

For loops

We just learned about conditional statements and how they can be developed to allow computers to make decisions.

But sometimes we need our programs to simply count or perform a task over and over again.

When it comes to tasks that are tedious, it's normal for humans to lose focus and energy.

It's in situations like these where computers can be especially helpful.

In this video, we'll examine how computers can perform repetitive tasks using iterative statements.

An iterative statement is code that repeatedly executes a set of instructions.

Iterative statements are also referred to as "loops." Setting up a loop allows us to repeatedly use a line of code without having to type it multiple times.

Before discussing the syntax, let's run a loop so you can experience what happens.

Notice how this code printed all the numbers in the list with only one print statement.

That's a loop.

There are two types of loops we'll explore: for loops and while loops.

We just ran a for loop, and we'll continue to focus on them in this video.

Later, we'll explore while loops.

for loops repeat code for a specified sequence.

An example of this would be using a for loop to print every item in a list.

For loops begin with the keyword for.

for signals the beginning of a for loop.

Similar to conditional statements, iterative statements consist of two main parts.

The parts of a loop are the loop header and the loop body.

Let's examine the for loop we just ran and use that to explore these parts.

The loop header is the line that contains the for keyword and ends with a colon.

It tells Python to start a loop.

It consists of the for keyword, a loop variable, and the sequence the loop will iterate through.

The loop variable is a variable that is used to control the iterations of a loop.

The loop variable comes directly after for.

A common name for it is the letter i, but you can give it any other name you want.

In for loops, this temporary variable is only used within the loop and not outside of it in the rest of the code.

The loop variable is followed by the in operator and the sequence the loop will iterate through.

In this example, this sequence is a list containing numbers from one through four.

It runs each of these numbers through a specified action.

We need to remember to put a colon at the end of the loop header to introduce this code.

The loop body refers to the indented lines after the loop header.

This represent the actions that are repeated while the loop iterates.

In this case, it will print each number in the list: first one, and then two, and so on.

Another important use of for loops is to repeat a specific process a set number of times.

This is done through combining it with the range function.

The range function generates a sequence of numbers.

As an example, range from zero to 10 sets a sequence that goes from zero, one, two all the way up

until the number nine.

When we use range, we start counting at the number in the first position; in this case, zero.

Then when we reach the number in the second position, it tells us where to stop.

That number is excluded.

In this case, where the number is 10, the sequence only goes up until nine.

An important detail about the range function is that if we don't provide a start point, it automatically starts from zero.

10 represents the stop point.

Since the stop point is excluded, the numbers included in the sequence start at zero and end at nine.

A sequence that starts at zero and ends at nine will iterate 10 times.

Let's run a for loop that incorporates the range function.

We'll use range to ask Python to repeat an action 10 times.

Then we'll indicate the action we want to repeat.

This action is printing an error message that indicates "cannot connect to the destination." Let's run this.

Using a for loop with the range function allowed us to repeat the same error message 10 times, instead of typing it over and over again ourselves.

In this video, we learned about the syntax and structure of iterative statements and worked with for loops as an example.

In the next video, we'll cover another type of iterative statement: the while loop.

Revision #1

Created 2023-12-11 07:20:35 UTC by naruzkurai

Updated 2023-12-19 03:38:35 UTC by naruzkurai