Função em Python

<https://wiki.python.org.br/PrincipiosFuncionais>

As funções em Python são definidas através da palavra chave def.

- Sintaxe:

def nome\_funcao(par1, par2, ... , parn):

<codigo> # indentação obrigatória.

. . .

return nome\_objeto\_a \_retornar

if \_\_name\_\_ == '\_\_main\_\_': # main + tecla <tab>

. . . # indentação obrigatória.

nome\_funcao(arg1, arg2, . . . , argn)

. . .

A convenção acima ( \_\_ ) é utilizada frequentemente em Python e, muitas vezes, a chamamos de "dunder" (vem da expressão em inglês "double-underscore" - sublinhado duplo em português).

- Sintaxe do uso de função:

def nome\_funcao (par1, par2, ... , parn): # par (parâmetro) - o que a função recebe

script # indentação obrigatória.

. . .

return val1, val2, ... , valn

def --> declara que estamos construindo uma funçao

nome\_funcao --> nome da função definido pelo desenvolvedor

script --> código da função

return --> indica o fim da função e o que estará sendo retornado

val1, val2, ... , valn --> Valores retornados

- Chamada da função:

nome\_funcao (arg1, arg2, ... , argn) # arg (argumento) - o que envio para a função

- Exemplo:

print() é uma função, ou seja, a ideia de criar uma função é a sua reutilização

em vários momentos do programa

Exemplos de funções:

Uma função nada mais é do que uma sub-rotina usada em um programa.

Uma função em Python é escrita da seguinte forma:

Digita-se a palavra chave **def nome\_da\_função()**

Nas funções existem também a palavra-chave **return**, que é responsável por retornar o resultado das operações realizadas no escopo da função.

- Declarar função vazia (é preciso escrever algo no escopo da função caso contrário o pycharm aponta um erro):

def funcao\_vazia():

pass

Chamando a função:

funcao\_vazia()

- Escrever uma função com algum conteúdo:

def funcao():

print(“Isso é uma função”)

Chamando a função:

funcao()

Saída: Isso é uma função

- Escrevendo uma função com um parâmetro:

def dobro(x):

dobro = a\*2

return dobro

Chamando a função:

print(“O dobro é = ”, dobro(2))

Saída: O dobro é = 4

- Escrevendo uma função com mais de um parâmetro:

def soma(a,b)

soma = a + b

return soma

Chamando a função:

print(“Soma = “, soma(5,5))

Saída: Soma = 10