

# Python - compreensão de lista

A compreensão de listas é uma ferramenta de programação muito poderosa. É semelhante à notação do construtor de conjuntos em matemática. É uma maneira concisa de criar uma nova lista executando algum tipo de processo em cada item da lista existente. A compreensão da lista é consideravelmente mais rápida do que processar uma lista por loop **for**.

## Exemplo 1

Suponha que queiramos separar cada letra de uma string e colocar todas as letras não vogais em um objeto de lista. Podemos fazer isso por meio de um loop **for** conforme mostrado abaixo -

```
chars=[]  
for ch in 'TutorialsPoint':  
    if ch not in 'aeiou':  
        chars.append(ch)  
print (chars)
```

O objeto da lista de caracteres é exibido da seguinte forma -

```
['T', 't', 'r', 'l', 's', 'P', 'n', 't']
```

## Técnica de compreensão de lista

Podemos facilmente obter o mesmo resultado por meio de uma técnica de compreensão de lista. Um uso geral da compreensão de lista é o seguinte -

```
listObj = [x for x in iterable]
```

Aplicando isso, a lista de caracteres pode ser construída pela seguinte instrução -

```
chars = [ char for char in 'TutorialsPoint' if char not in 'aeiou']  
print (chars)
```

A lista de caracteres será exibida como antes -

```
['T', 't', 'r', 'l', 's', 'P', 'n', 't']
```

## Exemplo 2

O exemplo a seguir usa compreensão de lista para construir uma lista de quadrados de números entre 1 e 10

```
squares = [x*x for x in range(1,11)]  
print (squares)
```

O objeto da lista de quadrados é -

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

## Loops aninhados na compreensão de lista

No exemplo a seguir, todas as combinações de itens de duas listas na forma de uma tupla são adicionadas em um terceiro objeto de lista.

### Exemplo 3

```
list1=[1,2,3]  
list2=[4,5,6]  
CombLst=[(x,y) for x in list1 for y in list2]  
print (CombLst)
```

Ele produzirá a seguinte **saída** -

```
[(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6)]
```

## Condição na compreensão da lista

A instrução a seguir criará uma lista de todos os números pares entre 1 e 20.

### Exemplo 4

```
list1=[x for x in range(1,21) if x%2==0]  
print (list1)
```

Ele produzirá a seguinte **saída** -

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```