



Funções

Uma função é um pedaço de código que faz alguma tarefa específica e pode ser chamado de qualquer parte do programa quantas vezes desejarmos.

Funções

Utilizamos funções para obter:

Clareza do código: separando pedaços de código da função main(), podemos entender mais facilmente o que cada parte do código faz. Além disso, para procurarmos por uma certa ação feita pelo programa, basta buscar a função correspondente. Isso torna muito mais fácil o ato de procurar por erros.

Funções

Utilizamos funções para obter:

Reutilização: muitas vezes queremos executar uma certa tarefa várias vezes ao longo do programa. Repetir todo o código para essa operação é muito trabalhoso, e torna mais difícil a manutenção do código: se acharmos um erro nesse código, teremos que corrigí-lo em todas as repetições do código. Chamar uma função diversas vezes contorna esses dois problemas.

Funções

Utilizamos funções para obter:

Independência: uma função é relativamente independente do código que a chamou. Uma função pode modificar variáveis globais ou ponteiros, mas limitando-se aos dados fornecidos pela chamada de função.

Funções

O princípio de uma função está, naturalmente, em permitir encapsular uma ideia ou operação, dando-lhe um nome, e então chamar essa operação de várias partes do resto de seu programa simplesmente usando o seu nome.

Funções

Em Python sua sintaxe é definida usando def e atribuindo um nome a ela, veja um exemplo:

```
def funcao():  
    print("Bloco de código")
```

Funções

Parâmetros

Além de executar código, funções também podem receber e retornar dados.

Podemos enviar dados para uma função através de seus parâmetros.

Observe o exemplo:

Funções

```
def imprime_nome(nome):  
    print(f"Nome: {nome}")
```

```
imprime_nome("Erickson")  
imprime_nome("Renan")  
imprime_nome("Daniel")
```

exercício 01

Faça um programa para imprimir:

1

2 2

3 3 3

.....

n n n n n n ... n

Funções

```
def exercicio_1(n):  
    for i in range(1,n+1):  
        print(str(i) * i )
```

exercício 02

Faça um programa para imprimir:

1

1 2

1 2 3

.....

1 2 3 ... n

Funções

```
def exercicio_2(n):  
    for i in range(1,n+1):  
        for x in range(1,i+1):  
            print(x,end=" ")  
        print()
```

Funções

faça uma função que conte quantas vogais tem num texto.

texto

O rato roeu a roupa do rei de Roma

Funções

```
def palavra(texto):  
    cont=0  
    y="aeiouAEIOU"  
    for x in texto:  
        if x in y:  
            cont = cont + 1  
    print(cont)
```

Funções

Crie uma função que recebe o nome de um **produto, a **quantidade** que tem no estoque e o **valor unitário** do produto. Retorne o valor total do meu estoque.**

Funções

```
def estoque(produto, quantidade, valor):  
    valor= quantidade*valor  
return valor
```

Desafio

Usando a função estoque que acabamos de criar, faça um código que cadastre os nomes dos produtos num array produtos, valor unitário em um array valorunitário e o retorno da função estoque no array valortotal.

Depois print os 3 arrays para mostrar que tudo foi registrado.

Funções

Faça um programa, com uma função que necessite de um argumento.

A função retorna o valor de caractere 'P', se seu argumento for positivo, e 'N', se seu argumento for negativo e Z se o argumento for zero.

Funções

Faça uma função que receba 2 argumentos e adicione cada um numa lista diferente. PRODUTO e PREÇO.

o sistema deve pedir os dados e após isso verificar se o usuário ainda pretende inserir mais produtos. Quando terminar de inserir, imprimir as listas

Funções

Quando precisamos definir uma função, e não sabemos a quantidade de parâmetros?

exemplo: faça uma função que receba uma quantidade não definida de parâmetros e some todos eles. E retorne a soma

Funções

***args**

def soma(*args):

**neste caso, args é recebido
como uma tupla..**

Funções

Faça uma função que possa receber um número não determinado de argumentos e some todos eles.

Funções

Faça uma função que receba um texto como argumento, mostre a quantidade de letras e também imprima o texto ao contrário

Funções

faça um função que recebe uma lista como argumento e crie uma nova lista, somente com números únicos.

exemplo:

a[1,2,2,3,4,4,5,3,6,7,6,8]

nova_lista[1,2,3,4,5,6,7,8]

Funções

```
def lista_unica(l):  
    nova_lista=[]  
    for x in l:  
        if x not in nova_lista:  
            nova_lista.append(x)  
    print(nova_lista)
```

```
a=[1,2,2,3,4,3,4,5,6,7,6,8,7,10]  
lista_unica(a)
```

Funções

**faça um função que recebe
um número como parâmetro
e verifique se este número é
primo**

Funções

```
def test_primo(n):  
    if (n==1):  
        return n,"Não é Primo"  
    elif (n==2):  
        return n,"É Primo";  
    else:  
        for x in range(2,n):  
            if(n % x==0):  
                return n,"Não é Primo"  
        return n,"É Primo"
```