Write a simple algorithm

In our everyday lives, we frequently follow rules for solving problems.
As a simple example, imagine you want a cup of coffee.
If you've made coffee many times, then you likely follow a process to make it.
First, you grab your favorite mug.
Then, you put water into the coffee maker and add your coffee grounds.
You press the start button and wait a few minutes.
Finally, you enjoy your fresh cup of coffee.
Even if you have a different approach to making coffee or don't drink coffee at all, you will likely follow a set of rules for completing similar everyday tasks.
When you complete these routine tasks, you're following an algorithm.
An algorithm is a set of rules that solve a problem.
In more detail, an algorithm is a set of steps that takes an input from a problem, uses this input to perform tasks, and returns a solution as an output.
Let's explore how algorithms can be used to solve problems in Python.
Imagine that you, as a security analyst, have a list of IP addresses.
You want to extract the first three digits of each IP address, which will tell you information about the networks that these IP addresses belong to.
To do this, we're going to write an algorithm that involves multiple Python concepts that we've covered so far: loops, lists, and strings.
Here's a list with IP addresses that are stored as strings.
For privacy reasons, in our example, we're not showing the full IP addresses.
Our goal is to extract the first three numbers of each address and store them in a new list.
Before we write any Python code, let's break down an approach to solving this problem with an algorithm.
What if you had one IP address instead of an entire list?
Well, then the problem becomes much simpler.
The first step in solving the problem will be to use string slicing to extract the first three digits from one IP address.
Now let's consider how to apply these to an entire list.
As the second step, we'll use a loop to apply that solution to every IP address on the list.
Previously, you learned about string slicing, so let's write some Python code to solve the problem for one IP address.
Here we're starting with one IP address that begins with 198.567.
And we'll write a few lines of code to extract the first three characters.
We'll use the bracket notation to slice the string.
Inside the print statement, we have the address variable, which contains the IP address we want to slice.
Remember that Python starts counting at 0.
To get the first three characters, we start our slice at index 0 and then continue all the way until index 3.
Remember, that Python excludes the final index.

In other words, Python will return the characters at index 0, 1, and 2.
Now, let's run this.
We get the first three digits of the address: 198.
Now that we're able to solve this problem for one IP address, we can put this code into a loop and apply it to all IP addresses in the original list.
Before we do this, let's introduce one more method that we'll be using in this code: the append method.
The append method adds input to the end of a list.
For example, let's say that my list contains 1, 2, and 3.
With this code, we can use the append method to add 4 to this list.
First, we are given the IP list.
Now, we're ready to extract the first three characters from each element in this list.
Let's create an empty list to store the first three characters of each IP from the list.
Now we can start the for loop.
Let's break this down.
The word "for" tells Python that we're about to start a for loop.
We then choose address as the variable inside of the for loop, and we specify the list called IP as the iterable.
As the loop runs, each element from the IP list will be stored temporarily in the address variable.
Inside the for loop, we have a line of code to add the slice from address to the networks list.
Breaking this down, we use the code we wrote earlier to get the first three characters of an IP address.
We'll use our append method to add an item to the end of a list.
In this case, we're adding to the networks list.
Finally, let's print the networks list and run the code.
The variable networks now contains a list of the first three digits of each IP address in the original list: IP.
That was a lot of information.
Designing algorithms can be challenging.
It's a good idea to break them down into smaller problems before jumping into writing your code.
We'll continue to practice this idea in the upcoming videos.
Meet you there.

Revision #1
Created 2023-12-24 12:48:30 UTC by naruzkurai
Updated 2023-12-27 11:35:24 UTC by naruzkurai