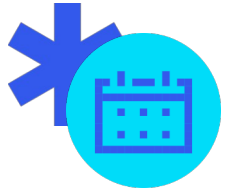




Listas

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





Agenda

O que vamos aprender?

Revisão

Listas



Relembrando variáveis?

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

Exemplo:

b = 10;	tipo inteiro
e = 1.3;	tipo float
t = "Olá alunos!!!"	tipo String

```
nome = input('Digite seu nome: ')
idade = input('Digite sua idade: ')
altura = input('Digite sua altura: ')
'Fulano da Silva Sauro'
29
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!!!!')
```





Listas

* Lista é um conjunto sequencial de valores, onde cada valor é identificado através de um índice. O primeiro valor tem índice 0. Uma lista em Python é declarada da seguinte forma:

Nome_Lista = [valor1, valor2, ..., valorN]

* Uma lista pode ter valores de qualquer tipo, incluindo outras listas.

Exemplo:

```
B = [7 , 'Blue' , 9.6 , [6,7,8] , 'Python' , (3 , 'j')]
```

```
print(B[2])
```

```
9.6
```

```
print(B[3])
```

```
[6,7,8]
```

```
print(B[3][1])
```

```
7
```





Listas

* Para alterar um elemento da lista, basta fazer uma atribuição de valor através do índice. O valor existente será substituído pelo novo valor.

*Exemplo:

```
L[3] = 'EdTech'  
print(L)
```

```
# Saída: B = [7 , 'Blue' , 9.6 , 'EdTech' , 'Python' , (3 , 'j')]
```





Função para manipulação de Listas

* A lista é uma estrutura mutável, ou seja, ela pode ser modificada. Na tabela a seguir estão algumas funções utilizadas para manipular listas.

Função	Descrição	Exemplo
len	retorna o tamanho da lista.	L = [1, 2, 3, 4] len(L) → 4
min	retorna o menor valor da lista.	L = [10, 40, 30, 20] min(L) → 10
max	retorna o maior valor da lista.	L = [10, 40, 30, 20] max(L) → 40
sum	retorna soma dos elementos da lista.	L = [10, 20, 30] sum(L) → 60
append	adiciona um novo valor na no final da lista.	L = [1, 2, 3] L.append(100) L → [1, 2, 3, 100]



Função para manipulação de Listas

Função	Descrição	Exemplo
extend	insere uma lista no final de outra lista.	<pre>L = [0, 1, 2] L.extend([3, 4, 5]) L → [0, 1, 2, 3, 4, 5]</pre>
del	remove um elemento da lista, dado seu índice.	<pre>L = [1,2,3,4] del L[1] L → [1, 3, 4]</pre>
in	verifica se um valor pertence à lista.	<pre>L = [1, 2 , 3, 4] 3 in L → True</pre>
sort()	ordena em ordem crescente	<pre>L = [3, 5, 2, 4, 1, 0] L.sort() L → [0, 1, 2, 3, 4, 5]</pre>
reverse()	inverte os elementos de uma lista	<pre>L = [0, 1, 2, 3, 4, 5] L.reverse() L → [5, 4, 3, 2, 1, 0]</pre>



Operações Listas

* Concatenação (+)

```
a = [0, 1, 2]
```

```
b = [3, 4, 5]
```

```
c = a + b
```

```
print(c)
```

```
# Saída: [0, 1, 2, 3, 4, 5]
```

* Repetição (*)

```
L = [1, 2]
```

```
R = L * 4
```

```
print(R)
```

```
# Saída: [1, 2, 1, 2, 1, 2, 1, 2]
```



Fatiamento em Listas



***O fatiamento de listas é semelhante ao fatiamento de strings.**

***Exemplo**

```
B = [7, 'Blue' , 9.6 , [6, 7, 8], 'Python', (3 , 'j')]
```

`B[1:4]` → seleciona os elementos das posições 1, 2, 3

```
# Saída: ['Blue', 9.6, [6, 7, 8]]
```

`L[2:]` → seleciona os elementos a partir da posição 2

```
# Saída: [9.6, [6, 7, 8], 'Python', (3, 'j')]
```

`L[:4]` → seleciona os elementos até a posição 3

```
# Saída: [7, 'Blue', 9.6, [6, 7, 8]]
```

Criação de listas com range ()



***A função range() define um intervalo de valores inteiros. Associada a list(), cria uma lista com os valores do intervalo.**

***A função range() pode ter de 1 a 3 parâmetros:**

- **range(n) → gera um intervalo de 0 a n-1**
- **range(i , n) → gera um intervalo de i a n-1**
- **range(i , n, p) → gera um intervalo de i a n-1 com intervalo p entre os números**

Exemplos:

```
L1 = list(range(5))
```

```
print(L1)
```

```
# Saída: [0, 1, 2, 3, 4]
```

```
L2 = list(range(3, 8))
```

```
print(L2)
```

```
# Saída: [3, 4, 5, 6, 7]
```

```
L3 = list(range(2, 11, 3))
```

```
print(L3)
```

```
# Saída: [2, 5, 8]
```



Botando para rodar!

**Vamos praticar todos
operadores conceituados!**

**Por hoje é
só!
Obrigado!
=)**

Até a próxima aula.