## **For Loops Recap**

For loops allow you to iterate over a sequence of values. Let's take the example from the beginning of the video:

for x in range(5):

  print(x)

Similar to if statements and while loops, for loops begin with the keyword **for** with a colon at the end of the line. Just like in function definitions, while loops and if statements, the body of the for loop begins on the next line and is indented to the right. But what about the stuff in between the for keyword and the colon? In our example, we’re using the range() function to create a sequence of numbers that our for loop can iterate over. In this case, our variable **x** points to the current element in the sequence as the for loop iterates over the sequence of numbers. Keep in mind that in Python and many programming languages, a range of numbers will start at 0, and the list of numbers generated will be one less than the provided value. So range(5) will generate a sequence of numbers from 0 to 4, for a total of 5 numbers.

Bringing this all together, the range(5) function will create a sequence of numbers from 0 to 4. Our for loop will iterate over this sequence of numbers, one at a time, making the numbers accessible via the variable **x** and the code within our loop body will execute for each iteration through the sequence. So for the first loop, **x** will contain 0, the next loop, 1, and so on until it reaches 4. Once the end of the sequence comes up, the loop will exit and the code will continue.

The power of for loops comes from the fact that it can iterate over a sequence of any kind of data, not just a range of numbers. You can use for loops to iterate over a list of strings, such as usernames or lines in a file.

Not sure whether to use a for loop or a while loop? Remember that a while loop is great for performing an action over and over until a condition has changed. A for loop works well when you want to iterate over a sequence of elements.