

Rodrigo Henrich

rodrigohenrich@faccat.br



Bibliotecas de funções em C

- Até o momento usamos algumas das bibliotecas de funções que a linguagem C disponibiliza
- Além das bibliotecas padrão, existem algumas disponibilizadas por terceiros

#include<stdio.h>

- É a biblioteca padrão para entrada (input) e saída (output) da linguagem C
- Algumas funções desta biblioteca
- **printf()** usada para imprimir dados na tela
- **scanf()** usada para capturar dados do usuário
- **fprintf()**, **fwrite()**, usada para imprimir dados em arquivo
- **fscanf()** usada para ler dados de arquivos
- **getchar()** usada para ler um caractere
- **gets()** usada para ler um vetor de caracteres

#include<stdio.h>

- **setbuf()** usada para setar o buffer do teclado
- Maiores detalhes em
<https://pubs.opengroup.org/onlinepubs/007908775/xsh/stdio.h.html>

#include<stdlib.h>

- Possui funções para alocação de memória, conversão de tipos, funções matemáticas
- **Funções de conversão de tipo**
- **atof()** converte uma String para float
- **atoi()** converte uma String para inteiro
- **atol()** converte uma String para um long int

#include<stdlib.h>

- **Funções de geração de números aleatórios**
- **rand()** gera um número aleatório
- **srand()** inicializa o gerador de números aleatórios com uma informação inicial

#include<stdlib.h>

- **Funções de alocação e liberação de memória**
- **malloc(), calloc(), realloc()** usadas para alocação dinâmica de memória
- **free()** usada para liberar a memória alocada

#include<stdlib.h>

- **Controle de processo**
- **abort()** força o término da execução
- **exit()** termina a execução do programa
- **getenv()** obtém uma variável de ambiente
- **system()** usada para chamar um programa externo

#include<stdlib.h>

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

int main() {
    // Nome da variável de ambiente que você deseja obter o valor
    char *env_var = "HOME";

    // Obtém o valor da variável de ambiente
    char *value = getenv(env_var);

    if (value == NULL) {
        // A variável de ambiente não foi encontrada
        printf("A variável de ambiente %s não está definida.\n", env_var);
    } else {
        // A variável de ambiente foi encontrada, imprime o valor
        printf("O valor da variável de ambiente %s é: %s\n", env_var, value);
    }
    return 0;
}
```

#include<stdlib.h>

- **Ordenamento e busca de dados**
- **bsearch()** realiza uma busca binária em um arquivo
- **qsort()** ordena um vetor usando o método Quick Sort

#include<stdlib.h>

- **Matemáticas**
- **abs()**, **labs()** retorna o valor absoluto de um número, ou seja sem o sinal
- **div()**, **ldiv()** divisão inteira entre dois números
- Mais informações em
<https://pubs.opengroup.org/onlinepubs/007908775/xsh/stdlib.h.html>

#include<stdlib.h>

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

int main() {
    long numerator = 10;
    long denominator = 3;

    // Usa a função ldiv para calcular quociente e resto
    ldiv_t result = ldiv(numerator, denominator);
    printf("Quociente: %ld\n", result.quot);
    printf("Resto: %ld\n", result.rem);
    return 0;
}
```

#include<math.h>

- **floor()** arredonda para baixo
- **ceil()** arredonda para cima
- **sqrt()** Calcula raiz quadrada
- **pow(variável, expoente)** potenciação
- **sin()** seno
- **cos()** cosseno
- **tan()** Tangente

#include<math.h>

- **log()** logaritmo natural
- **log10()** logaritmo base 10
- <https://pubs.opengroup.org/onlinepubs/007908775/xsh/math.h.html>

#include<string.h>

- **strcpy()** faz uma cópia de uma string para outra
- `strcpy(string_destino, string_origem);`

```
1  #include<stdio.h>
2  #include<string.h>
3  int main(){
4      char origem[] = "Palavra";
5      char destino[25];
6      strcpy(destino, origem);
7      printf("%s", destino);
8      //Equivale a
9      int i;
10     for(i=0; origem[i]!='\0'; i++){
11         destino[i] = origem[i];
12     }
13     destino[i]='\0';
14 }
```

#include<string.h>

- **strncpy()**; faz uma cópia de uma string para outra, especificando quantos caracteres devem ser copiados
- strncpy(string_destino, string_origem, tamanho);

```
1  #include<stdio.h>
2  #include<string.h>
3  int main(){
4      char origem[] = "Linguagem C";
5      char destino[10];
6      strncpy(destino, origem, 9);
7      printf("%s", destino);
8  }
```


#include<string.h>

- strcat(); concatena duas strings
- strcat(string_destino, string_origem);

```
1  #include<stdio.h>
2  #include<string.h>
3  int main(){
4      char strA[] = "Linguagem";
5      char strB[] = " C";
6      strcat(strA, strB);
7      printf("%s", strA);
8  }
```

#include<string.h>

- **strlen()**; retorna o tamanho de uma string
- `int strlen(String)`

```
1  #include<stdio.h>
2  #include<string.h>
3  int main(){
4      char palavra[] = "Linguagem C";
5      int tamanho = strlen(palavra);
6      printf("%s tem %d caracteres", palavra, tamanho);
7  }
```

#include<string.h>

- `int strcmp(str1, str2);` compara duas strings, seu resultado deve ser interpretado da seguinte forma
 - 0: conteúdo das strings são iguais
 - <0: conteúdo da string1 é menor do que string2
 - >0: conteúdo da string1 é maior do que string2

```
1  #include<stdio.h>
2  #include<string.h>
3  #include<locale.h>
4  int main(){
5      setlocale(LC_ALL, "");
6      char palavraA[] = "Linguagem C";
7      char palavraB[] = "Linguagem C";
8      if(strcmp(palavraA, palavraB)==0)
9          printf("São iguais");
10     else
11         printf("Não são iguais");
12 }
```

#include<string.h>

- `int strncmp(str1, str2, tamanho);`
compara duas strings, seu resultado deve ser interpretado da seguinte forma
 - 0: parte avaliada das strings é igual
 - <0: conteúdo da string1 é menor do que string2
 - >0: conteúdo da string1 é maior do que string2

```
1  #include<stdio.h>
2  #include<string.h>
3  #include<locale.h>
4  int main(){
5      setlocale(LC_ALL, "");
6      char palavraA[] = "Linguagem C";
7      char palavraB[] = "Linguagem C++";
8      if(strncmp(palavraA, palavraB, 9)==0)
9          printf("São iguais");
10     else
11         printf("Não são iguais");
12 }
```

#include<string.h>

- Maiores detalhes

<https://pubs.opengroup.org/onlinepubs/007908775/xsh/string.h.html>

#include<ctype.h>

- **Permite analisar dados relacionados a tipos**
- tolower() converte o caractere em minúsculo
- toupper() converte caractere minúsculo em maiúsculo.
- isalnum() verifica se o caracter é alfanumérico
- isalpha() verificar se o caracter é uma letra do alfabeto
- iscntrl() verificar se o caracter é um caracter de controle
- isdigit() verificar se o caracter é um dígito decimal

#include<ctype.h>

- **isgraph()** verifica se o caracter tem representação gráfica
- **islower()** verifica se o caracter é minúsculo
- **isprint()** verifica se o caracter é imprimível
- **ispunct()** verifica se o caracter é um ponto
- **isspace()** verificar se o caracter é um espaço em branco
- **isupper()** verifica se o caracter é uma letra maiúscula
- **isxdigit()** verifica se o caracter é um dígito hexadecimal

#include<stdbool.h>

- Adiciona o tipo bool a linguagem
- Variáveis deste tipo podem assumir apenas dois valores
 - true ou false;