













ESTRUTURA DE DADOS - 03A - 2022.2

Meus cursos <u>ESTRUTURA DE DADOS - 03A - 2022.2</u>

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):
 - o Ajusta a precisão do ponto flutuante para n casas decimais.
- std::setw(int n):

<u>Página inicial</u>

- 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 sáida cout.

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

```
#include <iostr
   #include
   using namespace std;
   int main()
       double values[] = { 1.23, 35.36, 653.7, 4358.24 };
        for( int i = 0; i < 4; i++ )
 9
10
           cout << setw(10);</pre>
11
           cout << values[i] << endl;</pre>
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:

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

Saída:

<u>N</u>

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

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

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

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

Saída:

```
574.72
5.75e+02
```

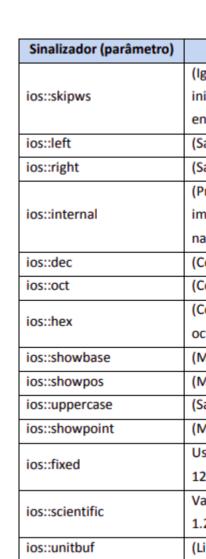
Como outro exemplo, para alinhar à esquerda os nomes no exemplo anterior e alinhar os números à direita, substitua o for loop da seguinte maneira:

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



B

Sinalizador (parâmetro)	Descrição		
	(Ignora o espaço em branco na entrada) caracteres de espaço em branco		
ios::skipws	iniciais (espaços, tabulações, novas linhas) são descartados ao efetuar a		
	entrada		
ios::left	(Saída ajustada à esquerda)		
ios::right	(Saída ajustada à direita)		
ios::internal	(Preenchimento após indicador de sinal ou base) Determina como um valor		
	impresso é posicionado dentro do seu campo (área reservada para o valor		
	na página impressa).		
ios::dec	(Conversão em decimal)		
ios::oct	(Conversão em octal)		
ios::hex	(Conversão em hexadecimal) Determina quando uma notação decimal,		
	octal ou hexadecimal será usada para imprimir e ler os valores.		
ios::showbase	(Mostra indicador de base na saída)		
ios::showpos	(Mostra o sinal "+" em inteiros positivos)		
ios::uppercase	(Saída hexadecimal maiúscula)		
ios::showpoint	(Mostra ponto decimal - saída em ponto flutuante).		
ios::fixed	Usa notação normal para exibir valores numéricos em ponto flutuante -		
	123.45)		
ios::scientific	Valores em ponto flutuante são exibidos usando-se notação científica -		
	1.23E2		
ios::unitbuf	(Libera flush - todas as streams depois da inserção)		
ios::stdio	(Libera flush - stdout, stderr depois da inserção)		
ios::boolalpha	(Valores booleanos podem ser recebidos ou mostrados)		

Configurando a precisão na impressão de números de ponto flutuante - std::setprecision()

O valor padrão para a precisão de ponto flutuante é seis. Por exemplo, o número 3466.9768 será impresso como 3466.98. Para alterar a forma como esse valor é impresso, use o manipulador std::setprecision. O manipulador tem dois sinalizadores: fixed e scientific. Se fixed for definido, o número será impresso como 3466,976800. Se scientific for definido, ele será impresso como 3.4669773 + 003.

Para exibir os números de ponto flutuante mostrados em alinhamento com um dígito significativo, substitua o for loop do exemplo anterior da seguinte maneira:

```
#include <iostream>
    #include <iomanip>
    using namespace std;
    int main( )
 6
        double values[] = { 1.23, 35.36, 653.7, 4358.24 };
        string names[] = { "Duda", "Joana", "Lia", "Sandra" };
 8
        for (int i = 0; i < 4; i++)
 9
             cout << setiosflags(ios::left)</pre>
10
                 << setw(6)
11
                 << names[i]
12
                 << resetiosflags(ios::left)</pre>
13
                 << setw(10)
14
                 << setprecision(1)</pre>
15
                 << values[i]</pre>
16
                 << endl;
17
18
        return 0;
19
```

Saída:

Duda	1
Joana	4e+01
Lia	7e+02
Sandra	4e+03

	Última atualização: terça, 4 mai 2021, 14:51			
	■ Entrada e Saída de dados	Seguir para	Comandos de Seleção ▶	
<u>N</u>				
Û				
(?)				

©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