

# collections.namedtuple

Biblioteca padrão #1



**Nome:**

Eduardo Mendes

**Email:**

mendesxeduardo@gmail.com

**GIT:**

github.com/z4r4tu5tr4

---

# O que é uma tupla em python?

- Sequência imutável
  - Não aceita resignação de novos itens após sua definição

```
In: t = (1,2,3,4)
```

```
In: t[0] = 5
```

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

# Tuplas são acessíveis via index/slice

**In:** `t = (1, 2, 3, 4)`

**In:** `t[0]`

**Out:** `1`

**In:** `t = (1, 2, 3, 4)`

**In:** `t[0:3]`

**Out:** `(1, 2, 3)`

**In:** `t = (1, 2, 3, 4)`

**In:** `t[::2]`

**Out:** `(1, 3)`

# Tuplas executam dois métodos

```
In: t = (1, 2, 3, 4)
```

```
In: t.count(1)
```

```
Out: 1
```

```
In: t = (1, 2, 3, 4)
```

```
In: t.index(3)
```

```
Out: 2
```



# Então, o que é uma namedtuple?

- Sequência imutável
  - Não aceita resignação de novos itens após sua definição
  - Os elementos estão acessíveis como atributos

```
In: n_t = namedtuple('jogador', ['nome', 'time', 'camisa'])
```

```
In: n_t('Ronaldo', 'Brasil', 9)
```

```
Out: jogador(nome='Ronaldo', time='Brasil', camisa=9)
```

# São acessíveis via index/slice/atributo

```
In: j = n_t('Ronaldo', 'Brasil', 9)
```

```
In: j[0]
```

```
Out: 'Ronaldo'
```

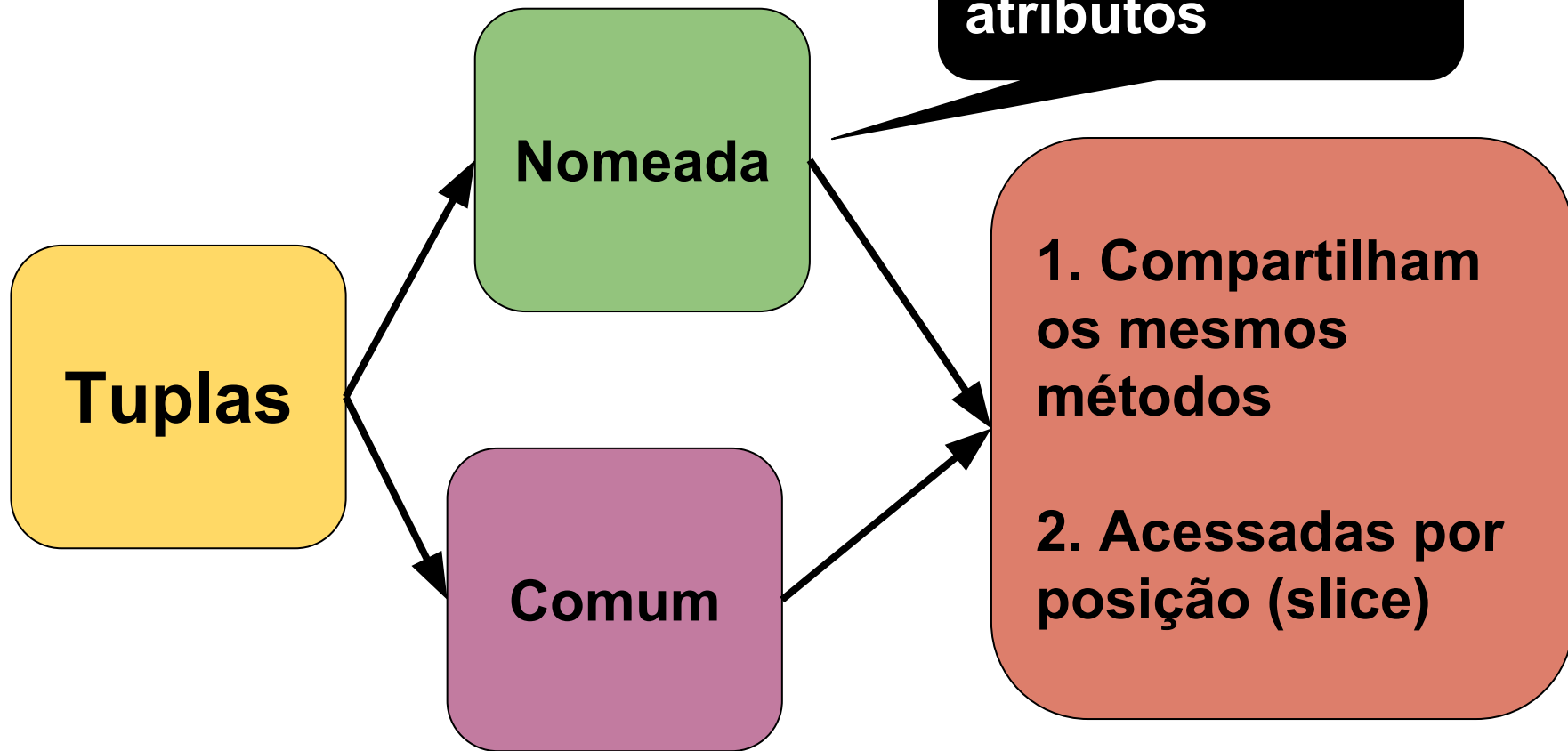
```
In: j[::2]
```

```
Out: ('Ronaldo', 9)
```

```
In: j.nome
```

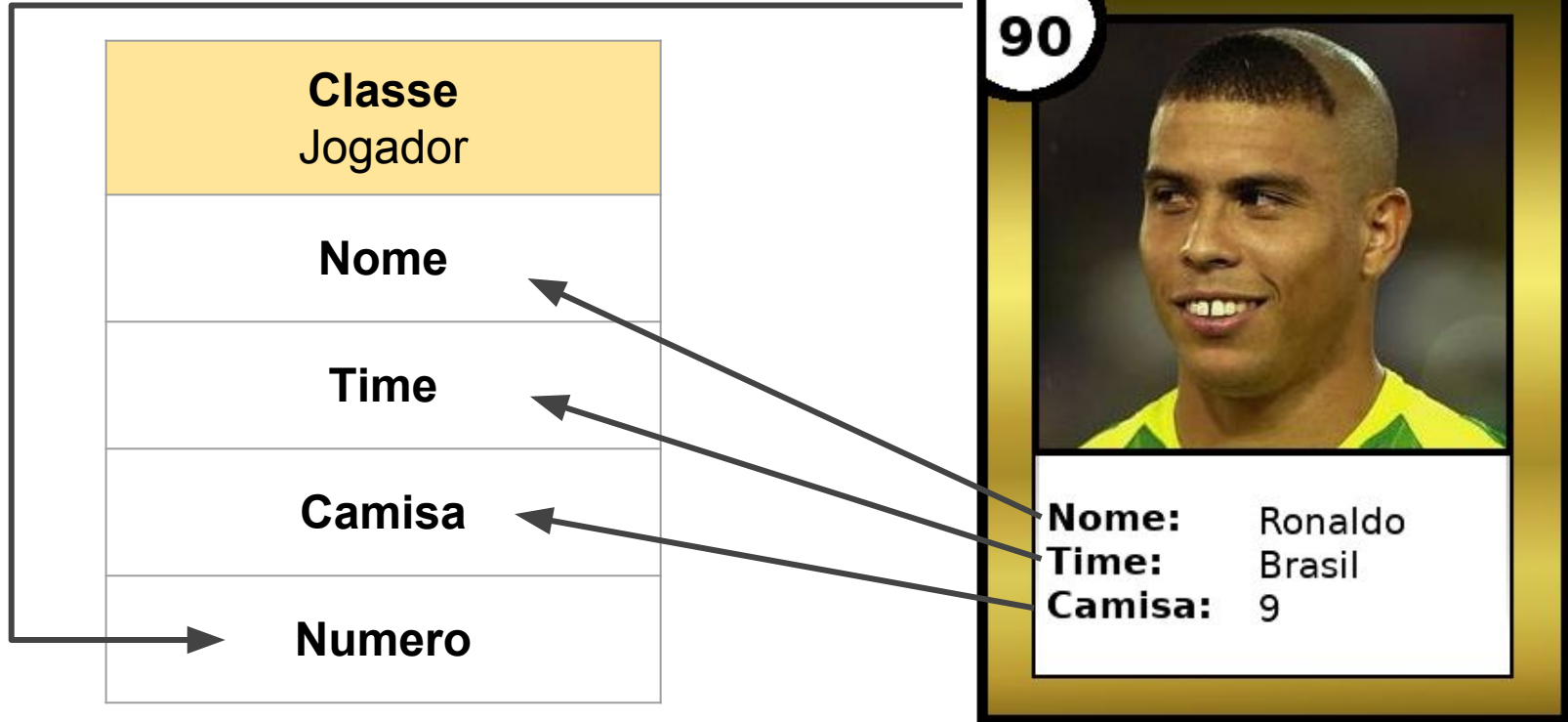
```
Out: 'Ronaldo'
```

# Tupla x namedtuple





# Exemplo: Figurinhas da copa [0]



# Inicializando uma namedtuple

**`namedtuple(Classe,  
Atributos)`**

Uma string com o 'tipo'  
ou 'classe' da qual a  
tupla pertence

Um iterável de strings  
com os atributos

# Exemplo: Figurinhas da copa [1]

```
namedtuple('jogador', ['nome',  
                        'time',  
                        'camisa',  
                        'Numero'])
```

Atributos

Classe Jogador
Nome
Time
Camisa
Numero

## Exemplo: Figurinhas da copa [2]



Classe Jogador	
Nome	Ronaldo
Time	Brasil
Camisa	9
Numero	90

## Exemplo: Figurinhas da copa [3]

```
jogador = t('Ronaldo',  
            'Brasil',  
            9,  
            90])
```

Classe Jogador	
Nome	Ronaldo
Time	Brasil
Camisa	9
Numero	90

MAAAAAAAAAAAAASSSSS,  
espera aí...

Posso fazer tudo isso  
com classes, não?

SIM

SIM

SIM

SIM

SIM

SIM

```
1 class jogador:
2     def __init__(self, nome, time, n):
3         self.nome = nome
4         self.time = time
5         self.n = n
6         self._t = (self.nome, self.time, self.n)
7
8     def __repr__(self):
9         return 'Jogador(nome={}, time={}, n={})'.format(self.nome,
10                                                         self.time,
11                                                         self.n)
12
13     def __getitem__(self, n):
14         return self._t[n]
15
16     def __len__(self):
17         return len(self._j)
18
19     def count(self, att):
20         return len([x for x in self._t if x == att])
21
22     def index(self, att):
23         for x, y in enumerate(self._t):
24             if y == att:
25                 return x
26
```

SIM

SIM

SIM

SIM

SIM

SIM



MAAAAAAAAAAAAASSSSS,

eu prefiro assim...



```
1  from collections import namedtuple
2
3  jogador = namedtuple('jogador', 'Nome Time Camisa Numero'.split())
4
```