**As funções em Python**

As funções podem fazer muitas coisas, como por exemplo causar um efeito, como mandar um texto para o terminal, criar um arquivo, tocar um som, ou validar valores, como a raiz quadrada de um valor ou a longuitude de uma string e retornar como o resultado da função.

**De onde vem as funções?**

Elas podem vir da própria linguagem, por exemplo o print vindo do próprio Python, de módulos que são add-ons que são instalados de fontes exteriores e do nosso próprio código, nós podemos escrever nossas próprias funções.

**Como funciona a evocação de uma função em Python?**

A invocação de uma função se da em cinco etapas:

1. Primeiro o Python checa se o nome específicado da função é legal(ele busca uma função com o nome mas se ele não acha ele aborta o código.
2. Segundo ele checa se os requerimentos para a função ser executada, por exemplo se uma função pede um valor inteiro para ser executada, quando a função for de fato executada esse valor precisa ser chamado também, se tiver mais ou menos valores do que na assinatura da função, ou se for um valor inválido ele aborta o código.
3. Terceiro, o Python larga o código por um momento e pula na função que você chamou, também pegando o argumento que você passou.
4. Quarto, a função executa o próprio código, e causa o efeito desejado(caso haja um) e valida o efeito desejado(caso haja um) e finaliza a tarefa.
5. E finalmente o Python volta ao seu código e finaliza a execução da função.

**Instruções em Python.**

No Python, diferente de outras linguas ele não pode conter mais de uma instrução na mesma linha, como em Java por exemplo, porém, ele pode extender a mesma instrução para mais de uma linha, assim facilitando a leitura do código.

**Python escape e caracteres newline**

A barra ao contrário em Python(\) é como um anúncio de alguma coisa que está por vir, por exemplo o \n, o Python *não lê* a barra, mas pula direto para o n, e o \n assim sendo chamado de newline.

Obs.:Por conta disso quando formos usar o "\" dentro de uma String temos que duplicar ele para que ele não acabe sendo ignorado.

**Argumentos dentro de print()**

A função print() tem alguns comportamentos especiais com seus argumentos, cada vez que você coloca um argumento extra dentro de print ele vai ser separado com um espaço por cada vírgula que você coloca dentro dele. Cada vez que você inicia um novo print() ele quebra a linha.

* Argumentos palavra-chave

Um argumento palavra-chave consiste de três elementos, a palavra-chave desse argumento; um sinal de igual; e o valor associado a esse argumento.

Todo argumento de palavra-chave precisa ser colocado na posição final dos argumentos de print().

Exemplo:

print("My name is ", end="")

print("Monty Python.")

Resultado:

My name is Monty Python.

Normalmente o default de end é "\n" mas como nós mudamos para uma String vazia ele vai simplesmente colocar tudo na mesma linha sem mudança nenhuma, porém como um espaço foi escrito dentro da primeira String ele vai ser escrito de forma correta.

Um outro exemplo:

print("My", "name", "is", sep="\_", end="\*")

print("Monty", "Python.", sep="\*", end="\*\n")

Resultado:

My\_name\_is\*Monty\*Python.\*

Além de usarmos o end nós também modificamos o sep, que é o comportamento que faz a vírgula separar as Strings com um espaço em branco.

O Python tem 69 funções próprias da linguagem, nós podemos encontrar uma lista delas em ordem alfabética no: <https://docs.python.org/3/library/functions.html>