

ASIMOV

Compreensão em listas

Além das operações de sequência e métodos de lista, o Python inclui uma operação mais avançada chamada de compreensão de lista.

As compreensões de lista nos permitem construir listas usando uma notação diferente. Você pode pensar nisso essencialmente como um loop construído dentro de colchetes. Um exemplo simples:

Exemplo 1

```
In [1]: # Pega todas as letras em uma string  
lst = [x for x in 'word']
```

```
In [2]: # Checa  
lst
```

```
Out[2]: ['w', 'o', 'r', 'd']
```

Esta é a idéia básica de uma lista de compreensão. Se você estiver familiarizado com a notação matemática, esse formato deve se sentir familiar, por exemplo: x^2 : x em $\{0,1,2 \dots 10\}$

Vejamos mais alguns exemplos de compreensões de lista em Python:

Exemplo 2

```
In [3]: # Eleva o quadrado itens no range e o transformam em lista  
lst = [x**2 for x in range(0,11)]
```

```
In [2]: lst
```

```
Out[2]: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Exemplo 3

Vamos ver como adicionar usando if:

```
In [4]: # Cria uma lista de números pares, utilizando para isso o if  
lst = [x for x in range(11) if x % 2 == 0]
```

```
In [6]: lst
```

```
Out[6]: [0, 2, 4, 6, 8, 10]
```

Exemplo 4

Também pode fazer operações aritméticas mais complicadas:

```
In [6]: # Converte Celsius para Fahrenheit  
celsius = [0,10,20.1,34.5]  
  
fahrenheit = [ ((float(9)/5)*temp + 32) for temp in celsius]  
  
fahrenheit
```

```
Out[6]: [32.0, 50.0, 68.18, 94.1]
```

Exemplo 5

Também podemos realizar compreensões de lista aninhadas, por exemplo:

```
In [8]: lst = [ x**2 for x in [x**2 for x in range(11)]]  
lst
```

```
Out[8]: [0, 1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000]
```

Mais tarde, no curso, aprenderemos sobre compreensões do gerador. Após esta palestra, você deve se sentir confortável em ler e escrever compreensões de lista básica.