

# Python For Loops: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2022

## Syntax

- Repeating a process using a for loop:

```
row_1 = ['Facebook', 0.0, 'USD', 2974676, 3.5]
for element in row_1:
    print(element)
```

- Appending values to a list:

```
a_list = [1, 2]
a_list.append(3)
```

- Opening a dataset file and using it to create a list of lists:

```
opened_file = open('AppleStore.csv')
from csv import reader
read_file = reader(opened_file)
apps_data = list(read_file)
```

- Converting a string to a float:

```
rating_sum = 0
for row in apps_data[1:]:
    rating = float(row[7])
    rating_sum = rating_sum + rating
```

## Concepts

- We can automate repetitive processes using **for loops**.
- We always start a for loop with `for` (like in `for element in app_ratings:` ).
- The indented code in the **body** gets executed the same number of times as elements in the **iterable variable**. If the iterable variable is a list containing three elements, the indented code in the body gets executed three times. We call each code execution an **iteration**, so there will be three iterations for a list that has three elements. For each iteration, the **iteration variable** will take a different value.

## Resources

- [Python For Loops](#)
- [A list of keywords in Python](#) — `for` and `in` are examples of keywords (we used `for` and `in` to write for loops)