

# FUNDAMENTOS DE MANIPULAÇÃO E ANÁLISE DE DADOS

# Agenda

Dia	Aula	Assunto
28/04	Aula 1	Introdução à Linguagem Python
29/04	Aula 2	Introdução à Linguagem Python
30/04	Aula 3	Manipulação e Análise de Dados
01/05	Aula 4	Manipulação e Análise de Dados

# Dados

**Por que só estão falando  
sobre dados nos dias atuais?**

# Dados

**Por que só estão falando  
sobre dados nos dias atuais?**

**Velocidade  $\equiv$**

# Dados

**Por que só estão falando  
sobre dados nos dias atuais?**

**Velocidade  $\equiv$**

**Armazenamento**

# Dados

**Os dados conversam com  
a gente e podem trazer informações úteis!**

# Dados

**Os dados conversam com a gente e podem trazer informações úteis!**

**Extração e manipulação**

# Dados

Os dados conversam com a gente e podem trazer informações úteis!

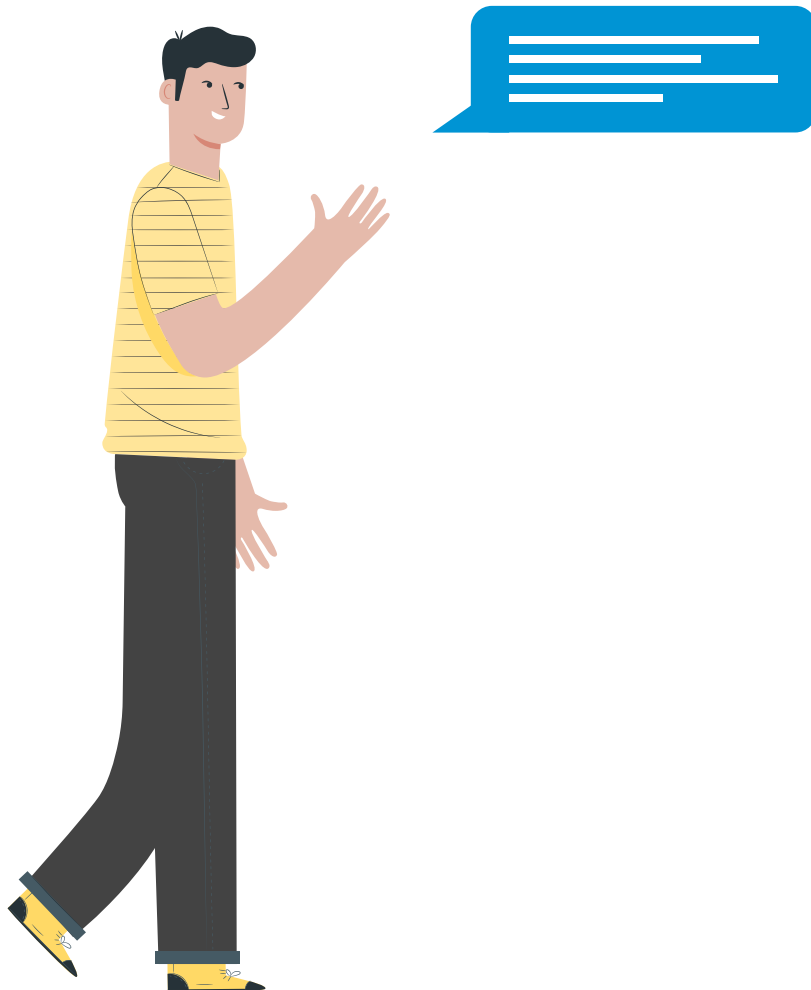
Extração e manipulação

# Análise!

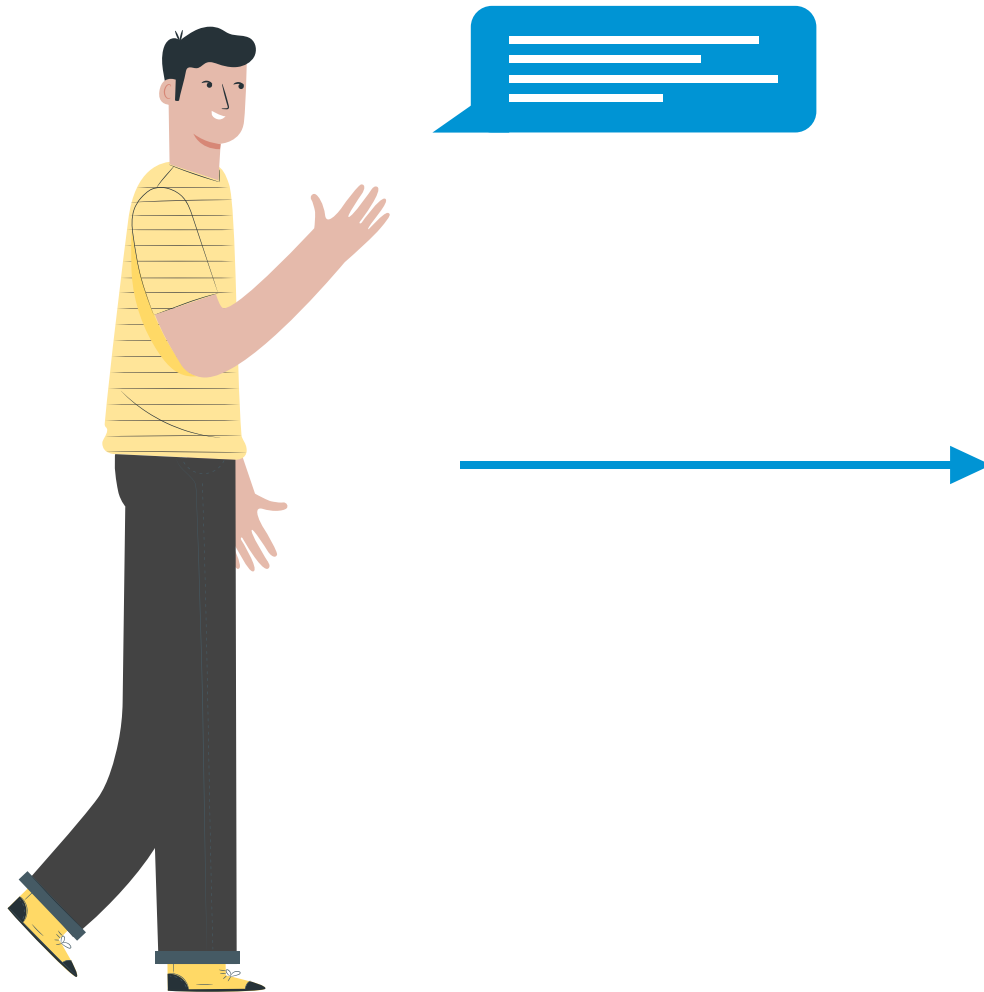




# Linguagens de Programação



# Linguagens de Programação



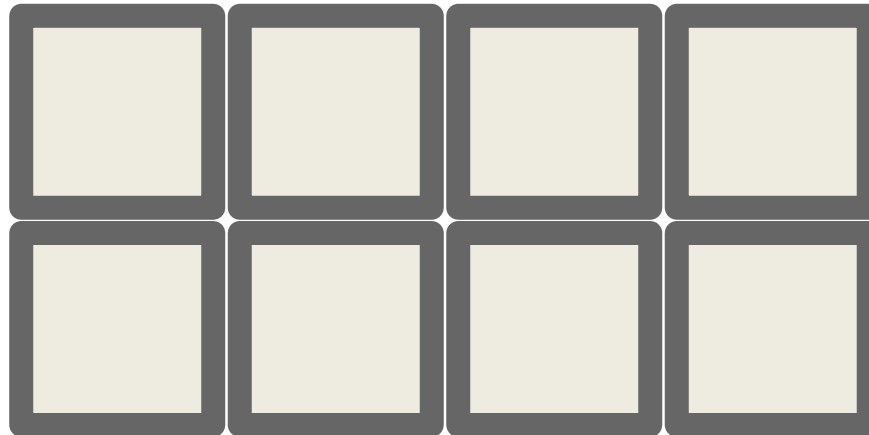
# Linguagens de Programação



# Linguagens de Programação

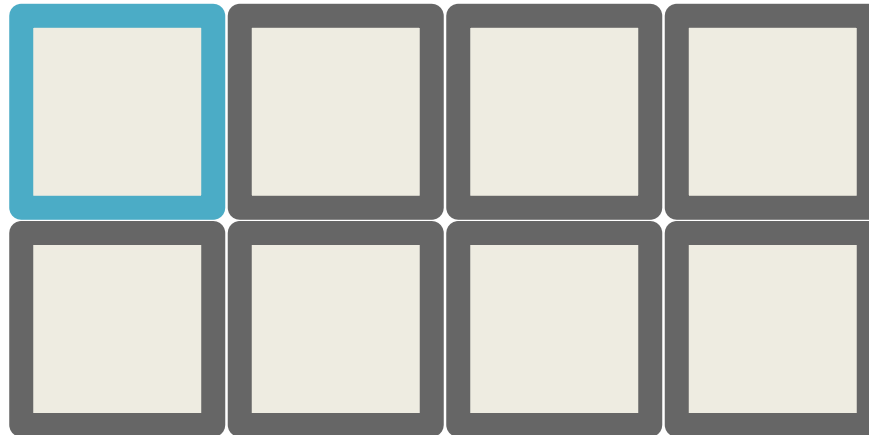


## Memória



# Variável

## Memória



# Variável

## Memória

idade



# Variável

## Memória

idade

15			



# Listas

L

# Listas

L =

# Listas

**L = [ ]**

# Listas

$L = [1 \ 2 \ 3 \ 4 \ 5]$

# Listas

$L = [1, 2, 3, 4, 5]$

# Listas

$L = [1, 2, 3, 4, 5]$

# Listas

$L = [1, 2, 3, 4, 5]$

0 1 2 3 4



# Listas

$L = [1, 2, 3, 4, 5]$

0 1 2 3 4



$L[1]$



# Listas

$L = [1, 2, 3, 4, 5]$

0 1 2 3 4




$L[1] \rightarrow 2$

# Listas

$L = [1, 2, 3, 4, 5]$


0 1 2 3 4



# Listas

$L = [1, 2, 3, 4, 5]$

0 1 2 3 4




$L[4]$

# Listas

$L = [1, 2, 3, 4, 5]$

0 1 2 3 4



$L[4] \rightarrow 5$

# Listas

$L = [1, 2, 3, 4, 5]$

$-5 \quad -4 \quad -3 \quad -2 \quad -1$

←

$L[-1]$

# Listas

$L = [1, 2, 3, 4, 5]$

$-5 \quad -4 \quad -3 \quad -2 \quad -1$

←

$L[-1] \rightarrow 5$

# Listas

$L = [1, 2, 3, 4, 5]$

$-5 \quad -4 \quad -3 \quad -2 \quad -1$

←

$L[-4]$

# Listas

$L = [1, 2, 3, 4, 5]$

$-5 \quad -4 \quad -3 \quad -2 \quad -1$

←


$L[-4] \rightarrow 2$



# Slicing

**L = [ 1 , 2 , 3 , 4 , 5 ]**

0      1      2      3      4



# Slicing

$L = [1, 2, 3, 4, 5]$

0 1 2 3 4




$L[1:3]$

# Slicing

$L = [1, 2, 3, 4, 5]$

0 1 2 3 4



$L[1:3] \rightarrow [2, 3]$

# Slicing

**L = [1, 2, 3, 4, 5]**

0      1      2      3      4



# Slicing

$L = [1, 2, 3, 4, 5]$

0 1 2 3 4



$L[0:4]$

# Slicing

$L = [1, 2, 3, 4, 5]$


0      1      2      3      4

$L[0:4] \rightarrow [1, 2, 3, 4]$

# Slicing

$L = [1, 2, 3, 4, 5]$

0      1      2      3      4



$L[:4] \rightarrow [1, 2, 3, 4]$

# Dicionários

**d =**



# Dicionários

**d** = { }

# Dicionários

$d = \{ 1 : \text{'oi'} \}$

# Dicionários

$d = \{ 1 : \text{'oi'} \quad 2 : \text{'olá'} \}$

# Dicionários

$d = \{ 1 : \text{'oi'}, 2 : \text{'olá'} \}$

$d = \{ \overset{\downarrow}{1} : \text{'oi'}, \overset{\downarrow}{2} : \text{'olá'} \}$

## CHAVES

$d = \{ 1 : \text{'oi'}, 2 : \text{'olá'} \}$

## VALORES

# Dicionários

$d = \{ 1 : \text{'oi'}, 2 : \text{'olá'} \}$

$d[1] \rightarrow \text{'oi'}$

# Dicionários

$d = \{ 1 : \text{'oi'}, 2 : \text{'olá'} \}$

$d[2] \rightarrow \text{'olá'}$



# Operadores relacionais

**==**

**!=**

**<**

**ou**

**<=**

**>**

**ou**

**>=**

# Operadores relacionais



IGUAL



ou



ou



# Operadores relacionais



IGUAL



DIFERENTE



ou



ou



# Operadores relacionais



IGUAL



DIFERENTE



OU



MENOR OU MENOR OU IGUAL



OU



# Operadores relacionais



IGUAL



DIFERENTE



OU



MENOR OU MENOR OU IGUAL



OU



MAIOR OU MAIOR OU IGUAL

# Operadores de lógica

**not**  
**and**  
**or**

# Operadores de lógica

**not**

nega uma expressão

**and**

**or**

# Operadores de lógica

**not**

nega uma expressão

**and**

só retorna **True** se ambas  
expressões forem **verdadeiras**

**or**



# Operadores de lógica

**not**

nega uma expressão

**and**

só retorna **True** se ambas expressões forem verdadeiras

**or**

só retorna **False** se ambas expressões forem falsas

# Operadores de lógica

**not**

nega uma expressão

**and**

só retorna True se ambas expressões forem verdadeiras

**or**

só retorna False se ambas expressões forem falsas

# Operadores de lógica

## Tabela Verdade not

Operador	Expressão	Resultado
not	True	False
not	False	True

# Operadores de lógica

not

nega uma expressão

and

só retorna True se ambas expressões forem verdadeiras

or

só retorna False se ambas expressões forem falsas

# Operadores de lógica

## Tabela Verdade and

Expressão 1	Operador	Expressão 2	Resultado
True	and	True	True
True	and	False	False
False	and	True	False
False	and	False	False

# Operadores de lógica

not

nega uma expressão

and

só retorna True se ambas expressões forem verdadeiras

or

só retorna False se ambas expressões forem falsas

# Operadores de lógica

## Tabela Verdade or

Expressão 1	Operador	Expressão 2	Resultado
True	or	True	True
True	or	False	True
False	or	True	True
False	or	False	False

# Desvio Condicional

Alguma condição importante?



# Desvio Condicional

Alguma condição importante?

**SIM**

# Desvio Condicional

Alguma condição importante?

**SIM**

**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM**

Executa o código abaixo

**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM**

Executa o código abaixo

**NÃO**

Segue a vida...

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código
```

# Desvio Condicional

↙  
**if** (**condição**):  
    # bloco de código  
    # bloco de código

# Desvio Condicional

**if** ( **condição** ) :  
    # bloco de código  
    # bloco de código

# Desvio Condicional



```
if (condição):  
    # bloco de código  
    # bloco de código
```



# Desvio Condicional

**if** (**condição**):  
    # bloco de código  
    # bloco de código

# Desvio Condicional

**if (condição):**

→ **# bloco de código**  
→ **# bloco de código**  
→ **# bloco de código**

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código
```

# Desvio Condicional

**if (condição):**  
    # bloco de código  
    # bloco de código

# Desvio Condicional

VERDADEIRA  
if (condição):  
    # bloco de código  
    # bloco de código

# Desvio Condicional

**VERDADEIRA**  
**if (condição):**  
    **# bloco de código**  
    **# bloco de código**

# Desvio Condicional

VERDADEIRA  
if (condição):  
# bloco de código  
# bloco de código ✓

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código
```



# Desvio Condicional

**if (condição):**  
# bloco de código  
# bloco de código

# Desvio Condicional

**FALSA**  
**if (condição):**  
    # bloco de código  
    # bloco de código

# Desvio Condicional

**FALSA**  
**if (condição):**  
# bloco de código  
# bloco de código **×**

# Desvio Condicional

Alguma condição importante?

# Desvio Condicional

Alguma condição importante?

**SIM**

# Desvio Condicional

Alguma condição importante?

**SIM**

**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM**

Executa o código abaixo

**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM**

Executa o código abaixo

**NÃO**

Executo o outro código



# Desvio Condicional

**if (condição):**  
    **# bloco de código**  
    **# bloco de código**

**else:**  
    **# bloco de código**  
    **# bloco de código**

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código  
else :  
    # bloco de código  
    # bloco de código
```

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código  
else :  
    # bloco de código  
    # bloco de código
```

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código
```

```
else :  
    # bloco de código  
    # bloco de código
```

# Desvio Condicional

**if (condição):**  
    **# bloco de código**  
    **# bloco de código** ✓

**else :**  
    **# bloco de código**  
    **# bloco de código**

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código  
else :  
    # bloco de código  
    # bloco de código
```

# Desvio Condicional

if (condição):  
# bloco de código  
# bloco de código



else :  
# bloco de código  
# bloco de código

# Desvio Condicional

if (**condição**):  
    # bloco de código  
    # bloco de código



else :  
    **# bloco de código**  
    **# bloco de código**





# Desvio Condicional

Alguma condição importante?

**SIM**

**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO** Alguma condição importante?

# Desvio Condicional

Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO** Alguma condição importante?

**SIM**  
**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO** Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO** Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO** Segue a vida...

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código  
elif (condição):  
    # bloco de código  
    # bloco de código
```

# Desvio Condicional

**if (condição):**  
    # bloco de código  
    # bloco de código

**elif (condição):**  
    # bloco de código  
    # bloco de código



# Desvio Condicional

**if (condição):**  
    **# bloco de código**  
    **# bloco de código**

**elif (condição):**  
    **# bloco de código**  
    **# bloco de código**

# Desvio Condicional

**if (condição):**  
    **# bloco de código**  
    **# bloco de código** ✓

**elif (condição):**  
    **# bloco de código**  
    **# bloco de código**

# Desvio Condicional

```
if (condição):  
    # bloco de código  
    # bloco de código  
elif (condição):  
    # bloco de código  
    # bloco de código
```

# Desvio Condicional

**if (condição):**  
    # bloco de código  
    # bloco de código

**elif (condição):**  
    # bloco de código  
    # bloco de código

# Desvio Condicional

**if (condição):**  
# bloco de código  
# bloco de código **×**

**elif (condição):**  
# bloco de código  
# bloco de código

# Desvio Condicional

if (**condição**):  
# bloco de código  
# bloco de código



elif (**condição**):  
# bloco de código  
# bloco de código

# Desvio Condicional

**if (condição):**  
# bloco de código  
# bloco de código **×**

**elif (condição):**  
# bloco de código  
# bloco de código

# Desvio Condicional

**if (condição):**  
# bloco de código  
# bloco de código 

**elif (condição):**  
# bloco de código  
# bloco de código 



# Desvio Condicional

**if (condição):**  
# bloco de código  
# bloco de código **×**

**elif (condição):**  
# bloco de código  
# bloco de código

# Desvio Condicional

**if (condição):**  
# bloco de código  
# bloco de código **×**

**elif (condição):**  
# bloco de código  
# bloco de código

# Desvio Condicional

**if (condição):**  
# bloco de código  
# bloco de código **×**

**elif (condição):**  
# bloco de código  
# bloco de código **×**

# Desvio Condicional

Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO** Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO**

# Desvio Condicional

Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO** Alguma condição importante?

**SIM** Executa o código abaixo

**NÃO** Executa outro código

# Desvio Condicional

**if (condição):**  
    # bloco de código  
    # bloco de código

**elif (condição):**  
    # bloco de código  
    # bloco de código

**else:**  
    # bloco de código  
    # bloco de código

# Desvio Condicional

**if (condição):**  
    # bloco de código  
    # bloco de código

**elif (condição):**  
    # bloco de código  
    # bloco de código

**else:**  
    # bloco de código  
    # bloco de código

# Desvio Condicional

**if (condição):**

- **# bloco de código**
- **# bloco de código**

**elif (condição):**

- **# bloco de código**
- **# bloco de código**

**else:**

- **# bloco de código**
- **# bloco de código**



# Função

```
def nome_da_função (p1, p2):  
    # bloco de código  
    return valor_retornado
```

# Função

↙  
**def** nome\_da\_função (p1, p2):  
 # bloco de código  
 **return** valor\_retornado

# Função

↙

```
def nome_da_função (p1, p2):  
    # bloco de código  
    return valor_retornado
```

# Função

```
def nome_da_função (p1, p2):  
    # bloco de código  
    return valor_retornado
```

# Função

```
def nome_da_função (p1, p2):  
    # bloco de código  
    return valor_retornado
```

# Função

```
def nome_da_função (p1, p2):  
    # bloco de código  
    return valor_retornado
```

Diagram illustrating function syntax with arrows pointing to parameters:

- Two blue arrows point to `p1` and `p2` in the function signature.
- A black arrow points to the colon `:` at the end of the function signature.

# Função

```
def nome_da_função (p1, p2):
```

```
    # bloco de código
```

```
    return valor_retornado
```

# Função

```
def nome_da_função (p1, p2):
```

```
→ # bloco de código
```

```
    return valor_retornado
```



# Função

```
def nome_da_função (p1, p2):
```

```
    # bloco de código
```

```
    return valor_retornado
```



# Função

```
def nome_da_função (p1, p2):
```

```
    # bloco de código
```

```
    return valor_retornado
```



# Função

```
def nome_da_função (p1, p2):
```

```
→ # bloco de código
```

```
→ return valor_retornado
```

# Função

```
def nome_da_função (p1, p2):  
    # bloco de código  
    return valor_retornado
```

# Função

```
def nome_da_função():  
    # bloco de código  
    return valor_retornado
```

# Função

```
def nome_da_função():  
    # bloco de código
```

# Função

```
def nome_da_função (p1, p2):  
    # bloco de código
```