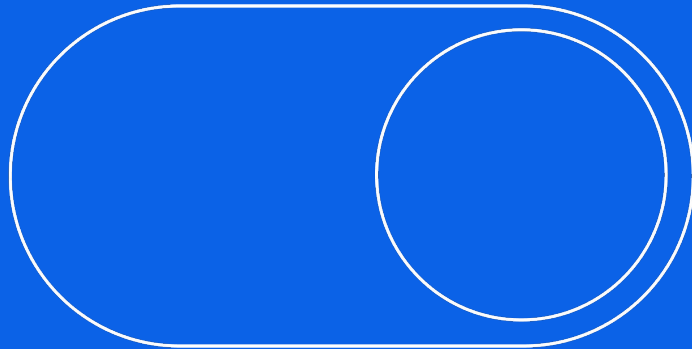


# Funções e pacotes

Módulo 1 Semana 4

FORMAÇÃO DESENVOLVEDOR  
PYTHON

ultima.



# Sherlon Almeida

Doutorando em Ciência da Computação e  
Matemática Computacional | USP

Mestre em Ciência da Computação | USP

Bacharel em Ciência da Computação

ultima.



# Vamos começar?

- Acomode-se em um lugar confortável
- Tenha água e um snack à mão
- Pegue caneta e caderno para fazer anotações
- Confira a bateria de seu dispositivo e carregue se for preciso

**ultima.**



# Agenda de Hoje

1. O que vimos até agora?
2. Tarefa da Semana
3. Dúvidas

ultima.



# O que vimos até agora?

- Noções básicas sobre funções e parâmetros
- Bibliotecas

ultima.





ultima.

# Funções

Como estruturar a implementação?



# Funções

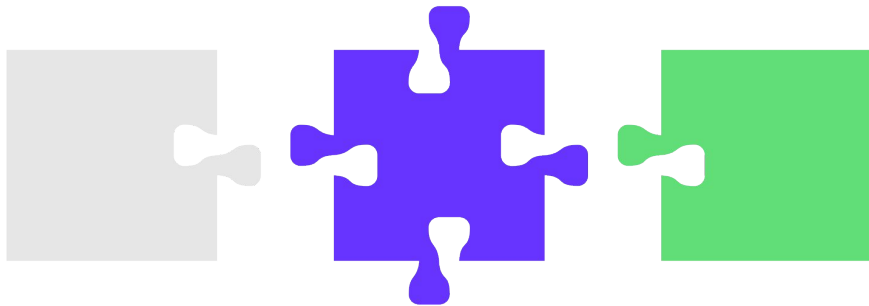


- Funções são **blocos de código** que realizam tarefas específicas;
- Permitem melhor **organização** e **reutilização** de código;
- Imagine-as como cada **peça** que compõe um Quebra-Cabeça.
- Programas **bem escritos** são compostos por funções.

Ler entrada

Soma

Mostrar Saída

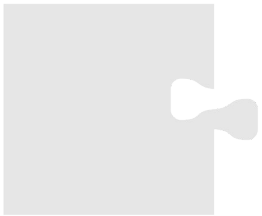


# Funções



## Exemplos

Ler entrada



Função input()

```
entrada = input ("Digite alguma coisa: ")
```

```
entrada = input ("Digite alguma coisa: ").split( )
```

```
entrada = int ( input ("Digite um número: ") )
```

```
entrada = float ( input ("Digite um número: ") )
```

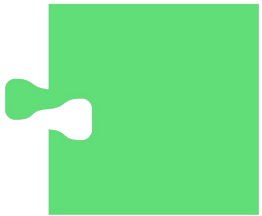


# Funções



## Exemplos

Mostrar Saída



Função print()

```
print ("Olá mundo!")
```

```
print ("Somando 5 + 3 = ", 5 + 3)
```

```
saida = "Imprimindo na tela"
```

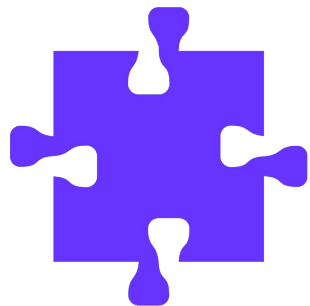
```
print ( saida )
```

# Funções



## Exemplos

Soma



Função soma()

```
resultado = soma (2, 5)
```

```
resultado = soma (-5, -4)
```

```
x = 8
```

```
y = 13
```

```
resultado = soma (x, y)
```

# Funções



```
#Definindo uma função
def nome_da_funcao ( parametros ):
    #Realiza todos os comandos necessários
    return algum_resultado
```

# Funções

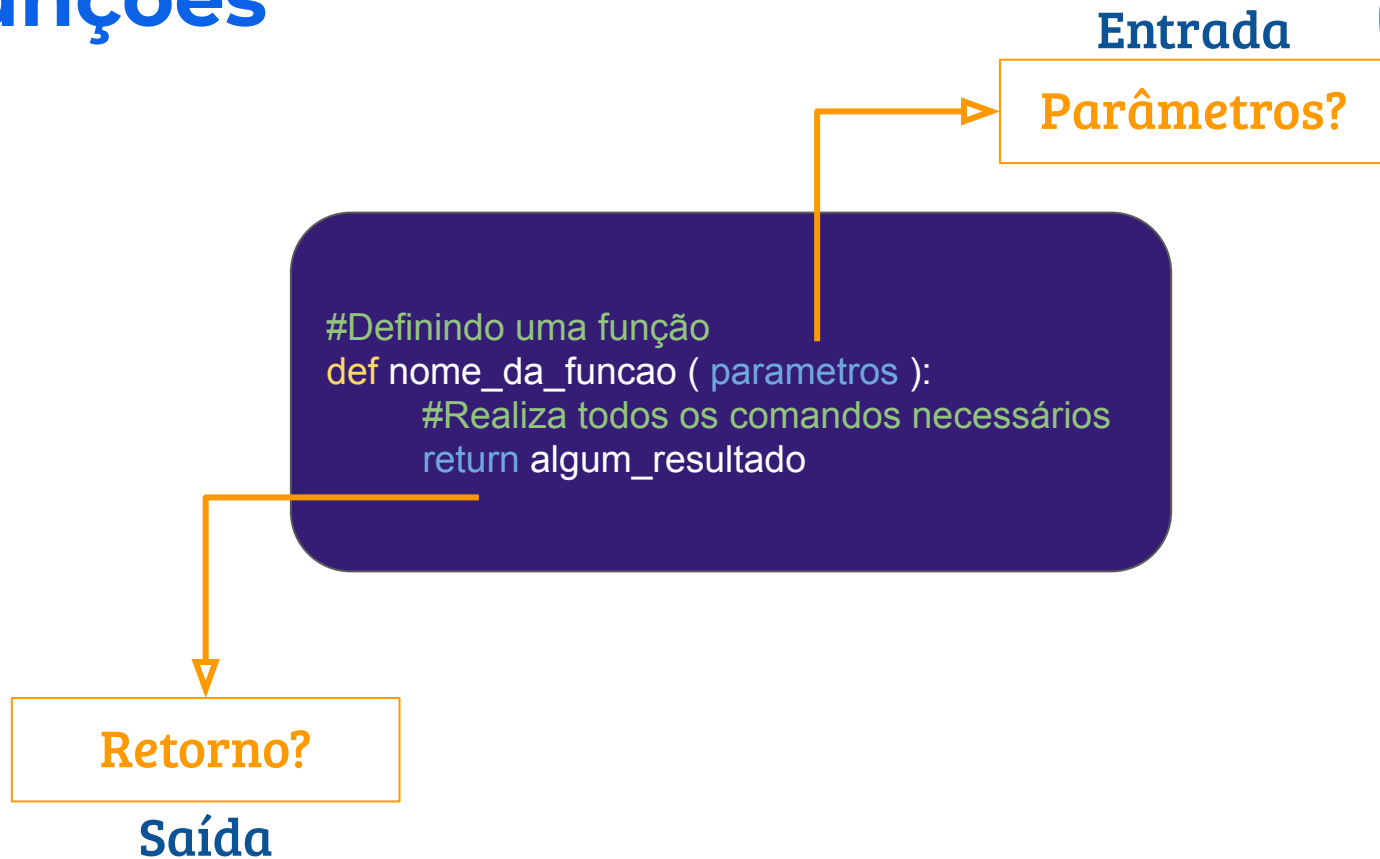


Parâmetros?

```
#Definindo uma função  
def nome_da_funcao ( parametros ):  
    #Realiza todos os comandos necessários  
    return algum_resultado
```

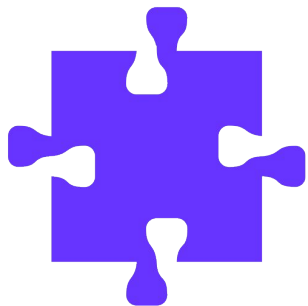
Retorno?

# Funções



# Funções

Soma



Função soma()

Função media3()

Função mediaN()

## Exemplos

#Função para somar 2 números

```
def soma (x, y):  
    return x + y
```

#Função para calcular média entre 3 números

```
def media3 (a, b, c):  
    resultado = (a + b + c) / 3  
    return resultado
```

#Função para calcular média entre N números

```
def mediaN (valores):  
    quantidade = len ( valores )  
    soma = 0  
    for i in range ( quantidade ):  
        soma = soma + valores [ i ]  
    resultado = soma / quantidade  
    return resultado
```



ultima.

# Instalando Bibliotecas

É preciso reinventar a roda?



# Bibliotecas

- É um local onde muitos livros são armazenados para serem usados posteriormente.
- Bibliotecas **implementam** e **disponibilizam** funções;
- Independentemente do que se deseja, é possível que exista uma **biblioteca pronta** a ser utilizada;
- Cada biblioteca possui uma **documentação** exemplificando o seu uso, com exemplos;
- É possível criar as **suas próprias** bibliotecas.





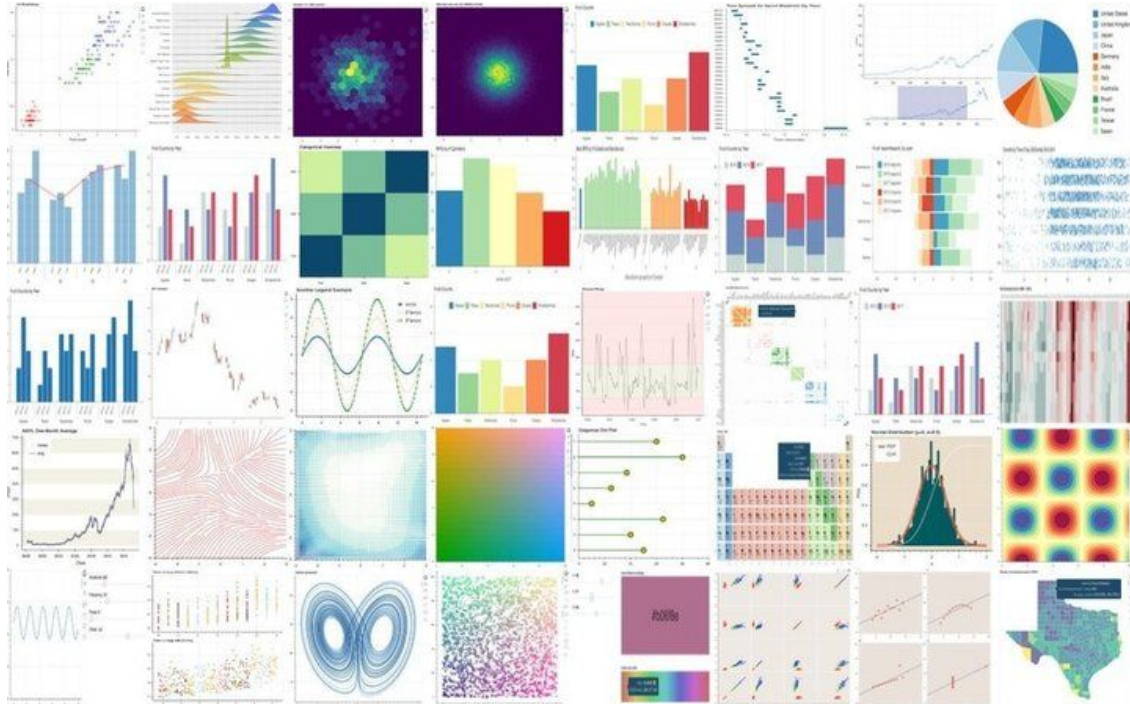
# Bibliotecas

```
import math
```

<code>ceil (x)</code>	Retorna o menor inteiro maior ou igual a x
<code>fabs (x)</code>	Retorna o valor absoluto de x
<code>floor (x)</code>	Retorna o maior inteiro menor ou igual a x
<code>exp (x)</code>	Retorna $e^{**} x$
<code>log2 (x)</code>	Retorna o logaritmo de base 2 de x
<code>log10 (x)</code>	Retorna o logaritmo de base 10 de x
<code>pow (x, y)</code>	Retorna x elevado à potência y
<code>sqrt (x)</code>	Retorna a raiz quadrada de x
<code>cos (x)</code>	Retorna o cosseno de x

# Bibliotecas

```
import matplotlib.pyplot as plt
```



# Bibliotecas

`import` QUALQUER COISA QUE VOCÊ QUISER

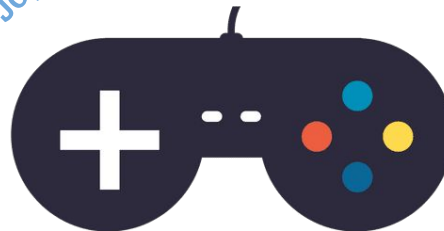
Vídeos



Áudio



Jogos



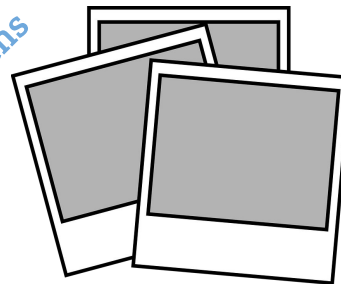
Mensagens



Inteligência Artificial



Imagens



# Instalação

- Anaconda
  - `conda install <biblioteca>`

PIP é um gerenciador de pacotes para projetos Python.

- Terminal
  - `pip install <biblioteca>`
- Colab
  - `%pip install <biblioteca>`



# Prática!

Os exercícios a seguir foram criados para te ajudar a fixar o conteúdo e praticar diretamente na plataforma tudo que foi visto durante a semana.

É muito importante que eles sejam realizados.

As soluções podem ser as mais simples possíveis.

**ultima.**



# Atividade 1

Crie uma função que, ao receber um número inteiro, retorna se um número é par ou ímpar.

## Dicas

- Utilize o comando `int(input())` para obter o número inteiro de entrada.
- Passe o número para uma função por parâmetro.
- Calcule e retorne o resultado.
- Mostre na tela se o número é 'Par' ou 'Impar', usando o comando `print()`.
- Extra: utilize `**kwargs` (Opcional)

ENTRADA	SAÍDA ESPERADA
2	Par
9	Impar
14	Par

ultima.

## Atividade 2

Crie uma função que recebe um número inteiro por parâmetro e então imprime na tela do número recebido até o zero.

### Dicas

- Utilize o comando `int(input())` para obter o número inteiro de entrada.
- Passe o número para uma função por parâmetro.
- Mostre na tela do número recebido até o zero, usando o comando `print()`.
- Extra: crie uma versão recursiva deste programa (Opcional)

ENTRADA	SAÍDA ESPERADA
2	2 1 0
5	5 4 3 2 1 0

ultima.

## Atividade 3

Crie um programa com uma função que necessite de três parâmetros e então que retorne a sua soma.

### Dicas

- Utilize o comando `int(input())` para obter os três números inteiros de entrada.
- Passe os três números para uma função por parâmetro.
- Calcule a soma e retorne o resultado.
- Mostre na tela a soma calculada, usando o comando `print()`.
- Extra: utilize `*args` (Opcional)

ENTRADA	SAÍDA ESPERADA
2 6 3	11
9 7 12	28
2 3 1	6



## Atividade 4

Crie um programa que seja capaz de obter e somar TODOS os números passados pelo usuário como entrada, permitindo somar qualquer quantidade de dados de entrada.

### Dicas

- Utilize uma estrutura de repetição para repetir a leitura de todos os números passados como entrada, até que encontre o valor '-1'.
- Ou seja, o número -1 será o critério de parada para a leitura dos dados de entrada.
- Utilize o comando `int(input())` para obter todos os números inteiros de entrada.
- Calcule a soma de todos os números.
- Mostre na tela a soma calculada, usando o comando `print()`.
- Extra: utilize Funções e `*args` (Opcional).

**ultima.**

ENTRADA	SAÍDA ESPERADA
2 7 -1	9
4 7 2 19 1 -1	33
2 8 5 13 4 3 16 -1	51

## Atividade 5

Faça um programa com uma função que necessite de um parâmetro. A função deve retornar "Positivo", se seu o número for maior que zero, "Negativo" se o número for menor que zero, e "Zero" se o número for igual a zero.

### Dicas

- Utilize o comando `int(input())` para obter o número inteiro de entrada.
- Implemente uma função que seja capaz de descobrir se um número é positivo, negativo ou zero, e retorne o resultado.
- Mostre na tela o resultado, usando o comando `print()`.

ENTRADA	SAÍDA ESPERADA
2	Positivo
-7	Negativo
0	Zero

ultima.

## Atividade 6

Escreva uma função que, dado o valor da conta de um restaurante e um percentual de taxa de serviço, calcule e exiba a gorjeta do garçom, considerando o percentual do valor da conta.

### Dicas

- Utilize o comando `float(input())` para obter o valor da conta.
- Utilize o comando `int(input())` para obter o valor da taxa de serviço.
- Implemente uma função que calcule a gorjeta do garçom, baseado no percentual do valor da conta definido.
- Mostre na tela o resultado com duas casas decimais, usando o comando `print(f"{resultado:.2f}")`.
- Lembrando que o cálculo de porcentagem é:  $\text{valor} * \text{porcentagem} / 100$ .

ENTRADA	SAÍDA ESPERADA
100 15	15.00
139.45 12	16.73
1529.32 27	412.92

## Atividade 7

Crie uma função que permita contar o número de vezes que aparece uma determinada letra em uma frase. A letra desejada e a frase a ser verificada serão passadas por parâmetro para a função, que retornará a quantidade da letra na frase.

### Dicas

- Utilize o comando `input()` para obter a letra desejada.
- Utilize o comando `input()` para obter a frase desejada.
- Implemente uma função que conte a quantidade de vezes que a letra aparece na frase e retorne este valor.
- Mostre na tela o resultado obtido, usando o comando `print()`.

ENTRADA	SAÍDA ESPERADA
a aaaaaa	6
p paralelepipedo	3
n banana	2

# Prática!

Agora é a vez de vocês!

Crie uma função que receba duas palavras e retorne "True" caso a primeira palavra seja um prefixo da segunda, e "False" caso contrário.

Em grupos, resolvam essa atividade, vocês tem 15 min.

Vamos lá?

**ultima.**



**E aí,  
como foi?**

**ultima.**



## Atividade em grupo

Crie uma função que receba duas palavras e retorne "True" caso a primeira palavra seja um prefixo da segunda, e "False" caso contrário.

Exemplo: 'programa' é prefixo de "programador", pois todas as letras de 'programa' correspondem às primeiras letras de "programador".

### Dicas

- Utilize o comando `input()` para obter a palavra1.
- Utilize o comando `input()` para obter a palavra2.
- Implemente uma função que identifique se a palavra1 é prefixo da palavra 2, e retorne o resultado obtido.
- Mostre na tela o resultado, usando o comando `print()`.
- Extra: utilize slicing para indexar as strings. (Opcional).

ultima.

## Exemplos de entrada e saída

ENTRADA	SAÍDA ESPERADA	EXPLICAÇÃO
programa programador	True	As 8 letras de programa são exatamente as 8 primeiras letras de programador, então é prefixo!
abcd abcdefghijkl	True	As 4 primeiras letras de "abcd" são exatamente as 4 primeiras letras de "abcdefghijkl", então é prefixo!
cana banana	False	Embora 3 das 4 letras de "cana" sejam iguais as 4 primeiras letras de "banana", era preciso que todas fossem iguais para ser prefixo. Ou seja, não é prefixo!
testa testemunha	False	Embora 4 das 5 letras de "testa" sejam iguais as 5 primeiras letras de "testemunha", era preciso que todas fossem iguais para ser prefixo. Ou seja, não é prefixo!



# Dúvidas

ultima.



**O que você achou da aula?**

**Conta para gente!**

ultima.

[Responda neste Form ↗](#)



**Boa noite,**  
**e até a próxima!**

**Obrigado!**

ultima.

