



**INSTITUTO FEDERAL**

Catarinense

Campus Camboriú

## AULA 3 (Tuplas e Conjuntos)

**Professora: Lidiane Visintin**

*lidiane.visintin@ifc.edu.br*

**Professor: Rafael de Moura Speroni**

*rafael.speroni@ifc.edu.br*

# Objetivo:



- Compreender o conceito de Tuplas.
- Compreender o conceito de Conjuntos.

# Tuplas

---

Vamos aos conceitos:

- **Tuplas** são uma estrutura de dados similar às listas, com a grande diferença de serem imutáveis.

```
tupla = (1, 2, 3, 4, 5)
```

## Tupla vs Listas

```
lista = [1, 2, 3, 4, 5]  
tupla = (1, 2, 3, 4, 5)
```

# Tuplas

Em Python, criamos tuplas utilizando parênteses ( );

- Como ficaria no Python:

```
t = ("a", "b", "c")
```

- Os parênteses são opcionais

```
t = "a", "b", "c"
```

irá exibir o mesmo resultado;

Tuplas **suportam a maior parte das operações de listas**, como fatiamento e indexação;

# Como declarar uma tupla?

```
t = ()
```

- Este comando cria uma tupla denominado de **t** e os parenteses ( ) após o símbolo de igualdade indicam que é uma tupla vazia;

```
t = "a", "b", "c"
```

- Este comando cria uma tupla com 3 elementos (**ato de empacotar**);  
**Quando criamos tuplas com um único elemento é preciso ter cuidado:**

```
t1 = (1)
```

```
t2 = (2,)
```

```
t3 = 1,
```

Neste exemplo somente t2 e t3 são tuplas.

# Como alterar um elemento de uma tupla?

---

```
t = "a", "b", "c"
```

Tuplas **não** podem ter seus valores alterados, mas suporta cálculos:

```
print(t * 2)
```

Gera como resultado:

```
('a', 'b', 'c', 'a', 'b', 'c')
```

# Encontrando um elemento de uma tupla?

---

- Além de consultar utilizando o índice;
- Podemos utilizar o **for**:

```
for elemento in t:  
    print(elemento)
```

# Ato de desempacotar?

- Podemos usar o \* para indicar vários valores a desempacotar:

```
*a, b = (1, 2, 3, 4, 5)
```

- No caso dizemos coloque o último valor em b e os demais valores em a.
- O conteúdo de a será igual à [1, 2, 3, 4], enquanto o de b será 5.
- Com o comando: `tuple(a)`, teremos a convertido para uma tupla e não mais uma lista.

**Desempacota como lista.**



# Conjuntos(set)

---

Vamos aos conceitos:

- **Conjuntos** são uma estrutura de dados que implementa operações de união, intersecção, diferença, entre outros.
- A principal diferença é **não admitir a repetição de elementos.**
- **Não** mantém a ordem dos elementos.

# Conjuntos

Em Python, criamos tuplas utilizando parênteses `set( )`;

- Como ficaria no Python:

```
C = set()
```

- Podemos adicionar valores com o método `add()`:

```
C.add(1)
```

```
C.add(2)
```

```
C.add(8)
```

- Ao exibirmos o resultado, teremos:

**{8, 1, 2}**

Mesmo o 8 sendo o último elemento a ser inserido no conjunto ele aparece na primeira posição ;

# Conjuntos - adicionando

---

- Adicionando mais valores:

```
C.add(-1)
```

```
C.add(0)
```

- Ao exibirmos o resultado, teremos:  
**{0, 1, 2, 8, -1}**

# Conjuntos

---

- Podemos utilizar o operador in para pesquisar em um conjunto:

```
print(1 in C) resultado será TRUE
```

- Um set(conjunto) pode ser criado a partir de listas, tuplas e qualquer outra estrutura que seja enumerável.

```
b = set([2, 3, 6, 12])
```

# Conjuntos - operações

```
b = {2, 3, 6, 12}
```

```
a = {1, 2, 6, 28}
```

Considerando os dois conjuntos **a** e **b**

```
print(a - b) resulta em: {1, 28}
```

```
print(a | b) resulta em: {1, 2, 3, 6, 12, 28}
```

```
print(a & b) resulta em: {2, 6}
```

- representa a diferença entre conjuntos;
- | representa a união entre conjuntos; e
- & representa a intersecção entre conjuntos;

# Referências



## Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. Pearson Prentice Hall. 2005

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de.. Algoritmos: lógica para desenvolvimento de programação de computadores.. 27. ed.. Érica. 2014

## Referências Complementares

DOWNEY, Allen B. **Pense em Python**. 2ª Ed. Novatec. 2016

MENEZES, Nilo Ney de Coutinho. **Introdução a programação com Python**. 3ª Ed. Novatec. 2019

CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. 2. ed. Elsevier, Campus,. 2002

## Referências na Internet

<https://docs.python.org/3/>

<https://www.w3schools.com/python/default.asp>