

# Listas comprehension

segunda-feira, 22 de agosto de 2022 17:07

Usando list comprehension em python

Podemos usar os laços dentro de uma lista:

```
l1 = [1,2,3,4,5,6]
ex1 = [variavel for variavel in l1]
print(ex1)
```

podemos também usar com strings:

```
l2 = ['geronimo','eunice','stanley']
ex2 = [v.replace('e','3').upper() for v in l2]
print(ex2)
```

podemos assim usar multiplicação ou outros mais que quiser:

```
l3 = [1,2,3,4,5,6]
ex3 = [v * 2 for v in l3]
print(ex3)
```

podemos usar mais de 1 variavel:

```
l4 = [1,2,3,4,5,6,7,8,9,10]
ex4 = [(v, v2) for v in l4 for v2 in range(5)]
print(ex4)
```

Podemos usar tuplas:

```
tuplas = (('nome','geronimo'),('morais','stanley'))

nova_tupla = [(y, x) for x,y in tuplas]

print(nova_tupla)
```

R-

```
[('geronimo', 'nome'), ('stanley', 'morais')]
```

Process finished with exit code 0

Pegando apenas valores pares usando if nas listas comprehension:

```

lista = list(range(100))

listanova = [v for v in lista if v % 2 == 0]
print(listanova)

```

Usando mais de 1 condição:

```

lista = list(range(100))

listanova = [v for v in lista if v % 2 == 0 if v % 8 == 0]
print(listanova)

```

Usando if e else:

```

lista = list(range(100))

listanova = [v if v % 3 == 0 else 'não é' for v in lista]
print(listanova)

```

Exercicio :

```

string = '012345678901234567890123456789012345678901234567890123456789'
c = 10
listanova = [string[i: i+c] for i in range(0, len(string), c)]
retorno = '.'.join(listanova)
print(listanova)
print(retorno)

```

podemos somar uma list comprehension com o sum antes da lista.

```

carrinho = []

carrinho.append(("Produto1", 20))
carrinho.append(("Produto1", 50))
carrinho.append(("Produto1", 300))
carrinho.append(("Produto1", 1000))
carrinho.append(("Produto1", 10))
carrinho.append(("Produto1", 25))

novo = sum([float(valor[1]) for valor in carrinho])
print(f"O valor total foi {novo}")

```