3 easy ways to traverse a list in Python

Iterating through a list is a task that you often have to do when programming on languages ??in general and Python in particular.

Iterating through a list is a task that you often have to do when programming on languages ??in general and Python in particular. Therefore, make sure that you are doing it in the best possible way.

Python is one of the fastest growing programming languages. Programmers use it for machine learning, data science, and a host of other fields. Before you start learning the more advanced aspects of the language, it's a good idea to master one of the most common data structures, lists.

Lists in Python are arrays, very familiar to other programming languages ??like C and C++. You can change the size of the list, and for convenience Python has several different methods available for lists. You can store multiple data types in a list, such as strings, objects, and even other lists.

Why do you need to know different iteration techniques?

You might be asking the question that a simple for loop can already iterate through a list in Python, why need to know other ways to do it?

Often using minification methods, such as list comprehensions or lambda functions, will make your code shorter and less cluttered. Also, with a complex list with many elements, you will often have to experiment to find the most efficient traversal.

More importantly, when you apply for a job as a programmer, recruiters may ask complex questions about listings. Therefore, if you know the different ways of browsing the list, you will be able to answer those questions more easily.

Method 1: Browse using For loop and Range . method

Like other programming languages, using a for loop is the most common method for traversing a list in Python.

```
arr = [10, 20, 30, 40] for val in arr: print('value', val)
```

Alternatively, you can also use the range() method to have more control over your for loop. The range() method takes three arguments:

1. start: Indicates the start index of the for loop traversal.
2. stop: Tells the program the last/stop index for the for loop traversal. It is common to use the length of the list (number of elements) as the stopping index.

3. step: The step size argument is optional. If provided, it sets how many for loops increment its running counter each time. By default, the step size is 1.

Iterate through a Python list using the range() method:

```
arr = [10, 20, 30, 40, 50, 60] for key in range(0, len(arr), 2): print('num', key
```

The above example runs a for loop from index 0 until the length of the array is exhausted and increments the loop counter by 2.

Method 2: Shorten browsing with List comprehension

One of Python's most intuitive features is list comprehension. It allows you to write simple, one-line solutions to many different problems.

Example: To calculate the square of the first 10 numbers, you just need to write this line of code:

```
sq = [x ** 2 for x in range(10)]
```

Given a list of numbers, you can print them out using a list comprehension like this:

```
arr = [1, 3, 5, 7, 9] [print(val) for val in arr]
```

Once mastered, list comprehension is an extremely powerful tool and can make coding a lot simpler.

Method 3: Use an inline lambda function

Normally, we declare functions in Python using the def keyword and must provide the body and header of the function. Lambda functions are another powerful feature of Python with the ability to make coding simpler, shorter, and easier. They have no names and can only contain a single expression. However, you can put any number of parameters into a lambda function.

When combined with the map() method, the lambda function can effectively act as a for loop. To print a list of numbers using a combination of the lambda and map() functions you write the following code:

```
arr = [1, 3, 4, 5, 6, 7, 8] myFun = list(map(lambda z:z, arr)) print(myFun)
```

Loops are essential in every programming language, and Python is no exception. Most of the programs you write will have one or more loops, not in one place then another, not in one form then another.

TipsMake.com hope that this article will be useful to you!