



Unidade 1: Iniciação à programação em Python

Lição 3: As funções em Python

Nesta terceira lição da Unidade 1, utilizará o editor de programas da aplicação TI-Python para criar funções e, depois, executar o programa e/ou as funções e observar os resultados no interpretador.

Objetivos:

- Descobrir o conceito de função em linguagem Python.
- Construir funções

Descobrimo o conceito de função em Python.

Implementar, no interpretador, o seguinte algoritmo:

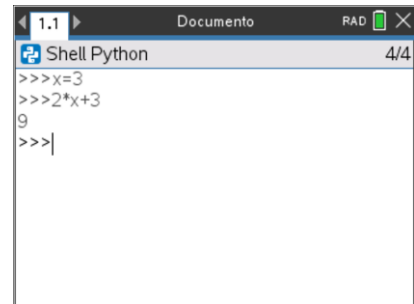
$$x \leftarrow 3$$

$$y \leftarrow 2 \times x + 3$$

É um processo relativamente simples.

Mas se pretender repetir o cálculo para outro valor de x, terá que escrever de novo toda a sequência de instruções. Assim, numa situação menos trivial pode tornar-se muito aborrecido e, com certeza, pouco eficaz.

Somos, portanto, levados a criar uma função que nos permita replicar facilmente o processo algorítmico.



Em algoritmia, uma função pode considerar-se como uma sequência de instruções, executando uma determinada tarefa, utilizando um ou mais **argumentos** (ou até nenhum, em alguns casos).

À função é atribuído um nome, de preferência sugestivo e simples.

- A programação de uma função começa sempre por **def** seguido pelo nome da função e depois pelos seus argumentos. Esta linha termina com o símbolo **:**
- Os dois pontos (..) assinalam o início do bloco das instruções que definem a função, sendo que todas as linhas com estas instruções são **indentadas**, ou seja, deslocadas para a direita em relação à 1ª linha. No início de cada linha, adicionamos o mesmo número de espaços.
- A função retorna um único resultado por intermédio do comando **return**. O resultado pode ser constituído por uma lista de resultados, uma cadeia de caracteres, ...

```
def nome_função(lista de argumentos) :
    ..bloco de instruções
    ..return (resultados)
```

Sintaxe da definição de função em Python

A correta indentação (recuo) das linhas de instrução, que se pode obter com a tecla de tabulação ou a tecla de espaço ou ainda no menu, é fundamental: qualquer instrução indentada após **def()** será executada como um bloco. A indentação não deve variar (número de espaços, alternância entre tabulações e espaços. . .) no bloco.





SUGESTÃO:

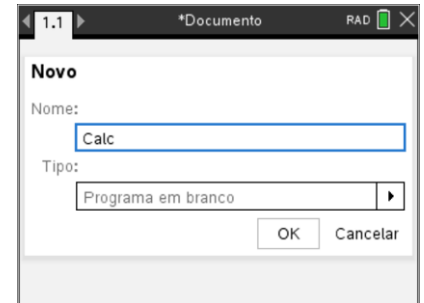
Uma função permite decompor o problema estudado noutros subproblemas e, assim, evitar a repetição de instruções. Uma vez definida uma função, ela pode ser "chamada" durante a execução do programa tantas vezes quanto as necessárias. Uma função pode não ter argumentos. Pode também ser "chamada" noutro programa: basta inseri-la numa instrução com os valores dos argumentos.

IMPLEMENTAÇÃO DE UM EXEMPLO:

- Criar um novo programa na aplicação TI-Python

Pressionar a tecla **[doc]** e escolher no menu a opção **A: Adicionar Python**, depois escolher **1: Novo**. De seguida, na janela que se abre, atribua um nome ao programa.

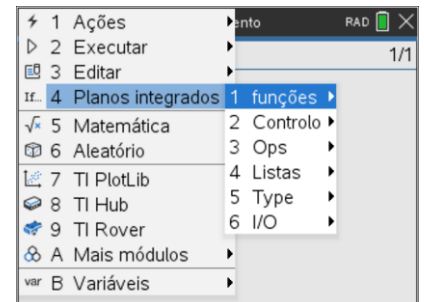
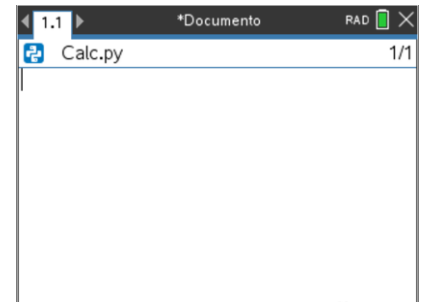
Para validar o nome do programa pode clicar sobre o botão OK, ou pressionar a tecla **[enter]**.



NOTA:

Por defeito, o editor de programas inicia com uma página em branco (sem incluir bibliotecas). No entanto, a edição pode ser facilitada se conhecer as bibliotecas necessárias para criar um novo projeto.

- O nome do programa surge na barra inicial da página de edição com a extensão **.py**.
- Pressionar a tecla **[menu]** para abrir a lista de submenus do editor.
- As instruções específicas da linguagem Python estão no menu **4: Planos integrados** e contém todas as instruções necessárias para escrever um programa, que são:
 - 1: Definição de funções
 - 2: Ciclos e condições
 - 3: Operadores Relacionais e Lógicos
 - 4: Listas
 - 5: Tipos de dados
 - 6: Entradas e saídas de dados



- Selecionar a instrução para construção de uma função no editor, selecionando a opção **1: Funções**, dos planos integrados, seguida da opção **1: def function():** .



- Observar a ação realizada pela calculadora, isto é, a exibição automática da estrutura de uma função em Python.
- Os campos de preenchimento surgem com texto acinzentado, podendo ser facilmente preenchidos usando a tecla `tab` da calculadora para se deslocar de campo para campo.
- Quando o cursor estiver num campo que pode ser preenchido, o texto passa a estar sombreado a azul claro.
- Para definir a função f que permita implementar o algoritmo acima, deverá obter o ecrã ao lado. Observará a indentação automática do cursor.
- Continuar de seguida com as instruções para implementação do algoritmo. Recordar que a atribuição de um valor a uma variável é realizada com o símbolo `=`.
- Pressionar a tecla `menu`, depois seleccionar **4: Planos integrados**, de seguida **1: Funções** e finalmente **2: return**, para completar o pequeno programa com a instrução de saída da imagem pela função.
- Pode também digitar a instrução manualmente, usando o teclado alfanumérico da calculadora, ficando destacada logo que seja reconhecida.

```

1.1 | *Documento | RAD | 2/2
* Calc.py
def função(argumento):
  *bloco
  
```

```

1.1 | *Documento | RAD | 3/3
* Calc.py
def f(x):
  y=2*x+3
  return y
  
```

Agora está pronto a executar o seu programa.

- Pressionar `ctrl R`, atalho para executar um programa.
- O interpretador é exibido e surge uma mensagem a dar conta do carregamento do programa
- Preencher a linha de comando com o nome da função e os argumentos, neste caso apenas um, e valide, pressionando a tecla `enter`.
- Utilizar a deslocação do cursor para cima, `▲`, para seleccionar um comando anterior e de seguida clicar na tecla `enter` pode a obter na atual linha de comando.

```

1.1 | 1.2 | *Documento | RAD | 7/7
Shell Python
>>>#Running Calc.py
>>>from Calc import *
>>>f(3)
9
>>>f(5)
13
>>>
  
```

SUGESTÃO:

A edição e a execução de um programa em Python, com a TI-Nspire CX, pode ser facilitada se dividir a página em duas aplicações, numa o Editor e noutra o Interpretador. Para tal clicar na tecla `docv`, depois opção **5 : Esquema da página.**, e seleccionar o formato que interessar, adicionando na segunda página o interpretador (**Shell**) da aplicação TI-Python.

Posteriormente, usando no editor o atalho `ctrl R` o cursor passará automaticamente para a janela do interpretador, que se encontra sob a do editor, e deste modo poderá a executar o programa vendo a edição do mesmo. Poderá separar as duas aplicações da divisão da página usando o atalho `ctrl 6`.

```

1.1 | 1.2 | *Documento | RAD | 1/1
* Calc.py
def função(argumento):
  bloco
  return y

Shell Python
>>>
  
```