

Constantes

Constantes em C++

Em **C**, além de variáveis, nós podemos usar também números ou caracteres cujos valores não mudam. Eles são chamados de **CONSTANTES**. Constantes não são associados a lugares na memória. Assim como variáveis, constantes também têm tipos.

Uma constante pode ser do tipo `int`, `char` , *etc.*

Forma literal – Utilizando a palavra chave `const`

```
const char hello[] = "Hello, literal constant!";
```

Como Macros

Para declarar uma constante em C usa-se o identificador define com tralha `#` na frente dele, e é definido logo depois do cabeçalho fora das funções. Geralmente usa-se palavras em MAIÚSCULAS sem acento, sem cedilha, sem traços, ...

Exemplo de uso de constantes em C:

```
#define HELLO "Hello, macro constant!"
```

Essa prática de declarar constantes como Macros é uma prática deve ser evitada em códigos Modernos da Linguagem C. De acordo com o histórico de erros em programas, percebeu-se que a maioria estava associada à essas constantes como macro. Logo, evite de usar constante como macros.

Constantes em C

Vejamos um exemplo agora que engloba as duas formas:

Analise o código e compile: `gcc constant-hello.c -o constant-hello`

```
#include <stdio.h>
#define HELLO "Hello, constant!"

int main(){
    const char hello[] = "Hello constant, again!";
    printf("Constant Macro HELLO: %s\n", HELLO);
    printf("Constant literal hello: %s\n", hello);
    return 0;
}
```

Constantes em C

Aproveitando que usamos bastante newline , vamos conhecer outros códigos de escape de caractere único . Existem diversos, vamos mostrar todos e uma breve descrição do que cada um faz:

Código de Escape	Descrição
<code>\n</code>	nova linha
<code>\r</code>	retorno de carro
<code>\t</code>	tab
<code>\v</code>	vertical tab
<code>\b</code>	backspace
<code>\f</code>	form feed (page feed)
<code>\a</code>	alert (beep)
<code>\'</code>	aspas simples (')
<code>\"</code>	aspas duplas (")
<code>\?</code>	interrogação (?)
<code>\\</code>	barras invertidas (\)

Constantes em C

Exemplo incluindo diversos códigos de escape:

```
#include <stdio.h>
#define HELLO "Hello, constant!"

int main(){
    const char hello[] = "Hello constant, again!";

    // Sem código de escapes
    printf("Constant Macro HELLO: %s", HELLO);
    printf("Constant literal hello: %s", hello);

    // Com o código de escape \n
    printf("Constant Macro HELLO: %s\n", HELLO);
    printf("Constant literal hello: %s\n", hello);

    // Com o código de escape \r
    printf("Constant Macro HELLO(com \\r): %s\r", HELLO);
    printf("Constant literal hello(com \\r): %s\r", hello);

    // Com o código de escape \t
    printf("Constant Macro HELLO(com \\t): %s\t", HELLO);
    printf("Constant literal hello(com \\t): %s\t", hello);

    // Com o código de escape \v
    printf("Constant Macro HELLO(com \\v): %s\v", HELLO);
    printf("Constant literal hello(com \\v): %s\v", hello);

    // Com o código de escape \b
    printf("Constant Macro HELLO(com \\b): %s\b", HELLO);
    printf("Constant literal hello(com \\b): %s\b", hello);

    // Com o código de escape \f
    printf("Constant Macro HELLO(com \\f): %s\f", HELLO);
    printf("Constant literal hello(com \\f): %s\f", hello);

    // Com o código de escape \a
    printf("Constant Macro HELLO(com \\a): %s\a", HELLO);
    printf("Constant literal hello(com \\a): %s\a", hello);

    return 0;
}
```