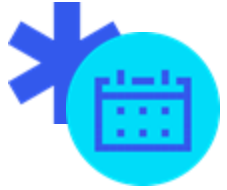




## Estrutura de Repetição FOR

< [thefutureisblue.me](https://thefutureisblue.me) />





# Agenda

O que vamos aprender?

Estrutura de Repetição FOR



# Relembrando variáveis?

\* **Variáveis** são pequenos espaços de memória, utilizados para armazenar e manipular dados.

Exemplo:

```
b = 10;
```

**inteiro**

```
e = 1.3;
```

**float**

```
t = "Olá alunos!!!"
```

**tipo String**

```
nome = input('Digite seu nome: ')
```

```
idade = input('Digite sua idade: ')
```

```
altura = input('Digite sua altura: ')
```

**tipo**

**'Fulano da Silva Sauro'**

**29**

**tipo**

**1.56**

# Relembrando

## Estrutura de controle?

\* O comando `if` é utilizado quando precisamos decidir se um trecho do programa deve ou não ser executado.

Exemplo:

```
idade = int(input("Qual sua idade?"))  
    if idade < 18:  
        print('Você ainda não pode começar as aulas de direção')
```



# Relembrando condicionais?

\* O comando `if/else` um trecho de código será executado se a condição for verdadeira e outro se a condição for falsa.

Exemplo:

```
idade = input("Qual sua idade?")
    if idade < 18:
        print('Você ainda não pode começar as aulas de direção')
    else
        print('Você já pode começar a tirar sua habilitação, parabéns!')
```

# Relembrando condicionais?

\* O comando `if/elif/else` um trecho de código será executado se a condição for verdadeira e outro se a condição for falsa.

Exemplo:

```
idade = input("Qual sua idade?")
    if idade < 15:
print('Você ainda não pode começar as aulas de direção')
    elif idade >= 18:
print('Você já pode começar a tirar sua habilitação, parabéns!')
    elif idade >60
print('Você está na flor da idade!')
    else
print('Está quase chegando nos 18!!!!')
```



# Estrutura de repetição: **FOR**

\* Um laço **FOR**, na linguagem Python, apresenta diferenças de iteração em relação a outras linguagens. Ele não itera sobre os índices de uma sequência e sim sobre os próprios elementos da sequência:

```
for <elemento> in <sequencia>:  
    <indentação> <comando> = 0
```



# Estrutura de repetição: **FOR**

\* Exemplo:

```
for a in range(5):  
    print(a)
```

# Saída  
0, 1, 2, 3, 4





# Estrutura de repetição: **FOR**

\* Exemplo:

```
for a in range(5, 10):  
    print(a)
```

# Saída  
5, 6, 7, 8, 9



# Estrutura de repetição: **FOR**

\* Exemplo:

```
lista = [1, 2, "blue", "red"]  
for a in lista:  
    print(a)
```

```
# Saída  
1, 2, "blue", "red"
```



# Estrutura de repetição: **FOR**

\* Exemplo:

```
variavel = "blue"  
for a in variavel:  
    print(a)
```

```
# Saída  
"b", "l" , "u", "e"
```



# Estrutura de repetição: **FOR**

Além disso, existem funções que podem ser utilizadas para limitar a execução do laço de repetição for, como a função **range(m, n, p)** (utilizada anteriormente), que cria uma lista de inteiros começando em m e terminando em n - 1 sendo incrementada de p em p; e a função **len()** que retorna o tamanho de determinado vetor.

\* Exemplo:

m = -1

n = +1

p = 1

```
for i in range(m, n, p):  
    print(i)
```

//Saída

-1, 0



# Estrutura de repetição: **FOR**

Geralmente, a função **len()** é utilizada quando se quer percorrer uma lista, de tal forma que o valor da variável iteradora não assuma cada valor da lista e sim cada posição da lista. Ou seja, não importa o conteúdo de cada posição da lista, a variável iteradora irá assumir o valor de cada posição. O exemplo abaixo exemplifica isso.

\* Exemplo:

```
x = ['Blue', 'é', 'nota', 1000]
```

```
for i in range(len(x)):
```

```
    print(i)
```

```
for i in x:
```

```
    print(i)
```





# Botando para rodar!

Vamos praticar todos operadores conceituados!

# Por hoje é só! Obrigado! =)

Até a próxima aula.