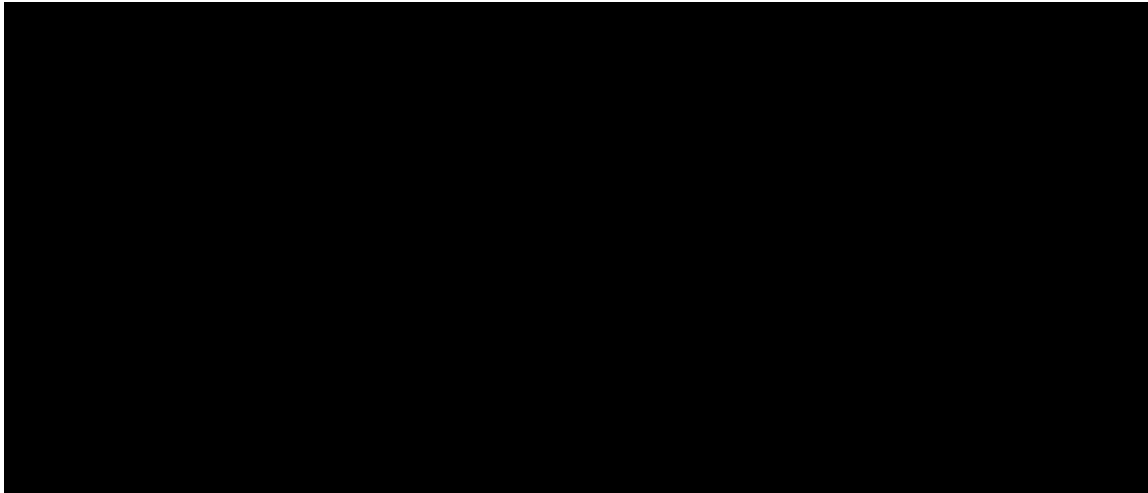


## LISTAS E TUPLAS

### TUPLAS

Neste vídeo, Allan apresenta a manipulação de dados em sequências através de Tuplas.



04:35

### CRIAÇÃO DE TUPLAS

Tuplas são uma estrutura básica de sequência mais conservadora, que não permite alterações dos seus elementos após criada. Por esse motivo, é chamada de sequência imutável ou estática.

Assim como as listas podemos criar as tuplas de duas maneiras, chamando o construtor tuple ou com **operador parênteses** em vez de colchetes. Vamos ver na prática abaixo:

```
x = tuple([1, 2, 3, 4, 5, 6])
type(x)
tuple
y = (1, 2, 3)
type(y)
tuple
```

## LISTAS E TUPLAS

como uma tupla.

```
x = 1, 2, 3  
type(x)  
tuple
```

Esse comportamento é complementado pela função de **desempacotamento de variáveis**, que armazena em um grupo de variáveis os valores de uma tupla caso essa tenha o mesmo número de valores em si.

Imagine a seguinte tupla:

```
minha_tupla = (1,2,3)
```

Como essa tupla tem três elementos, precisamos de três variáveis para **desempacotá-la**. Veja abaixo:

```
x,y,z = minha_tupla
```

Com o comando anterior agora a variável x tem o valor 1, a variável y tem o valor 2 e z tem o valor 3

## TUPLAS VS LISTAS

A principal **vantagem** da tupla sobre a lista é que, por ser uma estrutura estática, é muito mais simples de armazenar em memória, gastando menos espaço. A lista precisa de muito mais variáveis para ser eficiente, além de ter métodos específicos de edição.

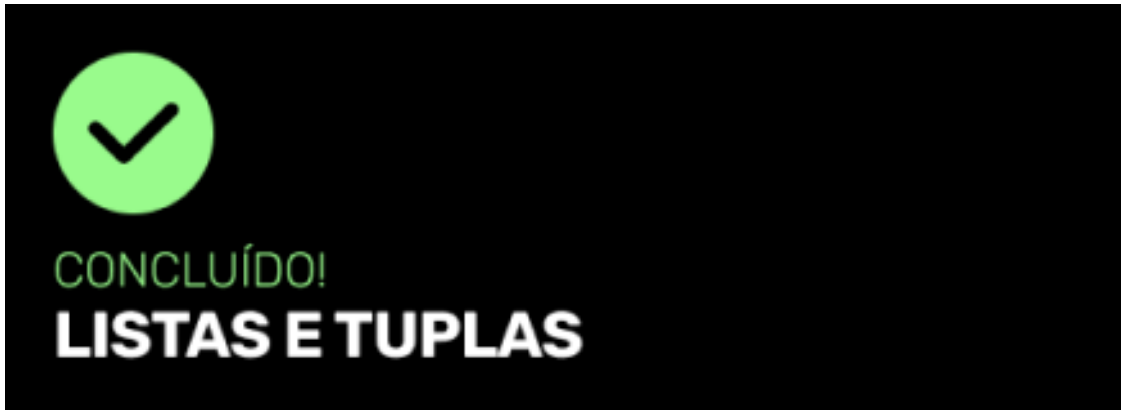
A principal **desvantagem** da tupla é que toda vez que se quer alterar, é necessário construir uma nova tupla com a modificação. Isso pode pesar na memória e tornar qualquer programa ineficiente muito rápido, principalmente quando grandes quantidades de dados estão na tupla.

LISTAS E TUPLAS



## LISTAS E TUPLAS

### CONCLUSÃO



Parabéns! Que bom que você chegou até aqui. **Agora você já conhece as Listas e Tuplas no Python.** Revise os aprendizados desta aula e avalie se você já tem segurança para realizar essas ações.

### CHECKLIST

Ao final desta aula, você deve sentir segurança para:

- Criar lista através do construtor: **list()**
- Criar lista através de variável = **[ ]**
- **Adicionar elementos** numa lista
- Acessar os elementos de uma lista pela **posição**. Ex: lista[0]
- Acessar porções da lista através do **Slice**. Ex: lista[0:3]
- Realizar **operações** com uma lista
- Criar uma **lista Multidimensional**
- Compreender as **vantagens e desvantagens** entre listas e tuplas

## LISTAS E TUPLAS

- **Desempacotar tuplas** em variáveis  $a, b, c = (1, 2, 3)$

Caso ainda tenha dúvidas, não hesite em voltar aos conteúdos e buscar ajuda, esses fundamentos são super importantes para você construir uma base de trabalho sólida. Esta aula ficará disponível para você rever quando quiser.

[Notebook utilizado no vídeo e no texto](#)

Até breve!

---

CONCLUIR