

# For em Python

---

## Usando For em uma Lista

Utilizamos o For para andarmos dentro de uma Lista e pegarmos individualmente os dados internos.

Assim como vimos no IF, usamos alguns testes condicionais para verificarmos:

```
in # Verificar se o elemento está dentro da Lista
not in # Verificar se o elemento NÃO está dentro da Lista
```

Com esse teste condicional, podemos andar por toda a Lista e imprimir os dados, como abaixo:

```
caracteres = ['a', 'b', 'c', 'd']

for caracter in caracteres:
    print("Letra: " + caracter)
```

No exemplo acima, sempre que tiver um **caracter** na lista de **caracteres** podemos pegar esse dado e imprimir

Podemos agora modificar nossa Lista como quisermos:

### 1. Removendo um item específico da Lista

```
lista1 = ["Pepe", "A", "B", "Pepe", "C"]

for item in lista1:
    if "Pepe" in lista1:
        lista1.remove("Pepe")
print(lista1)
# Saida: ["A", "B", "C"]
```

### 2. Criando uma lista de valores de um intervalo de numeros

Usamos a Função **range(inicio,fim)** para pegarmos valores de um limite de valores disponiveis

Devemos definir qual é o valor de inicio e o de fim+1 para a Função, onde ele vai pegar esses valores para o nosso for:

```
range(0, 11) # Vai ser pego do número 0 até 10
```

com isso, se quisermos criar uma lista de valores que vão de 1 á 10, construímos o For da seguinte maneira:

```
lista = []

for number in range(0,11):
    lista.append(number)

print(lista)
#Saida: [1,2,3,4,5,6,7,8,9,10]
```

### 3. Criando uma lista de somente valores Pares ou Impares

Podemos criar listas com somente valores pares ou impares, mas para isso devemos construir um IF que verifica se o resto da divisão entre dois numeros é igual a zero(pares) ou diferente de zero(impares)

- Pares

```
lista = []

for number in range(1,11):
    if number%2 == 0:
        lista.append(number)
print(lista)
#Saida: [2, 4, 6, 8, 10]
```

- Impares

```
lista = []

for number in range(1,11):
    if number%2 != 0:
        lista.append(number)
print(lista)
#Saida: [1, 3, 5, 7, 9]
```

### 4. Criando uma lista nova com o quadrado da primeira lista

quando usamos \*\* em Python significa que estamos fazendo uma potenciação, onde o temos na esquerda a Base e na direita o Expoente:

```
2**2 # 2 elevado na 2 que vai dar 4
```

Então com essa informação, podemos criar uma nova lista somente com o dobro de cada número da lista original

```
values = [1,2,3,4,5,6,7,8]
double = []

for number in values:
    number = number**2
    double.append(number)
print(double)
```

## 5. Pegando um pedaço da lista

Podemos pegar somente um pedaço da lista usando `[]` junto da lista como abaixo:

```
lista = [1,2,3,4,5,6,7,8,9,10]

for number in lista[:5]:
    print(str(number))

#Saida:
# 1
# 2
# 3
# 4
# 5
```

Com esse código, podemos pegar somente os valores da lista que vão da primeira até a posição 4, onde vai de 1 á 5, se colocarmos os numeros como o numero das posições ficam assim:

```
lista = [0,1,2,3,4,5,6,7,8,9]

for number in lista[:5]:
    print(str(number))

#Saida:
# 0
# 1
# 2
# 3
# 4
```

---

# Usando um For para outras coisas

---

Podemos utilizar um For para outras coisas além de Listas

## 1. somas consecutivas

Vamos por exemplo multiplicar quatro vezes por 2, para isso iremos construir um for que vai até 3, já que um **range()** sem definir um valor inicial começa no zero como abaixo:

```
range(3) # 0,1,2
```

Então o nosso For para calcular 4 vezes o 2 usando somas consecutivas fica assim:

```
value = 2

for i in range(3):
    value = value + value

print(value) #Saida: 16
```

dessa forma multiplicamos 4 vezes o 2 (222\*2) utilizando somas consecutivas

```
2 + 2 = 4
4 + 4 = 8
8 + 8 = 16
```

## 2. somando +1 por um determinado tempo

podemos colocar o valor em uma variável e chamar ele dentro do **range()**

```
maximo = 10
valor = 0

for i in range(maximo):
    valor = valor + 1
print(valor)
```

---