sequence.

Solution:

The first and second elements of the Fibonacci sequence are 0 and 1 respectively. Each successive element is obtained by adding the two elements before it. For instance, the 3rd element is obtained by adding the 1st and 2nd elements, the 4th is obtained by adding the 2nd and 3rd elements, and so on. Thus we can conclude the following for `myArray` in a Fibonacci sequence:

$$myArray[i] = myArray[i-2] + myArray[i-1]$$

We used this logic in lines 11 to 14 of the following code to calculate the Fibonacci sequence up to a certain range `n`.

```python
class MyRange:
  def __init__(self, n):
    self.n = n

  def __iter__(self):
    return self

  def next(self):
    myArray = []
    for i in range(self.n): # from n to 0
      if i == 0 or i == 1:
        myArray.append(i) # adds the even number to the list
      else:
        myArray.append(myArray[i-2] + myArray[i-1])
    return myArray

myrange = MyRange(8)
print(myrange.next())
```

This code is illustrated in the figure below:

i = 0

i = 0

0

i = 1

0

i = 1

| 0 | 1 |

i = 2

| 0 | 1 |

i = 2

| 0 | 1 |

$i = 2$
$0+1=1$

0	1	1

$i = 2$

0	1	1

$i = 3$

0	1	1

$i = 3$

0	1	1

$i = 3$
$1+1=2$

0	1	1	2

$i = 3$

0	1	1	2

i = 4

0	1	1	2

i = 4

0	1	1	2

i = 4
1+2=3

0	1	1	2	3

i = 4

0	1	1	2	3

i = 5

0	1	1	2	3

i = 5

0	1	1	2	3

i = 5
2+3=5

0	1	1	2	3	5

i = 5

0	1	1	2	3	5

i = 6

0	1	1	2	3	5

i = 6

0	1	1	2	3	5

i = 6
3+5=8

0	1	1	2	3	5	8

i = 6

0	1	1	2	3	5	8

0	1	1	2	3	5	8

i = 7

0	1	1	2	3	5	8

i = 7
5+8=13

0	1	1	2	3	5	8	13

i = 1

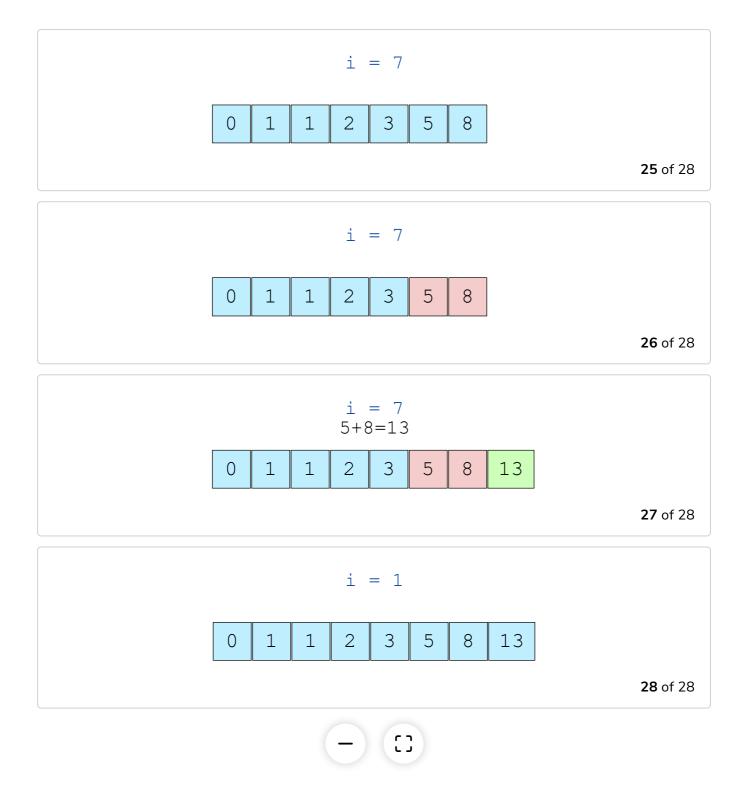| 0	1	1	2	3	5	8	13

Now that you've learnt and practised iterators, let's discuss generators in the next chapter.