

List Comprehension e Generator Expression

Vamos começar com **List Comprehension**, e basicamente é um recurso do Python, onde podemos criar uma lista a partir de outra lista (ou coleção), aplicando uma **transformação** e/ou **filtro**.

Dito isso, é uma forma de criar listas transformadas ou filtradas **BEM MAIS SIMPLIFICADA**, ao invés de utilizar um **for looping** padrão junto com as funções de **map()** e/ou **filter()** que seria um código mais complexo e robusto!

Trabalhando com listas auxiliares

E para exemplificar melhor, vamos primeiro mostrar como que faremos a transformação de uma lista para outra utilizando o **for looping** padrão, veja:

```
# Trabalhando com listas auxiliares
frutas = ['banana', 'acai', 'morango']
frutas_result = []

for fruta in frutas:
    if 'b' in fruta:
        frutas_result.append(fruta)

print(frutas_result)
```

['banana']

Trabalhando com list comprehension

Veja que foi necessário criar uma lista “auxiliar”, e cada iteração do for fazemos uma validação para aí sim adicionar o item correto nessa nova lista transformada!

Agora vamos realizar a mesma transformação de uma lista para outra, mas utilizando o **list comprehension**:

```
frutas = ['banana', 'acai', 'morango']
frutas_result = [fruta for fruta in frutas if 'b' in fruta]

print(frutas_result)
```

['banana']

Generator Expression

É **QUASE** igual ao **List Comprehension**, ou seja, o **Generator Expression** também é um recurso do **Python**, onde podemos criar uma lista a partir de outra lista (ou coleção), aplicando uma **transformação** e/ou **filtro**.

Porém a diferença de ambos é que o **List Comprehension** cria uma lista em memória, já o **Generator Expression** cria um objeto **generator**, que seria um objeto que gera valores sob demanda, conhecido como objetos **lazy**!

E a sintaxe de um **Generator Expression** é muito simples, e em relação ao **List Comprehension** basta trocarmos os `[]` por `()`, ficando dessa forma:

```
# Generator Expression
frutas = ['banana', 'acai', 'morango']
frutas_result = (fruta for fruta in frutas if 'b' in fruta)

print(frutas_result)
print(list(frutas_result))

<generator object <genexpr> at 0x00000205FDCF05F0>
['banana']
```

Vale destacar então que utilize **List Comprehension** sempre que quiser trabalhar com **Lists**, ou seja com coleções menores. Já o **Generator Expression** utilize sempre quando quiser trabalhar com coleções maiores, sempre pensando na melhor performance!