

# Funções e Procedimentos

Prof: M.Sc Mário Angel Praia Garcia

# 1. Funções e Procedimentos

- Via de Regra
  - Sem saber como escrever uma função. Utilizamos uma, como printf, scanf, puts, que fazem parte da biblioteca standard de C.
  - SEMPRE a função main() deve estar presente em um código, independente do número e da variedade de funções contidas.

```
printf("1- Inserir\n");  
printf("2- Remover\n");  
printf("Opcao? ");
```

# 1. Funções e Procedimentos

- Características de uma função:
  - Deve ter um nome único.
  - Pode ser invocada a partir de outras funções.
  - Realiza uma única tarefa bem definida.
  - Não importa como é seu funcionamento mas sim o seu resultado final.
  - O seu código deve ser o mais independente e o mais genérico do resto do programa.
  - Um função retorna apenas um único valor como resultado mas pode conter várias instruções **return**.

# 1. Funções e Procedimentos

- Sintaxe de uma função:

tipo\_retorno **nome\_da\_função** (parâmetros)

{

comando 1;

comando 2;

.....

}

# 1. Funções e Procedimentos

- **Exemplo:**

```
int soma (int x, int y)
{
    int resultado;
    resultado = x + y;
    return resultado;
}
```

# 1. Funções e Procedimentos

- **Exemplo:**

```
int soma (int x, int y)
```

```
{
```

```
    int resultado;
```

```
    resultado = x + y;
```

```
    return resultado;
```

```
}
```



Tipo de retorno da função

# 1. Funções e Procedimentos

- **Exemplo:**

```
int soma (int x, int y)
```

```
{
```

```
    int resultado;
```

```
    resultado = x + y;
```

```
    return resultado;
```

```
}
```



Nome da função

# 1. Funções e Procedimentos

- **Exemplo:**

```
int soma (int x, int y)
```

```
{
```

```
    int resultado;
```

```
    resultado = x + y;
```

```
    return resultado;
```

```
}
```



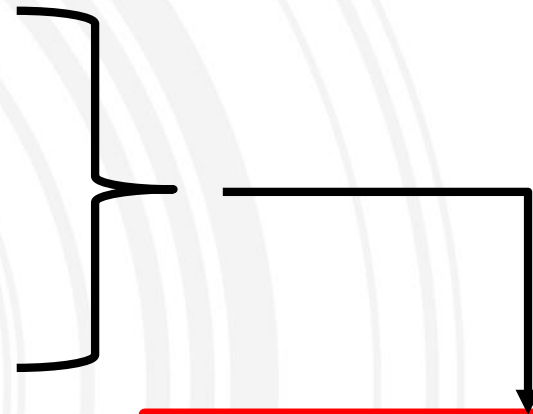
Parâmetros da função



# 1. Funções e Procedimentos

- **Exemplo:**

```
int soma (int x, int y)
{
    int resultado;
    resultado = x + y;
    return resultado;
}
```



Comandos da função /  
Corpo da função /  
Bloco de comandos da função

# 1. Funções e Procedimentos

- Exemplo prático – Sem utilizar Função:
- Faça um programa que leia dois valores e imprima a soma deles.

# 1. Funções e Procedimentos

- Exemplo prático – Sem utilizar função:
- Faça um programa que leia dois valores e imprima a soma deles.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a , b;
```

```
    scanf ( "%d %d", &a, &b);
```

```
    printf("%d + %d = %d", a, b, (a+b) );
```

```
    return 0;
```

```
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
int soma (int a, int b)    int main()
{
    int ret;
    ret = a + b;
    return ret;
}

{
    int a , b;
    scanf ( "%d %d", &a, &b);
    printf("%d + %d = %d", a, b,
soma(a, b) );
    return 0;
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
int soma (int a, int b)
{
    int ret;
    ret = a + b;
    return ret;
}
```

```
int main()
```

```
{
    int a , b;
    scanf ( "%d %d", &a, &b);
    printf("%d + %d = %d", a, b,
soma(a, b) );
    return 0;
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
int soma (int a, int b)
{
    int ret;
    ret = a + b;
    return ret;
}
```

```
int main()
{
    int a , b;
    scanf ( "%d %d", &a, &b);
    printf("%d + %d = %d", a, b,
soma(a, b) );
    return 0;
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
int soma (int a, int b)
{
    int ret;
    ret = a + b;
    return ret;
}
```

```
int main()
{
    int a , b;
    scanf ( "%d %d", &a, &b);
    printf("%d + %d = %d", a, b,
    soma(a, b) );
    return 0;
}
```

**a = 2;**  
**b = 2;**

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
int soma (int a, int b)
```

```
{
```

```
    int ret;
```

```
    ret = a + b;
```

```
    return ret;
```

```
}
```

```
int main()
```

```
{
```

```
    int a , b;
```

```
    scanf ( "%d %d", &a, &b);
```

```
    printf("%d + %d = %d", a, b,  
soma(a, b) );
```

```
    return 0;
```

```
}
```

**a = 2;**

**b = 2;**



# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
int soma (int a, int b)
{
    int ret;
    ret = a + b;
    return ret;
}
```

```
int main()
{
    int a , b;
    scanf ( "%d %d", &a, &b);
    printf("%d + %d = %d", a, b,
    soma(a, b) );
    return 0;
}
```

**a = 2;**  
**b = 2;**

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
int soma (int a, int b)
{
    int ret;
    ret = a + b;
    return ret;
}
```

```
int main()
{
    int a , b;
    scanf ( "%d %d", &a, &b);
    printf("%d + %d = %d", a, b,
    soma(a, b) );
    return 0;
}
```

**a = 2;**  
**b = 2;**

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
int soma (int a, int b)
{
    int ret;
    ret = a + b;
    return ret;
}
```

```
int main()
{
    int a , b;
    scanf ( "%d %d", &a, &b);
    printf("%d + %d = %d", a, b,
    soma(a, b) );
    return 0;
}
```

**Saída do programa**  
**a + b = 4**

# 1. Funções e Procedimentos

- Exercícios:

1. Crie uma função que receba como parâmetro o número inteiro e devolva o seu dobro.
2. Crie uma função que receba um número inteiro e devolva 1, se ele for par.
3. Crie uma função que receba 3 valores e devolva a soma entre eles.
4. Crie uma função que receba 2 números e devolva 0, se os números forem iguais senão retorna o número que for de maior valor.

# 1. Funções e Procedimentos

- Na linguagem C apenas existem funções.
- Outras linguagens tipificam esses dois tipos de rotinas.

Ex: Linguagem Pascal

- FUNCTION – funções.
- PROCEDURE – procedimento.

Então, onde está a diferença entre uma função e um procedimento que é abordado pela linguagem C ?

# 1. Funções e Procedimentos

Então, onde está a diferença entre uma função e um procedimento que é abordado pela linguagem C ?

- **Função** sempre tem um tipo e um valor de retorno associado a ela.
- **Procedimento** possui um tipo “neutro” e não devolve qualquer valor.

- EXEMPLOS:

- `int soma(int a, int b) { return a+b }`
- `void imprima() { }`

**OBS:** Se não for colocado o tipo de retorno em uma função, automaticamente será considerado o tipo `int` a ela.

# 1. Funções e Procedimentos


- Sintaxe de um procedimento:

```
void nome_do_procedimento (parâmetros)
{
    comando 1;
    comando 2;
    .....
}
```

# 1. Funções e Procedimentos

- **Exemplo:**

```
void imprima() { }
```



“tipo neutro” que indica que a função não possui valor de retorno.

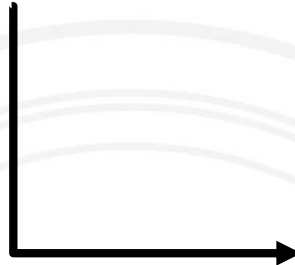
Uma função que possui como “tipo” Void é chamada de **procedimento**.



# 1. Funções e Procedimentos

- **Exemplo:**

```
void imprima() { }
```



Nome do procedimento

# 1. Funções e Procedimentos

- **Exemplo:**

```
void imprima( ) { }
```

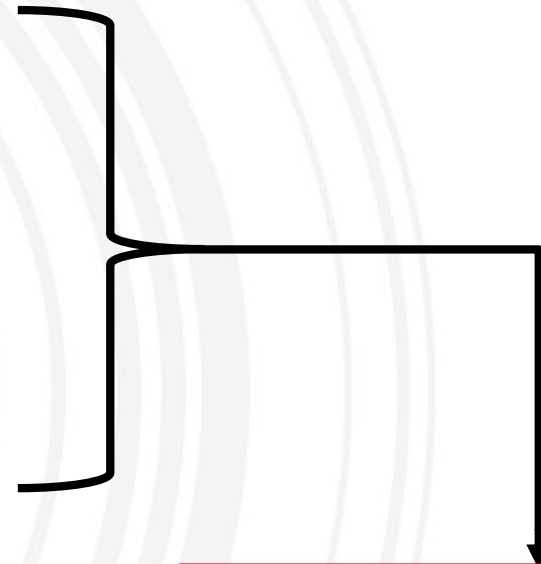


Lista de parâmetros

# 1. Funções e Procedimentos

- **Exemplo:**

```
void imprima( )  
{  
    int i;  
    for ( i = 1 ; i <= 10; i++)  
    {  
        printf("%d ", i);  
    }  
}
```



Corpo do procedimento

# 1. Funções e Procedimentos

- Exemplo prático – Sem utilizar função:
- Faça um programa que imprima uma sequência de 10 números.

```
#include <stdio.h>

int main()
{
    int i;
    for(i = 1; i <= 10; i++) {
        printf("%d ", i);
    }
    return 0;
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
void imprima ();  
int main()  
{  
    imprima();  
    return 0;  
}
```

```
void imprima ()  
{  
    int i;  
    for(i = 1; i <= 10; i++) {  
        printf("%d ", i);  
    }  
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
void imprima ();
```

```
int main()
```

```
{
```

```
    imprima();
```

```
    return 0;
```

```
}
```

```
void imprima ()
```

```
{
```

```
    int i;
```

```
    for(i = 1; i <= 10; i++) {
```

```
        printf("%d ", i);
```

```
    }
```

```
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
void imprima ();  
int main()  
{  
    imprima();  
    return 0;  
}
```

```
void imprima ()  
{  
    int i;  
    for(i = 1; i <= 10; i++) {  
        printf("%d ", i);  
    }  
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
void imprima ();  
int main()  
{  
    imprima();  
    return 0;  
}
```

```
void imprima()  
{  
    int i;  
    for(i = 1; i <= 10; i++) {  
        printf("%d ", i);  
    }  
}
```



# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
void imprima ();  
int main()  
{  
    imprima();  
    return 0;  
}
```

```
void imprima ()  
{  
    int i;  
    for(i = 1; i <= 10; i++) {  
        printf("%d ", i);  
    }  
}
```

# 1. Funções e Procedimentos

- Exemplo prático – Utilizando função:

```
#include <stdio.h>
```

```
void imprima ();  
int main()  
{  
    imprima();  
    return 0;  
}
```

```
void imprima ()  
{  
    int i;  
    for(i = 1; i <= 10; i++) {  
        printf("%d ", i);  
    }  
}
```

Saída do programa  
1 2 3 4 5 6 7 8 9 10

# 1. Funções e Procedimentos

- **Observações Finais:**

```
#include <stdio.h>
```

```
void imprima ();
```

```
int main()
```

```
{
```

```
    imprima();
```

```
    return 0;
```

```
}
```

## Protótipo de função

- Corresponde ao cabeçalho da função.
- Serve para identificar toda a estrutura da função.
- Indica ao compilador quais os parâmetros e o tipo de retorno da função

# 1. Funções e Procedimentos

- **Observações Finais:**

```
#include <stdio.h>
```

```
void imprima ()
```

```
{
```

```
    int i;
```

```
    for(i = 1; i <= 10; i++)
```

```
    {
```

```
        printf("%d ", i);
```

```
    }
```

```
}
```



## Váriaveis locais

- São declaradas e conhecidas **SOMENTE** dentro da função.
- Todas são destruídas ao final da execução da função.

# 1. Funções e Procedimentos

- **Observações Finais:**

```
#include <stdio.h>
```

```
int var;
```

```
void imprima ()
```

```
{
```

```
.....
```

```
}
```

```
int main()
```

```
{
```

```
.....
```

```
}
```

Váriaveis globais

- São declaradas e conhecidas por TODAS as funções existentes.

# 1. Funções e Procedimentos

- Exercícios:

1. Crie uma função que desenha uma linha na tela usando vários símbolos(Ex: =====). A função recebe por parâmetro quantos símbolos serão mostrados.
2. Crie uma função que receba um número inteiro e gere como saída n linhas com pontos de exclamação.
3. Crie uma função que receba 2 valores reais A e B, troca o valor delas e imprima seus novos valores.