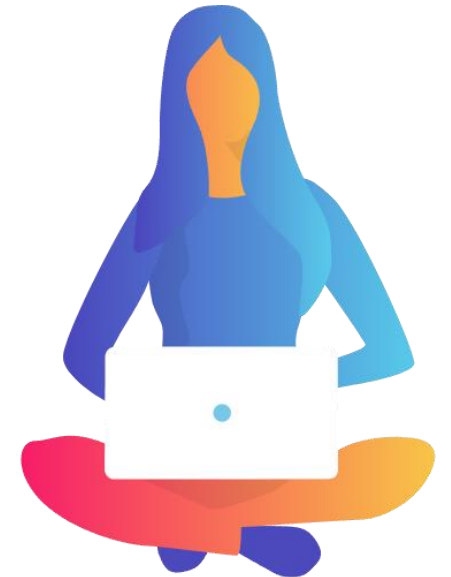


# Laboratório de Programação

## Aula 06 - Tuplas

Uma tupla (tuple), como uma lista, é um sequência de items de qualquer tipo. Entretanto, diferentemente de listas, tuplas são **imutáveis**. Sintaticamente, uma tupla é uma sequência de valores separadas por uma vírgula. Apesar de não ser necessário, há a convenção de se envolver uma tupla entre parênteses.



# Fatiamento (ou Slicing) de Tuplas

## Exemplos

```
meses = ('Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio', 'Junho',  
        'Julho', 'Agosto', 'Setembro', 'Outubro', 'Novembro', 'Dezembro')
```

```
print(meses)  
print(meses[1])  
print(meses[0:2]) #imprime de onde até onde?  
print(meses[0:14]) #e se extrapolar pra mais?  
print(meses[:2]) #os dois primeiros  
print(meses[2:]) #tudo menos os dois primeiros  
print(meses[-1]) #o último  
print(meses[-2]) #o penúltimo  
print(meses[-2:]) #os dois últimos  
print(meses[:-2]) #todos menos os dois últimos
```

- Criando Tupla vazia

```
tuplaVazia = ()
```

- Criando Tupla com somente um valor

```
tupla1 = (1,)
```

```
tupla2 = (1)
```

- Tamanho da Tupla: len

```
print (len (tuplaVazia))
```

```
print (len (tupla1))
```

```
print (len (tupla2))
```

- E se quiser manipular o valor da Tupla?

```
dados = (1, 2)
```

```
dados[1] = 7
```

Traceback (most recent call last):

File "tuplas.py", line 7, in <module>

`dados[1] = 7`

**TypeError: 'tuple' object does not support item assignment**

# Atribuição de tuplas

## Exemplos

```
pessoa = ("Fulano", "de Tal", 25)
```

```
nome = pessoa[0]
```

```
sobrenome = pessoa[1]
```

```
idade = pessoa[2]
```

---

```
pessoa = ("Fulano", "de Tal", 25)
```

```
(nome, sobrenome, idade) = pessoa
```

```
print(nome, sobrenome, idade)
```

**Saída: Fulano de Tal 25**

**Também funciona com  
listas :)**

# Atribuição de tuplas

## Exemplos

```
a = 1
```

```
b = 2
```

```
temp = a
```

```
a = b
```

```
b = temp
```

```
a = 1
```

```
b = 2
```

```
(a, b) = (b, a)
```

```
print(a, b)
```

**Saída: 2 1**



**Desafio!!!**





Faça uma função que receba uma lista de inteiros como parâmetro e retorne o número que mais aparece na lista, seguido da quantidade de vezes que esse número aparece.

Ex.:

```
lista = [1, 2, 3, 4, 5, 6, 1]
```

```
retorno da função: (1, 2)
```

# Tuplas como valor de retorno

## Exemplos

```
def numeroMaisFrequente(numeros):  
    numeroMaisFrequente = None  
    maiorFrequencia = 0  
    for numero in numeros:  
        frequencia = numeros.count(numero)  
        if frequencia > maiorFrequencia:  
            maiorFrequencia = frequencia  
            numeroMaisFrequente = numero  
  
    return (numeroMaisFrequente, maiorFrequencia)
```

```
listaNumeros = [1, 2, 3, 4, 5, 6, 1]  
(numeroMaisFrenquente, frequencia) = numeroMaisFrequente(listaNumeros)  
print(f"Número mais frequente: {numeroMaisFrenquente}, Frequência:  
{frequencia}")
```

# Tuplas com argumentos de comprimento variável

As funções podem receber um número variável de argumentos. Um nome de parâmetro que comece com `*` reúne vários argumentos em uma **tupla**.

Ex.:

```
def somaNumeros(*args):  
    soma = 0  
    for numero in args:  
        soma = soma + numero  
    return soma  
  
print(somaNumeros(1, 2, 3))
```

# Tuplas aninhadas e For

```
peessoas = (  
    ("João", 20),  
    ("Maria", 25),  
    ("James", 20)  
)
```

```
for nome, idade in pessoas:  
    print(f"Nome: {nome}, idade: {idade}")
```

**Agora testem  
com listas!**

# Laboratório de Programação

## Aula 06 - Tuplas