Rodrigo Henrich

rodrigohenrich@faccat.br

- 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

Para criar uma função em C usamos a seguinte sintaxe

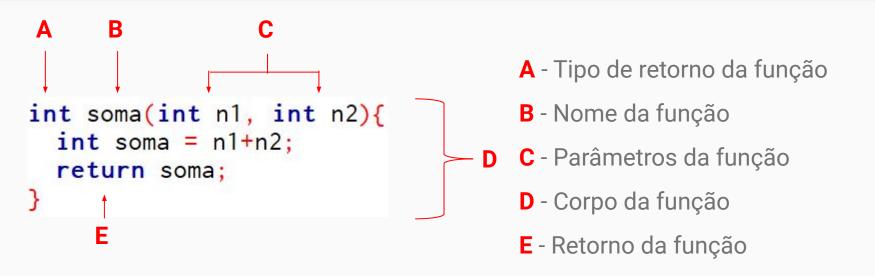
tipo_retorno nome_funcao(tipo_parametro nome_paramentro, tipo_paramentro2 nomeParametro2)

- 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.

tipo_retorno nome_funcao(tipo_parametro nome_paramentro, tipo_paramentro2 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

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



- 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

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

- 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

```
#include<stdio.h>
    #include<locale.h>
    int numeroUm, numeroDois, resultado;
    void soma(){
      resultado = numeroUm + numeroDois;
8
9
    pint main(){
      setlocale(LC_ALL, "");
11
      printf("Entre com o primeiro número para somar ");
      scanf("%d",&numeroUm);
14
      printf("\nEntre com o segundo número para somar ");
15
      scanf("%d",&numeroDois);
16
      soma();
      printf("%d+%d=%d", numeroUm, numeroDois, resultado);
18
       return 0:
```

- 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

```
#include<stdio.h>
    #include<locale.h>
    int soma(int n1, int n2);
   □int main(){
       int numeroUm, numeroDois, resultado;
       setlocale(LC_ALL, "");
      printf("Entre com o primeiro número para somar ");
       scanf("%d",&numeroUm);
11
      printf("\nEntre com o segundo número para somar ");
      scanf("%d",&numeroDois);
13
      resultado = soma(numeroUm, numeroDois);
14
      printf("%d+%d=%d", numeroUm, numeroDois, resultado);
15
      return 0:
16
17
    □int soma(int n1, int n2){
      int soma = n1+n2;
19
       return soma;
```