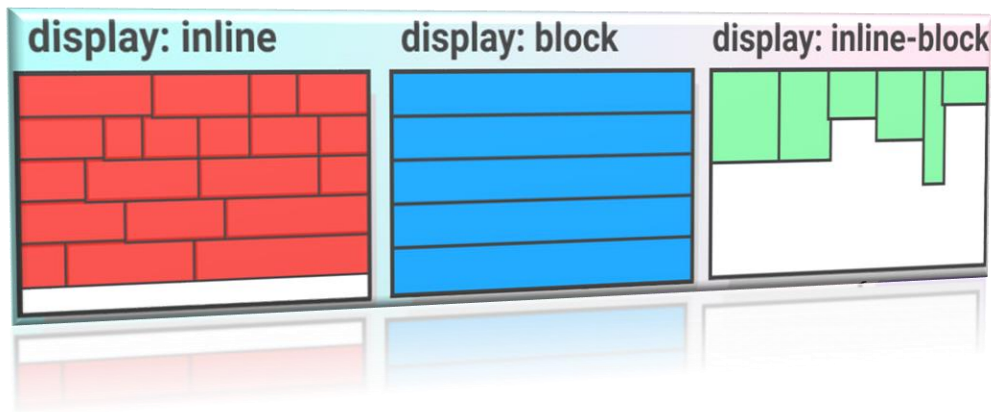


Autor: João vitor Quirino Sarti

GitHub: [NULLBYTE-RGH \(github.com\)](https://github.com/NULLBYTE-RGH)



CSS (Display):Block, Inline e Inline-Block



O que é:

A propriedade display nada mais é o que a forma como um determinado elemento irá se comportar na árvore DOM, nesse caso, o comportamento é como o elemento irá ser renderizado e se comportará em relação aos elementos ao seu redor.

Existem vários tipos de Displays, uns mais flexíveis e outros menos, nesse momento serão abordados 3 tipos:

- Block
- Inline
- Inline-Block

Display: Block

O display Block é o mesmo que foi utilizado até o momento, que é o padrão para a maioria das TAGs HTML (se utilizado o style sheet padrão do navegador) ele funciona com as propriedades mencionadas na lista passada (Box-Model), ou seja, content, padding, margin e border.

Esse tipo de display tem a característica de exibir apenas 1 conteúdo por “Linha”, independente de outro conteúdo caber ou não ao lado.

```
display: block;
```

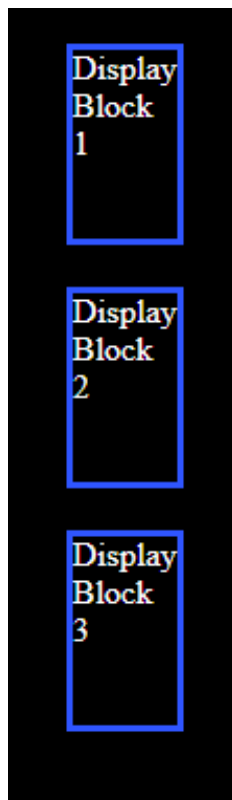
Exemplo:

-Criando Divs com display Block:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Displays</title>
  </head>
  <body style="background-color:black; color:white">
    <div
      style="
        display: block;
        border: 3px solid;
        width: 50px;
        height: 90px;
        margin: 20px;
        border-color: rgb(47, 85, 255);
      "
    >
      Display Block 1
    </div>
    <div
      style="
        display: block;
        border: 3px solid;
        width: 50px;
        height: 90px;
        margin: 20px;
        border-color: rgb(47, 85, 255);
      "
    >
      Display Block 2
    </div>
    <div
      style="
        display: block;
        border: 3px solid;
        width: 50px;
        height: 90px;
        margin: 20px;
        border-color: rgb(47, 85, 255);
      "
    >
      Display Block 3
```

```
</div>  
</body>  
</html>
```

O resultado esperado é:



Esse resultado não é nada mais do que você já sabia e estava aplicando até agora. Cada Box-Model tem seu espaço reservado na linha, independente da largura da caixa. Agora e se você quiser aplicar várias TAGs na mesma linha? Bom, para isso existe o tipo de display (**Inline**).

Display:Inline

Esse display é utilizado quando se quer mostrar várias TAGs em sequência, o que é muito útil quando se quer fazer um menu horizontal por exemplo, porém o display **Inline** apenas de ser um Box-Model, as propriedades altura e largura, não tem efeito sobre o conteúdo dessas caixas, isso porque o intuito é realmente se comportar como se fosse tudo um “texto” contínuo.

```
display: inline;
```

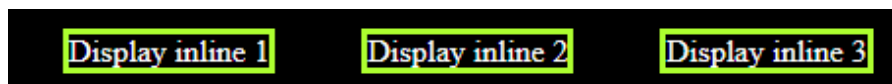
```
width: 50px; →Não tem efeito
```

```
height: 90px; →Não tem efeito
```

Exemplo:

```
<div
  style="
    display: inline;
    border: 3px solid;
    width: 50px;
    height: 90px;
    margin: 20px;
    border-color: greenyellow;
  "
>
  Display inline 1
</div>
<div
  style="
    display: inline;
    border: 3px solid;
    width: 50px;
    height: 90px;
    margin: 20px;
    border-color: greenyellow;
  "
>
  Display inline 2
</div>
<div
  style="
    display: inline;
    border: 3px solid;
    width: 50px;
    height: 90px;
    margin: 20px;
    border-color: greenyellow;
  "
>
  Display inline 3
</div>
```

A saída é:



Agora temos diferentes blocos alinhados atuando como um único bloco!

Display:Inline-Block

O display Inline-Block é similar ao Inline, porém sua diferença fica no quesito de os blocos no Display Inline não atuarem como fazendo parte de um único Box model, e sim como cada bloco tendo seu próprio Box-Model. Em suma, esse display é como se os blocos que antes ficavam cada um em uma linha, agora podem ocupar o mesmo espaço em uma linha.

E pelo fato de cada bloco ser individualizado todas as propriedades do display Block estão valendo, ou seja, largura e altura agora podem ter efeito sobre os elementos.

```
display: inline-block;
```

Exemplo:

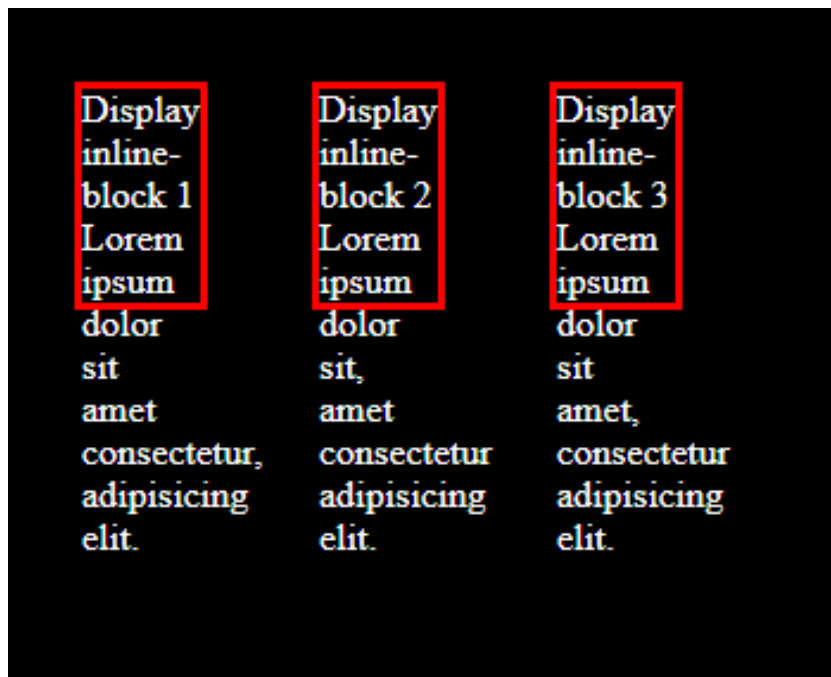
```
<div
  style="
    display: inline-block;
    border: 3px solid;
    width: 50px;
    height: 90px;
    margin: 20px;
    border-color: red;
  "
>
  Display inline-block 1 Lorem ipsum dolor sit amet consectetur,
adipisicing
  elit.
</div>
<div
  style="
    display: inline-block;
    border: 3px solid;
    width: 50px;
    height: 90px;
    margin: 20px;
    border-color: red;
  "
>
  Display inline-block 2 Lorem ipsum dolor sit, amet consectetur
adipisicing
  elit.
</div>
<div
  style="
    display: inline-block;
    border: 3px solid;
    width: 50px;
    height: 90px;
```

```

        margin: 20px;
        border-color: red;
    "
    >
        Display inline-block 3 Lorem ipsum dolor sit amet, consectetur
adipisicing
        elit.
    </div>

```

Resultado:



Vemos que agora nossas caixas estão alinhadas e até mesmo o sentido do overflow do texto, já que limitamos a largura do Box-Model e sua altura. Esse vazamento pode ser resolvido pelo método Overflow, abordado na lista anterior ([Box_Model](#)).

Juntando todo o código temos um exemplo de fácil visualização:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Displays</title>
  </head>
  <body style="background-color: black; color: white">
    <p style="display: flex; border: 4px dotted; justify-content: center">

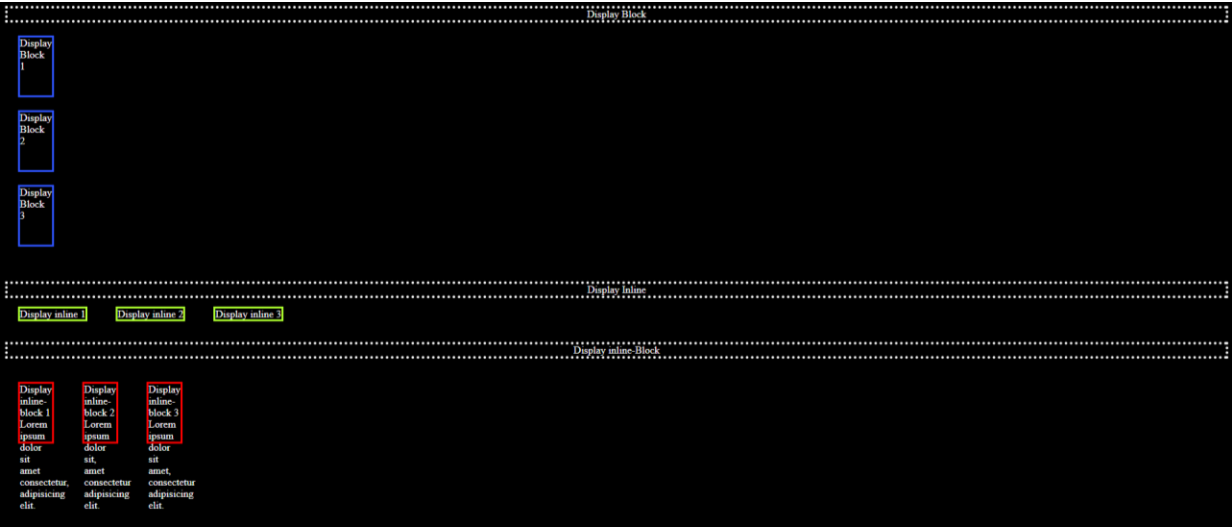
```

```
    Display Block
  </p>
  <div
    style="
      display: block;
      border: 3px solid;
      width: 50px;
      height: 90px;
      margin: 20px;
      border-color: rgb(47, 85, 255);
    "
  >
    Display Block 1
  </div>
  <div
    style="
      display: block;
      border: 3px solid;
      width: 50px;
      height: 90px;
      margin: 20px;
      border-color: rgb(47, 85, 255);
    "
  >
    Display Block 2
  </div>
  <div
    style="
      display: block;
      border: 3px solid;
      width: 50px;
      height: 90px;
      margin: 20px;
      border-color: rgb(47, 85, 255);
    "
  >
    Display Block 3
  </div>
  <br />
  <p style="display: flex; border: 4px dotted; justify-content:
center">
    Display Inline
  </p>
  <div
    style="
      display: inline;
      border: 3px solid;
      width: 50px;
      height: 90px;
```

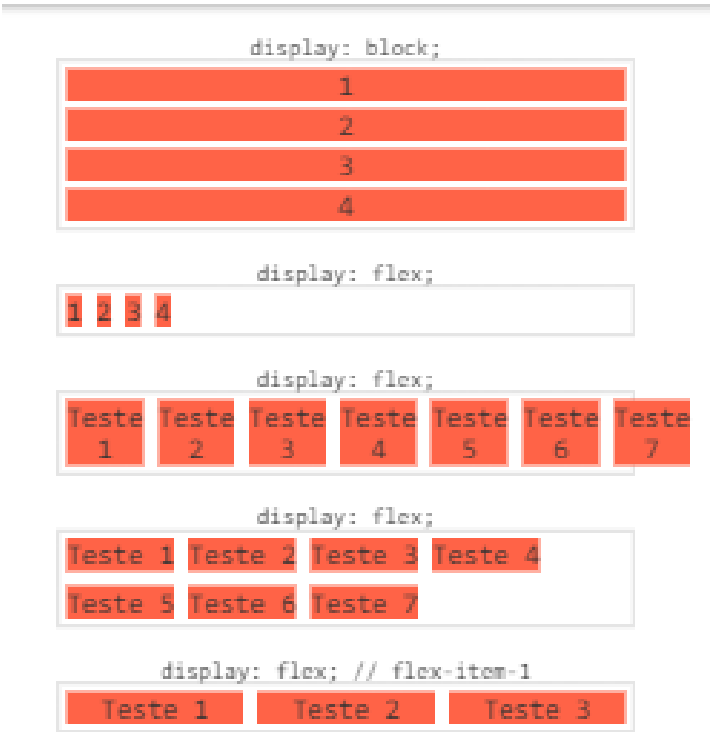


```
</div>
<div
  style="
    display: inline-block;
    border: 3px solid;
    width: 50px;
    height: 90px;
    margin: 20px;
    border-color: red;
  "
>
  Display inline-block 2 Lorem ipsum dolor sit, amet consectetur
adipisicing
  elit.
</div>
<div
  style="
    display: inline-block;
    border: 3px solid;
    width: 50px;
    height: 90px;
    margin: 20px;
    border-color: red;
  "
>
  Display inline-block 3 Lorem ipsum dolor sit amet, consectetur
adipisicing
  elit.
</div>
</body>
</html>
```

Saída:



Nos títulos foi utilizado o Display do Tipo Flex, que não será abordado nesse momento, mas caso queira saber mais: [Guia completo de Flexbox - CSS - display: flex; \(origamid.com\)](https://origamid.com/guia-completo-de-flexbox-css-display-flex/)



Sites recomendados:

[CSS display property \(w3schools.com\)](https://www.w3schools.com/css/css_display_property.asp)

[CSS Height, Width and Max-width \(w3schools.com\)](https://www.w3schools.com/css/css_height_width_max.asp)