



IN

dev full stack- Python

Aula 1 - função

Funções

- Sequência de instruções que computa um ou mais resultados que chamamos de parâmetros
- Podem ser reaproveitados por diversas plataformas diferentes
- Ex:
 - `input()`
 - `print()`
 - `format()`
 - `type()`

Pense comigo...

$$v_m = \frac{\Delta x}{\Delta t} \rightarrow v_m = \frac{x_2 - x_1}{t_2 - t_1} = \frac{100 - 10}{10 - 0} = \frac{90}{10} = 9 \text{ m/s}$$



t = 10s



A
10m

B
100m

$$\text{velocidade média} = \frac{\Delta s}{\Delta t} \rightarrow v_m = \frac{\Delta s}{\Delta t}$$

Pense comigo...

- Quais são os recursos que preciso para calcular a velocidade média do carro?

Parâmetros

Pense comigo...

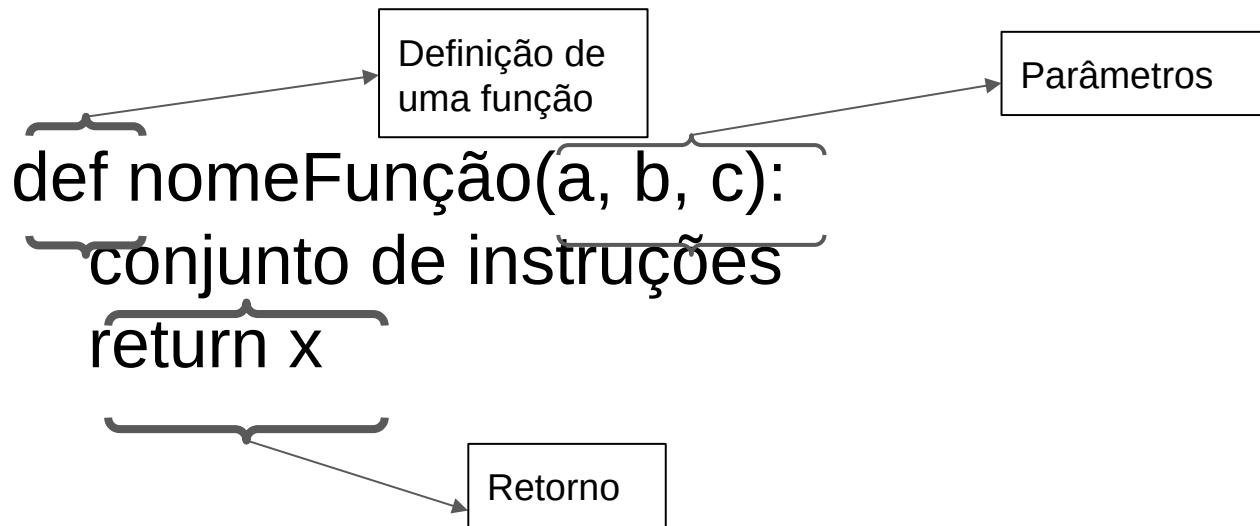
- Qual o resultado final que espero após realizar a função?

Retorno

Quando usar função?

- Trechos de código do que se repetem;
- Ações/tarefas específicas para um determinado objetivo;

Estrutura da Função



Funções

São formadas por:

- **Parâmetros**
- **Retorno**



Funções embutidas

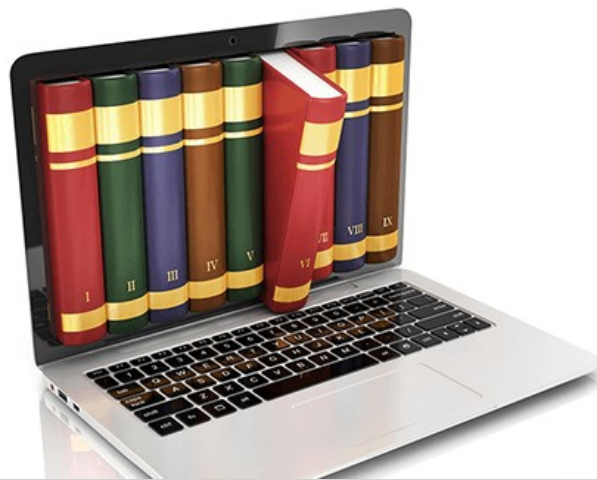
- Não são necessárias incluir o “import”
- Estão embutidas no interpretador
- Ex:
 - `print()`
 - `input()`
 - `int()`
 - `float()`
 - `format()`

Funções embutidas

<code>abs()</code>	<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>
<code>all()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>any()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>ascii()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>bin()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bool()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>breakpoint()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	

Bibliotecas Padrão

- Tipos de dados
- Módulos Matemáticos e Numéricos
- Acesso a arquivos e diretórios
- Manuseio de dados da internet
- Muitos outros...



Exemplo

```
{ def calc_velocidade_media(distancia, tempo):  
    return distancia/tempo }
```

Função definida

```
d = float(input("Informe a distância percorrida em quilômetros: "))  
t = int(input("Informe o tempo percorrido em horas: "))  
r = calc_velocidade_media(d, t) Chamada da função  
print("A velocidade média foi de: {:.2f}".format(r))
```

Parâmetros da função

- Lista com nenhum ou mais elemento que podem ser obrigatórios ou opcionais;
- Passa valores para a função sem necessidade de serem solicitados dentro dela;
- Os parâmetros em python são passados valores por referência.

Parâmetros opcionais

- Quando o parâmetro é opcional, pode-se atribuir valores e permitir que o parâmetro opcional seja nulo;
- Ex: medida para saber se virá em m/s ou em km/h

km/h -> m/s (/3,6)

```
def calc_velocidade_media(distancia, tempo, medida = None):  
    if (medida != None):  
        if(medida.lower() == "m/s"):  
            return distancia/tempo * 3.6  
    return distancia/tempo  
  
d = float(input("Informe a distância percorrida:"))  
t = int(input("Informe o tempo percorrido: "))  
m = input("Informe as unidades que você informou (km/h ou m/s)")  
r = calc_velocidade_media(d, t, m)  
print("A velocidade média foi de: {:.2f}".format(r))
```

```
Informe a distância percorrida:40  
Informe o tempo percorrido: 2  
Informe as unidades que você informou (km/h ou m/s)m/s  
A velocidade média foi de: 72.00  
...
```

```
Informe a distância percorrida:40  
Informe o tempo percorrido: 2  
Informe as unidades que você informou (km/h ou m/s)km/h  
A velocidade média foi de: 20.00  
,
```

Retorno

- Retorna um ou mais valores para quem chamou a função.

Retorno

Ex:

```
def soma(a, b):  
    return (a+b)
```

```
var1 = int(input("Informa o primeiro valor"))  
var2 = int(input("Informa o segundo valor"))
```

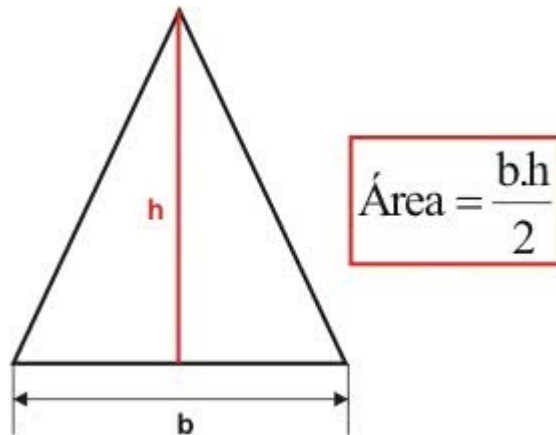
```
r = soma(var1, var2)  
print(r)
```

Atividade 1

Crie um algoritmo em Python, que dado dois números informados pelo usuário e sua operação (das 4 operações básicas da matemática), realize os cálculos adequados dentro de funções.

Atividade 2

Elaborar um algoritmo que calcule a área de triângulos , quantos o usuário quiser calcular
separe a função com o nome “calcula_triângulo”.



Argumentos default

- É possível dar valores default a argumentos;
- Se o chamador não especificar valores para esses argumentos, os defaults são usados;
- Formato: `def nome (arg1=default1, ..., argN=defaultN)`
- Se apenas alguns argumentos têm default, esses devem ser os últimos

Argumentos default

Ex:

```
def calc_peso_ideal(alt, sexo = 'f'):
    if(sexo.lower() == 'f'):
        return 62.1 * alt - 44.7
    else:
        return 72.7 * alt - 58

altura = float(input("Informe a sua altura: "))
sexo = input("Informe o seu sexo (m/f)")
print("O seu peso ideal é: ", calc_peso_ideal(altura, sexo))
```

Documentação de funções

- Ao invés de usar comentários para descrever o que uma função, é mais vantajoso usar docstrings
- Uma constante string escrita logo após o cabeçalho da função (comando def)
- Permite o acesso à documentação a partir do interpretador, usando a notação função . `__doc__`

```
def calc_peso_ideal(alt, sexo = 'f'):
    "Esta função realiza o calculo do peso ideal baseado na fórmula 62.1 x alt
    if(sexo.lower() == 'f'):
        return 62.1 * alt - 44.7
    else:
        return 72.7 * alt - 58
```

```
altura = float(input("Informe a sua altura: "))
sexo = input("Informe o seu sexo (m/f)")
print("O seu peso ideal é: ", calc_peso_ideal(altura, sexo))
```

```
>>> calc_peso_ideal.__doc__
'Esta função realiza o calculo do peso ideal baseado na fórmula 62.1 x alt - 44.7'
```

Retorno de mais de um valor

Em Python podemos retornar várias variáveis, objetos, expressões , etc;

```
def cadastro():  
    name = input("Qual seu nome: ")  
    age  = int(input("Idade: ") )  
  
    return name, age  
  
print("Iniciando cadastro...")  
nome, idade = cadastro()  
  
print("Cadastro realizado com sucesso:")  
print("Seu nome é", nome, "e você tem", idade, "anos de idade.")
```


Atividade 3

Faça um procedimento que recebe 3 valores inteiros por parâmetro e retorna-os ordenados em ordem crescente.

Atividade 4

Escreva uma função que recebe, por parâmetro, um valor inteiro e positivo e retorna o somatório desse valor

Atividade 5

Faça uma função que receba um valor inteiro e positivo e calcula o seu fatorial.

Atividade 6 (desafio)

Crie uma função que dado uma data digitada pelo usuário, informe há quantos dias atrás aconteceu aquela data.

Leitura Complementar

Como ordenar listas e dicionários (importante: o link vai mais longe do que o que foi demonstrado em sala):

<https://academiahopper.com.br/lambda-em-python-como-usar-para-ordenar-de-listas-e-dicionarios/>

A estrutura de repetição FOR usando RANGE():

<http://excript.com/python/for-in-range-python.html>

Trabalhando com datas em Python:

<https://vaiprogramar.com/como-trabalhar-com-data-hora-python-datetime/>

The logo consists of the letters 'IN' in a white, serif font, centered within a solid red square. The background of the entire image is a solid red color with abstract, darker red geometric shapes and lines creating a sense of depth and movement.

IN

71 3901 1052 | 71 9 9204
0134

@infinity.school

www.infinityschool.com.br

Salvador Shopping Business | Torre Europa Sala 310
Caminho das Árvore, Salvador - BA CEP: 40301-155