

# Manipulação de String

## Programação Estruturada em C



# Roteiro

---

- **String**

  - Definição

  - Declaração

  - Caracteres especiais

  - Ler e Impressão

- **Funções da string.h**

- **Outras Funções**

- **Duvidas**

- **Exercícios**

## String - Definição

---

- Uma String, para programação, é um tipo de dado representando uma palavra, uma frase ou um texto, normalmente composto de caracteres.
- Em C, não há um tipo de dado específico para strings (ao contrário de Java e C++, por exemplo), mas há suporte para seu uso.
- A definição de String, para a linguagem C, é:  
  
**“Um vetor de caracteres terminado pelo caractere ‘\0’, o caractere nulo, de valor inteiro 0”.**
- Há suporte para várias operações com strings na linguagem C, através de funções da biblioteca **string.h**.

# String - Declaração

---

- Podemos declarar uma string de duas formas:  
Da mesma forma que declaramos um vetor de caracteres, terminando com o caractere `'\0'`:

```
char faculdade[] = {'U', 'F', 'P', 'E', '\0'};  
char curso[3] = {'E', 'C', '\0'};  
char disciplina[20] = {'I', 'P', '\0'};
```

Usando strings constantes\*:

```
char faculdade[] = "UFPE";  
char curso[3] = "EC";  
char disciplina[20] = "IP";
```

# String – Caracteres especiais

- Existem caracteres especiais, usados nas strings:

Caractere	Uso
\\	Barra
\"	Aspas duplas.
\'	Aspas simples.
\n	Nova linha, com cursor no início.
\r	Retorno ao começo da linha.
\b	Backspace.
\t	Tabulação horizontal (TAB).
\f	Nova linha, com cursor abaixo de onde estava.
\a	Alerta (bipe).
\v	Tabulação vertical.
\nnn ou \xnn	Caractere de valor <i>nnn</i> (octal) ou <i>nn</i> (hexadecimal)

## String – Ler e Impressão

---

- Para ler uma string do usuário usamos a função `scanf()`, com `%s` :

```
scanf("%s %s", nome, sobrenome);
```

- Para imprimir uma string, usamos a `printf()`, com `%s` :

```
printf("%s %s", nome, sobrenome);
```

- Observação: O `scanf` com formato `"%s"` só captura uma palavra por vez. Para capturar uma linha inteira, se usa a formatação `"%[^\n]"`, ou use a função `gets(char* string)`.

## String – Ler e Impressão

---

```
// ler com espaço na palavra
#include <stdio.h>
#include <stdlib.h>

void main(){

    char pnome[20];
    char snome[20];

    printf("Digite o nome:");

    scanf("%[^\n]s", pnome); // manda ignorar enter
    setbuf(stdin, NULL); // elimina o enter ignorado da linha acima

    printf("Veio:%s",pnome);
}
```

## Funções da string.h

---

- Pelo fato que strings são vetores, sua manipulação é complicada, é feita acessando cada posição.
- Como operações com strings são bastante comuns, e manipular vetores é uma tarefa cansativa, há a biblioteca “string.h” com funções para facilitar essa manipulação.



## Funções da string.h

---

```
char* strcpy(char* destino, char* fonte)
```

- Usada para copiar a string **fonte** para o vetor **destino**.
- É uma alternativa ao operador "=", que não pode ser usado com vetores e, portanto, strings.
- É tarefa do programador garantir que o vetor **destino** seja grande o suficiente para armazenar a string em **fonte**.
- Retorna o endereço da string **destino**.

## Funções da string.h

---

```
char* strcat(char* destino, char* fonte);
```

- Usada para concatenar a string **fonte** com a string **destino**.
- A string **fonte** será copiada para o final da string **destino**.
- É tarefa do programador garantir que o vetor **destino** seja grande o suficiente para armazenar as duas strings.
- Retorna o endereço da string **destino**.

## Funções da string.h

---

```
char* strcmp(char* string1, char* string2);
```

- Usada para comparar as duas strings.
- Se **string1** for lexicamente anterior à **string2**, strcmp() retorna um inteiro negativo.
- Se **string1** for lexicamente posterior à **string2**, strcmp() retorna um inteiro positivo.
- Se as duas strings forem iguais, strcmp() retorna 0.

## Funções da string.h

---

```
int strlen(char* string);
```

- Retorna o tamanho da string **string**.
- O tamanho não inclui o '\0' do fim da string.

## Funções da string.h

---

```
char* strstr(char* palheiro, char agulha);
```

- Procura a string **agulha** dentro da string **Palheiro**.
- Se achar, retorna o endereço da primeira ocorrência de **agulha** na string **palheiro**.
- Se não achar, retorna NULL.

## Outras Funções

---

- Temos, na “stdio.h”, duas funções interessantes para strings:

```
int sprintf (char * string, char * format, ... );  
int sscanf (char * string, char * format, ...);
```

- sprintf() é usada como a printf(), mas imprime na string, e não na saída padrão.
- sscanf() é usada como a scanf(), mas captura dados da string, e não da entrada padrão.

## Outras Funções

---

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

int main()
{
    char str[80];

    sprintf(str, "Value of Pi = %f", M_PI);
    puts(str);

    return(0);
}
```

## Outras Funções

---

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

int main()
{
    int day, year;
    char weekday[20], month[20], dtm[100];

    strcpy( dtm, "Saturday March 25 1989" );
    sscanf( dtm, "%s %s %d %d", weekday, month, &day, &year );

    printf("%s %d, %d = %s\n", month, day, year, weekday );

    return(0);
}
```



---

**Dúvidas?**

## Exercício 1

---

- **Faça um programa que peça ao usuário:**
  - (i) Uma string `s`,
  - (ii) Um caractere `ch1`,
  - (iii) Um caractere `ch2`.
- **E troque todos os caracteres, iguais a `ch1`, da string `s` pelo caractere `ch2`.**

## Exercício 2

---

- Escreva um programa que receba uma sequência de strings (um nome e 4 sobrenomes) numa combinação de letras maiúsculas e minúsculas. A saída deve ser uma string com o nome completo com as primeiras letras de cada palavra maiúsculas.

Entrada:

- JOAO cArLOS DA silVA Junior

Saída:

- Joao Carlos Da Silva Junior

## Exercício 3

---

- **Elabore um programa que receba uma linha de texto e conte as vogais apresentando o respectivo histograma na seguinte forma:**

Exemplo:

- Entrada: “Na próxima quarta-feira é feriado.”
- Saída:
  - a: \*\*\*\*\* (6)
  - e : \*\*\* (3)
  - i : \*\*\* (3)
  - o : \*\* (2)
  - u : \* (1)

## Opcional

---

- Dada uma string, inverter a ordem das palavras.

Exemplo:

Entrada: "uma frase qualquer"

Saída: "qualquer frase uma"