

# TUPLAS

Tuplas são semelhantes às listas, porém, são imutáveis.

Não podemos acrescentar, apagar ou fazer atribuições aos seus itens

As tuplas são criadas usando-se parênteses em vez de colchetes, porém, os parênteses não são obrigatórios

Veja a criação de tuplas com e sem parênteses

```
linguagens = ("Assembly", "Cobol", "C", "C++")  
print(linguagens)  
#Resultado: ('Assembly', 'Cobol', 'C', 'C++')
```

```
linguagens = "Python", "Java", "Go", "C#"  
print(linguagens)  
#Resultado: ('Python', 'Java', 'Go', 'C#')
```

# TUPLAS

É possível criar uma tupla vazia

```
Command Prompt - python
>>> vazio = ()
>>> vazio
()
>>> len(vazio)
0
>>> _
```

Criando uma tupla com um único elemento

```
Command Prompt - python
>>> tupla1 = (1)
>>> tupla1
1
>>> tupla1 = (1,)
>>> tupla1
(1,)
```

# ACESSANDO TUPLAS

Podemos acessar os elementos da tupla pelo índice e usar fatiamento

países

```
países = "Brasil", "Paraguai", "Uruguai", "México"  
pais = países[0]  
print(pais) # Brasil
```

```
fatia = países[1:3]  
print(fatia) # ('Paraguai', 'Uruguai')
```

# ALTERAR UMA TUPLA

Se tentarmos alterar um item da tupla, é gerado o erro "O objeto não suporta a atribuição de itens"

```
países = "Brasil", "Paraguai", "Uruguai", "México"  
países[1] = "Colômbia"
```

Traceback (most recent call last):

File ".../tupla.py", line 2, in <module>

países[1] = "Colômbia"

TypeError: 'tuple' object does not support item assignment

# IMPRIMIR TUPLAS

Podemos imprimir os elementos de uma tupla usando o comando for

```
países = "Brasil", "Paraguai", "Uruguai", "México"
```

```
for país in países:  
    print(país)
```

Como eu já disse, melhor não usar acentuação. Nem todas linguagens aceitam....


Resultado:  
Brasil  
Paraguai  
Uruguai  
México

# CONVERTENDO UMA LISTA EM UMA TUPLA

```
lista_carros = ["Gol", "Corolla", "Ranger", "Kadett",  
               "Fusca", "Clio"]  
tupla_carros = tuple(lista_carros)  
print(f"Tuplea carros: {tupla_carros}")
```

Resultado:

Tupla carros: ('Gol', 'Corolla', 'Ranger', 'Kadett', 'Fusca', 'Clio')



# CONVERTENDO UMA TUPLA EM UMA LISTA

```
tupla_carros = "Gol", "Corolla", "Ranger", "Kadett",  
               "Fusca", "Clio"  
lista_carros = list(tupla_carros)  
print(f"Lista carros: {lista_carros}")
```

Resultado:

Lista carros: ['Gol', 'Corolla', 'Ranger', 'Kadett', 'Fusca', 'Clio']



# DESEMPACOTANDO ELEMENTOS DA TUPLA

```
tupla_carros = "Golf", "Corolla", "Civic"  
carro1, carro2, carro3 = tupla_carros  
print(f"Carro1: {carro1}")  
print(f"Carro2: {carro2}")  
print(f"Carro3: {carro3}")
```

Resultado:

Carro1: Golf

Carro2: Corolla

Carro3: Civic



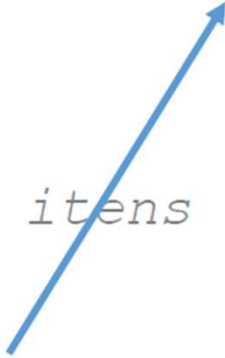
# DESEMPACOTANDO ELEMENTOS DA TUPLA USANDO ATRIBUIÇÃO MÚLTIPLA

```
tupla_carros = "Golf", "Corolla", "Civic", "Opala", \
               "Tucson", "Elantra"
carro1, *carros = tupla_carros
print(f"Carro1: {carro1}")
print(f"Carros: {carros}") # Uma lista com os itens restantes
```

Resultado:

Carro1: Golf

Carros: ['Corolla', 'Civic', 'Opala', 'Tucson', 'Elantra']



Barra invertida indica que a linha continua.

# ATRIBUIÇÃO MÚLTIPLA NÃO PRECISA ESTAR NO FIM DA SEQUÊNCIA

```
tupla_carros = "Golf", "Corolla", "Civic", "Opala", \
               "Tucson", "Elantra"
*carros, tucson, elantra = tupla_carros
print(f"Carros: {carros}")
print(f"Tucson: {tucson}")
print(f"Elantra: {elantra}")
```

Resultado:

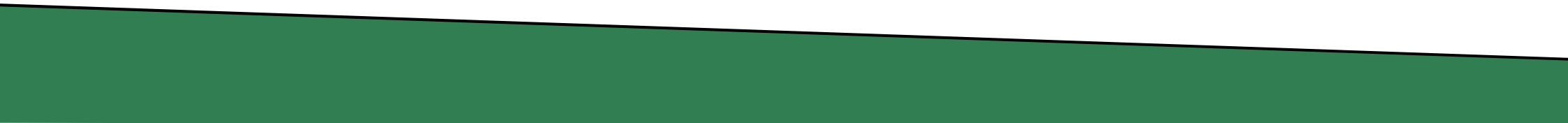
Carros: ['Golf', 'Corolla', 'Civic', 'Opala']

Tucson: Tucson

Elantra: Elantra

# EXERCÍCIO 1

Dada uma lista L de n valores inteiros, escreva um programa que remova os números pares da lista



## EXERCÍCIO 2

Dadas duas listas P1 e P2, ambas com  $n$  valores reais que representam as notas de uma turma na prova 1 e na prova 2, respectivamente, escreva um programa que calcule a média da turma nas provas 1 e 2, imprimindo em qual das provas a turma obteve a melhor média.