



Strings

- Conjunto de caracteres (char)
- Armazenados em formato de matrizes unidimensionais (arrays, vetores)
- Toda string encerra com um valor nulo (\0) em seu final, incluído automaticamente pelo compilador
- Seu tamanho deve ser declarado levando em conta o caractere de encerramento

```
char strExemplo[18] = "C e na softblue!!";
```

C	e	n	a	s	o	f	t	b	l	u	e	!	!	\0			
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]

Declarando strings

- Especificando o tamanho
- Não especificando o tamanho

```
char minhaStr[4] = "ola";
printf("%s", minhaStr);      // ola

char minhaStr[3] = "ola";
printf("%s", minhaStr);      // olaa##

char minhaStr[4] = {'o', 'l', 'a', '\0'};
printf("%s", minhaStr);      // ola
```

```
char minhaStr[] = "ola";
printf("%s", minhaStr);      // ola

char minhaStr[];
minhaStr = "ola";
printf("%s", minhaStr);      // Erro

char *minhaStr;
minhaStr = "ola";
printf("%s", minhaStr);      // ola
```

Interagindo com o usuário



- Biblioteca **stdio.h**
- Capturando valores
 - gets
 - gets vs. scanf
- Imprimindo valores
 - puts
 - puts vs. printf

```
char minhaString[50];  
gets(minhaString);  
puts(minhaString);
```

Matriz de strings



- Matriz dentro de matriz
- Matriz bidimensional

```
char matrizDeStrings[3][20];  
gets(matrizDeStrings[0]);  
gets(matrizDeStrings[1]);  
gets(matrizDeStrings[2]);  
  
for(x=0; x<3; x++)  
{  
    puts(matrizDeStrings[x]);  
}
```

Funções interessantes



- Biblioteca **string.h**
 - **strcpy**: Copia uma string
 - **strcat**: Concatena duas strings
 - **strlen**: Retorna o tamanho da string
 - **strcmp**: Compara se duas strings são iguais
 - **strchr**: Retorna o ponteiro para a primeira ocorrência de um caractere
 - **strrchr**: Retorna o ponteiro para a última ocorrência de um caractere
 - **strstr**: Retorna o ponteiro para a primeira ocorrência de uma string

strcpy



- Copia uma string em outra
- Altera o valor de uma string

Sintaxe

```
(char *) strcpy(char *destino, char *origem);
```

Exemplo

```
char str1[] = "Minha string 1";
char str2[] = "Minha string 2";
strcpy(str2, str1);
printf("%s", str2); // Minha string 1

strcpy(str2, "Softblue");
printf("%s", str2); // Softblue
```

strcat



- Concatena a segunda string na primeira

Sintaxe

```
(char *) strcat(char *destino, const char *conteudo);
```

Exemplo

```
char str1[] = "Minha string 1";
char str2[] = "Minha string 2";
strcat(str2, str1);
printf("%s", str2); // Minha string 2Minha string 1
```

strlen



- Retorna o tamanho da string

Sintaxe


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

M	i	n	h	a		s	t	r	i	n	g		\0	
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]

Exemplo

```
char str1[] = "Minha string 1";
printf("%d", strlen(str1)); // 14
```

strcmp



- Compara se duas strings são iguais
- Operador `==` não funciona


Sintaxe

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

Exemplo

```
char str1[] = "Minha string 1";
if(!strcmp(str1, "Minha string 1"))
{
    printf("Iguais.");
}
else
{
    printf("Diferentes.");
}
```

strchr



- Retorna o ponteiro da primeira ocorrência

Sintaxe


```
(char *) strchr(char *string, char caractere);
```

M	i	n	h	a		s	t	r	i	n	g		1	\0
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]

Exemplo

```
char str1[] = "Minha string 1";
char *pos = strchr(str1, 'n');
printf("%s", pos); // nha string 1
```

strrchr



- Retorna o ponteiro da última ocorrência

Sintaxe

```
(char *) strrchr(char *string, char caractere);
```

M	i	n	h	a		s	t	r	i	n	g		1	\0
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]

Exemplo

```
char str1[] = "Minha string 1";
char *pos = strrchr(str1, 'n');
printf("%s", pos); // ng 1
```

Códigos de escape



- Caracteres especiais utilizados em strings

Código	Descrição
\a	Beep (som)
\b	Backspace (BS)
\f	Alimentação de formulário (FF)
\n	Nova linha (LF)
\r	Retorno (CR)
\t	Tabulação horizontal
\v	Tabulação vertical
\'	Aspas simples
\"	Aspas duplas
\0	Nulo
\\	Barra invertida

Códigos de escape



- Utilizando códigos de barra invertida

Exemplos

```
printf("Te\nste"); // Te
                  // ste
printf("\'Teste\"); // 'Teste'
printf("\"Teste\""); // "Teste"
printf("Te\tste"); // Te  ste
```

Conversão entre textos e números



- Funções para conversão
 - Biblioteca **stdlib.h**
 - **atoi**: string para inteiro
 - **atof**: string para ponto flutuante