

Iteration: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2023

Syntax

- Iterate over a range of integers using the `range()` function:

```
for index in range(0, 5):  
    print(index)
```

- Iterate over the elements of a list directly with a `for` loop:

```
compensation = [124267, 236889, 74321, 62593, 36288]  
for comp in compensation:  
    print(comp * 1.34)
```

- Import a specific function from a Python module:

```
from csv import reader
```

- Use the `open()` function and a `with` statement to safely open a `.csv` file:

```
with open('kaggle2021-short.csv') as f:  
    read_file = reader(f, delimiter=",")  
    kaggle_data = list(read_file)
```

Concepts

- A `for` loop has three elements:
 - a sequence of values, such as a list
 - a **loop variable** that holds the information from a single element in the list as we iterate
 - a code block that is repeated for each item in the sequence
- There are different methods for iterating over a list. One way is to iterate over indices, and another is to directly iterate over the elements of the list itself. We recommend using the latter method since it produces cleaner code.
- Python modules are collections of code that can be reused. Modules often contain useful functions that are frequently used in data analysis. The `csv` module is an example of this, and we imported the `reader()` function from this module to use it.
- Data cleaning is the process of identifying and correcting errors, inconsistencies, and inaccuracies in data to make it suitable for further analysis and use.

Resources

- [More information on for loops](#)