



CSS Display: definindo como um elemento HTML será renderizado!

CSS Display: definindo como um elemento HTML será renderizado!

Última atualização 24 de junho de 2021

Dentro do [estilo CSS](#), a propriedade **CSS display** é utilizada para indicar a forma como os elementos [HTML](#) serão dispostos na página ou em espaços delimitados. Por essa característica, é um recurso essencial para manter a organização dos elementos na página.

Outra característica dessa propriedade é a facilidade para criar layouts responsivos, que são facilmente adaptados para diferentes tamanhos de tela. Para demonstrar como esse recurso funciona e o que podemos fazer com ele, preparamos este conteúdo com os seguintes tópicos:

Índice

01 | O que é a propriedade display?

02 | Qual é a sua sintaxe?



CSS Display: definindo como um elemento HTML será renderizado!

Continue com a gente e boa leitura!

O que é a propriedade display?

Para entender o que faz a propriedade **display**, é preciso lembrar que existem dois tipos de [elementos HTML](#). O elemento em bloco ou **block** ocupa 100% da largura disponível e para ele é sempre iniciado em uma nova linha. Um exemplo são os títulos, [divs](#) e parágrafos nos quais, mesmo se os seus conteúdos forem o de apenas uma palavra, toda a linha correspondente deverá ser reservada a cada elemento.

Já os elementos em linha ou *inline* ocupam apenas o espaço necessário para exibir o seu conteúdo. Por exemplo: [imagens](#), links, entre outros. A propriedade display permite modificar a forma como esses elementos são renderizados. Dessa forma, podemos organizá-los de diferentes maneiras na página HTML.

Qual é a sua sintaxe?

A sintaxe da propriedade display é:

```
display: value;
```

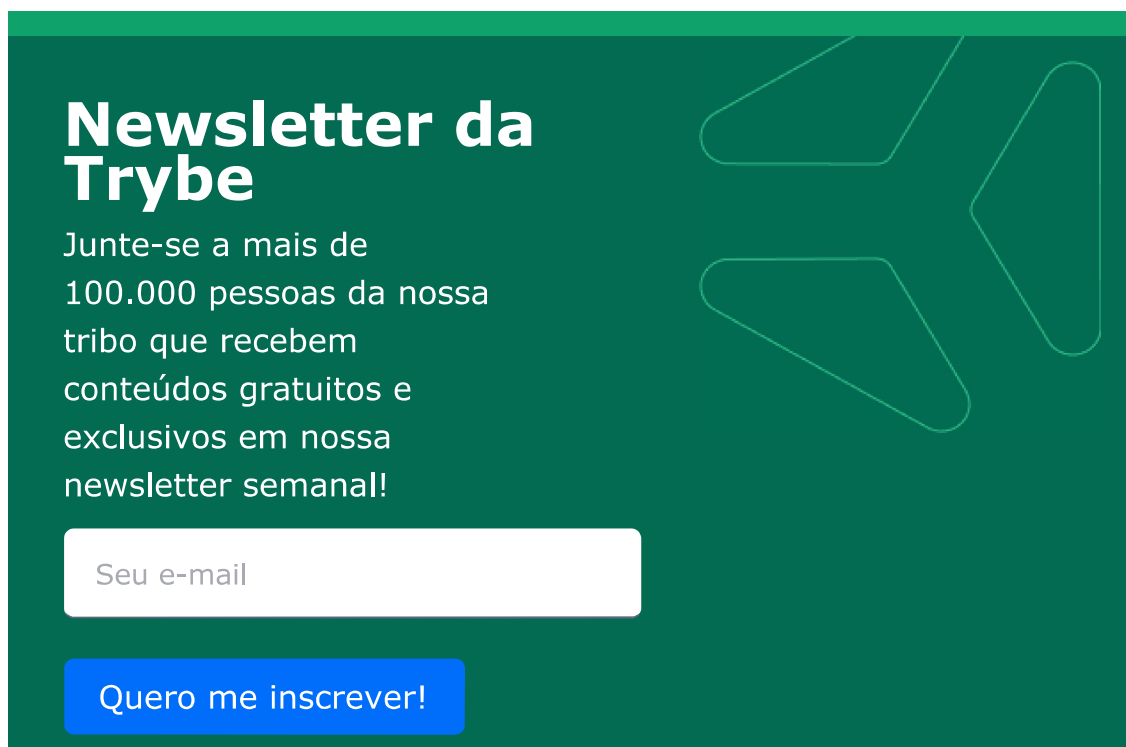
No qual o atributo **value** pode conter os seguintes valores:

- inline;
- block;
- contents;



CSS Display: definindo como um elemento HTML será renderizado!

- inline-table;
- list-item;
- run-in;
- table;
- table-caption;
- table-column-group;
- table-header-group;
- table-footer-group;
- table-row-group;
- table-cell;
- table-column;
- table-row;
- none;
- initial;
- inherit.



Quais os valores da propriedade display?



CSS Display: definindo como um elemento HTML será renderizado!

Renderizando o elemento HTML embutido no bloco: inline

O valor **CSS display inline** é utilizado quando queremos que os elementos fiquem posicionados lado a lado. É importante dizer que, ao utilizar esse valor, as propriedades **width** (largura) e **height** (altura) serão ignoradas se forem definidas no [código](#) CSS.

```
display: inline;
```

Renderizando o elemento HTML como um bloco: block

Quando atribuímos o valor **CSS display block** à propriedade, significa que o elemento será posicionado em uma nova linha.

```
display: block;
```

Definindo propriedades de largura e altura para o elemento HTML: inline-block

Ao utilizarmos o valor **CSS display inline-block** podemos organizar os elementos lado a lado e definirmos as propriedades de largura e altura, pois elas não serão ignoradas pelo navegador.

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



CSS Display: definindo como um elemento HTML será renderizado!

Quando a propriedade **display** é definida como **contents**, significa que as características da caixa delimitadora do espaço do elemento serão ignoradas e ele fará parte do elemento pai. Vale ressaltar que outras propriedades definidas para o elemento continuam valendo, como a cor e o estilo da fonte.

```
display: contents;
```

Renderizando o elemento HTML de forma responsiva e flexível: flex

O valor **CSS display flex** é utilizado para organizar os elementos HTML de [forma responsiva](#). Na prática, ao definirmos um elemento com esse valor, ele funciona como um container para agrupar os elementos filhos, que são considerados **flex-itens** e são organizados nos sentidos horizontal ou vertical.

Ao definirmos o display como flex, podemos utilizar outras propriedades adicionais para estabelecer como os elementos serão dispostos na página ou em um espaço específico. São elas:

- **flex-direction**: determina a direção em que os elementos serão dispostos, que pode ser **row** (linha) ou **column** (coluna), ou ainda, **row-reverse** e **column-reverse** para exibir os itens na posição invertida;
- **flex-wrap**: para definir como o conteúdo de um elemento será exibido dentro do espaço. Em caso de texto, por exemplo, ele pode ser forçado a permanecer na mesma linha ou dividido conforme a necessidade. Os valores possíveis são: **nowrap** para não quebrar a linha, **wrap** para quebrar e **wrap-reverse**, em que o conteúdo é preenchido no sentido de baixo para cima;
- **flex-flow**: representa uma forma abreviada de escrever os valores dos atributos **flex-direction** e **flex-wrap**;



CSS Display: definindo como um elemento HTML será renderizado!

pertencentes ao container. Os valores possíveis são: **flex-start**, **flex-end**, **stretch**, **center** e **baseline**;

- **align-content**: funciona semelhante ao **align-items**, entretanto, alinha o conteúdo verticalmente em relação às linhas.

Renderizando o elemento HTML como um container de grade: grid

O **CSS display grid** permite organizar os elementos da página de forma horizontal e forma vertical, simultaneamente. Para isso, ele também conta com uma série de propriedades adicionais para ajudar a definir como os itens serão dispostos na página ou no espaço disponível.

Basicamente, o **grid** é formado por um conjunto de linhas e colunas. Se nada for especificado, os elementos são organizados em bloco, ou seja, um em cada linha. Por isso, é preciso definir outras propriedades para organizar no formato desejado. São elas:

- **column-gap**: define o espaço entre as colunas;
- **row-gap**: define o espaço entre as linhas;
- **gap**: estabelece o mesmo espaço de distanciamento entre linhas (**row-gap**) e colunas (**column-gap**);
- **grid-area**: determina a área do elemento, ou seja, quantas linhas e colunas o item ocupará no grid;
- **grid-auto-columns**: define o valor automático para a coluna;
- **grid-auto-flow**: indica o sentido em que os itens serão inseridos automaticamente na grade;
- **grid-auto-rows**: define o valor automático para a linha do grid;
- **grid-column-start**: especifica o início do grid;
- **grid-column-end**: especifica o final do grid;
- **grid-column**: define a junção das propriedades **grid-column-start** e **grid-column-end**;
- **grid-row-gap**: indica o tamanho do espaço entre as linhas;



CSS Display: definindo como um elemento HTML será renderizado!

- **grid-row-end**: indica em que linha será o fim do item;
- **grid-row**: representa a união das propriedades **grid-row-start** e **grid-row-end**;
- **grid-template**: representa a união das propriedades **grid-template-rows**, **grid-template-columns** e **grid-areas**;
- **grid-template-areas**: especifica a área do grid com o nome dos itens do grid;
- **grid-template-columns**: estabelece o tamanho das colunas e a quantidade para o layout grid;
- **grid-template-rows**: indica o tamanho das linhas do grid.

Renderizando o elemento HTML como uma tabela: table

O **CSS display table** faz com que o elemento tenha as características de uma tabela. Ela funciona de forma semelhante à [definição de tabelas por código HTML](#), entretanto, a formatação é feita por meio de [propriedades CSS](#).

Portanto, o valor display igual a **table** indica a estrutura da tabela. Será preciso definir outros elementos, de que falaremos a seguir, para construir a formatação da tabela conforme a quantidade de linhas, colunas e demais complementos.

```
display: table;
```

Configurando o elemento título de uma tabela: table caption

O display **table-caption** equivale à [tag HTML <caption>](#) e é utilizado para indicar que o elemento corresponde a um texto descritivo que será inserido na linha superior externa à tabela.



CSS Display: definindo como um elemento HTML será renderizado!

uma tabela: table-column-group

O display igual a **table-column-group** funciona de forma equivalente à tag HTML **<colgroup>** e é utilizado para delimitar um grupo de colunas.

```
display: table-column-group;
```

Configurando um bloco de linhas para o tipo de título das colunas: table-header-group

O display **table-header-group** equivale à tag HTML **<thead>** e indica que o elemento corresponde ao cabeçalho para a tabela ou para um trecho dela.

```
display: table-header-group;
```

Configurando um bloco de linhas para o tipo de rodapés de coluna: table-footer-group

O display **table-footer-group** é utilizado para indicar que o elemento corresponde ao rodapé da tabela.

```
display: table-footer-group;
```




CSS Display: definindo como um elemento HTML será renderizado!

O display **table-row-group** funciona de forma similar à tag HTML **<tbody>** e é utilizado para delimitar grupos de linha na tabela.

```
display: table-row-group;
```

Configurando o preenchimento de espaço em uma célula: table-cell

O display **table-cell** corresponde ao espaço disponível para o conteúdo exibido na tabela e é equivalente às tags HTML **<td>** e **<th>**.

```
display: table-cell;
```

Configurando coluna(s) de uma tabela: table-column

O **table-column** indica quais colunas pertencem a um grupo de colunas. Essa propriedade é utilizada em conjunto com a **table-column-group** para indicar cada coluna especificamente.

```
display: table-column;
```

Configurando linha(s) de uma tabela: table-row

O **table-row** indica que o elemento corresponde a uma linha da tabela. Ele é equivalente à tag HTML **<tr>**.



CSS Display: definindo como um elemento HTML será renderizado!

um item de lista: list-item

O elemento indicado com o display igual a **list-item** é exibido com as características de um item de uma lista. Ele pode ser utilizado em conjunto com as propriedades **list-style-type** e **list-style-position** para definir o estilo do item.

```
display: list-item;
```

Renderizando o elemento HTML como um bloco ou embutido: run-in

O display **run-in** funciona semelhante ao display **block** ou **inline**. Sua característica depende do contexto em que está inserido. Se for definido apenas como **run-in**, ele funciona como o **block**. Já, se for associado ao display **inline**, assume essa característica.

```
display: run-in;
```

Desativando a renderização do elemento HTML no bloco: none

Quando definimos um elemento com **CSS display none**, significa que ele e os seus elementos filhos não são exibidos na tela.

```
display: none;
```



CSS Display: definindo como um elemento HTML será renderizado!

Como mencionamos, por padrão, os elementos podem ser **block** ou **inline**. Ao definirmos o valor de display como **initial**, ele assume o valor original correspondente ao tipo de elemento.

```
display: initial;
```

Herdando a propriedade de seu elemento pai: inherit

Para que um elemento herde as características da propriedade **display** do elemento pai, devemos definir o seu valor como **inherit**.

```
display: inherit;
```

Qual a compatibilidade com os navegadores?

A propriedade **display** contém uma série de valores possíveis. Por isso, é importante avaliar a compatibilidade com os navegadores para evitar que [falhas](#) comprometam a [usabilidade](#) da [aplicação web](#). Confira a partir de qual versão esse recurso é suportado nos principais navegadores:

- Internet Explorer: suporte parcial a partir da versão 10;
- Edge: 12;
- Firefox: suporte parcial a partir da versão 2 e total a partir da 28;
- Chrome: suporte parcial a partir da versão 4 e total a partir da 21;
- Safari: suporte parcial a partir da versão 3.1 e total a partir da 6.1;
- Opera: 12.1;



CSS Display: definindo como um elemento HTML será renderizado!

- Chrome for Android: 89;
- Firefox for Android: 86;
- Samsung Internet: 4.

O **CSS display** é uma propriedade utilizada para organizar os elementos na página HTML. Ela pode conter diferentes valores, entre eles o **flex** e o **grid**, que possibilitam a criação de layouts responsivos. Dessa forma, podemos [construir aplicações](#) que são facilmente adaptadas aos diferentes tamanhos de tela.

Gostou do nosso conteúdo sobre CSS display? Então, confira o nosso post sobre [o que é desenvolvimento web e saiba como ingressar nessa carreira!](#)

Michelle Horn

DEIXE UM COMENTÁRIO

Você precisa fazer o [login](#) para publicar um comentário.

CSS: o que é, guia sobre como usar e vantagens!

CSS Border

CSS Color

CSS Background-image

CSS Display

CSS Gradient Linear e Radial

CSS Hover

CSS Margin

CSS Padding



CSS Display: definindo como um elemento HTML será renderizado!

Currículo

Dúvidas

Trabalhe Conosco

Gerador de CPF

Pague só quando trabalhar

Guia HTML

Guia Javascript

Guia Soft skills

Carreira

Tecnologia

Desenvolvimento Web

Linguagens de Programação

Framework de Programação

TXN

Ferramentas

Desabilitar cookies

Política de Privacidade