

# Funções Básicas



**João Frederico Roldan Viana**

[jfredrv@gmail.com](mailto:jfredrv@gmail.com)

(85)99231.2777

# Agenda

---

- Funções Básicas
  - Biblioteca STDIO
    - printf, scanf
  - Biblioteca STDLIB
    - abs, trunc,
    - rand, srand
    - exit, abort, system
  - Biblioteca MATH
    - sin,cos, tan
    - log , log2, log10
    - pow, sqrt
    - ceil, floor

# Funções Básicas

## ■ Biblioteca

- Conjunto de funções que vem com o C
- Como inserir uma biblioteca para que o programa possa utilizar suas funções:
  - *#include <nome\_biblioteca>*
  - Obs: a linha deve está no cabeçalho do programa
- Exemplos:
  - *#include <stdio.h>*
  - *#include <stdlib.h>*
  - *#include <math.h>*

# Funções Básicas

## ■ printf

- Biblioteca stdio.h
- Escrever dados no dispositivo de saída padrão
- Sintaxe

➤ *int printf(const char \*cadeia, . . .);*

- Retorno: Quantidade de caracteres impressos
- Exemplos:

➤ `printf("Gosto de programar em C");`

➤ `printf("Gosto de %s em %c", "programar", 'C');`

# Funções Básicas

---

## ■ printf

- Especificadores de formato:
  - Informam como os outros parâmetros (...) da função serão interpretados e apresentados
  - Definidos pelo símbolo % seguido pelo código
  - Mesma quantidade de especificadores e argumentos
  - Relacionados da esquerda para direita

# Funções Básicas

## ■ printf

- Especificadores de formato:

Código	Formato		
c	Caractere		
s	Cadeia de caracteres		
d, i	Inteiro	decimal	
ld, li			longo
ll			longo longo
u		sem sinal	

# Funções Básicas

## ■ printf

- Especificadores de formato:

Código	Formato		
f	Ponto Flutuante		
Lf		longo (long double)	
e, E		Notação Científica	
Le, LE			Longo
g		Usa e ou f (o mais curto)	
%	Escreve o símbolo %		

# Funções Básicas

## ■ printf

- Largura Mínima de Campo:
  - Número colocado entre o % e o código
  - Preenche a saída com espaços
  - Caso coloque um 0 (zero) antes da largura, os espaços serão preenchidos com 0 (zeros)
  - Exemplos:
    - ✓ `printf("%f\n", 2.34);` Saída: 2.34
    - ✓ `printf("%6f\n", 2.34);` Saída: 2.34
    - ✓ `printf("%06f\n", 2.34);` Saída: 002.34



# Funções Básicas

## ■ printf

- Precisão:

- Segue o especificador de largura (se houver), separados por ponto (.)

- Exemplos:

- ✓ `printf("%.2f\n", 2.346);` Saída: 2.35

- ✓ `printf("%.4d\n", 100);` Saída: 0100

- ✓ `printf("%14s\n", "exemplo da utilização");`  
Saída: exemplo da uti

# Funções Básicas

## ■ scanf

- Biblioteca `stdio.h`
- Ler dados no dispositivo de entrada padrão
- Sintaxe
  - *`int scanf(const char *cadeia, . . .);`*
- Retorno: Quantidade de caracteres escritos
- Exemplos:
  - `scanf("%d", &num);`
  - `scanf("%d %f", &qtd, &valor);`

# Funções Básicas

---

## ■ scanf

- Especificadores de formato:
  - Informam como os outros parâmetros (...) da função serão interpretados e armazenados
  - Definidos pelo símbolo % seguido pelo código
  - Mesma quantidade de especificadores e argumentos
  - Relacionados da esquerda para direita

# Funções Básicas

## ■ scanf

- Especificadores de formato:

Código	Formato
c	Caractere
s	Cadeia de caracteres
d, i	Inteiro decimal com sinal
u	Inteiro decimal sem sinal
f, e, g	Ponto flutuante decimal

# Funções Básicas

## ■ scanf

- Definindo o formato da entrada:
  - Espaços em branco ou não-brancos indicam caracteres que devem ser descartados na entrada
  - Se o caractere não for encontrado, a função termina
  - Exemplos:
    - ✓ `scanf("%d %c", &num, &ch);` Entrada: 12 b
    - ✓ `scanf("%d,%c", &a, &b);` Entrada: 12,3
    - ✓ `scanf("%d/%d/%d", &dia, &mes, &ano);`  
Entrada: 12/10/2009

# Funções Básicas

## ■ abs

- Biblioteca `stdlib.h`
- Determina o valor absoluto de um número inteiro
- Sintaxe
  - *int abs(int num);*
  - *int abs(float num);*
- Retorno: Valor absoluto de *num* (int) ou parte inteira de *num* (float)
- Exemplos:
  - `abs(-100);` Retorno: 100
  - `abs(15.970);` Retorno: 15

# Funções Básicas

## ■ rand

- Biblioteca `stdlib.h`
- Gera um número pseudo-aleatório entre 0 e `MAX RAND`
- Sintaxe
  - *`int rand(void);`*
- Retorno: um número entre 0 e `MAX RAND`
- Exemplos:
  - `int a = rand();` (a entre 0 e `MAX RAND`)
  - `int b = rand() % (max + 1);` (b entre 0 e max)
  - `int c = (rand() % (max - min + 1)) + min;`  
(c entre min e max)

# Funções Básicas

## ■ srand

- Biblioteca `stdlib.h`
- Muda a semente que gera os números pseudo-aleatórios da função *rand*
- Sintaxe
  - *`void srand(unsigned semente);`*
- Ação: *semente* passa a ser o ponto de partida para a seqüência de número da função *rand*
- Exemplos:
  - `srand(1234567);`



# Funções Básicas

---

## ■ srand

- Uma forma de garantir que não seja gerado a cada execução o mesmo número pseudo-aleatório é passar como parâmetro da função srand uma semente que varie constantemente.
- Para isso podemos usar a função time(0) da biblioteca time.h
  - `srand(time(0));`

# Funções Básicas

---

## ■ exit

- Biblioteca `stdlib.h`
- Permite o abandono do processo ou programa, onde a função é evocada
- Sintaxe
  - *`void exit(int status);`*
- Status: Valor de retorno ao processo que chamou o processo que será terminado. Zero indica a não ocorrência de erro.
- Exemplos:
  - `exit(0);`

# Funções Básicas

## ■ abort

- Biblioteca `stdlib.h`
- Termina anormalmente um programa
- Sintaxe
  - *`void abort(void);`*
- Um vez chamada, a função envia a mensagem (“Abnormal program termination”) para o *`stderr`* e termina a execução do programa através da chamada da função `exit(3)`.
- Exemplos:
  - `abort();`

# Funções Básicas

## ■ system

- Biblioteca `stdlib.h`
- Executa um comando do sistema operacional
- Sintaxe
  - *`int system(const char *comando);`*
- Retorno: Zero caso a execução tenha ocorrido com sucesso e -1, caso contrário
- Exemplos:
  - `system("cls");`
  - `system("pause");`
  - `system("dir");`

# Funções Básicas

## ■ sin

- Biblioteca math.h
- Retorna o valor do seno
- Sintaxe
  - *double sin(double valor\_radiano);*
- Retorno: Seno do valor recebido em radiano
- Exemplo:
  - `sin(3.141592653589793/2);` Retorno: 1.000000
  - `sin(3.141592653589793);` Retorno: 0.000000

# Funções Básicas

## ■ cos

- Biblioteca math.h
- Retorna o valor do co-seno
- Sintaxe
  - *double cos(double valor\_radiano);*
- Retorno: Co-seno do valor recebido em radiano
- Exemplo:
  - `cos(3.141592653589793/2);` Retorno: 0.000000
  - `cos(3.141592653589793);` Retorno: 1.000000

# Funções Básicas

## ■ tan

- Biblioteca math.h
- Retorna o valor da tangente
- Sintaxe
  - *double tan(double valor\_radiano);*
- Retorno: Tangente do valor recebido em radiano
- Exemplo:
  - `tan(3.141592653589793/4);` Retorno: 1.000000
  - `tan(3.141592653589793/6);` Retorno: 0.577350

# Funções Básicas

## ■ log

- Biblioteca `math.h`
- Retorna o valor do logaritmo na base  $e$  (número de Euler)
- Sintaxe
  - *`double log(double num);`*
- Retorno: O logaritmo de *num* na base  $e$
- Exemplo:
  - `log(2.71828182845904523);` Retorno: 1.000000



# Funções Básicas

## ■ log2

- Biblioteca math.h
- Retorna o valor do logaritmo na base 2
- Sintaxe
  - *double log2(double num);*
- Retorno: O logaritmo de *num* na base 2
- Exemplo:
  - `log2(4.0);` Retorno: 2.000000

# Funções Básicas

## ■ log10

- Biblioteca math.h
- Retorna o valor do logaritmo na base 10
- Sintaxe
  - *double log10(double num);*
- Retorno: O logaritmo de *num* na base 10
- Exemplo:
  - `log10(10.0);` Retorno: 1.000000

# Funções Básicas

## ■ pow

- Biblioteca math.h
- Retorna o valor da base elevada ao expoente
- Sintaxe
  - *double pow(double base, double expoente);*
- Retorno: O valor de *base* elevado a *expoente*
- Exemplo:
  - `pow(2.0, 4.0);` Retorno: 16.000000
  - `pow(25.0, 0.5);` Retorno: 5.000000

# Funções Básicas

## ■ sqrt

- Biblioteca math.h
- Retorna o valor da raiz quadrada de um número
- Sintaxe
  - *double sqrt(double num);*
- Retorno: O valor da raiz quadrada de *num*
- Exemplo:
  - `sqrt(25.0);` Retorno: 5.000000
  - `sqrt(100.0);` Retorno: 10.000000

# Funções Básicas

## ■ ceil

- Biblioteca math.h
- Retorna, em double, o primeiro número inteiro acima
- Sintaxe
  - *double ceil(double num);*
- Retorno: O primeiro número inteiro acima de *num*
- Exemplo:
  - `ceil(45.98234);` Retorno: 46.000000
  - `ceil(45.18234);` Retorno: 46.000000

# Funções Básicas

## ■ floor

- Biblioteca math.h
- Retorna, em double, o primeiro número inteiro abaixo
- Sintaxe
  - *double floor(double num);*
- Retorno: O primeiro número inteiro abaixo de *num*
- Exemplo:
  - floor(45.98234); Retorno: 45.000000
  - floor(45.18234); Retorno: 45.000000