



List Comprehension in Python-Complete Tutorial

Python / March 25, 2020



This **List Comprehension in Python-Complete Tutorial** is made for you to give complete knowledge of the **List Comprehension in Python**. At the end of this tutorial, you will have full knowledge of the **List Comprehension in Python**.

Hello, & Welcome! to the **List Comprehension in Python-Complete Tutorial**.

Part 16- List Comprehension in Python-Complete Tutorial

In this article of Python Tutorial, you will learn following-

1. How to Create a List using For Loop.
2. Using List Comprehension.

How to Create a List using For Loop.

You can create a list using For Loop. Here, in this tutorial, I will teach you with the help of an example. Suppose, there is a list-

```
prizes=[5,10,50,100,1000]
```

This is the list of employees prize, but boss announces a double bonanza offer, so to update the list with double bonanza, you don't need to override the list, you can do it with for loop.

Let's see how you can do it by using for loop-

```
prizes=[5,10,50,100,1000]
dbl_prize=[]
for prize in prizes;
    dbl_prize.append(prize*2)
print(dbl_prize)
```

OUTPUT-

```
[10,20,100,200,2000]
```

Here, we have created an empty list `dbl_prize`, then we use for loop. Inside for loop, we use the `append` method to append a new list in the multiple of 2. Therefore, we are getting output as `[10,20,100,200,2000]`.

Using List Comprehension.

By using List Comprehension, you can create a list in python in one line only. Unlike in for loop we have written two to three lines of code, but in the list comprehension, we can perform the same task in one line.

First, let's see the syntax of List Comprehension-

```
list_name=[expression for item in list]
```

Here, an **expression** may be anything, which you wanna give. Like in for Loop, we give expression as "prize*2". So you can pass any expressions like that. The expression must return a value.

Item is the object or value in the list. It may be anything. Like in for loop example, the item is a "prize". You can give it as 'i' or anything.

The **list** is the name of **list** that can return its elements only one at a time. In for loop example, list is "prizes".

Now, if you have to perform the same task as we did by for loop, you can do it by **List Comprehension**.

Let's see how you update the prize list with double bonanza with the help of list comprehension-

```
prizes=[5,10,50,100,1000]
dbl_prizes=[prize*2 for prize in prizes]
print(dbl_prizes)
```

OUTPUT-

```
[10,20,100,200,2000]
```

Here, we have written in one line as "**dbl_prizes=[prize*2 for prize in prizes]**". Unlike in for loop, we have to write two or three lines of code.

Here, prize*2 is an expression.

prize is item

and prizes is name of the list.

One More Example-

Let's take one more example to understand List Comprehension. Suppose you have a list of numbers like-

```
nums=[1,2,3,4,5,6,7,8,9,10]
```

and you have to convert this list into squared even numbers. So you can do this with two methods.

The first method is by using For Loop. And the second method is List Comprehension.

I will discuss both methods here. Let's start with the For Loop method.

```
nums=[1,2,3,4,5,6,7,8,9,10]
squared_even_nums=[]
for num in nums:
    if(num**2)%2==0:
        squared_even_nums.append(num**2)
print(squared_even_nums)
```

OUTPUT -

```
[4,16,36,64,100]
```

Here, first, we create an empty list as "**squared_even_nums**". Then we use For Loop method, inside **For Loop** we use **if condition** to check that number is even or odd. And if the number is even then we perform **multiplication with 2**. Then print the list.

By List Comprehension-

The same task is performed in less line of code with the List Comprehension method. Let's see how to perform in List Comprehension-

```
nums=[1,2,3,4,5,6,7,8,9,10]
squared_even_nums=[num*2 for num in nums if(num**2)%2==0]
print(squared_even_nums)
```

OUTPUT -

```
[4, 16, 36, 64, 100]
```

Here, the same task is performed with one line of code. That's why List Comprehension is easy as compare to For Loop.

That's all for the **List Comprehension in Python**. I hope now you have a better understanding of the **List Comprehension in Python**.

Congratulations! You successfully learned the **List Comprehension in Python**.



In the next tutorial, we will start learning **Map, Filter, and Lambda in Python**.

Till then, Enjoy Learning Python!

All the Best!

Are you ML Beginner and confused, from where to start ML, then read my BLOG – [How do I learn Machine Learning?](#)

If you are looking for Machine Learning Algorithms, then read my Blog – [Top 5 Machine Learning Algorithm](#).

If you are wondering about Machine Learning, read this Blog- [What is Machine Learning?](#)

Thank YOU!

Though of the Day...

“Live as if you were to die tomorrow. Learn as if you were to live forever.”

– Mahatma Gandhi

[← Previous Post](#)

[Next Post →](#)

I thought on “List Comprehension in Python-Complete Tutorial”

Pingback: [Modules and Packages in Python-Complete Tutorial for Everyone\(2020\)](#)

Leave a Comment

Your email address will not be published. Required fields are marked *

Type here..

☐ Save my name, email, and website in this browser for the next time I comment.

[Post Comment »](#)

Follow Us on Twitter-



Follow us on Pinterest-



Our Facebook Group-



Copyright © 2020 MLTut

Powered by MLTut