

Introdução à Programação #13

Funções

Uma **função** é um bloco de código que, tal como o próprio nome indica, tem uma função própria, ou seja, que serve para aquela finalidade. As funções permitem-nos, por exemplo, reutilizar código.

Ao invés de colocarmos um determinado trecho de código diversas vezes, basta criarmos uma função com esse código e, de cada vez que for necessário, precisamos apenas de invocar a função.

Durante os seus anos de programador, irá criar milhares de funções para as mais diversas funções. Então, recomendamos que vá guardando as funções que utiliza porque, futuramente, poderá necessitar delas.

Existe uma função que tem vindo a ser sempre utilizada: a função *main*. É a função principal do programa, aquela que é executada automaticamente quando um programa é executado. A sintaxe das funções é a seguinte:

```
1. tipo_dado_retorno nome_da_funcao(argumentos) {
2.     //Codigo da Funcao
3. }
```

Vejamos então o exemplo de uma função simples que nos permite mostrar a mensagem “Olá Mundo!”:

```
1. #include <stdio.h>
2.
3. int main() {
4.     sayHello();
5.     return 0;
6. }
7.
8. void sayHello() {
9.     printf("Olá Mundo!\n");
10. }
```

A função chama-se “sayHello” e não retorna nenhuns dados (“void”). O que está dentro das chavetas (“{}”) é o código que é executado quando a função é chamada.

Argumentos e Parâmetros

No exemplo acima é possível visualizar que tanto na invocação da função, como na sua escrita foi colocado um par de parênteses sem nada. Isto acontece porque não existem parâmetros.

Vamos criar uma função que efetua uma soma: como enviamos os valores para a função? Definindo o nome e o tipo de parâmetros a serem recebidos. Veja o seguinte exemplo.

```
1. void soma(int n1, int n2) {
2.     int soma = n1 + n2;
3. }
```

A função acima, denominada “soma”, tem dois **parâmetros**, o “n1” e o “n2”, ambos do tipo inteiro. Os valores que a eles irão ser atribuídos chamam-se **argumentos**.

Para utilizar a função, basta então proceder do seguinte modo:

```
1. soma(4, 6);
```

Substituindo os números 4 e 6 por quaisquer números que desejar e a função irá armazenar, na variável “soma” que apenas está disponível dentro da função, o valor da soma entre os dois números.

Retorno

Na função anterior somámos dois números mas nada mais aconteceu. E se quisermos, por exemplo, que a função retorne o valor da soma?

Para isso temos que substituir o “void” pelo tipo de dados que, neste caso é “int” e utilizar o comando “return”. Ficaria algo semelhante ao seguinte:

```
1. int soma(int n1, int n2) {
2.     int soma = n1 + n2;
3.     return soma;
4. }
```

Ou seja, enviamos dois números para a função; esta cria a variável “soma” com a soma dos dois números e de seguida retorna esta variável que é do tipo “int”. Podemos simplificar esta função ficando apenas com o seguinte:

```
1. int soma(int n1, int n2) {
2.     return n1 + n2;
3. }
```

Podemos, por exemplo, utilizar esta função no seguinte contexto:

```
1. #include <stdio.h>
2.
3. int main() {
4.     int a, b;
5.
6.     printf("Insira o primeiro número: ");
7.     scanf("%d", &a);
8.
9.     printf("Insira o segundo número: ");
10.    scanf("%d", &b);
11.
12.    printf("A soma é %d", soma(a,b));
13.    return 0;
14. }
15.
16. int soma(int n1, int n2) {
17.     return n1 + n2;
18. }
```

<http://pplware.sapo.pt/tutoriais/vamos-programar-introducao-a-programacao-13>