

# Entrada e Saída Console

Prof. Gabriel Barbosa da Fonseca  
Adaptado de Profa. Rosilane Mota

## Saída

---

A saída padrão de um programa desenvolvido para Console é a tela.

Como os dados armazenados na memória do computador são apenas 0's e 1's, o tipo indica como representar a informação. Mas a disposição dela na saída também pode ser formatada.

---

## Em C – printf (stdio.h)

`printf(.....);`

Escreve a informação entre parênteses a partir da posição atual do cursor no Console

`printf("....%?...%?.....",var1, var2,...);`

Nas interrogações os conteúdos das variáveis na ordem indicada é integrado ao texto (conjunto de caracteres).


# Saída formatada

---

printf ( ... ).

| Especificador | Tipo                               |
|---------------|------------------------------------|
| %d            | int                                |
| %.xf          | float (opcional x casas decimais)  |
| %.xlf         | double (opcional x casas decimais) |
| %s            | char[]                             |
| %b            | bool                               |
| %c            | char                               |

Indica-se com o especificador a posição na string de saída na qual o dado será formatado, sendo que a ordem dos especificadores corresponde à ordem das variáveis indicadas após a string



printf ("O carro possui %d portas e sua velocidade é %.2lf km/h", quantidade, veloc);

1º      2º

# Saída formatada

printf ( ).

| Especificador | Tipo                               |
|---------------|------------------------------------|
| %d            | int                                |
| %.xf          | float (opcional x casas decimais)  |
| %2.xf         | double (opcional x casas decimais) |
| %s            | char[]                             |
| %b            | bool                               |
| %c            | char                               |

| Códigos de formatação para printf() | Significado  |
|-------------------------------------|--|
| %c                                  | Caractere simples.                                 |
| %d                                  | Inteiro decimal com sinal.                         |
| %i                                  | Inteiro decimal com sinal.                         |
| %e                                  | Notação científica (e minúsculo).                  |
| %E                                  | Notação científica (E maiúsculo).                  |
| %f                                  | Ponto flutuante em decimal.                        |
| %g                                  | Usa %e ou %f, o que for menor.                     |
| %G                                  | Usa %E ou %f, o que for menor.                     |
| %o                                  | Inteiro octal sem sinal.                           |
| %s                                  | String de caracteres.                              |
| %u                                  | Inteiro decimal sem sinal.                         |
| %x                                  | Inteiro hexadecimal sem sinal (letras minúsculas). |
| %X                                  | Inteiro hexadecimal sem sinal (letras maiúsculas). |
| %p                                  | Ponteiro (endereço).                               |
| %n                                  | Ponteiro inteiro.                                  |
| %%                                  | Imprime um caractere %.                            |

## Exemplos

---

```
printf("Digite um número: ");
```

Digite um número: 5\_

```
printf("Digite um número: \n");
```

Digite um número: 5  
\_

```
printf("Os números %d e %.2f foram digitados", semCasa, com2Casas);
```

---

Os números 6 e 3.14 foram digitados\_

## Caracteres especiais

---

Dentro da string (entre os aspas), pode-se incluir após uma ‘\’ caracteres que assumem funções especiais

|    |            |
|----|------------|
| \n | Nova linha |
| \t | Tabulação  |
| \\ | Exibe a \  |
| \" | Exibe o "  |

---

## Em C

```
printf("Esse é\num exemplo\n");
```

Esse é  
um exemplo

—

## Concatenação String

---

Uma string deve começar e terminar em uma mesma linha, ou seja, o abre aspas e o fecha aspas devem estar na mesma linha.

Para textos maiores, pode-se utilizar mais de uma string em sequência, delimitadas por aspas.

---

## Em C

```
printf("Esse é um texto "  
      "muito extenso\n");
```

Esse é um texto muito extenso

—

## Em C

### Entrada – scanf (stdio.h)

---

A entrada padrão de um programa desenvolvido para Console é o teclado.

O comando scanf possui o seguinte formato geral:

```
scanf("%? %? %?",&var1,&var2,&var3);
```

---

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

```
int main(void) {
```

```
    int a, b;
```

```
    double val;
```

```
    float val2;
```

```
    char c;
```

```
    printf("Hello"
```

```
    " World\n");
```

```
    scanf("%d %d "
```

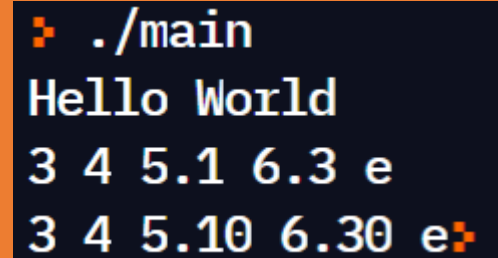
```
    "%lf %f %c",&a, &b,
```

```
    &val,&val2, &c);
```

```
    printf("%d %d %.2lf %.2f %c",a, b,val,val2,c);
```

```
    return 0;
```

```
}
```

A terminal window with a dark background and orange border. It shows the execution of a C program. The prompt is './main'. The program outputs 'Hello World' followed by a newline. Then it prompts for input with '3 4 5.1 6.3 e'. The user enters '3 4 5.10 6.30 e' and presses enter, indicated by a cursor arrow at the end.

```
> ./main
Hello World
3 4 5.1 6.3 e
3 4 5.10 6.30 e>
```



# Entrada e Saída Console

Prof. Gabriel Barbosa da Fonseca

Adaptado de Profa. Rosilane Mota