

CAPÍTULO 3

O capítulo 3 dedica-se ao estudo de funções, sua estrutura e utilidade. Na programação funções são uma sequência de comandos que fazem ações desejadas. Para isso a função recebe do programa um argumento e retorna uma ou nenhuma saída. Funções sem retorno são chamadas de função do tipo void. Como exemplo de função temos:

```
>>> type(42)
<class 'int'>
```

Nesse exemplo a função de nome “type” recebe como argumento o inteiro 42 e retorna qual o tipo desse valor, no caso inteiro. Em Python às vezes é possível converter valores, como em:

```
>>> pi=3.14159
>>> int(pi)
3
```

```
>>> a=str(5.6)
>>> a*3
'555'
```

```
>>>int('Vai Brasil!')
ValueError: invalid literal for int(): 'Vai Brasil!'
```

No primeiro caso a função int recebe como argumento o valor da variável tipo float “pi” e transforma-a num tipo int. No segundo caso a função str recebe o float 5.6 como argumento e retorna a string ‘5.6’ que é atribuída como valor da variável “a”. No último caso não foi possível transformar a string num inteiro.

O primeiro exemplo nos remete ao uso de operações matemáticas na computação e leva à dúvida se existem funções em Python específicas para realizar operações matemáticas comuns envolvendo constantes matemáticas como o Pi. De fato, existe um conjunto de funções em Python que performam operações matemáticas e elas estão dentro de um módulo que precisa ser previamente importado para ser utilizado no texto. O objeto módulo é constituído por variáveis e funções definidas neste. No caso de operações matemáticas temos como exemplo:

```
>>> import math
>>> 10*math.sin(24)/math.log10(math.sqrt(100))
-9.055783620066238
```

Note no exemplo anterior que funções podem ser compostas e o resultado de uma função pode ser usado como argumento para outra. Além das funções disponíveis automaticamente ou por meio de módulos o programador pode definir funções de seu interesse para executarem comandos desejados. Isso pode ser feito da seguinte maneira:

```
>>> def derik(n):
    print(str(n)+ “tigres tristes para ” + str(n) + “pratos de trigo”)
```

Usamos a palavra reservada def para definir a função de nome derik que recebe como argumento o número n e escreve na tela a frase “N tigres tristes para N pratos de trigo”. Dentro do programa a função criada pode ser usada a qualquer momento bastando escrevermos seu nome e passarmos o argumento de maneira correta. Definir uma função cria um objeto função de tipo function.

É importante ressaltar que quando um valor é passado a uma função ele é passado por meio de parâmetros, ou seja, a função cria uma variável chamada de parâmetro que recebe o valor do argumento. É válido lembrar também que as variáveis declaradas dentro de uma função e os parâmetros tem escopo local, ou seja, quaisquer alterações nos parâmetros ou nas variáveis só tem efeito dentro da função, não sendo modificado dentro do programa.