



# PROGRAMAÇÃO EM PYTHON



python



AULA 09

# EXPRESSÕES LAMBDA

- O Python permite a criação de funções anônimas (funções que não estão vinculadas a um nome) em tempo de execução usando o construtor "lambda".
- Ao executar uma expressão lambda o Python retorna uma função, ao invés de atribuí-la à um nome como acontece quando usamos def, por isso são anônimas.
- Estas funções são úteis quando precisamos de uma função para ser passada como parâmetro para outra função e que não será mais utilizada após isso, sendo descartada em seguida.
- Uma expressão lambda permite criar uma função em uma única linha de código.
- Por ser uma expressão, pode aparecer em lugar que um def não pode, como por exemplo, dentro de uma lista ou nos argumentos de chamada de uma função.

# SINTAXE EXPRESSÃO LAMBDA

lambda arg1, arg2, ...argN : expressão usando os argumentos

Podemos passar vários argumentos separados por vírgula. E o retorno será uma função com o resultado do código informado.

```
def a(x):  
    return x ** 2  
print(a(4))  
b = lambda x: x ** 2  
print(b(4))
```

# EXEMPLO 1

A variável **soma** recebe o valor de uma função **lambda** que tem como parâmetro uma **lista** que vai retornar a função **sum**.

```
soma = lambda lista: sum(lista)
numeros = [1, 5, 10, 50]
total = soma(numeros)
print(total)
numeros = [80, 74, 25]
total = soma(numeros)
print(total)
```

# FUNÇÃO TYPE

A função type retorna o tipo de uma variável, função ou objeto.

```
>>> a = 10
>>> type(a)
<class 'int'>
>>> b = "Olá mundo!"
>>> type(b)
<class 'str'>
>>> type(print)
<class 'builtin_function_or_method'>
>>> type(True)
<class 'bool'>
>>> lista = [1,2,3,4]
>>> type(lista)
<class 'list'>
>>>
```

```
>>> type(15.4)
<class 'float'>
>>> dict = {"a":1,"b":2,"c":3}
>>> type(dict)
<class 'dict'>
>>> def funcao():
...     return True
...
>>> type(funcao)
<class 'function'>
>>>
```

# FUNÇÃO MAP

- O mapeamento é feito através da função `map()`, que consiste em aplicar uma função a todos os itens de uma sequência, seja uma lista, tupla, etc., gerando uma outra sequência contendo os resultados e com o mesmo tamanho da sequência inicial. A função `map()` sempre retorna um gerador.
- A sintaxe da função `map` é: `map(função, objeto_iterável, ...)`
- Onde:
  - função: `map()` passa cada item do objeto iterável para a função informada.
  - objeto\_iterável: Objeto iterável que deve ser mapeado. Podemos passar mais de um objeto iterável

# EXEMPLOS FUNÇÃO MAP

```
lista = [1, 2, 3, 4, 5, 6]
lista1 = list(map(lambda x: x**2, lista))
#eleva ao quadrado os elementos de lista
para criar lista1
print("lista = ", lista, "\n\nlista1 = ",
      lista1)
```

```
def calculaQuadrado(numero):
    return f"O quadrado de {numero} é: {numero ** 2}"
lista_numeros = [1, 2, 3, 4, 5, 6]
calculo = map(calculaQuadrado, lista_numeros)
for n in calculo:
    print(n)
```

# FUNÇÃO REDUCE()

- A função reduce() aplica uma função em todos de uma lista, e retornar apenas um valor. O formato geral é:

**reduce**(função a ser aplicada, lista)

```
from functools import reduce

lista = [1, 2, 3, 4, 5, 6]
produto = reduce(lambda x,y: x * y, lista)
#retorna o produto de todos os elemento de lista
print("lista = ", lista, "\n\nproduto = ", produto)
```



# FUNÇÃO FILTER()

- A função filter() aplica um filtro sobre os elementos da lista, de acordo com a função passada como primeiro argumento. O formato geral é: **filter(função a ser aplicada, lista)**

```
lista = [1, 2, 3, 4, 5, 6]
lista1 = list(filter(lambda x: x % 2 != 0, lista))
#retorna a lista de números ímpares
print("lista = ", lista, "\n\nlista1 = ", lista1)
```

# DESAFIO CONVERSOR DE TEMPERATURA

- Faça um programa que receba o usuário escolha qual unidade de medida esta inserindo a temperatura e converta para a temperatura requerida

Conversão de		Para	Fórmula
Celsius	→	Fahrenheit	$^{\circ}\text{F} = ^{\circ}\text{C} \times 1,8 + 32$
Fahrenheit	→	Celsius	$^{\circ}\text{C} = (^{\circ}\text{F} - 32) : 1,8$
Celsius	→	Kelvin	$\text{K} = ^{\circ}\text{C} + 273,15$
Kelvin	→	Celsius	$^{\circ}\text{C} = \text{K} - 273,15$
Fahrenheit	→	Kelvin	$\text{K} = (^{\circ}\text{F} + 459,67) : 1,8$
Kelvin	→	Fahrenheit	$^{\circ}\text{F} = \text{K} \times 1,8 - 459,67$

# EXERCÍCIOS

1. Faça um programa que solicite o nome do usuário e a idade do usuário, depois disso exiba a mensagem: "{nome} possui {idade} anos.". Esta mensagem deve ser escrita em uma função lambda.
2. Faça um programa que solicite dez números ao usuário, depois disso, exiba todos números pares e só então exiba todos os números ímpares. Utilize a função filter para fazer isso.
3. Escreva uma função que imprime todos os números primos entre 1 e 500
  - Dica: um número é primo se ele for divisível apenas por 1 e ele mesmo, use o operador % (resto da divisão) para isso.