

# Listas

- Listas são estruturas de dados capazes de armazenar múltiplos elementos.

## Declaração:

Para a criação de uma lista, basta colocar os elementos separados por vírgulas dentro de colchetes [], como no exemplo abaixo:

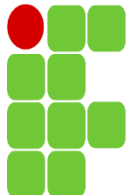
```
>>> nomes_frutas = ["maçã", "banana", "abacaxi"]
>>> nomes_frutas
['maçã', 'banana', 'abacaxi']

>>> numeros = [2, 13, 17, 47]
>>> numeros
[2, 13, 17, 47]
```

A lista pode conter elementos de tipos diferentes:

```
>>> ['lorem ipsum', 150, 1.3, [-1, -2]]
['lorem ipsum', 150, 1.3, [-1, -2]]

>>> vazia = []
>>> vazia
[]
```



# Listas (Índices)

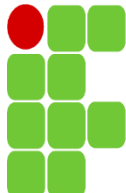
- Assim como nas strings, é possível acessar separadamente cada item de uma lista a partir de seu índice:

```
>>> lista = [100, 200, 300, 400, 500]
>>> lista[0]  # os índices sempre começam em 0
100

>>> lista[2]
300

>>> lista[4]  # último elemento
500

>>> lista[-1] # outra maneira de acessar o último elemento
500
```



# Listas (Índices)

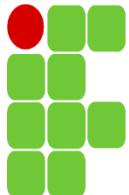
- Conforme visto anteriormente, ao utilizar um índice negativo os elementos são acessados de trás pra frente, a partir do final da lista:

```
>>> lista[-2]  # penúltimo elemento
400

>>> lista[-3]  # terceiro
300

>>> lista[-4]  # segundo
200

>>> lista[-5]  # primeiro
100
```



# Listas (Índices)

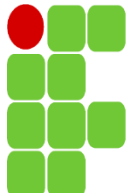
- Ou pode-se acessar através de slices (fatias), como nas strings

```
>>> lista[2:4]  # da posição 2 até a 4 (não inclusa)
[300, 400]

>>> lista[:3]   # até a posição 3 (não incluso)
[100, 200, 300]

>>> lista[2:]   # da posição 2 até o final
[300, 400, 500]

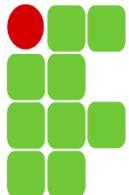
>>> lista[:]    # do começo até o final
[100, 200, 300, 400, 500]
```



# Listas (Índices)

- Podemos avaliar se os elementos estão na lista com a palavra **in**:

```
>>> lista_estranha = ['duas palavras', 42, True, ['batman', 'robin'], -0.84, 'hipófise',  
↳ '']  
>>> 42 in lista_estranha  
True  
  
>>> 'duas palavras' in lista_estranha  
True  
  
>>> 'dominó' in lista_estranha  
False  
  
>>> 'batman' in lista_estranha[3] # note que o elemento com índice 3 também é uma_  
↳ lista  
True
```



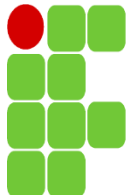
# Listas (Índices)

- É possível obter o tamanho da lista utilizando o método `len()`:

```
>>> len(lista)
5

>>> len(lista_estranha)
6

>>> len(lista_estranha[3])
2
```



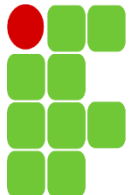
# Removendo itens da lista

- Devido à lista ser uma estrutura mutável, é possível remover seus elementos utilizando o comando **del**:

```
>>> lista_estranha
['duas palavras', 42, True, ['batman', 'robin'], -0.84, 'hipófise']

>>> del lista_estranha[2]
>>> lista_estranha
['duas palavras', 42, ['batman', 'robin'], -0.84, 'hipófise']

>>> del lista_estranha[-1]  # Remove o último elemento da lista
>>> lista_estranha
['duas palavras', 42, ['batman', 'robin'], -0.84]
```

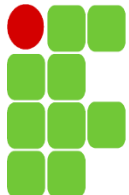


# Trabalhando com listas

O operador + concatena listas:

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> c = a + b
>>> c
[1, 2, 3, 4, 5, 6]
```

O operador \* repete a lista dado um número de vezes:



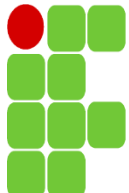


# Trabalhando com listas

O método `append()` adiciona um elemento ao final da lista:

```
>>> lista = ['a', 'b', 'c']
>>> lista
['a', 'b', 'c']

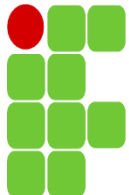
>>> lista.append('e')
>>> lista
['a', 'b', 'c', 'e']
```



# Trabalhando com listas

Temos também o `insert()`, que insere um elemento na posição especificada e move os demais elementos para direita:

```
>>> lista.insert(3, 'd')
>>> lista
['a', 'b', 'c', 'd', 'e']
```



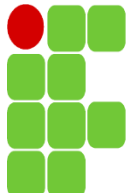
# Trabalhando com listas

`extend()` recebe uma lista como argumento e adiciona todos seus elementos a outra:

```
>>> lista1 = ['a', 'b', 'c']
>>> lista2 = ['d', 'e']
>>> lista1
['a', 'b', 'c']

>>> lista2
['d', 'e']

>>> lista1.extend(lista2)
>>> lista1
['a', 'b', 'c', 'd', 'e']
```

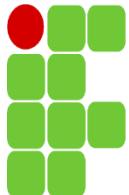


# Trabalhando com listas

O método `sort()` ordena os elementos da lista em ordem ascendente:

```
>>> lista_desordenada = ['b', 'z', 'k', 'a', 'h']
>>> lista_desordenada
['b', 'z', 'k', 'a', 'h']

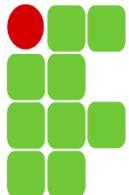
>>> lista_desordenada.sort()
>>> lista_desordenada # Agora está ordenada!
['a', 'b', 'h', 'k', 'z']
```



# Trabalhando com listas

Para fazer uma cópia de uma lista, devemos usar o método `copy()`:

```
>>> lista1 = ['a', 'b', 'c']
>>> lista2 = lista1.copy()
>>> lista1
['a', 'b', 'c']
>>> lista2
['a', 'b', 'c']
>>> lista2.append('d')
>>> lista1
['a', 'b', 'c']
>>> lista2
['a', 'b', 'c', 'd']
```



F I M