

# Python For Loop with Index

## Python For Loop with Index

To access index in [Python For Loop](#), you can use `enumerate()` function or `range()` function.

In this tutorial, we will go through example programs that demonstrate how to iterate over an iterable and access index as well in the loop.

### Python For Loop with Index using `enumerate()`

`enumerate()` function keeps track of the count of elements we are iterating in the loop.

In the following example, we have used `enumerate()` function, to access both current index and element during an iteration with for loop.

#### Python Program

```
list_1 = ["apple", "banana", "orange", "mango"]

for index, element in enumerate(list_1):
    print(index, element)
```

#### Output

```
0 apple
1 banana
2 orange
3 mango
```

### Python For Loop with Index using `range()`

`range()` function returns an iterable with range of numbers. So, `range()` function can be used to iterate over the range of indexes of the iterable.

Please note that, using `range()` function, to get index and accessing element using index, is not in the spirit of python. `enumerate()` is relatively recommended if you would like to access index in the for loop.

In the following example, we will use `range()` function, to access index of iterable.

#### Python Program

```
list_1 = ["apple", "banana", "orange", "mango"]

for index in range(len(list_1)):
    print(index, list_1[index])
```

#### Output

```
0 apple  
1 banana  
2 orange  
3 mango
```

## Conclusion

Concluding this [Python Tutorial](#), we learned how to access index of the iterable in for loop during each iteration.

## Python Programming

‡ [Python Tutorial](#)

‡ [Install Python](#)

‡ [Install Anaconda Python](#)

‡ [Python HelloWorld Program](#)

‡ [Python Variables](#)

‡ [Python Variable Data Type Conversion](#)

‡ [Python Comments](#)

## Control Statements

‡ [Python If](#)

‡ [Python If Else](#)

‡ [Python While Loop](#)

‡ [Python For Loop](#)

## Python String

‡ [Python String Methods](#)

‡ [Python String Length](#)

‡ [Python String Replace](#)

‡ [Python Split String](#)

‡ [Python Count Occurrences of Sub-String](#)

‡ [Python Sort List of Strings](#)

## Functions

‡ [Python Functions](#)

## Python Collections

‡ [Python List](#)

‡ [Python Dictionary](#)

## Advanced

‡ [Python Multithreading](#)

## Useful Resources

‡ [Python Interview Questions](#)