

Rodrigo Henrich

rodrigoenrich@faccat.br



Criando funções em C

- Quando temos um trecho de código que precisa ser executado várias vezes ao longo programa
- Podemos transformar esse trecho de código em uma função
- Deixa o código mais simples
- Mais fácil de compreender
- Além de dividir o programa em pequenos pedaços de código, o que facilita a identificação de possíveis bugs.
- As funções também são chamadas de subrotinas

Criando funções em C

- Para criar uma função em C usamos a seguinte sintaxe

`tipo_retorno nome_funcao(tipo_parametro nome_parametro, tipo_parametro2 nomeParametro2)`

- Chamamos essa parte da função de assinatura da função
- **tipo_retorno**, é obrigatório em todas as funções, caso a função não retorne nenhum valor o tipo de retorno deve ser **void**
- se a função possuir um tipo de retorno especificado, usamos a palavra `return` seguida do valor ou variável que queremos retornar como resposta da função.

Criando funções em C

`tipo_retorno nome_funcao(tipo_parametro nome_parametro, tipo_parametro2 nomeParametro2)`

- **nome_funcao**, é o nome da função ele vamos usar para chamar nossa função em outras funções ou na função `main()`;
- Entre parênteses segue os parâmetros que nossa função recebe
- Os parâmetros são separados por ,
- Cada parâmetro tem um tipo e um nome
- Os parâmetros de uma função são variáveis locais, ou seja elas só existem dentro da função, assim como todas as funções declaradas dentro do escopo da função

Criando funções em C

- Exemplo de função para somar dois números inteiros e retornar para a função principal

Criando funções em C

The diagram illustrates the components of a C function definition. The code is: `int soma(int n1, int n2){
 int soma = n1+n2;
 return soma;
}`. Red arrows and labels point to specific parts: 'A' points to the return type 'int' at the start; 'B' points to the function name 'soma'; 'C' has two arrows pointing to the parameters 'int n1' and 'int n2'; 'D' is a bracket on the right side of the function body; 'E' has an arrow pointing to the closing curly brace '}'.

```
int soma(int n1, int n2){  
    int soma = n1+n2;  
    return soma;  
}
```

A - Tipo de retorno da função

B - Nome da função

C - Parâmetros da função

D - Corpo da função

E - Retorno da função

Criando funções em C

- Como chamar a função criada
- Para chamar uma função criada temos que usar o nome da função;
- No exemplo a seguir vamos ler dois números do teclado e realizar a soma deles usando nossa função

Criando funções em C

```
1  #include<stdio.h>
2  #include<locale.h>
3
4  int soma(int n1, int n2){
5      int soma = n1+n2;
6      return soma;
7  }
8
9  int main(){
10     int numeroUm, numeroDois, resultado;
11     setlocale(LC_ALL, "");
12     printf("Entre com o primeiro número para somar ");
13     scanf("%d",&numeroUm);
14     printf("\nEntre com o segundo número para somar ");
15     scanf("%d",&numeroDois);
16     resultado = soma(numeroUm, numeroDois);
17     printf("%d+%d=%d", numeroUm, numeroDois, resultado);
18     return 0;
19 }
```


Criando funções em C

- Funções modificando variáveis globais
- Como lembram é possível declarar variáveis globais, elas irão existir ao longo de toda a execução do programa e serão acessíveis por todas as funções do programa
- Vamos realizar o mesmo exemplo de soma anterior, porém agora sem usar retorno e parâmetros na função

Criando funções em C

```
1  #include<stdio.h>
2  #include<locale.h>
3
4  int numeroUm, numeroDois, resultado;
5
6  void soma(){
7      resultado = numeroUm + numeroDois;
8  }
9
10 int main(){
11     setlocale(LC_ALL, "");
12     printf("Entre com o primeiro número para somar ");
13     scanf("%d",&numeroUm);
14     printf("\nEntre com o segundo número para somar ");
15     scanf("%d",&numeroDois);
16     soma();
17     printf("%d+%d=%d", numeroUm, numeroDois, resultado);
18     return 0;
19 }
```

Criando funções em C

- Todas as funções devem aparecer antes da função main()
- Caso a função seja criada depois da função main() será preciso trazer sua assinatura para antes da função principal
- Para isso vamos usar o mesmo exemplo anterior

Criando funções em C

```
1  #include<stdio.h>
2  #include<locale.h>
3
4  int soma(int n1, int n2);
5
6  int main(){
7      int numeroUm, numeroDois, resultado;
8      setlocale(LC_ALL, "");
9      printf("Entre com o primeiro número para somar ");
10     scanf("%d",&numeroUm);
11     printf("\nEntre com o segundo número para somar ");
12     scanf("%d",&numeroDois);
13     resultado = soma(numeroUm, numeroDois);
14     printf("%d+%d=%d", numeroUm, numeroDois, resultado);
15     return 0;
16 }
17
18 int soma(int n1, int n2){
19     int soma = n1+n2;
20     return soma;
21 }
```