

Introdução à Programação #20

Henrique Dias @ Pplware

14 de janeiro de 2015

1 *Strings*

Strings, como já mencionado anteriormente, são sequências de caracteres e, em C, são *arrays* de caracteres com um delimitador que indica o final da *string*, o delimitador `\0`. Daí termos abordado primeiro os *arrays*.

O \0 é ASCII para 0 e é utilizado para finalizar "strings".

```
1 "Pplware" -> String
2 "P" -> String
3 'P' -> Character
```

Como mostrado acima, os caracteres, já abordados, são colocados dentro de plicas enquanto que todas as *strings* são colocadas dentro de aspas. Um único caracter inserido dentro de aspas é considerado uma *string*.

Sintaxe Em C podemos declarar *strings* de diversas formas. Podemos declarar um *array* do tipo *char* com um número de caracteres definidos:

```
1 char nome[8] = "Pplware";
```

Nesse *string* podemos colocar a palavra “Pplware” que tem 7 caracteres, mas qual é a razão que nos leva a colocar 8 quando declaramos a variável? Não se esqueça do caracter `\0` que conta como 1 caracter.

Podemos ainda declarar e inicializar *strings* das seguintes formas:

```
1 char nome[8] = {'P', 'p', 'l', 'w', 'a', 'r', 'e', '\0'}; /* Forma mais trabalhosa */
2 char nome[] = "Pplware"; /* Não há necessidade de colocar o seu tamanho. */
3 char *nome = "Pplware";
```

Este “tipo” de dados pode, à semelhança de todos os outros, ser integrado com a função *printf* utilizando “%s”, uma abreviatura para *string*.

Exemplo No exemplo seguinte declaramos uma *string* que depois é imprimida recorrendo à função *printf*.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char *nome = "Vamos Programar? – Introdução à Programação";
7
8     printf("%s\n", nome);
9     return 0;
10 }
```