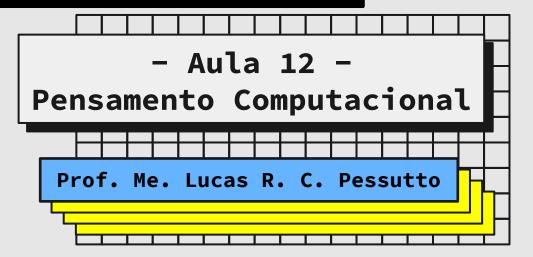
# Outras Estruturas de Dados: Tuplas e Conjuntos





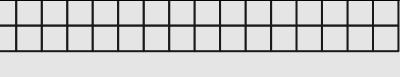




\_ 凸 ×



### Na aula de hoje...







Concatenação Cópia de Listas Fatiamento



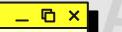
Aninhamento de Listas



#### **List Comprehensions**

Sintaxe Mapeamento Filtragem

 $\mathbb{Z}$ 



### Recapitulando

Estrutura de dados	Uso
Listas (list)	Coleção mutável com itens ordenados/indexáveis mutáveis; pode haver duplicatas
Strings (str)  Coleção imutável com itens ordenados/indexáv imutáveis; pode haver duplicatas	
Dicionários (dict)	Coleção mutável com itens não-ordenados com chaves imutáveis e valores mutáveis; não pode haver chaves duplicadas
Tuplas (tuple)	?
Conjuntos (set)	?

# Tuplas (tuple)

```
Д
```

```
>>> t = ("abacaxi", "uva", "morango")
>>> type(t)
<class 'tuple'>
```

- É uma coleção de elementos (com possíveis duplicatas) ordenados/indexáveis, mas imutáveis;
- \* Muito similar a listas: comandos de acesso e fatiamento são possíveis; mas os de atribuição falham!
- \* Geralmente mais rápidas que listas!



#### Tuplas

```
Sintaxe: tupla = (elem1, elemem2, ..., elemN)
```

Tuplas Especiais:

Tupla Vazia:

```
tupla = ()
```

Tupla com 1 elemento:



**Vírgula Obrigatória**! Caso contrário o Python interpreta como uma expressão entre parênteses.



#### Acesso aos valores

O acesso aos valores de uma tupla é feito usando a notação de colchetes, da mesma maneira que as listas:

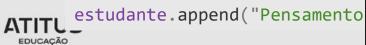
```
estudante = ("João da Silva", 22, "Ciência da Computação", 9.5)
print(f"Aluno: {estudante[0]}")
```

Qual o resultado das seguintes operações?

```
print(estudante[2])

Ciência da Computação

TypeError: 'tuple' object doesn't support item deletion
```



AttributeError: 'tuple' object has no attribute 'append'

### Outras Operações

$$s = (10, 20, 30)$$

Qual o resultado das seguintes operações?



# Listas x Strings x Tuplas

$$S = [1, 2, 3, 4]$$
  
 $W = [5, 6]$ 

$$S = (1, 2, 3, 4)$$
  
 $W = (5, 6)$ 

\_ © ×

s.count(4)

s[1:4]

Tamanho

#### Contagem s.count('e')

\_ © ×





#### Listas x Strings x Tuplas

$$s = [1, 2, 3, 4]$$
  
 $w = [5, 6]$ 

$$s = (1, 2, 3, 4)$$
  
 $w = (5, 6)$ 

#### Índice

#### Pertencimento

True

False

False

#### Concatenação

"hello!"

[1, 2, 3, 4, 5, 6]

(1, 2, 3, 4, 5, 6)



\_ © ×



### Listas x Strings x Tuplas

$$s = [1, 2, 3, 4]$$
  
 $w = [5, 6]$ 

$$S = (1, 2, 3, 4)$$
  
 $W = (5, 6)$ 

Valor Mínimo

min(s)



Soma

sum(s)

10

10





### Problema 1: Horóscopo Chinês



#### Enunciado de um problema:

Faça um programa para mostrar o animal do horóscopo chinês correspondente ao ano de nascimento informado pelo usuário.

Ao contrário do horóscopo ocidental, os signos são anuais, ou seja, todas as pessoas nascidas naquele ano possuem o mesmo signo.

Sabe-se que o ano de 1900 foi o ano do rato, 1901 do boi, ..., 1912 foi o ano do porco e 1913 foi o ano do rato novamente.

Número <b>≑</b>	Signo \$
1	Rato
2	Boi
3	Tigre
4	Coelho
5	Dragão
6	Serpente
7	Cavalo
8	Cabra
9	Macaco
10	Galo
11	Cão
12	Porco



### Problema 1: Horóscopo Chinês

```
zodiaco = ('Rato', 'Boi', 'Tigre', 'Coelho', 'Dragão', 'Cobra', 'Cavalo', 'Cabra',
'Macaco', 'Galo', 'Cachorro', 'Porco')
print("Este programa mostra o seu signo no Horóscopo Chinês")
fim = False
while not fim:
    ano nasc = int(input('Informe o ano de nascimento (yyyy): '))
    while ano nasc < 1900:
        ano nasc = int(input('ENTRADA INCORRETA! Informe o ano de nascimento (yyyy): '))
    signo =
                       ??
    print(f"\nSeu signo é {zodiaco[signo].upper()}\n")
    fim = input("Informar outro ano [S]im ou [N]ão? ").lower() != "s"
```

### Д

# Conjuntos (set)

```
>>> c = {"abacaxi", "uva", "morango"}
>>> type(c)
<class 'set'>
```

- \* É uma coleção de elementos não-ordenados/não-indexáveis, imutáveis e sem duplicatas;
- \* A coleção é mutável (pode-se adicionar e remover elementos...);
- \* Essa estrutura implementa o conceito matemático de conjuntos!



### Conjuntos

Sintaxe:

```
conjunto = {elem1, elemem2, ..., elemN}
```

Conjuntos Especiais:

Conjunto Vazio:

```
conjunto = set()
Não é possível criar um conjunto fazendo
conjunto = {}, pois essa sintaxe é usada
para criar dicionários.
```

Conjunto Unitário:





### Operações em Conjuntos

- \* A estrutura de dados set implementa uma série de operações de Teoria dos Conjuntos
  - → Pertencimento
  - → Inserção
  - → Remoção
  - → União
  - → Intersecção
  - → Diferença
  - → Diferença Simétrica
  - → Tamanho



# Listas x Strings x Tuplas

Python **Operador** 

**Função** 

 $B = \{3, 4, 5, 6\}$ Pertencimento  $1 \in A$ 

1 in A

A.union(B)

True

\_ © ×

Inserção

Remoção

União

**EDUCAÇÃO** 

 $A = \{1, 2, 3\}$ 

 $A \cup B$ 

A.add(4)

{1, 2, 3, 4}

A.remove(2)

{1, 3}

## Listas x Strings x Tuplas

Int	torsoccã	i 0			_		
В	= {3,	4,	5,	<b>6</b> }			
$\overline{}$	_ ( <u>+</u> )	ر کے	<b>-</b> )				

B = \(\frac{1}{2}\), \(\frac{4}{2}\), \(\frac{1}{2}\), \(\frac{1}{2}\)			
Intersecção	$A \cap B$	A & B	A.in



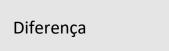








\_ © ×



Diferença Simétrica

$$A - B$$



len(A)





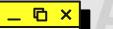
$$A \triangle B$$

|A|





Cardinalidade



#### Problema 2: Alunos



#### Enunciado de um problema:

Faça um programa que pede ao usuário que informe os alunos que cursam as disciplinas de "Pensamento Computacional", "Inteligência Artificial" e "Desafios da Profissão". Ao fim, queremos listar todos os alunos que cursam:

- \* "Inteligência Artificial" e "Desafios da Profissão"
- \* "Pensamento Computacional" ou "Inteligência Artificial"
- \* "Pensamento Computacional" mas não fazem "Desafios da Profissão"



Informe um aluno de Inteligência Artificial (ou nada para terminar): Informe um aluno de Desafios da Profissão (ou nada para terminar): João Informe um aluno de Desafios da Profissão (ou nada para terminar): Ana Informe um aluno de Desafios da Profissão (ou nada para terminar): Joana Informe um aluno de Desafios da Profissão (ou nada para terminar): Alunos que cursam Inteligência Artificial E Desafios da Profissão: - Ana Alunos que cursam Pensamento Computacional OU Inteligência Artificial: - Maria - João - José - Ana Alunos que cursam Pensamento Computacional mas NÃO Desafios da Profissão: - Maria

Informe um aluno de Pensamento Computacional (ou nada para terminar): Ana Informe um aluno de Pensamento Computacional (ou nada para terminar): João Informe um aluno de Pensamento Computacional (ou nada para terminar): Maria

Informe um aluno de Pensamento Computacional (ou nada para terminar):

Informe um aluno de Inteligência Artificial (ou nada para terminar): Ana Informe um aluno de Inteligência Artificial (ou nada para terminar): José Informe um aluno de Inteligência Artificial (ou nada para terminar): Maria

```
ia = set()
des prof = set()
nome = " "
while nome != "":
    nome = input("Informe um aluno de Pensamento Computacional (ou nada para terminar): ")
    if nome != "":
        pcomp.add(nome)
nome = " "
while nome != "":
    nome = input("Informe um aluno de Inteligência Artificial (ou nada para terminar): ")
    if nome != "":
        ia.add(nome)
nome = " "
while nome != "":
    nome = input("Informe um aluno de Desafios da Profissão (ou nada para terminar): ")
    if nome != "":
        des prof.add(nome)
```

pcomp = set()



```
print("Alunos que cursam Inteligência Artificial E Desafios da Profissão:")
for aluno in ia.intersection(des_prof):
    print(f" - {aluno}")

print("Alunos que cursam Pensamento Computacional OU Inteligência Artificial:")
for aluno in pcomp.union(ia):
    print(f" - {aluno}")

print("Alunos que cursam Pensamento Computacional mas NÃO Desafios da Profissão:")
for aluno in pcomp.difference(des_prof):
    print(f" - {aluno}")
```





#### Resumo de Estruturas de Dados

Estrutura de dados	Uso
Listas (list)	Coleção mutável com itens ordenados/indexáveis mutáveis; pode haver duplicatas
Strings (str)	Coleção imutável com itens ordenados/indexáveis imutáveis; pode haver duplicatas
Dicionários (dict)	Coleção mutável com itens não-ordenados com chaves imutáveis e valores mutáveis; não pode haver chaves duplicadas
Tuplas (tuple)	Coleção imutável com itens ordenados/indexáveis imutáveis; pode haver duplicatas
Conjuntos (set)	Coleção mutável com itens não-ordenados imutáveis; não pode haver duplicatas