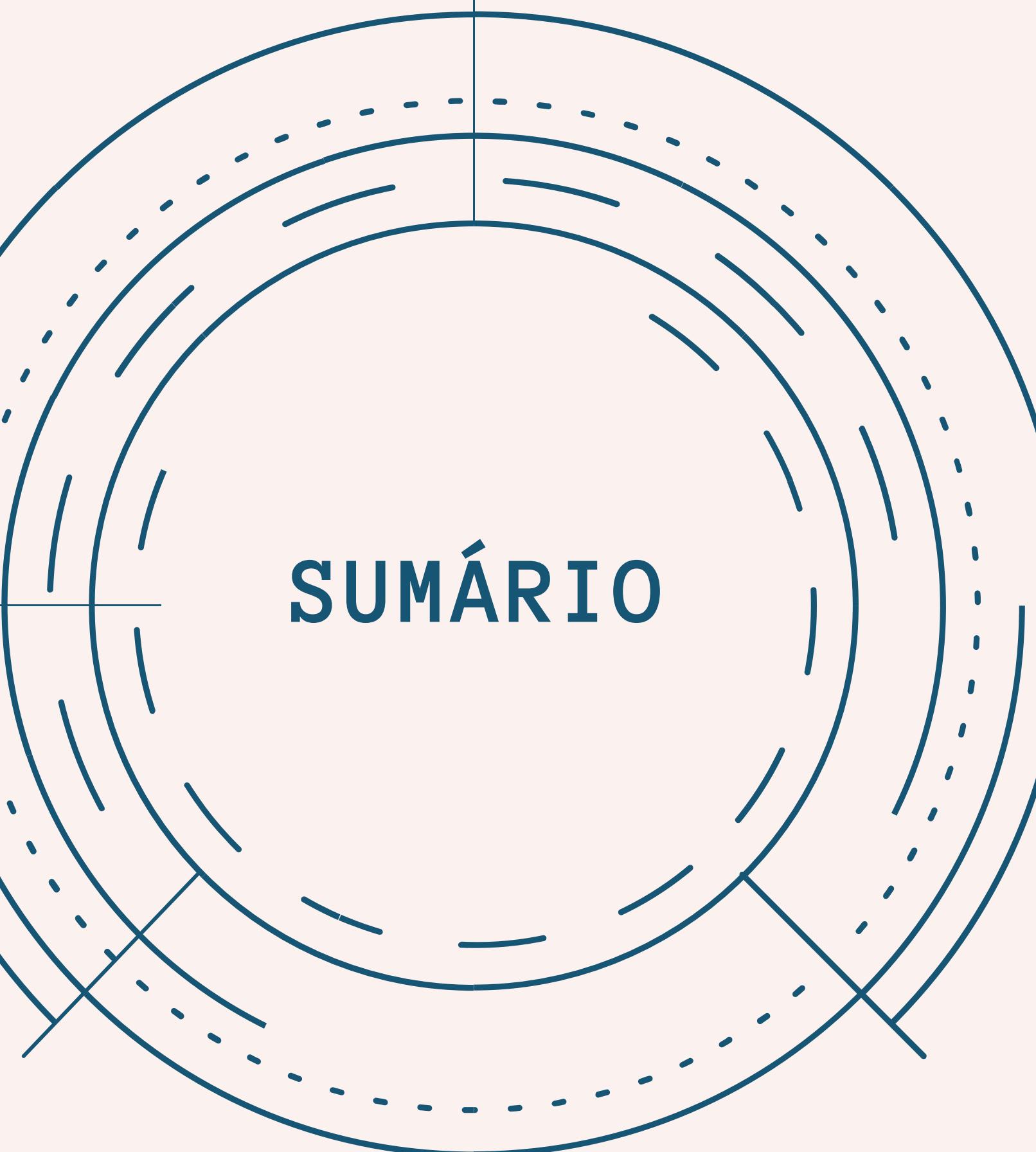




PYTHON

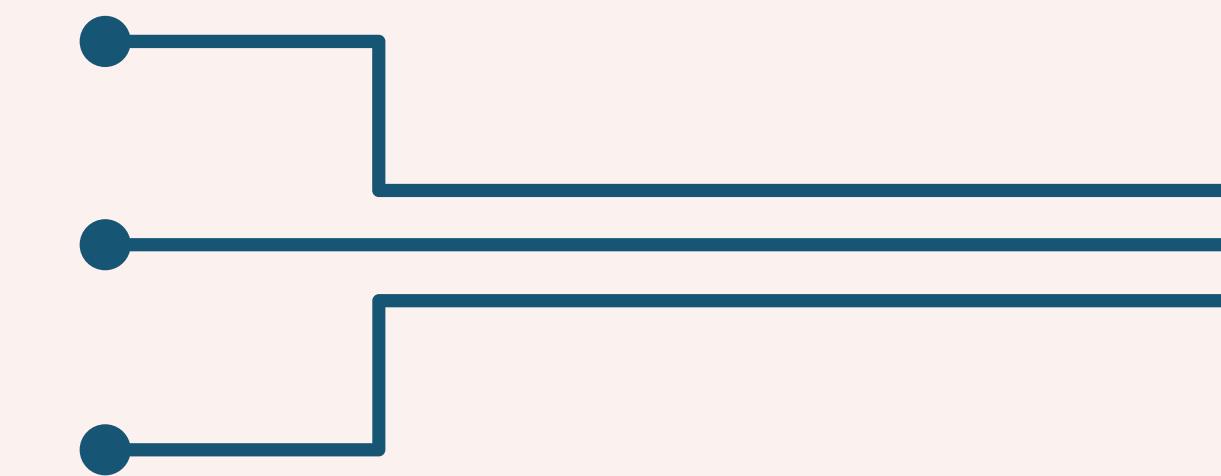
UMA INTRODUÇÃO À MELHOR LINGUAGEM DE PROGRAMAÇÃO



SUMÁRIO

O QUE VEREMOS

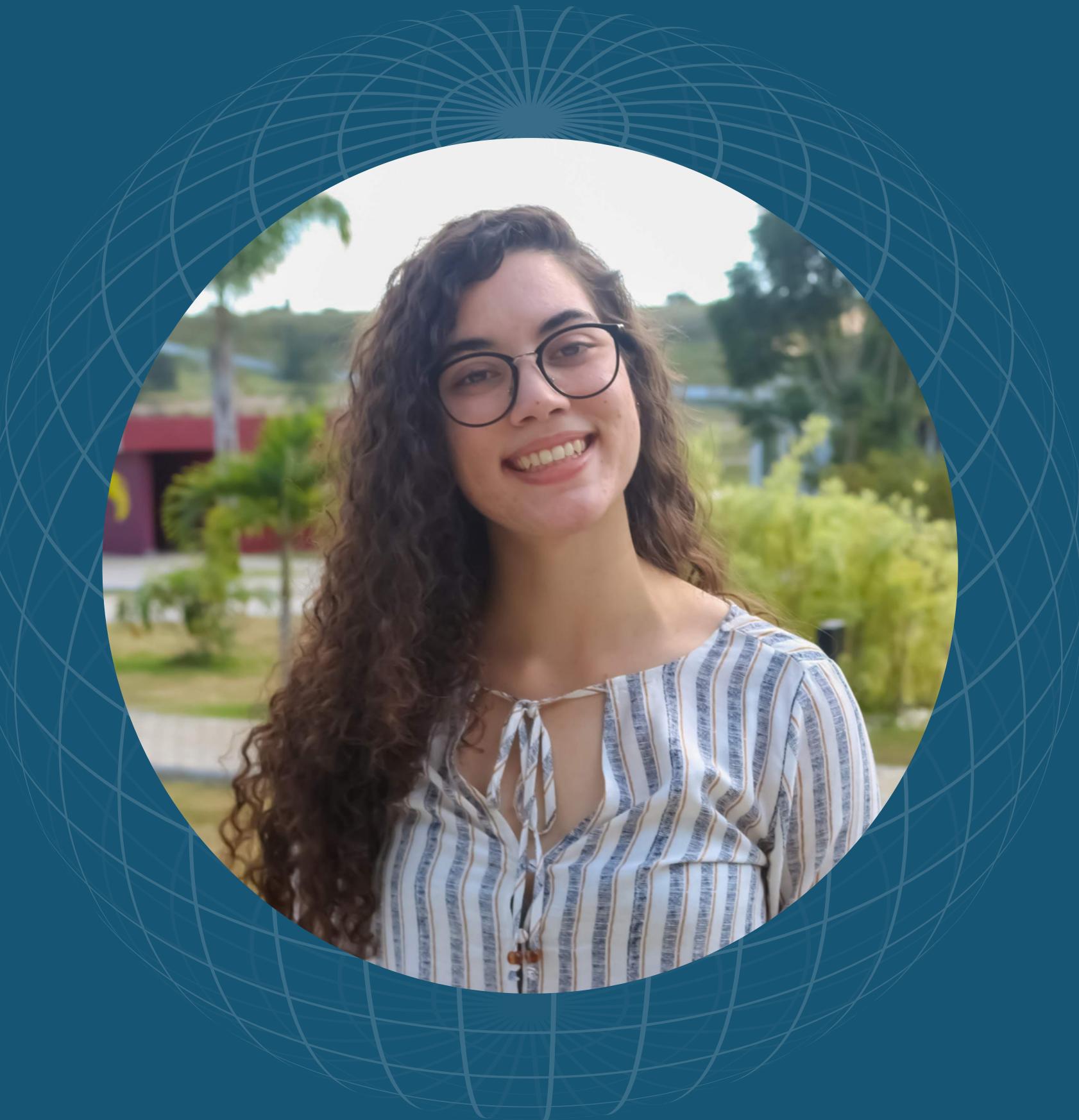
- Quem sou eu?
- O que é uma linguagem de programação?
- Pq Python?
- Sobre a linguagem
- Considerações finais (?)





QUEM SOU EU?

PRAZER, LUÍSA MOURA



POR ONDE COMEÇAR?

Luísa Moura

22 anos

Técnica em Informática pelo IFBA

O QUE FAÇO?

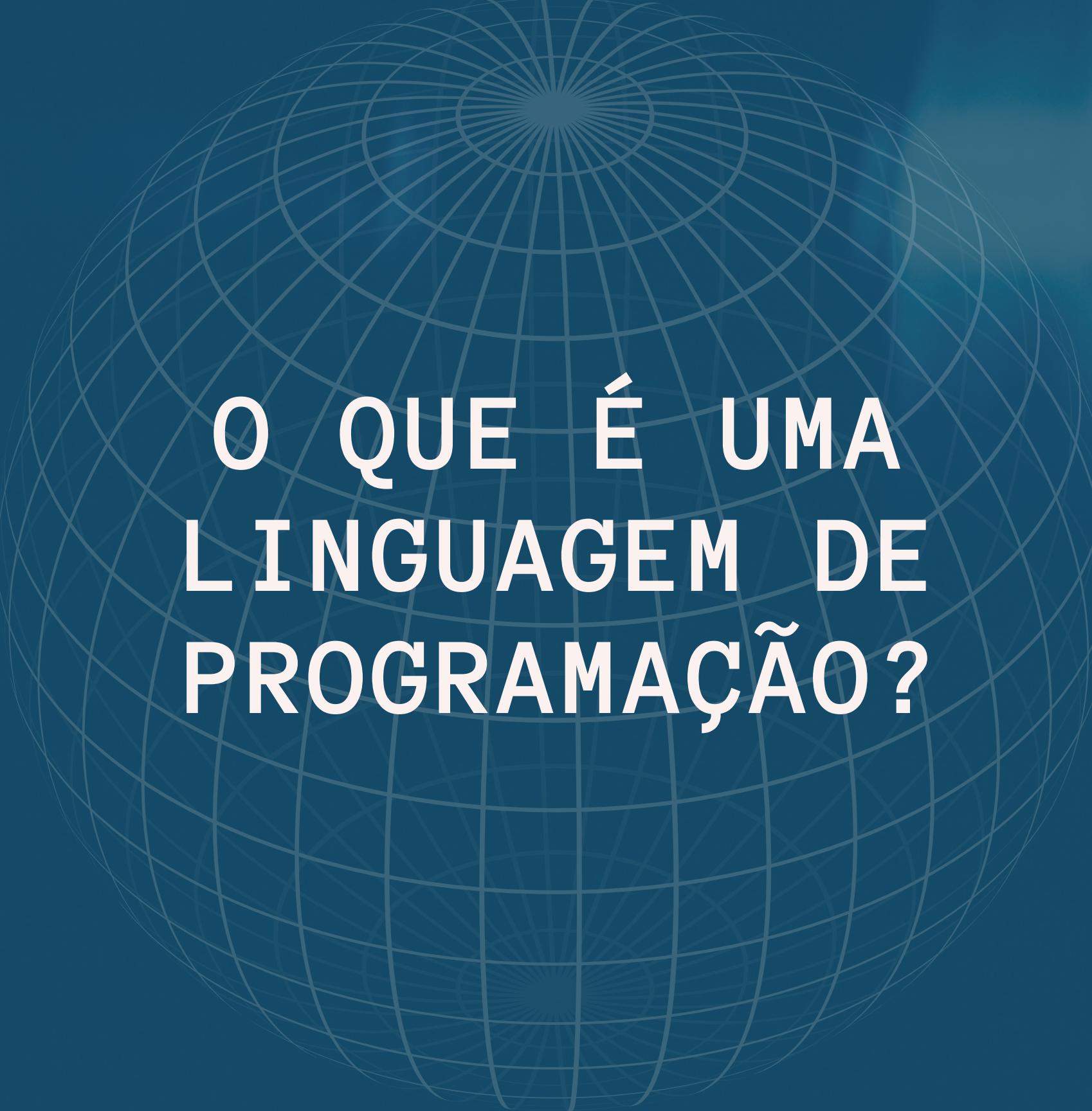
Um bocado de coisa

Ciências de Computação no ICMC/USP

Estagiária no Facebook

FALTA UMA PERGUNTA

não sei o que falar sobre mim mais

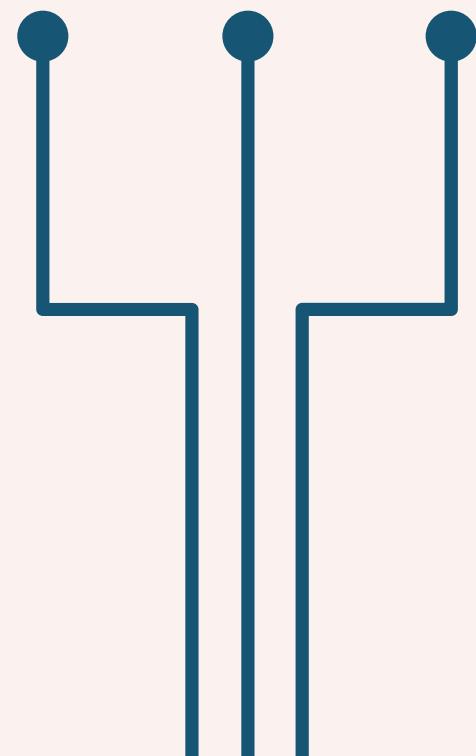


O QUE É UMA
LINGUAGEM DE
PROGRAMAÇÃO?

#DESCUBRA

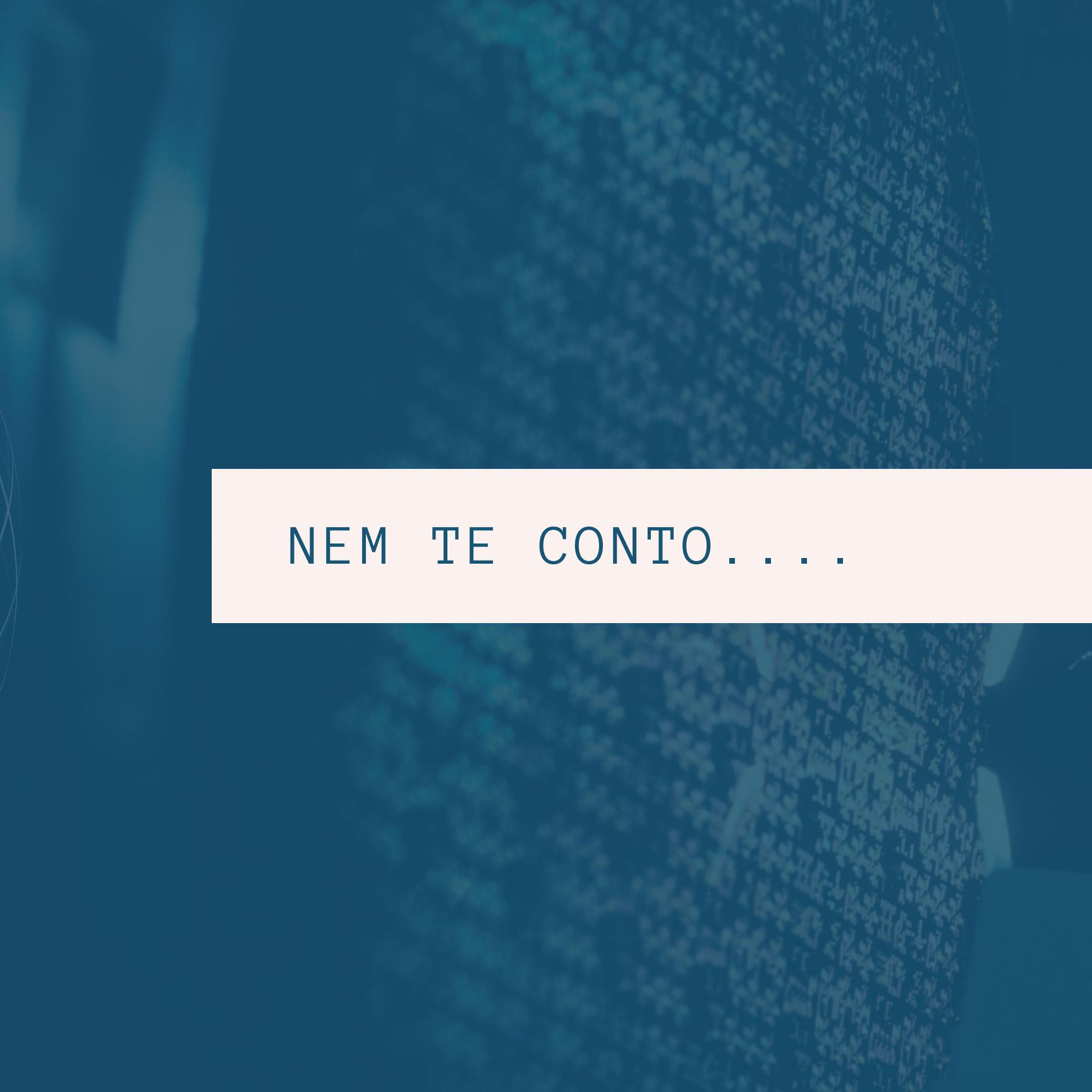
O QUE É UMA LINGUAGEM DE PROGRAMAÇÃO?

- Forma de conversar com o computador
- Linguagens de baixo nível
`01111000101010`
- Linguagens de alto nível





POR QUE
PYTHON?



NEM TE CONTO . . .

LINGUAGEM MASSA

Simples e de fácil aprendizagem. Versátil e já conta com muitas bibliotecas e coisas prontas.

MUITO USADA

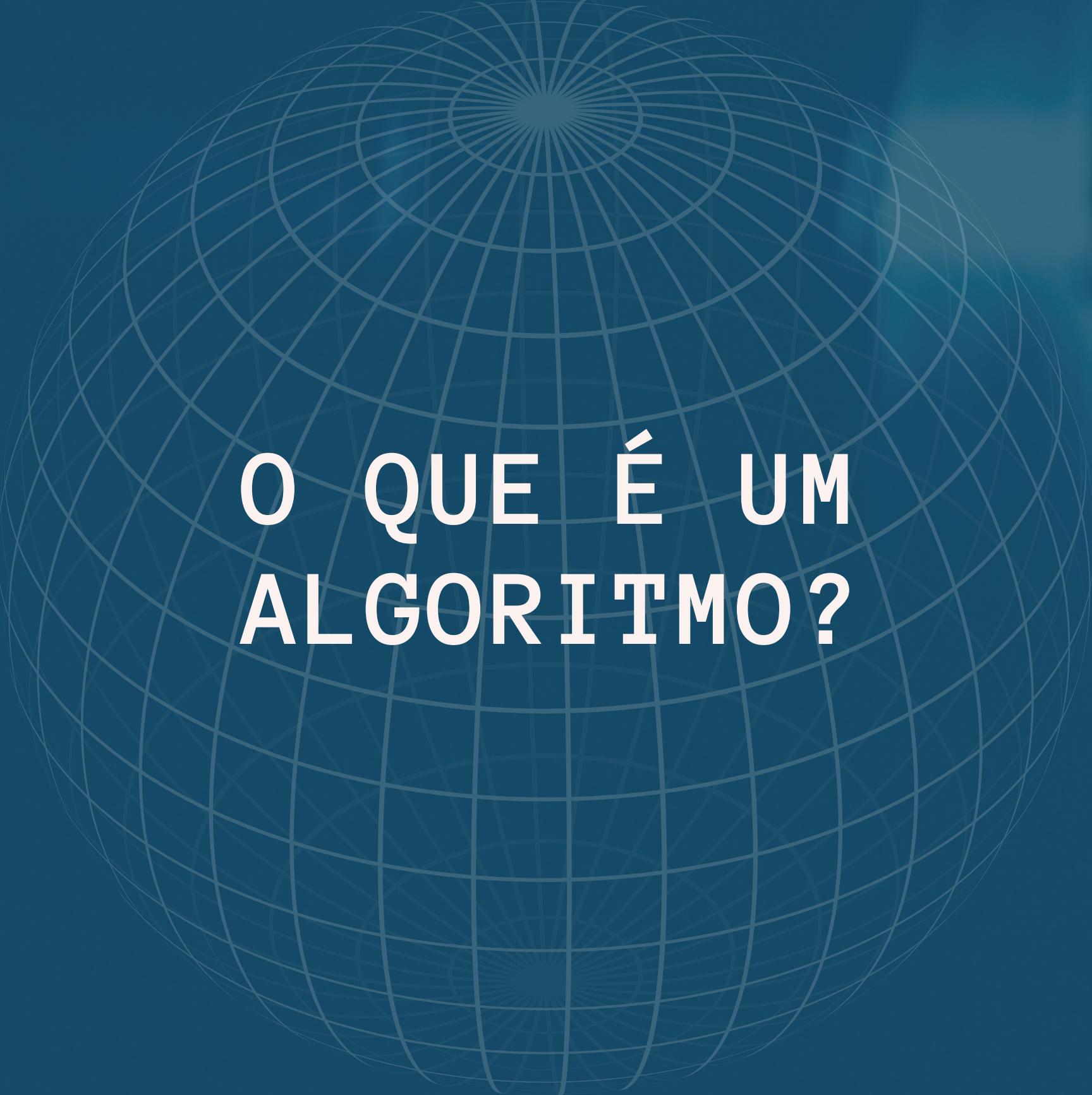
3ª linguagem de programação mais utilizada. Empresas como Google, Microsoft, Facebook e Netflix utilizam.

ÓTIMA COMUNIDADE

Muitos posts e tutoriais (vídeos e textos) sobre as mais diferentes formas de usar a linguagem.

FAZER PROJETOS !!

Tem de tudo na internet, então isso abre portas para a imaginação! Nem a internet é mais o limite.



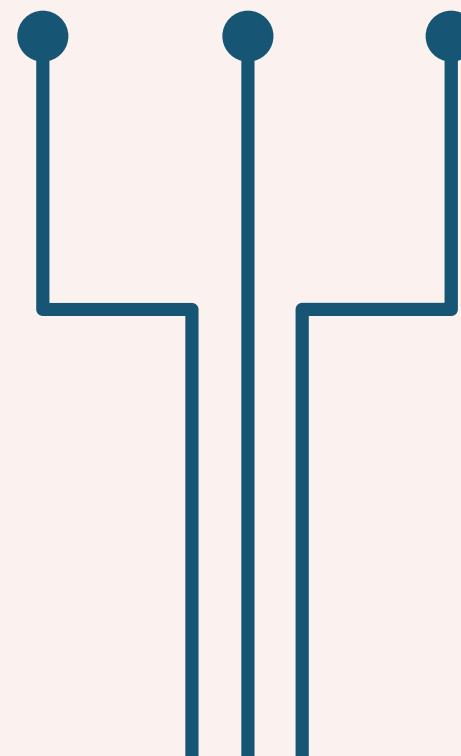
O QUE É UM
ALGORITMO?



VEREMOS

O QUE É UM ALGORITMO?

- Sequência de passos para resolver um problema
- Exemplo: receita de bolo





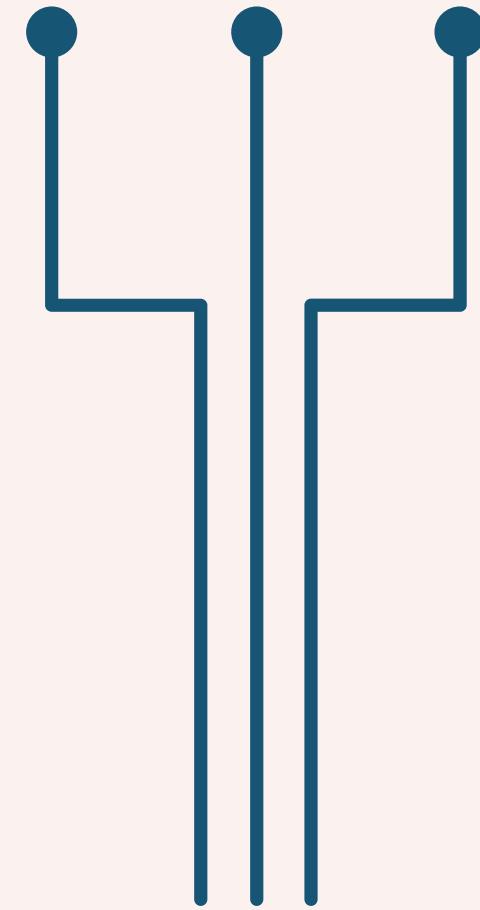
PYTHON ! !



FINALMENTE

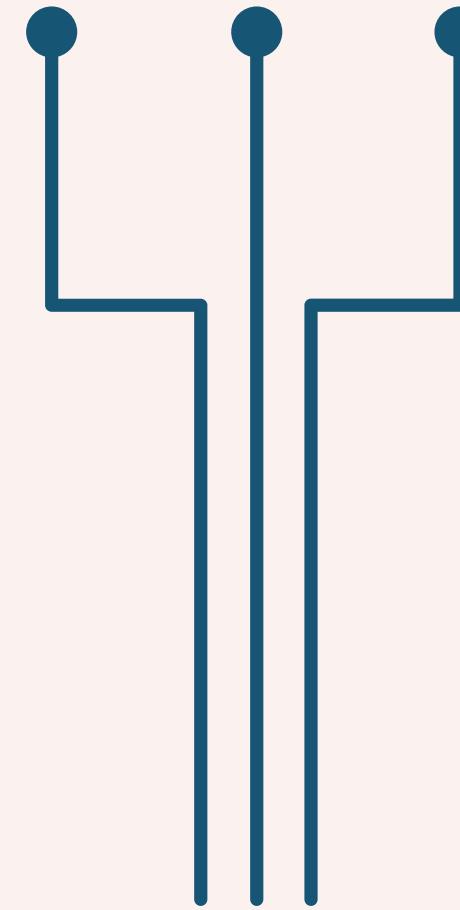
PRIMEIROS PASSOS

- Onde programar:
 - repl.it
 - VSCode
- O que é a Main?
- Como rodar o código?
 - python nome_do_arquivo.py



PRIMEIRO PASSO – DE VDD

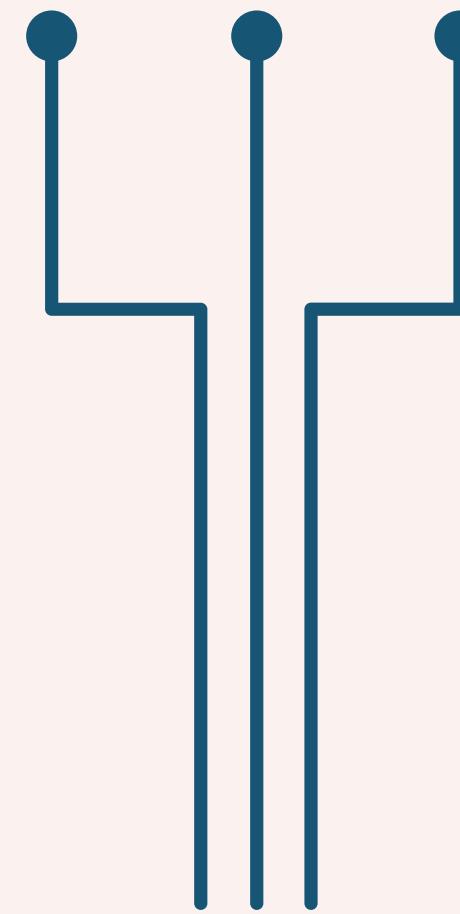
- Para que haja uma interação com o usuário, é possível imprimir valores na tela.
- Chama-se Saída.
- Utilizamos o comando **print()**
- Imprimir na tela "Hello World!"
 - `print('Hello World!')`



VARIÁVEIS

- Pequenos espaços de memória utilizados para armazenar dados.
- As variáveis podem ser de diferentes tipos e, no caso de Python, não é preciso declarar que tipo é esse.
- Como funciona:

```
nome = valor
```

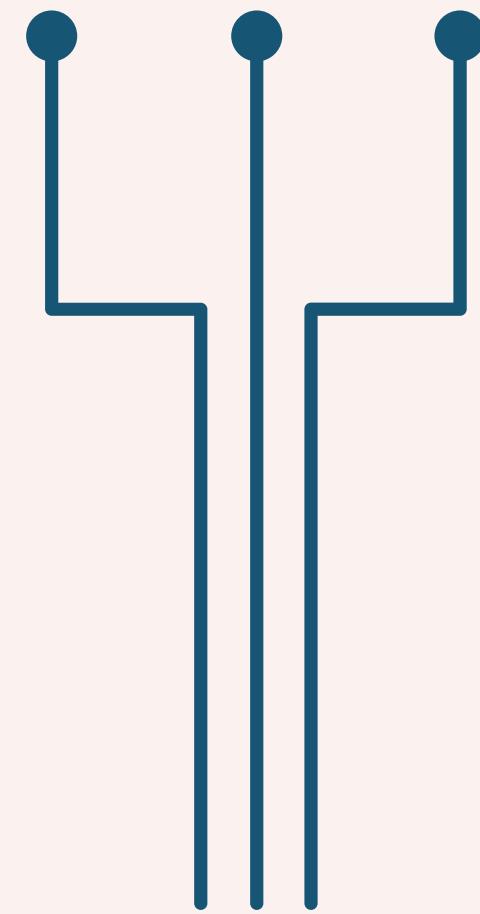


VARIÁVEIS

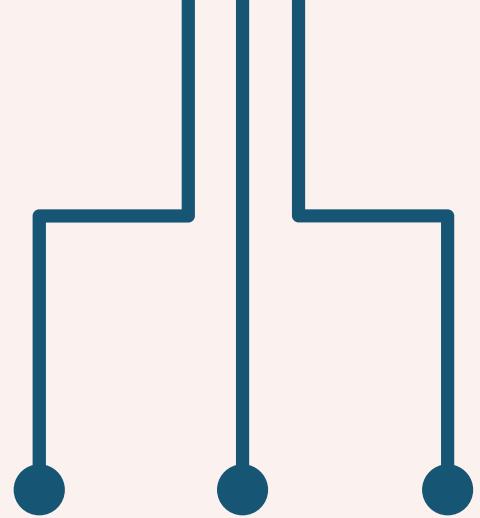
```
inteiro = 42  
print(inteiro)  
>>> 42
```

```
real = 3.14  
print(real)  
>>> 3.14
```

```
palavra = 'tchop'  
print(palavra)  
>>> 'tchop'
```



Variáveis - Prática



CRIAR DUAS VARIÁVEIS DE CADA TIPO

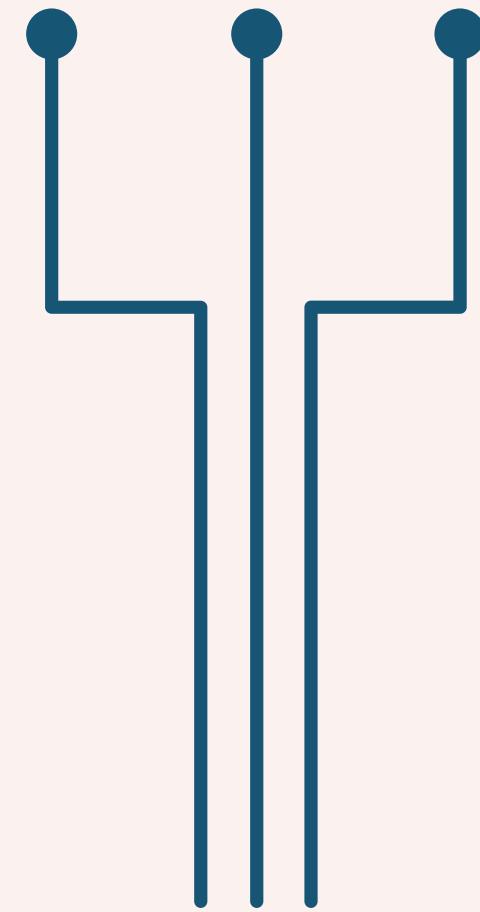
- INTEIRO
- NÚMERO REAL
- PALAVRA

LEMBRANDO QUE:

NOME = VALOR

ENTRADA

- É muito importante interagir com os usuários, recebendo dados como entrada.
- A atribuição de valor pode ser feita utilizando o comando `input()`.



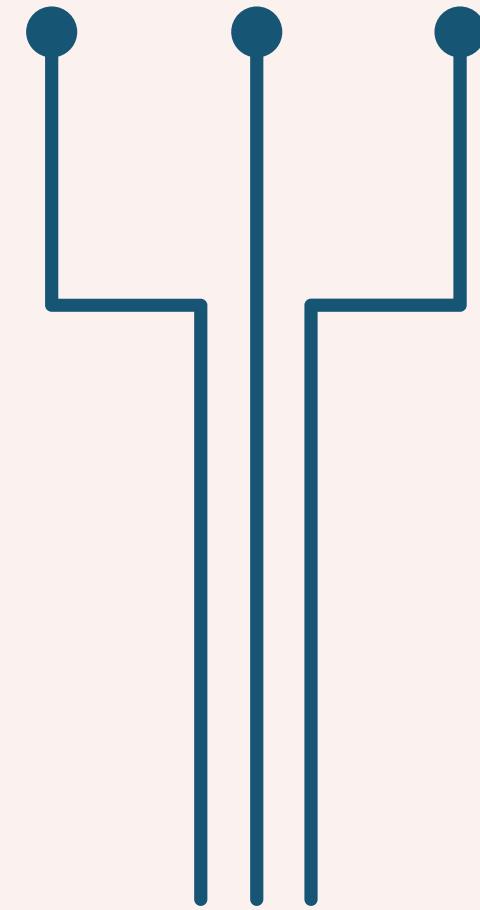
ENTRADA

```
nome = input()  
>>> luisa  
print('Meu nome é: ', nome)  
>>> 'Meu nome é luisa'
```

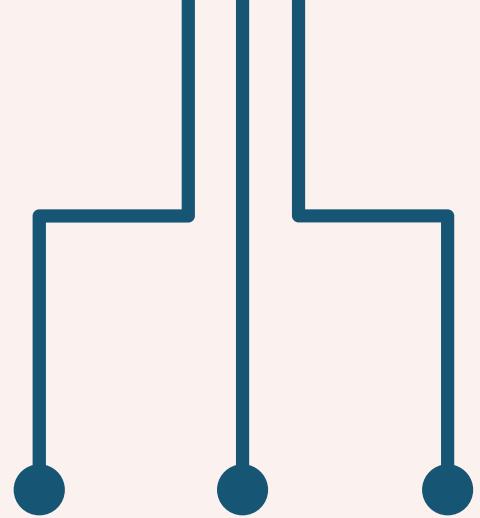
```
idade = int(input())  
>>> 22  
print(idade)  
>>> 22  


---

cidade = input('Digite sua cidade: ')  
>>> Digite sua cidade: Conquista  
print(cidade)  
>>> 'Conquista'
```



Entrada – Prática



RECEBER COMO ENTRADA

- SEU ANO DE NASCIMENTO
- NOME DO SEU ESTADO

IMPRIMIR NA TELA

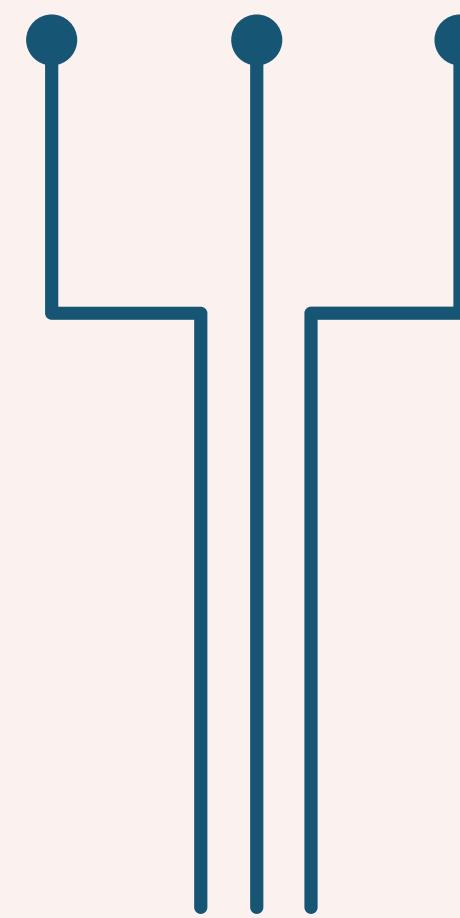
- NASCI EM <ANO> NO ESTADO <ESTADO>

LEMBRANDO QUE:

NOME = TIPO(INPUT())

OPERAÇÕES

- Em muitos casos precisamos manipular variáveis e valores, para isso podemos utilizar operações:
 - Adição +
 - Subtração -
 - Multiplicação *
 - Divisão /
 - Divisão inteira //
 - Resto %
 - Lógicos and, or e not
- Algumas das operações também são válidas para *strings*.

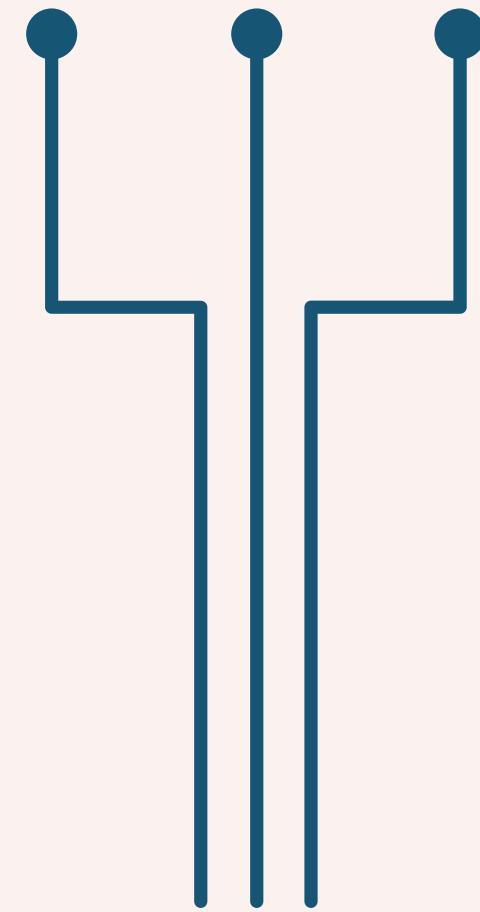


OPERAÇÕES

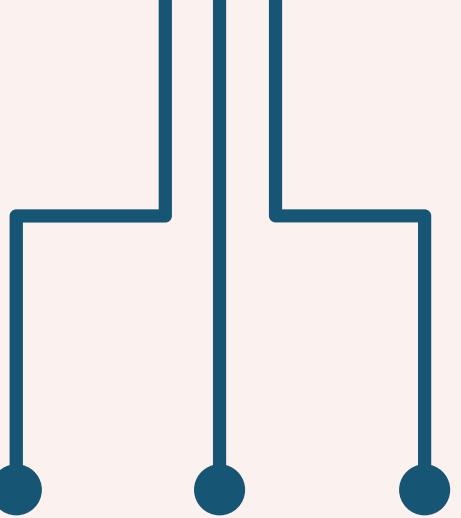
```
altura = 2  
largura = 3  
area = altura*largura  
print(area)  
>>> 5
```

```
nome = 'Luisa'  
sobrenome = 'Moura'  
nome_completo = nome + ' ' + sobrenome  
print(nome_completo)  
>>> 'Luísa Moura'
```

```
total = 50.50  
pessoas = 10  
print('Valor por pessoa: ', total/pessoas)  
>>> 'Valor por pessoa: 5.05'
```



Operações - Prática



1

-
- RECEBER COMO ENTRADA
- COTAÇÃO DO DÓLAR
 - VALOR EM REAIS

IMPRIMIR NA TELA

- VALOR EM DÓLARES

2

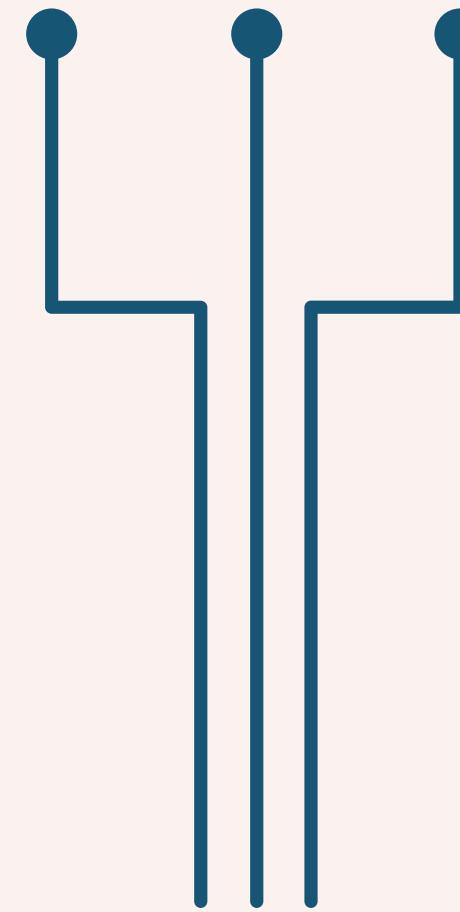
-
- RECEBER COMO ENTRADA
- DOIS NÚMEROS

IMPRIMIR NA TELA:

- MÉDIA DOS NÚMEROS

CONDIÇÕES

- Em muitos casos precisamos de estruturas para decidir o fluxo de execução do algoritmo.
- Exemplo:
 - Se não tiver manteiga, use óleo.
- Instruções:
 - if
 - elif
 - else
- Operações:
 - >
 - <
 - ==
 - not
- Indentação!!



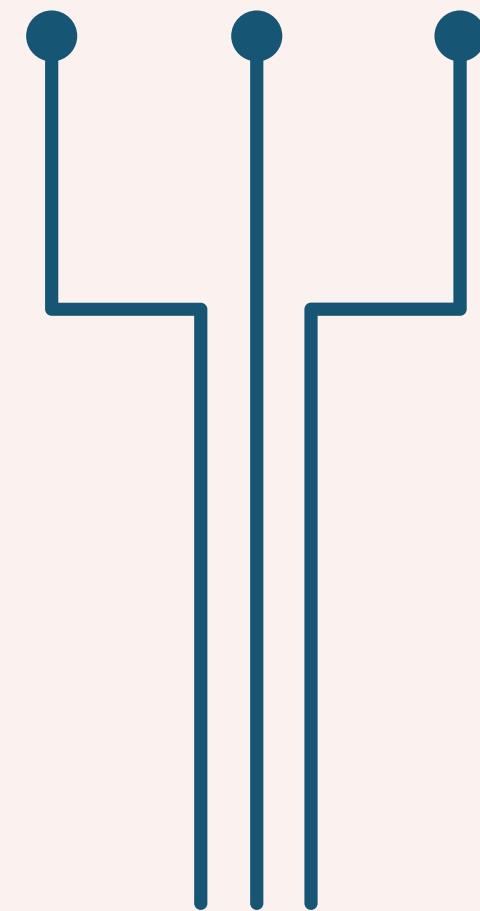
CONDIÇÕES

```
idade = 17
```

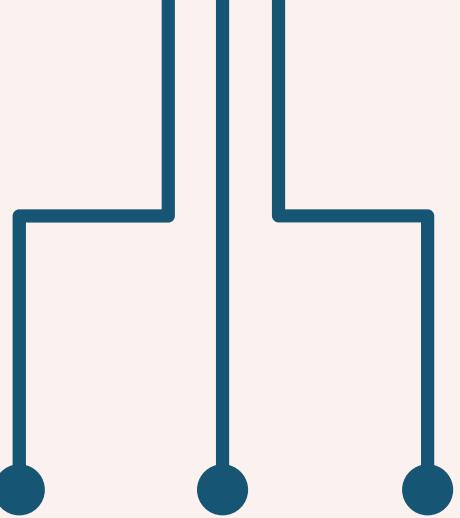
```
if idade >= 18:  
    print('A pessoa pode dirigir')  
else:  
    print('A pessoa não pode dirigir')
```

```
qtd_pessoas = 1
```

```
if qtd_pessoas == 1:  
    print('Ta valendo')  
elif qtd_pessoas == 2:  
    print('Ainda ta valendo')  
else:  
    print('Não vale mais')
```



Condições - Prática



1

RECEBER COMO ENTRADA

- UM NÚMERO INTEIRO

IMPRIMIR NA TELA

- SE O NÚMERO FOR POSITIVO: "YAY"
- SE NÃO FOR: "AFF"

2

RECEBER COMO ENTRADA

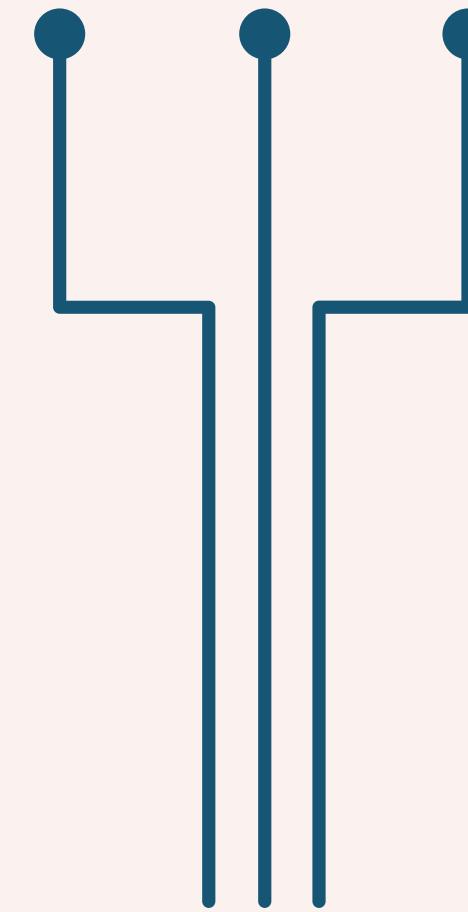
- UM NOME

IMPRIMIR NA TELA

- SE FOR SEU NOME OU SOBRENOME: "SOU EU"
- SE NÃO FOR NENHUM: "NÃO SOU EU"

REPETIÇÃO

- Executar a mesma instrução várias vezes
- Exemplo:
 - Calcular a média das notas de cada aluno.
- Instruções:
 - `for`
 - `while`
- Indentação!!



REPETIÇÃO

```
saldo = 100
```

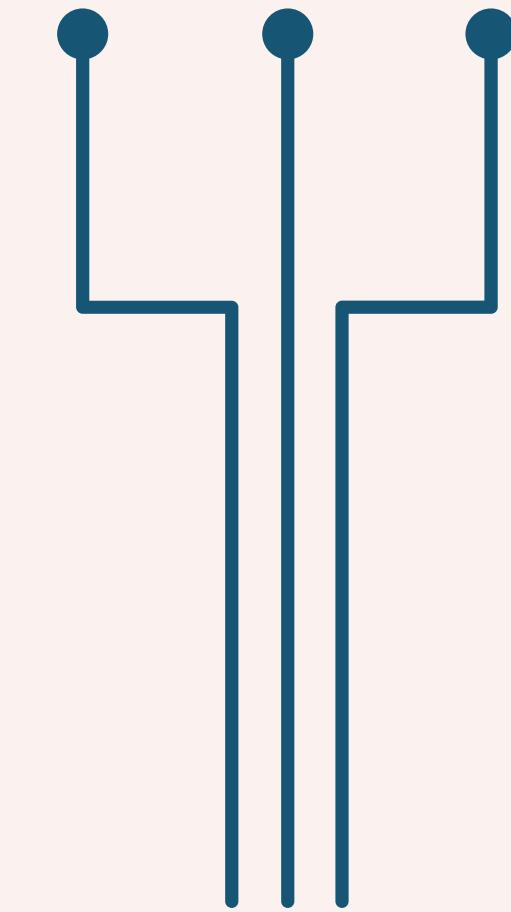
while <condição verdadeira>:
 comandos

```
while saldo > 0:  
    saldo -= 10
```

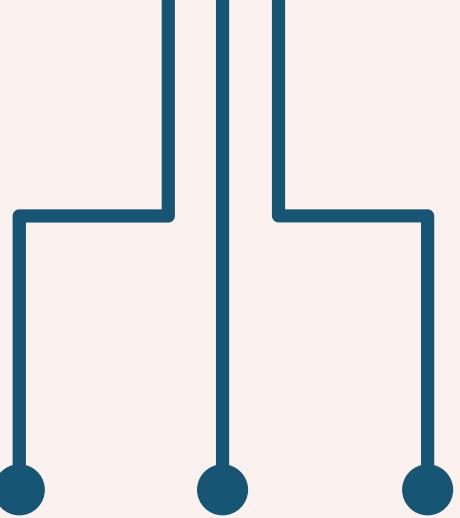
```
soma = 0
```

for <variável> in range(<inicio>, <fim>):
 comandos

```
for num in range(0, 10):  
    soma += num
```



Repetição - Prática



1

RECEBER COMO ENTRADA
• UM NÚMERO INTEIRO

IMPRIMIR NA TELA
• TODOS OS NÚMEROS MENORES QUE ELE

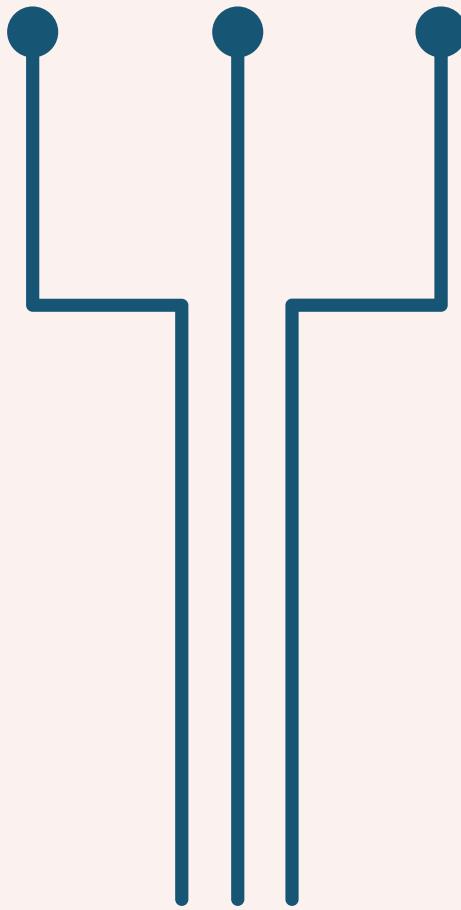
2

RECEBER COMO ENTRADA
• UM NÚMERO INTEIRO

IMPRIMIR NA TELA
• SUA TABUADA

LIS~~T~~AS

- Estrutura de dados para armazenar vários valores.
- Como uma array/vector
- Acesso por posição/índice
 - Começa no 0!!



LIS~~T~~AS

```
alunos_presentes = ['Luisa', 'David', 'Zé', 'Ana']
```

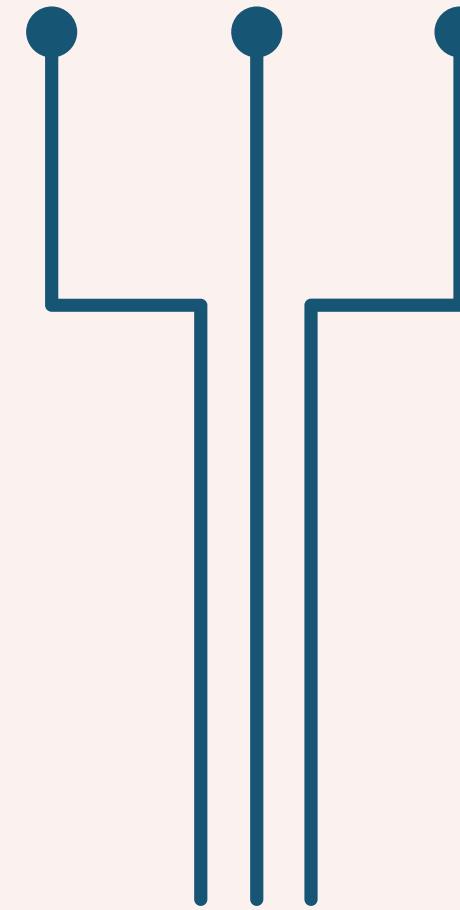
```
primeiro = alunos_presentes[0]  
ultimo = alunos_presentes[3]
```

```
lista_notas = [9, 8, 10, 7]  
soma_notas = 0
```

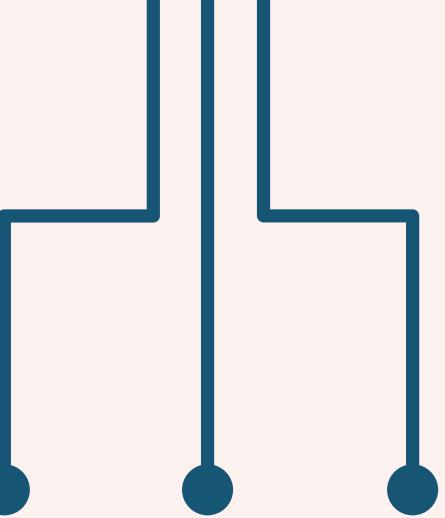
```
for nota in lista_notas:  
    soma_notas += nota
```

```
media = soma_notas / len(lista_notas)
```

repetição na lista: passar por todos os elementos
`for <variavel> in <lista>:
 comandos`



Listas - Prática



1

CRIAR UMA LISTA DE NUMEROS
ENCONTRAR O MENOR NÚMERO
IMPRIMIR NA TELA
• O MENOR NÚMERO

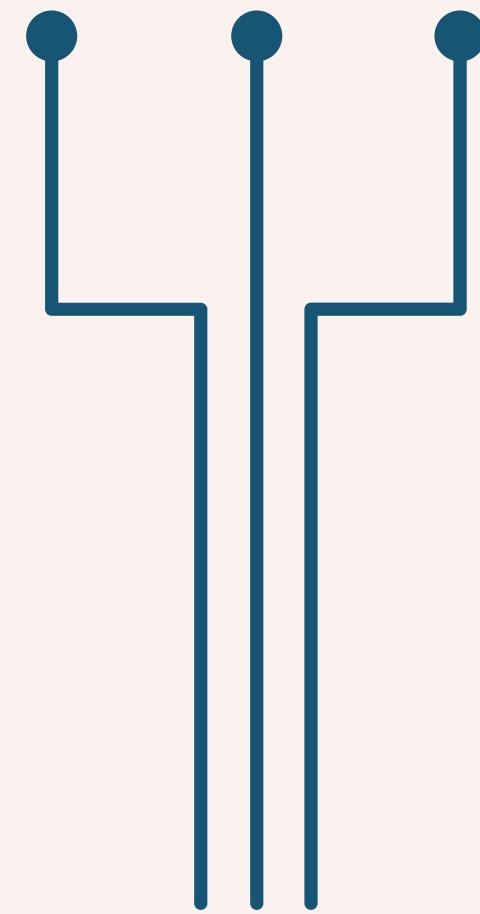
2

RECEBER COMO ENTRADA
• UM NUMERO X
• X NOMES

IMPRIMIR NA TELA
• A QUANTIDADE DE LETRAS DE CADA NOME

DICIONÁRIOS

- Estrutura de dados para armazena pares de valores, dos quais um é chave e outro valor.
- Acesso por chave
 - Não precisa ser numérica



DICIONÁRIOS

```
localizacao = {'Brasil': 'América',  
               'Portugal': 'Europa',  
               'Espanha': 'Europa'}
```

```
continente_brasil = localizacao['Brasil']
```

```
precos = {'Banana': 2,  
          'Sapato': 100,  
          'Gasolina': 10}
```

```
total_precos = 0
```

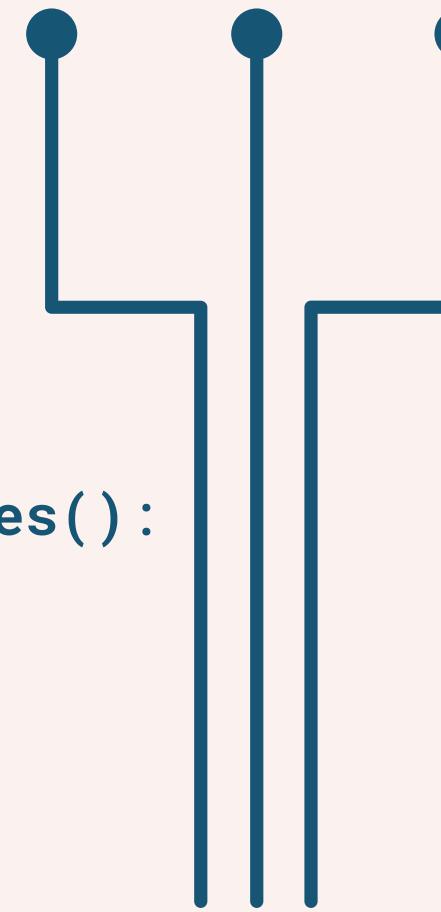
```
for produto in precos:  
    total_precos += precos[produto]
```

- passar por todas as chaves

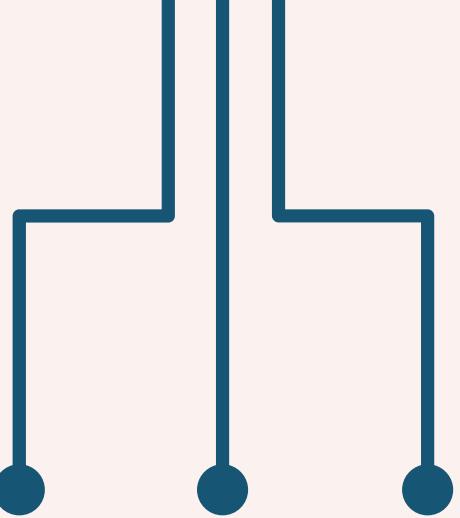
for <variavel> in <dict>:
 comandos

- passar por todos os valores

for <variavel> in <dict>.values():
 comandos



Dicionário - Prática



1

CRIAR UM DICIONARIO QUE REPRESENTA
CARACTERÍSTICAS DE UM ABIENTE

IMPRIMIR NA TELA CADA UMA DELAS:

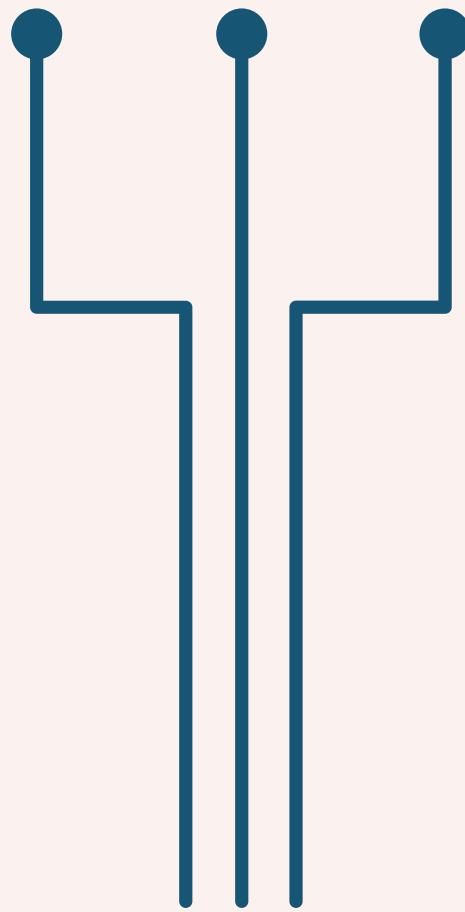
- ACESSANDO-AS PELA CHAVE
- UTILIZANDO REPETICAO

2

– CRIAR UM DICIONÁRIO COM NOMES DE PESSOAS
E QUANTO DINHEIRO ELAS TEM
– CALCULAR A SOMA DOS VALORES
– RECEBER UM VALOR COMO ENTRADA
– IMPRIMIR SE É POSÍVEL COMPRAR UM PRODUTO
NO VALOR RECEBIDO COMO ENTRADA

FUNÇÕES

- Estrutura para encapsular tarefas.
- Após a declaração da função, o programador define tudo que ela fará.
- Já usamos algumas funções!
 - `print()`
 - `input()`



DICIONÁRIOS

```
def area_quadrado(lado): → definição da função
```

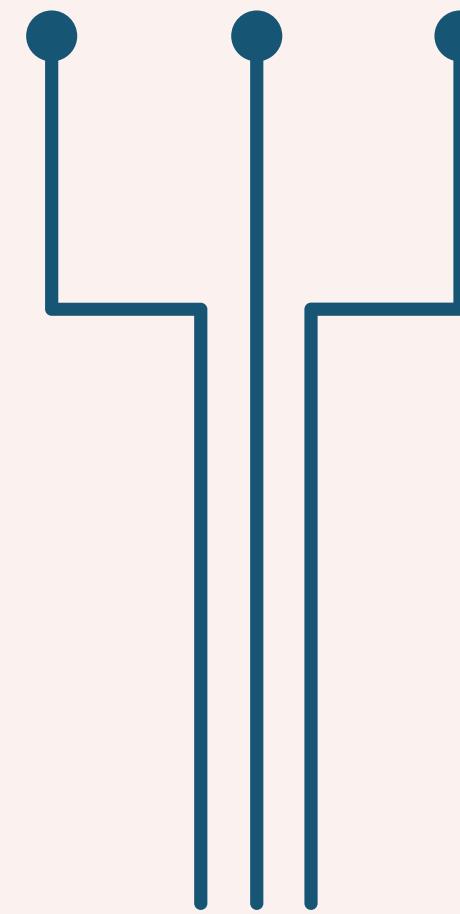
```
    area = lado*lado  
    return area
```

```
area = area_quadrado(10)
```

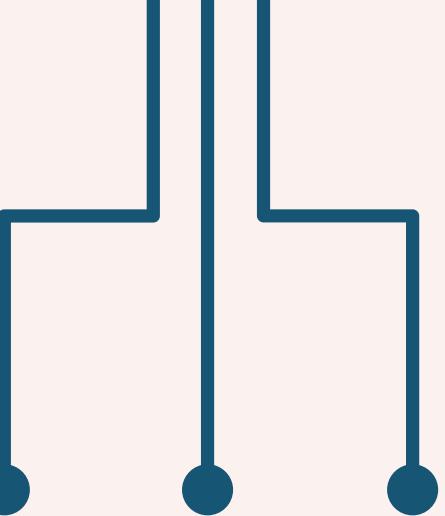
```
def area_triangulo(base, altura):
```

```
    area = base*altura  
    return area →
```

- retorno da função
- forma de comunicação da função com o código principal



Dicionário - Prática



1

CRIAR UMA FUNÇÃO QUE RECEBE UMA LISTA DE NÚMEROS E RETORNE A MÉDIA DELES

2

CRIAR UMA FUNÇÃO QUE RECEBE UM NÚMERO E RETORNA UM BOOLEANO INDICANDO SE O NÚMERO É PAR OU NÃO

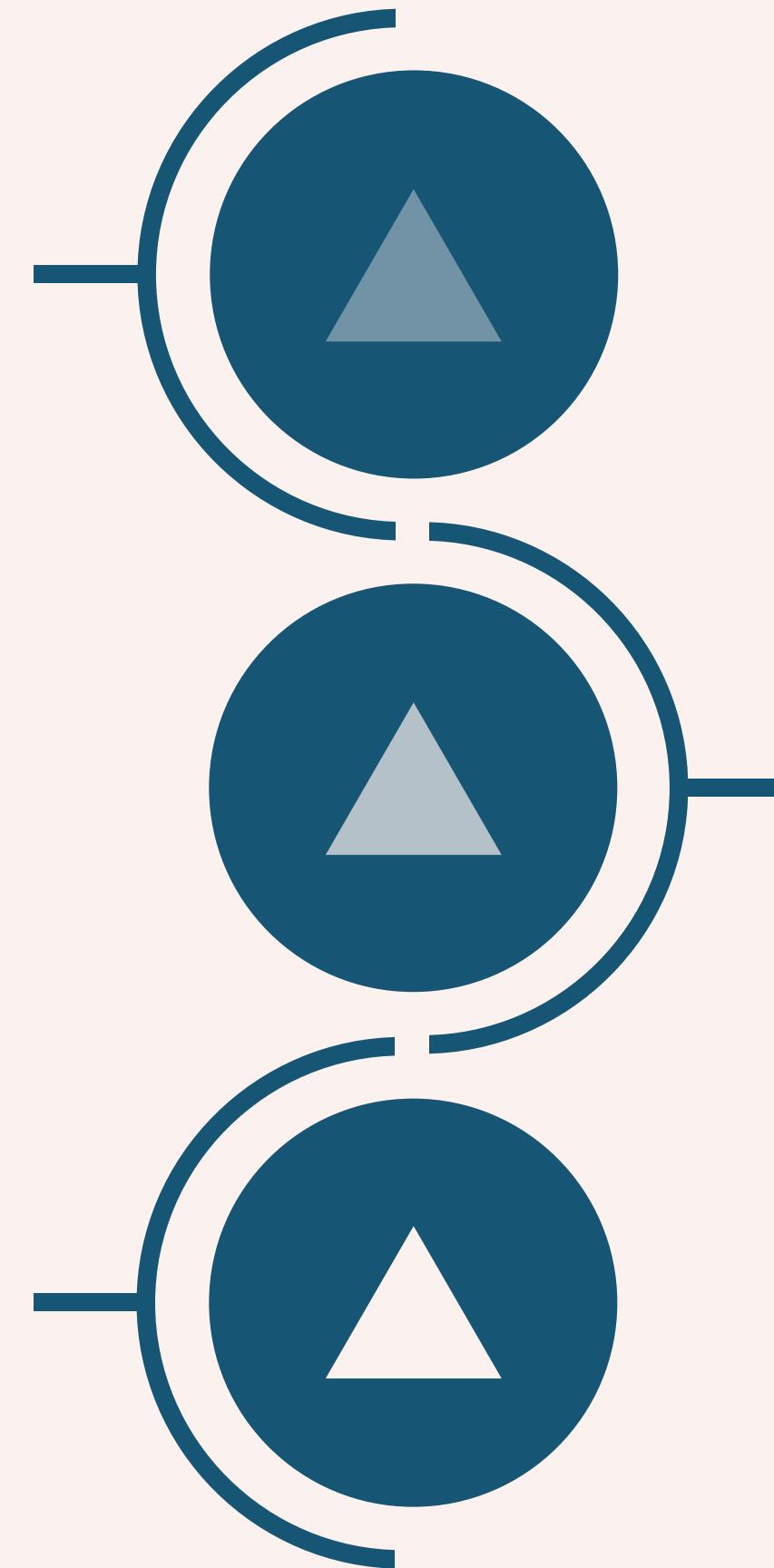


CONSIDERAÇÕES FINAIS

UHUUUUUUU

COMEÇAR COM O SIMPLES

Apesar de existirem muuuitas ferramentas, a mais simples pode ser a mais poderosa



NÃO PARAR POR AQUI

Agora você tem toda a internet para explorar!!

TESTAR, TESTAR, TESTAR

Infelizmente a gente só sabe o que funciona testando, então isso é um passo super importante

MUITO OBRIGADA!!

“

LUÍSA MOURA

Só vai

REFERÊNCIAS SUPER TOPS PRA CONTINUAR

- http://antigo.scl.ifsp.edu.br/portal/arquivos/2016.05.04_Apostila_Python_-_PET ADS_S%C3%A3o_Carlos.pdf
- <https://www.cursoemvideo.com/curso/python-3-mundo-1/>
- <https://www.cursoemvideo.com/curso/python-3-mundo-2/>
- <https://www.cursoemvideo.com/curso/python-3-mundo-3/>
- Como instalar python em video: <https://www.youtube.com/watch?v=pDBnCDuL-dc>
- Como instalar vscode em video: https://www.youtube.com/watch?v=_R6Ys1WRUFk