

Introdução a Programação - Prof. Mauro Gil

mgil@unicarioca.edu.br
[@maurocgil](#)

Função printf()

- Sintaxe:

```
printf("expressão de controle",argumentos);
```

É uma função de I/O, que permite escrever no dispositivo padrão (tela). A expressão de controle pode conter caracteres que serão exibidos na tela e os códigos de formatação que indicam o formato em que os argumentos devem ser impressos. Cada argumento deve ser separado por vírgula.

Função printf()

\n	nova linha	%c	caractere simples
\t	tab	%d ou %i	decimal
\b	retrocesso	%e	notação científica
\"	aspas	%f	ponto flutuante
\\	barra	%o	octal
\f	salta formulário	%s	cadeia de caracteres
\0	nulo	%u	decimal sem sinal
%x	hexadecimal		

Função printf()

exemplo02_01

```
main()
{
    printf("Este é o numero quatro: %d\n",4);
    printf("%s está a %d bilhões de km\ndo sol", "Saturno",2);
}
```

Função printf()

exemplo02_02: (Tamanho de campos na impressão)

```
main()
{
    printf("\n%2d",350);
    printf("\n%4d",350);
    printf("\n%6d",350);
}
```

Função printf()

exemplo02_03 (arredondamento)

```
main()
{
    printf("%4.2f\n",3456.78);
    printf("%3.2f\n",3456.78);
    printf("%3.1f\n",3456.78);
    printf("%011.3f\n",3456.78);
}
```

Função printf()

exemplo02_04 (alinhamento)

```
main()
{
    printf("%10.2f %10.2f %10.2f\n",8.0,15.3,584.13);
    printf("%10.2f %10.2f %10.2f\n",834.0,1500.55,4890.21);
}
```

Função printf()

exemplo02_05 (complementando com zeros à esquerda)

```
main()
{
    printf("\n%04d",21);
    printf("\n%06d",21);
    printf("\n%6.4d",21);
    printf("\n%6.0d",21);
}
```


Função printf()

exemplo02_06 (imprimindo caracteres)

```
main()
{
    printf("%d %c %x %o\n",'A','A','A','A');
    printf("%c %c %c %c\n",'A',65,0x41,0101);
}
```

A tabela ASCII possui 256 códigos de 0 a 255, se imprimirmos em formato caractere um número maior que 255, será impresso o resto da divisão do número por 256; se o número for 3393 será impresso A pois o resto de 3393 por 256 é 65.

A linguagem C adota a seguinte convenção:

Na representação decimal, o primeiro dígito é diferente de 0;

Na representação octal, o primeiro dígito é 0;

Na representação hexadecimal, os dois primeiros dígitos caracteres são 0x.

Assim, 0101 é a representação octal e 0x41 é a representação hexadecimal de sessenta e cinco.

Função scanf()

- Sintaxe:

```
scanf("expressão de controle", argumentos);
```

Também é uma função de I/O implementada em todos compiladores C. Ela é o complemento de printf() e nos permite ler dados formatados da entrada padrão (teclado).

Função scanf()

- A lista de argumentos deve consistir nos endereços das variáveis. C oferece um operador para tipos básicos chamado operador de endereço e referenciado pelo símbolo "&" que retorna o endereço do operando.

Operador de endereço &:

- A memória do computador é dividida em bytes, e são numerados de 0 até o limite da memória. Estas posições são chamadas de endereços. Toda variável ocupa uma certa localização na memória, e seu endereço é o primeiro byte ocupado por ela.

Função scanf()

exemplo02_07

```
main()
{
    int num;
    printf("Digite um número: ");
    scanf("%d",&num);
    printf("\no número é %d",num);
    printf("\no endereço e %u\n",&num);
}
```

Função getchar ()

- É a função original de entrada de caractere dos sistemas baseados em UNIX. getchar() armazena a entrada até que ENTER seja pressionada.

exemplo02_08

```
main()
{
    char ch;
    ch=getchar();
    printf("%c\n",ch);
}
```

Função putchar ()

- Escreve na tela o argumento de seu caractere na posição corrente.

exemplo02_09 e exemplo02_10 (*)

```
main()
{
    char ch;
    printf("digite uma letra minúscula : ");
    ch=getchar();
    putchar(toupper(ch));
    putchar('\n');
}
```