
Ricardo Sérgio
Introdução à Programação e Algoritmia
API B – 12º Ano
MacBook Pro

Python

09 {

#Arrays

}

Ricardo Sérgio

Arrays{

- Em Python não existem os arrays tradicionais.
- Para serem utilizados é necessário a importação de uma biblioteca do género NumPy
- Existem porém listas que são em tudo semelhantes aos arrays e que podem ser utilizadas diretamente (nativas)
- Mas afinal o que são arrays ou listas????

}

Arrays{

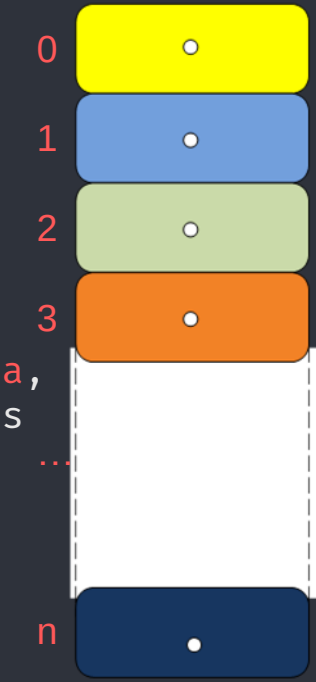
- Arrays ou listas são variáveis multidimensionais
- Não seria bom podermos guardar os nomes de todos os alunos de uma turma numa só variável?
- Imaginem as linhas de código que pouparíamos!
- Bem é isso mesmo que se consegue utilizando arrays (neste caso listas)

}

Listas{

- Imaginem a nossa variável do tipo lista como um armário com gavetas.
- Na primeira podemos ter meias, na segunda camisolas, na terceira calças, etc
- Relativamente ao conteúdo, em Python podemos ter mistura, i.e. podemos ter strings numa "gaveta" e números noutras
- Em Python as listas não necessitam de ser previamente dimensionados.
- A primeira posição da lista é a zero.

}



Listas{

- Sintaxe:

- Com conteúdo:

```
nomeLista = [conteudo1, conteudo2,…,conteudoN]
```

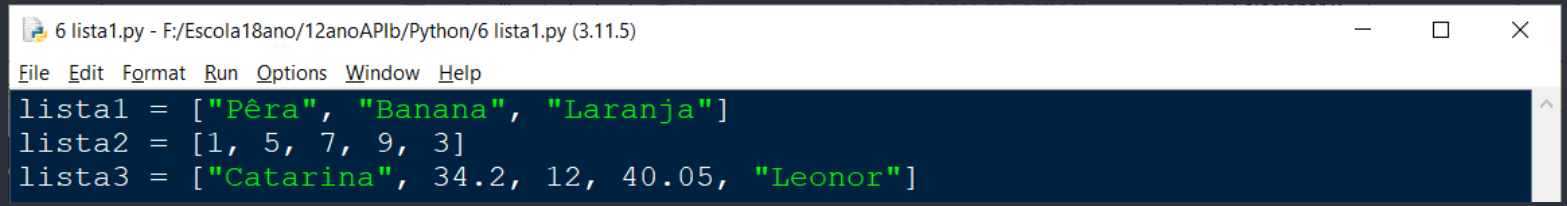
- Vazia:

```
nomeLista = []
```

}

Listas{

- Exemplo:



- A lista 1 apenas com strings
- A lista 2 apenas com inteiros
- A lista 3 com strings, inteiros e floats

}

Listas{

Imprimir

6 lista1.py - F:/Escola18ano/12anoAPIb/Python/6 lista1.py (3.11.5)

File Edit Format Run Options Window Help

```
lista1 = ["Pêra", "Banana", "Laranja"]
lista2 = [1, 5, 7, 9, 3]
lista3 = ["Catarina", 34.2, 12, 40.05, "Leonor"]
print(lista1)
print(lista2)
print(lista3)
```

Resultado

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
>>> ========== RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista1.py ==========
['Pêra', 'Banana', 'Laranja']
[1, 5, 7, 9, 3]
['Catarina', 34.2, 12, 40.05, 'Leonor']
>>>
```

Ln: 20 Col: 0

}

Listas{

- A forma de imprimir que nos interessa mais é a de imprimir apenas o conteúdo das posições da lista
- Ou seja saber o que está dentro de cada "gaveta"
- Como fazer? A partir das posições numéricas.

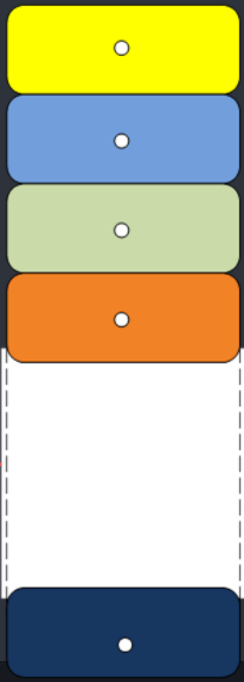
```
print(lista[nº posição])
```

6 lista1.py - F:/Escola18ano/12anoAPIb/Python/6 lista1.py (3.11.5)

File Edit Format Run Options Window Help

```
lista1 = ["Pêra", "Laranja", "Banana"]
print(lista1[2])
```

}



Listas{

Exemplo:

6 lista1.py - F:/Escola18ano/12anoAPIb/Python/6 lista1.py (3.11.5)

File Edit Format Run Options Window Help

```
lista1 = ["Pêra", "Laranja", "Banana"]
lista2 = [1, 5, 7, 9, 3]
lista3 = ["Catarina", 34.2, 12, 40.05, "Leonor"]
print(lista1[1])
print(lista2[3])
print(lista3[0])
```

Resultado?

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
>>> 
========== RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista1.py ==========
Laranja
9
Catarina
>>> 
```

Ln: 30 Col: 0

}

Listas{

- Como substituir valores em listas?
- Atribuir à posição em questão um novo valor:

6 lista1.py - F:\Escola18ano\12anoAPIb\Python\6 lista1.py (3.11.5)

File Edit Format Run Options Window Help

```
lista = ["Pêra", "Laranja", "Banana"]
lista[1]="Tangerina"
print(lista)
```

- Resultado:

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
>>> ========== RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista 2.py ==========
['Pêra', 'Tangerina', 'Banana']
>>>
```

Ln: 18 Col: 0

}

Listas{

- Se eu pretender adicionar um novo elemento à lista no decorrer do código eu posso fazê-lo da seguinte forma:

- Fim da lista:

```
nomeLista.append(valor_a_adicionar)
```

- Posição específica da lista:

```
nomeLista.insert(posição, valor_a_adicionar)
```

}

Listas{

Exemplo:

6 lista 2.py - F:/Escola18ano/12anoAPIb/Python/6 lista 2.py (3.11.5)

File Edit Format Run Options Window Help

```
lista = ["Pêra", "Laranja", "Banana"]
lista.append("Tangerina")
print(lista)
```

Resultado:

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
>>> ========== RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista 2.py ==========
['Pêra', 'Laranja', 'Banana', 'Tangerina']
>>>
```

Ln: 12 Col: 0

}

Listas{

Exemplo:

6 lista 2.py - F:/Escola18ano/12anoAPIb/Python/6 lista 2.py (3.11.5)

File Edit Format Run Options Window Help

```
lista = ["Pêra", "Laranja", "Banana"]
lista.insert(1,"Tangerina")
print(lista)
```

Resultado:

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
>>> ========== RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista 2.py ==========
['Pêra', 'Tangerina', 'Laranja', 'Banana']
>>>
```

Ln: 15 Col: 0

}

Listas{

- Se eu pretender remover um elemento da lista no decorrer do código eu posso fazê-lo da seguinte forma:

- Fim da lista:

```
nomeLista.pop()
```

- Posição específica da lista:

```
nomeLista.pop(posição a remover)
```

}

Listas{

Exemplo:

6 lista 2.py - F:/Escola18ano/12anoAPIb/Python/6 lista 2.py (3.11.5)

File Edit Format Run Options Window Help

```
lista = ["Pêra", "Laranja", "Banana"]
lista.pop()
print(lista)
```

Resultado:

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
>>> ========== RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista 2.py ==========
['Pêra', 'Laranja']
>>>
```

Ln: 24 Col: 0

}

Listas{

Exemplo:

6 lista 2.py - F:/Escola18ano/12anoAPIb/Python/6 lista 2.py (3.11.5)

File Edit Format Run Options Window Help

```
lista = ["Pêra", "Laranja", "Banana"]
lista.pop(1)
print(lista)
```

Resultado:

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
>>> ============= RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista 2.py ============
['Pêra', 'Banana']
>>>
```

Ln: 21 Col: 0

}

Listas{

- Uma vez que podemos aceder ao conteúdo das listas através de números os ciclos encaixariam aqui como uma luva

- Podemos fazer o ciclo percorrer as posições da lista através do contador

6 lista1.py - F:/Escola18ano/12anoAPIb/Python/6 lista1.py (3.11.5)

File Edit Format Run Options Window Help

```
lista = ["Catarina", 34.2, 12, 40.05, "Leonor"]
for i in range(len(lista)):
  print(lista[i])
```

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
>>> 
========== RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista1.py ==========
Catarina
34.2
12
40.05
Leonor
>>> 
```

Ln: 43 Col: 0

}

Listas{

- No caso anterior utilizamos len(lista) para saber o tamanho da lista e assim parar o ciclo
- Podemos fazer o ciclo parar em qualquer posição, bem como iniciar

6 lista1.py - F:/Escola18ano/12anoAPIb/Python/6 lista1.py (3.11.5)

File Edit Format Run Options Window Help

```
lista = ["Catarina", 34.2, 12, 40.05, "Leonor"]
for i in range(2,4):
  print(lista[i])
```

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
========== RESTART: F:/Escola18ano/12anoAPIb/Python/6 lista1.py ==========
12
40.05
>>>
```

Ln: 62 Col: 0

}

```
Para Hoje{

}
```

Realiza os guiões práticos existentes no Website da tua disciplina

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**