

1. Uma função é um bloco independente de código que realiza uma tarefa específica. Elas podem ou não conter entrada e retorno de valor, podendo ser chamadas múltiplas vezes no código, com isso permitindo que o código da função seja reutilizado várias vezes e que a secção de código da função seja testado separadamente mais facilmente para garantir que é funcional.

2. Parâmetros servem para passar valores para funções para elas poderem usá-los. Eles são passados quando a função é chamada, sendo declarados na função. Isso permite que a função seja generalizada, pois permite que valores diferentes sejam usados cada vez que a função for chamada, sendo o valor uma palavra que vai ser escrita, ou um valor que será usado em uma operação matemática, etc.

3. Pode se classificar as funções em quatro grupos:

As que não possuem parâmetros e retorno, sendo chamadas de procedimentos;

As que não possuem parâmetros mas possuem retorno;

As que possuem parâmetros mas não retorno;

E as que possuem ambos.

4. O *void* representa o vazio e quando usado na declaração de uma função, significa que a função não retornará um valor.

5. Um parâmetro pode ser passado de duas maneiras: Por valor, onde uma cópia do valor é passado para a função, separando a variável na função da variável de onde o valor foi copiado, sendo o padrão da linguagem C/C++; E por referência, onde o endereço da memória onde o valor se localiza é passado para a função, com isso a função manipula o conteúdo da variável original.

6. O tipo de retorno de uma função é especificado no início da declaração da função, exemplificando qual tipo de valor que a função irá retornar quando terminar. Os tipos de retorno seguem o mesmo padrão dos tipos de variáveis, podendo ser: int, float, double, string, char, bool, etc.