



# ESTRUTURA DE DADOS – 03A – 2022.2

[Página inicial](#)[Meus cursos](#)[ESTRUTURA DE DADOS – 03A – 2022.2](#)[Tópico 2. Breve introdução ao C++](#)[Manipulando a saída com a biblioteca iomanip](#)

## Manipulando a saída com a biblioteca iomanip

### Biblioteca iomanip

A biblioteca `iomanip` é utilizada para formatar a saída das informações em programas C++. O arquivo de cabeçalho `iomanip` fornece manipuladores parametrizados para manipulação das informações de saída que serão exibidas no console.

Esta biblioteca dispõe das seguintes [funções](#):

- **`std::setprecision(int n)`:**
  - Ajusta a precisão do ponto flutuante para `n` casas decimais.
- **`std::setw(int n)`:**
  - Define a largura do campo para `n`, incluindo espaços em branco à esquerda do valor que será impresso até que o comprimento da string tenha o valor `n`.
- **`std::setfill(char n)`:**
  - Define o caractere de preenchimento como `n`. Os espaços em branco definidos no `setw` serão substituídos pelo caractere `n`.
- **`std::setiosflags ios_base::fmtflags mask)`:**
  - Define a apresentação entre ponto decimal, notação científica, etc.

### Configurando a largura da saída – `std::setw` e `std::setfill`

Para alinhar a saída, especifique a largura de saída para cada item colocando o manipulador `std::setw` no fluxo de saída `cout`.

O exemplo abaixo alinha à direita os valores em uma coluna com pelo menos 10 caracteres de largura:

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main()
6 {
7     double values[] = { 1.23, 35.36, 653.7, 4358.24 };
8     for( int i = 0; i < 4; i++ )
9     {
10         cout << setw(10);
11         cout << values[i] << endl;
12     }
13     return 0;
14 }
15
```

Saída:

```
1.23
35.36
653.7
4358.24
```

Os espaços em branco são adicionados a qualquer valor menor do que 10 caracteres de largura.



Para preencher um campo, use a função `std::setfill`, que define o valor do caractere de preenchimento para campos que têm uma largura especificada. O padrão é um espaço em branco. Para preencher a coluna de números com asteriscos, modifique o loop anterior do `for` seguinte maneira:

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main()
6 {
7     double values[] = { 1.23, 35.36, 653.7, 4358.24 };
8     for (int i = 0; i < 4; i++)
9     {
10         cout << setw(10) << setfill('*');
11         cout << values[i] << endl;
12     }
13     return 0;
14 }
```

Saída:

```
*****1.23
*****35.36
*****653.7
***4358.24
```

Para especificar as larguras de elementos de dados na mesma linha, use o manipulador `setw`:

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main( )
6 {
7     double values[] = { 1.23, 35.36, 653.7, 4358.24 };
8     string names[] = { "Duda", "Joana", "Lia", "Sandra" };
9     for( int i = 0; i < 4; i++ )
10         cout << setw( 7 ) << names[i]
11             << setw( 10 ) << values[i] << endl;
12     return 0;
13 }
```

Na saída, as strings são impressas em um campo de largura 6 e inteiros em um campo de largura 10:

Duda	1.23
Joana	35.36
Lia	653.7
Sandra	4358.24

## Configurando os sinalizadores – `std::setiosflags()`

Ao utilizarmos as formatações de saída, devemos também considerar as possíveis flags (sinalizadores) utilizados pelo `cout`. A exibição de um valor `float`/`double` pode ser feita em formato científico (`5.7e+002`) ou na forma de número real (`574.71`). Este formato é definido pela função `std::setiosflags()`.

Exemplo:



```

1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main() {
6      float valorTotal = 574.718765;
7      // Imprime 574.71
8      cout << setiosflags(ios::fixed);
9      cout << setprecision(2) << valorTotal << "\n\n";
10
11     // Imprime 5.75e+02
12     cout << resetiosflags(ios::fixed);
13     cout << setiosflags(ios::scientific);
14     cout << valorTotal << "\n\n";
15     return 0;
16 }

```

Saída:

574.72

5.75e+02

Como outro exemplo, para alinhar à esquerda os nomes no exemplo anterior e alinhar os números à direita, substitua o `for` loop da seguinte maneira:

```

1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main( )
6  {
7      double values[] = { 1.23, 35.36, 653.7, 4358.24 };
8      string names[] = { "Duda", "Joana", "Lia", "Sandra" };
9      for (int i = 0; i <4; i++)
10         cout << setiosflags(ios::left)
11             << setw(6) << names[i]
12             << resetiosflags(ios::left)
13             << setw(10) << values[i] << endl;
14     return 0;
15 }

```

Saída:

Duda	1.23
Joana	35.36
Lia	653.7
Sandra	4358.24

Abaixo segue tabela com os 16 parâmetros aceitos pelo `std::setiosflags()`:





Última atualização: terça, 4 mai 2021, 14:51



◀ Entrada e Saída de dados

Seguir para...

Comandos de Seleção ▶

©2020 - Universidade Federal do Ceará - Campus Quixadá.

Todos os direitos reservados.

Av. José de Freitas Queiroz, 5003

Cedro - Quixadá - Ceará CEP: 63902-580

Secretaria do Campus: (88) 3411-9422

📱 Obter o aplicativo para dispositivos móveis

