

pyladies

**Seja mais Pythônico:
*List Comprehension***

Elloá B. Guedes

ebgcosta@uea.edu.br

www.elloaguedes.com

List Comprehension

- Jeito **sucinto** de criar listas baseando-se em listas existentes
- Maneira elegante
- Menos linhas de código
- Maneira de ser mais pythônico
- Você consegue o mesmo resultado sem *list comprehension*

Exemplo 1

- Quero separar as letras de uma *string* em uma lista

```
frase = "paralelepipedo"  
lista = []  
for letra in frase:  
    lista.append(letra)  
print(lista)
```

Exemplo 1

- Quero separar as letras de uma *string* em uma lista

```
frase = "paralelepipedo"  
lista = []  
for letra in frase:  
    lista.append(letra)  
print(lista)
```

```
['p', 'a', 'r', 'a', 'l', 'e', 'l',  
'e', 'p', 'i', 'p', 'e', 'd', 'o']
```

Exemplo 1

- Quero separar as letras de uma *string* em uma lista com *list comprehension*

```
frase = "paralelepipedo"  
lista = [letra for letra in frase]  
print(lista)
```

Exemplo 1

- Quero separar as letras de uma *string* em uma lista com *list comprehension*

```
frase = "paralelepipedo"  
lista = [letra for letra in frase]  
print(lista)
```

```
['p', 'a', 'r', 'a', 'l', 'e', 'l',  
'e', 'p', 'i', 'p', 'e', 'd', 'o']
```

List Comprehension

- Ideia geral

```
umaLista = [x for x in iterável]
```

List Comprehension

- Ideia geral

```
umaLista = [x for x in iterável]
```



```
Listas,  
strings e  
tuplas
```


Exemplo 2

- Quero formar uma lista nova a partir de uma lista antiga, excluindo os números pares

```
lista = [x for x in listaOriginal if x % 2 !=0]
```

Exemplo 2

- Quero formar uma lista nova a partir de uma lista antiga, excluindo os números pares

```
lista = [x for x in listaOriginal if x % 2 !=0]
```



Checagem de
condição

Exemplo 3

- Quero uma lista com todos os caracteres de uma string em maiúsculo
- *Pensando...*
- *List comprehension pra obter os caracteres*
- *Depois faço iteração pra transformar todos em maiúsculos*

Exemplo 3

- Quero uma lista com todos os caracteres de uma string em maiúsculo
- *Você pode fazer tudo de uma única vez!*

```
lista = [x.upper() for x in frase]
```

Exemplo 3

- Quero uma lista com todos os caracteres de uma string em maiúsculo
- *Você pode fazer tudo de uma única vez!*

```
lista = [x.upper() for x in frase]
```

Lembre-se: strings são imutáveis!

Exemplo 4

- Quero uma lista com o fatorial de todos os números ímpares de uma lista original
- *List comprehension + função + if*

Exemplo 4

- Quero uma lista com o fatorial de todos os números ímpares de uma lista original
- *List comprehension + função + if*

```
import math  
lista = [math.factorial(x) for x in listaOriginal if x % 2 != 0]
```



Exemplo 5

- Quero todos os números ímpares de 1 a 100

Exemplo 5

- Quero todos os números ímpares de 1 a 100
- *Preciso criar então uma lista com os números de 1 a 100 e depois usar list comprehension + if?*

Exemplo 5

- Quero todos os números ímpares de 1 a 100

```
lista = [x for x in range(1,100,2)]
```

Exemplo 5

- Quero todos os números ímpares de 1 a 100


```
lista = [x for x in range(1,100,2)]
```

Exemplo 5

- Quero todos os números ímpares de 1 a 100

```
lista = [x for x in range(1,100,2)]
```

Início: opcional
Se omitido,
considera-se zero




Exemplo 5

- Quero todos os números ímpares de 1 a 100

```
lista = [x for x in range(1,100,2)]
```

Limite: Obrigatório
Não será incluído

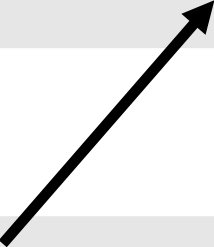


Exemplo 5

- Quero todos os números ímpares de 1 a 100

```
lista = [x for x in range(1,100,2)]
```

Salto: Opcional
Se omitido, igual a 1



Exemplo 6

- Quais são os números entre 1 e 100 divisíveis por 3 e por 5?

Exemplo 6

- Quais são os números entre 1 e 100 divisíveis por 3 e por 5?


```
lista = [x for x in range(1,101) if ((x % 3)==0) and ((x%5)==0)]
```


Exemplo 6

- Quais são os números entre 1 e 100 divisíveis por 3 e por 5?

```
lista = [x for x in range(1,101) if ((x % 3)==0) and ((x%5)==0)]
```

Expressão
lógica



Exemplo 6

- Quais são os números entre 1 e 100 divisíveis por 3 e por 5?

```
lista = [x for x in range(1,101) if ((x % 3)==0) if ((x%5)==0)]
```

Exemplo 6

- Quais são os números entre 1 e 100 divisíveis por 3 e por 5?

```
lista = [x for x in range(1,101) if ((x % 3)==0) if ((x%5)==0) ]
```

Ifs aninhados



Exemplo 7

- Faça uma lista com números de 1 a 100 em que:
 - Se o número for par, armazene o seu cubo
 - Em caso contrário, armazene o seu quadrado
-
- Noção de if-else

Exemplo 7

- Faça uma lista com números de 1 a 100 em que:
- Se o número for par, armazene o seu cubo
- Em caso contrário, armazene o seu quadrado

```
lista = [x**3 if (x%2 ==0) else x**2 for x in range(1,101)]
```

Exemplo 7

- Faça uma lista com números de 1 a 100 em que:
- Se o número for par, armazene o seu cubo
- Em caso contrário, armazene o seu quadrado

```
lista = [x**3 if (x%2 ==0) else x**2 for x in range(1,101)]
```

Exemplo 8

- List comprehension em loops aninhados

```
lista = []  
for x in [20, 40, 60]:  
    for y in [2, 4, 6]:  
        lista.append(x * y)  
print(lista)
```

Exemplo 8

- List comprehension em loops aninhados

```
lista = []  
for x in [20, 40, 60]:  
    for y in [2, 4, 6]:  
        lista.append(x * y)  
print(lista)
```

```
[40, 80, 120, 80, 160, 240,  
120, 240, 360]
```


Exemplo 8

- List comprehension em loops aninhados

```
lista = [x * y for x in [20, 40, 60] for y in [2, 4, 6]]
```

```
[40, 80, 120, 80, 160, 240,  
120, 240, 360]
```

Exemplo 9

- Às vezes temos duas listas de mesmo tamanho (sincronizadas)
- Queremos mapear os elementos de uma lista para outra
- Ideia de dicionário

```
# Quantas pessoas cada uma convidou para o pyladies  
lista1 = ["Jessica", "Juliany", "Elloa", "Daniele"]  
lista2 = [15, 17, 10, 22]
```

Exemplo 9

- Às vezes temos duas listas de mesmo tamanho (sincronizadas)
- Queremos mapear os elementos de uma lista para outra
- Ideia de dicionário

```
# Quantas pessoas cada uma convidou para o pyladies
lista1 = ["Jessica", "Juliany", "Elloa", "Daniele"]
lista2 = [15, 17, 10, 22]
d= {}
for i in range(len(lista1)):
    d[lista1[i]] = lista2[i]
print(d)
```

Exemplo 9

- Às vezes temos duas listas de mesmo tamanho (sincronizadas)
- Queremos mapear os elementos de uma lista para outra
- Ideia de dicionário

```
# Quantas pessoas cada uma convidou para o pyladies
lista1 = ["Jessica", "Juliany", "Elloa", "Daniele"]
lista2 = [15, 17, 10, 22]
d= {}
for i in range(len(lista1)):
    d[lista1[i]] = lista2[i]
print(d)
```

```
{'Jessica': 15, 'Daniele': 22,
'Juliany': 17, 'Elloa': 10}
```

Exemplo 9

- Às vezes temos duas listas de mesmo tamanho (sincronizadas)
- Queremos mapear os elementos de uma lista para outra
- Ideia de dicionário

```
# Quantas pessoas cada uma convidou para o pyladies
lista1 = ["Jessica", "Juliany", "Elloa", "Daniele"]
lista2 = [15, 17, 10, 22]
d = {lista1[i]: lista2[i] for i in range(len(lista1))}
print(d)
```

```
{'Jessica': 15, 'Daniele': 22,
 'Juliany': 17, 'Elloa': 10}
```

Exemplo 10

- Um exemplo interessante com strings
- Determinar quais strings de uma lista são palíndromas
- Pense com *list comprehension*!

Exemplo 10

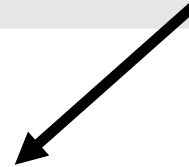
- Um exemplo interessante com strings
- Determinar quais strings de uma lista são palíndromas
- Pense com *list comprehension*!

```
novaLista = [x for x in lista if x == x[::-1]]
```

Exemplo 9

- Um exemplo interessante com strings
- Determinar quais strings de uma lista são palíndromas
- Pense com *list comprehension*!

```
novaLista = [x for x in lista if x == x[::-1]]
```



Fatiamento da string

Finalizando...

- **List comprehension**: uma maneira de ser mais pythônico!
 - Menos linhas de código
 - Aproxima da notação matemática
 - Melhoria na legibilidade
- Autores argumentam que há **melhorias no tempo de execução**
 - ~ 35% mais rápido que laços com for
 - ~ 45% mais rápido que laços com map

pyladies
Obrigada!

Elloá B. Guedes

ebgcosta@uea.edu.br

www.elloaguedes.com