

Atividade1

April 20, 2025

1 Atividade 1 da disciplina de LINGUAGENS FORMAIS E AUTÔMATOS

1.1 INSTALAÇÕES DE BIBIOTECAS

```
[30]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
[ ]: from google.colab import drive
drive.mount('/content/drive') # Monte o Drive
!ls "/content/drive/MyDrive/Colab Notebooks" # Liste os arquivos
```

```
[36]: from google.colab import files
files.download('/content/drive/MyDrive/Colab Notebooks/Atividade1.ipynb')
```

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

```
[ ]: !apt-get update
!apt-get install -y pandoc texlive-xetex texlive-fonts-recommended
↳ texlive-generic-recommended
```

```
[56]: !cp "/content/drive/MyDrive/Colab Notebooks/Atividade1.ipynb" "/content/"

!jupyter nbconvert --to pdf "/content/Atividade1.ipynb"

!ls -l "/content/Atividade1.pdf"

from google.colab import files
files.download('/content/Atividade1.pdf')
```

[NbConvertApp] Converting notebook /content/Atividade1.ipynb to pdf

[NbConvertApp] Writing 42108 bytes to notebook.tex

[NbConvertApp] Building PDF

[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']

[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']

```
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 48495 bytes to /content/Atividade1.pdf
-rw-r--r-- 1 root root 48495 Apr 20 20:00 /content/Atividade1.pdf

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>
```

1.2 RESOLUÇÕES:

1. Listagem de Elementos de um Conjunto Como você pode representar o conjunto ($V = \{a, e, i, o, u\}$) em Python e imprimir seus elementos?

```
[10]: V = {'a', 'e', 'i', 'o', 'u'}
print("Elementos do conjunto V:", V)
```

Elementos do conjunto V: {'i', 'a', 'o', 'e', 'u'}

2. Dado o conjunto ($V = \{a, e, i, o, u\}$), como você pode verificar se o elemento 'a' está presente em (V)?

```
[13]: V = {'a', 'e', 'i', 'o', 'u'}
print("'a' está em V?", 'a' in V) # Retorna True
print("'b' está em V?", 'b' in V) # Retorna False
```

'a' está em V? True
'b' está em V? False

3. Criação de um Conjunto com Propriedades Específicas

```
[43]: B = {x for x in range(11, 20) if x % 2 == 0}
B = set(range(12, 20, 2))
print("Conjunto B:", B)
```

Conjunto B: {16, 18, 12, 14}

4. Se ($V = \{a, e, i, o, u\}$) e ($C = \{i, o, u\}$), como você pode verificar se todos os elementos de (C) também estão em (V)?

```
[44]: V = {'a', 'e', 'i', 'o', 'u'}
C = {'i', 'o', 'u'}
print("C é subconjunto de V?", C.issubset(V)) # Retorna True
print("C está contido em V?", C <= V) # Retorna True
```

C é subconjunto de V? True
C está contido em V? True

5. Como você pode representar um conjunto (D) em Python que contém todos os números inteiros de 1 a 10 que são divisíveis por 3?

```
[45]: D = {x for x in range(1, 11) if x % 3 == 0}
print("Conjunto D:", D) # {3, 6, 9}
```

Conjunto D: {9, 3, 6}

6. Dados dois conjuntos ($A = \{1, 2, 3\}$) e ($B = \{3, 4, 5\}$), como você pode obter a união de (A) e (B)?

```
[46]: A = {1, 2, 3}
B = {3, 4, 5}
uniao = A.union(B)
# Ou alternativamente:
uniao = A | B
print("União de A e B:", uniao) # {1, 2, 3, 4, 5}
```

União de A e B: {1, 2, 3, 4, 5}

7. Dado os conjuntos ($A = \{1, 2, 3\}$) e ($B = \{3, 4, 5\}$), como você pode encontrar a interseção entre (A) e (B)?

```
[47]: A = {1, 2, 3}
B = {3, 4, 5}
intersecao = A.intersection(B)
# Ou alternativamente:
intersecao = A & B
print("Interseção de A e B:", intersecao) # {3}
```

Interseção de A e B: {3}

8. Como você pode determinar os elementos que estão em ($A = \{1, 2, 3\}$) mas não estão em ($B = \{3, 4, 5\}$)?

```
[48]: A = {1, 2, 3}
B = {3, 4, 5}
diferenca = A.difference(B)
diferenca = A - B
print("Diferença A - B:", diferenca) # {1, 2}
```

Diferença A - B: {1, 2}

9. Se ($A = \{1, 2, 3\}$) e ($B = \{3, 4, 5\}$), como você pode obter a diferença simétrica entre (A) e (B)?

```
[49]: A = {1, 2, 3}
B = {3, 4, 5}
diff_simetrica = A.symmetric_difference(B)
diff_simetrica = A ^ B
print("Diferença simétrica:", diff_simetrica) # {1, 2, 4, 5}
```

Diferença simétrica: {1, 2, 4, 5}

10. Dado ($A = \{1, 2, 3\}$) e ($B = \{1, 2, 3, 4, 5\}$), como você pode verificar se (A) é um subconjunto de (B) e se (B) é um superconjunto de (A)?

```
[50]: A = {1, 2, 3}
      B = {1, 2, 3, 4, 5}
      print("A é subconjunto de B?", A.issubset(B)) # True
      print("B é superconjunto de A?", B.issuperset(A)) # True
      print("A <= B?", A <= B) # True
      print("B >= A?", B >= A) # True
```

A é subconjunto de B? True
B é superconjunto de A? True
A <= B? True
B >= A? True

11. Números Pares Maiores que 10

```
[51]: pares_maiores_10 = {x for x in range(11, 100) if x % 2 == 0}
      print("Números pares > 10:", pares_maiores_10)
```

Números pares > 10: {12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98}

12. Números Primos Menores que 20

```
[52]: def is_prime(n):
      if n < 2:
          return False
      for i in range(2, int(n**0.5)+1):
          if n % i == 0:
              return False
      return True

      primos_menor_20 = {x for x in range(2, 20) if is_prime(x)}
      print("Números primos < 20:", primos_menor_20) # {2, 3, 5, 7, 11, 13, 17, 19}
```

Números primos < 20: {2, 3, 5, 7, 11, 13, 17, 19}

13. Números Ímpares Divisíveis por 3 até 30

```
[53]: impares_div3 = {x for x in range(1, 31) if x % 2 != 0 and x % 3 == 0}
      print("Ímpares divisíveis por 3 até 30:", impares_div3) # {3, 9, 15, 21, 27}
```

Ímpares divisíveis por 3 até 30: {3, 9, 15, 21, 27}

14. Quadrados Perfeitos Menores que 100

```
[54]: quadrados_perfeitos = {x**2 for x in range(1, 10)} # 10² = 100 (não incluso)
      print("Quadrados perfeitos < 100:", quadrados_perfeitos) # {1, 4, 9, 16, 25, 36, 49, 64, 81}
```

Quadrados perfeitos < 100: {64, 1, 4, 36, 9, 16, 49, 81, 25}

15. Múltiplos de 5 entre 10 e 50

```
[55]: multiplos_de_5 = set(range(10, 51, 5))  
print("Múltiplos de 5 entre 10 e 50:", multiplos_de_5)  # {10, 15, 20, 25, 30, 35, 40, 45, 50}
```

Múltiplos de 5 entre 10 e 50: {35, 40, 10, 45, 15, 50, 20, 25, 30}