

# Iteration and Loops

---

Remix CS 2019-20

# For Loops

- In Python for loops allow you to iterate over the items in a iterable.
- An iterable is something like a String or List, as well as dictionaries and Sets, which we will cover later.

# For Loop Syntax

This sets the name of the variable that is used to reference the item in the iterable that the loop is at

For loop line starts with the keyword “for”

Iterable that you are looping over goes here

For loop line ends with a semicolon

Code	Output
1 <code>iterable = [1, 2, 3, 4]</code>	1
2	2
3 <code>for item in iterable:</code>	3
4 <code>print(item)</code>	4

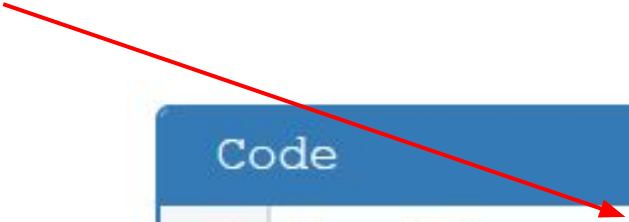
The variable created on the first line is assigned the value of each the iterable at each step along the loop, and can be called to get the value of the iterable at that step

# Practice Exercises

1. Find the sum of the elements in the list [2, 5, 18, 13, 8].
2. Find the average of the elements in the list [2, 5, 18, 13, 8].
3. Challenge Problem: Find the greatest element in the list [2, 5, 18, 13, 8] using a for loop.

# Iterating through values in a range

range(n) creates a  
iterable with every value  
Greater than or equal to  
zero and less than n.



Code	Output
1 for i in range(4): 2     print(i)	0 1 2 3

You can also specify the starting value used to create the range.

Code		Output
1	<code>for i in range(2, 4):</code>	2
2	<code>print(i)</code>	3

Code		Output
1	<code>for i in range(1, 5, 2):</code>	1
2	<code>print(i)</code>	3

You can also specify a step size that is used for the range, in this example every second element is used in the range.

You can use the start, end and step size properties to create a range that decreases in value as you iterate through it.

Code		Output
1	<code>for i in range(3, -1, -1):</code>	3
2	<code>print(i)</code>	2
		1
		0

## Why is range useful?

Situation: Say you wanted to loop through 2 lists of the same length, and create a new list where the first number in the new list was the sum of the first number in each of the other 2 lists, and the second number in the new list was the sum of the second number in each of the other 2 lists, etc, etc, until the end of the lists.

# You could do something like this...

But there's a problem.

You can get each element in the first list, but how do you get the element from the second list?

Code

```
1 list1 = [2, 8, 51]
2 list2 = [7, 13, 4]
3
4 for number in list1:
5     # Do something
```



# But what about this:

Code	Output
<pre>1 list1 = [2, 8, 51] 2 list2 = [7, 13, 4] 3 4 new_list = [] 5 6 for index in range(len(list1)): 7     number1 = list1[index] 8     number2 = list2[index] 9     sum = number1 + number2 10    new_list.append(sum) 11 12 print(new_list)</pre>	<pre>[9, 21, 55]</pre>

# Practice Exercises

1. Use a for loop with range to print out the sequence 13, 10, 7, 4.
2. Given `list1 = [5, 18, -9, 22]` and `list2 = [2, 34, -3, -88]` create a new list where the first element in the new list is the greatest of the first elements of `list1` and `list2`, and the second element of the new list is the greatest of the second elements of `list1` and `list2`, etc, etc.

# While Loops

- While loops loop while a condition is met.
- While loops are useful if you want a section of a program to run until something happens, but you may not know exactly when it when it will occur.

# While Loop Syntax

The while loop executes while the condition (a boolean) is True

The while loop line begins with the keyword “while”

While loop line ends with a semicolon

Code	Output
1 number = 0	0
2	1
3 while number < 5:	2
4     print(number)	3
5     number = number + 1	4

Something must change within the while loop so that the condition becomes false, otherwise the loop will repeat infinitely and never progress

# Practice Exercises

1. Use a while loop to loop through every number starting at 8 that is greater than or equal to zero.
2. Use a while loop to loop through the list [8, 9, 24, -7, 13, 22] and print the index of the first item that is less than 0.