



Capítulo 2. Entrada e Saída

Tópico 1. Saída

Explicação 1. A saída em C++ é dada através do **objeto** `cout` e o operador `<<`. Observe bem a direção dos operadores: é como se eles estivessem apontando para a esquerda. Isso significa que qualquer coisa que esteja à direita do operador será “jogado” para o objeto `cout`. Assuma por enquanto que `cout` representa a tela do computador. Como a tela é um periférico de saída, as informações deve ir em direção à tela.

Exemplo 1. Exibindo coisas na tela.

```
cout << "Olá!";  
cout << 5;
```

Explicação 2. É possível concatenar informações e exibir na saída de uma única vez. Basta usar vários operadores `<<`. Além disso, você pode misturar tipos de dados.

Exemplo 1. Concatenando informações e jogando tudo para `cout`.

```
cout << "Olá! Eu tenho " << 5 + 5 << " borrachas";
```

Exemplo 2. Observe que o texto que está entre aspas é uma string, ou seja, uma frase. Significa que tudo que está entre as aspas é entendido como uma informação única, que é jogada para `cout` do jeito que está. Quando temos palavras sem aspas, isso é entendido pela linguagem como nome de variáveis. Neste exemplo, o programa irá exibir o valor de uma variável, junto com uma frase.

```
int x;  
x = 50 - 25;  
cout << "O valor de x é " << x;
```

Explicação 3. Para pular de linha, use a palavra **`endl`**, também concatenada.

Exemplo 1. Pulando de linha. O termo técnico é “quebra de linha”, mas nós somos programadores, e não precisamos falar que nem gente¹.

```
cout << "Linha um. Ela vai ser quebrada... " << endl;
cout << "Pronto. Foi quebrada! ";
```

Ou seja, o resultado do código acima será:

```
Linha um. Ela vai ser quebrada...
Pronto. Foi quebrada!
```

Exemplo 2. Pulando várias linhas de uma única vez.

```
cout << "Vão ser puladas duas linhas." << endl << endl;
cout << "Este" << endl << "texto" << "pula" << endl << "linhas";
```

O resultado do código acima será:

```
Vão ser puladas duas linhas.

Este
texto
pula
linhas
```

Exemplo 3. Se você não der **endl**, não adianta fazer o programa em várias linhas, pois nenhuma linha será pulada.

```
cout << "Aqui até parece";
cout << "Que várias linhas foram puladas " << endl << "mas ";
cout << "isso é enganoso";
```

O resultado será:

```
Aqui até pareceQue várias linhas foram puladas
mas isso é enganoso
```

No exemplo, algumas coisas devem ser observadas. Por exemplo, na primeira linha de código, “Aqui até parece” não termina com um **endl**, o que significa que nenhuma linha será quebrada. Além disso, observe que não colocamos nenhum espaço depois da palavra “parece”, e isso vai fazer com que a próxima coisa saia “grudada” na palavra parece, como, de fato, aconteceu no resultado.

Exemplo 4. Não confunda nome de variáveis com conteúdo de strings, que são frases. As strings vêm entre aspas duplas; as variáveis, vêm sem aspas.

```
int atchim = 5;
cout << "atchim" << atchim << endl;
```

O resultado será:

```
atchim 5
```

¹ Em homenagem ao Escudeiro.

Tópico 2. Entrada

Explicação 4. A entrada de dados é feita *a partir do* objeto **cin**. Assuma que cin represente o seu teclado. Temos que **cin** “joga” as informações para dentro de uma variável qualquer já existente. Para facilitar as coisas, vamos trabalhar apenas com valores numéricos. O operador que faz com que **cin** jogue informações para dentro de uma variável é o **>>**, visto que a informação vem do **cin**, e vai em direção à variável.

Exemplo 1. Lendo um valor numérico e exibindo o mesmo.

```
int valor;
cout << "Digite o valor numérico, cara! ";
cin >> valor;
cout << "Você digitou " << valor << endl;
```

Na primeira linha, declaramos uma variável int chamada valor. Na segunda linha, exibimos uma mensagem na tela. Pressupõe-se que o usuário saiba ler. Na terceira linha, o valor que o usuário digitou é armazenado na variável **valor**. Na última linha, está claro o que acontece.

Exemplo 2. Lendo dois valores numéricos e exibindo a soma entre eles.

```
cout << "Digite o primeiro valor: ";
int x;
cin >> x;
cout << "Digite o segundo valor: ";
int y;
cin >> y;
int soma = x + y;
cout << "A soma é " << soma << endl;
```

Observe que neste exemplo, as variáveis **x** e **y** não foram declaradas no início do programa; em vez disso, foram declaradas conforme necessitou-se de tais variáveis.

Exemplo 3. Lendo dois valores numéricos em seqüência, e exibindo o produto deles.

```
cout << "Digite dois números, dando <ENTER> após cada um deles: ";
int a, b;
cin >> a;
cin >> b;
int x = a * b;
cout << "O produto é " << x << endl;
```

Vale observar que, assim como no exemplo anterior, a gente declarou uma variável para armazenar o resultado. Mas isso não é necessário. A gente pode fazer a multiplicação na mesma hora que joga para cout:

```
cout << "O produto é " << a * b << endl;
```

Exemplo 4. Descobrimo quando o cara nasceu. Como é que descobre? Não, não é adivinhando². É perguntando. Essa é velha, mas serve para nosso propósito de dominar o mundo.

```
cout << "Digite o ano em que estamos: ";
int ano_atual;
cin >> ano_atual;
cout << "Agora, digite a sua idade: ";
int idade;
cin >> idade;
cout << "Você nasceu em " << ano_atual - idade << endl;
```

² Muito obrigado, prof. Escudeiro!