

[◀ VOLTAR](#)

Layout responsivo

Neste tópico iremos aprender o que são como trabalhar com layouts responsivos, frameworks, templates e validadores HTML e CSS.

NESTE TÓPICO

- › O layout responsivo
 - › Trabalhando com viewport (visor)
 - › Exemplos de utilização do viewport
- Marcar tópico



O layout responsivo

Os layouts responsivos são úteis pois permitem que sua página em HTML se adapte a todos os dispositivos de visualização tais como desktop, celular ou tablet por exemplo.

Para a criação de layouts responsivos utilizamos apenas HTML e CSS para redimensionar, ocultar, diminuir, ampliar ou mover o conteúdo para que ele pareça bem em qualquer tela (W3SCHOOLS, ONLINE).



Trabalhando com viewport (visor)

O viewport é a área visível para o usuário de uma página na internet e ela varia de acordo com o dispositivo.

Como antes as páginas eram visíveis apenas em monitores não era necessária nenhuma adaptação mas com o surgimento de novas possibilidades de visualização foi necessário criar uma forma de fazer com que o conteúdo diminuísse para se ajustar as telas. (W3SCHOOLS, ONLINE)

Como uma solução rápida à esta necessidade foi criada no HTML5 o valor viewport para a tag meta.

Exemplos de utilização do viewport

- Para adaptar o viewport para um layout móvel definido em 320px.

```
1. <meta name="viewport" content="width=320">
```

- Para layouts flexíveis devemos basear a largura do viewport para o dispositivo em questão ou seja, igualar a largura do layout à largura do dispositivo utilizando:

```
1. <meta name="viewport" content="width=device-width">
```

- Para ter certeza que o layout será mostrado como planejado devemos definir o nível de zoom inicial da seguinte forma:

```
1. <meta name="viewport" content="initial-scale=1">
```

- Se precisar que após a abertura, o layout será exibido em escala 1:1, ou seja, de que não será aplicado nenhum zoom devemos definir a linha de comando abaixo (neste caso tenha certeza de que os usuários conseguirão ler o conteúdo sem precisar de zoom, ok?):

```
1. <meta name="viewport" content="maximum-scale=1">
```

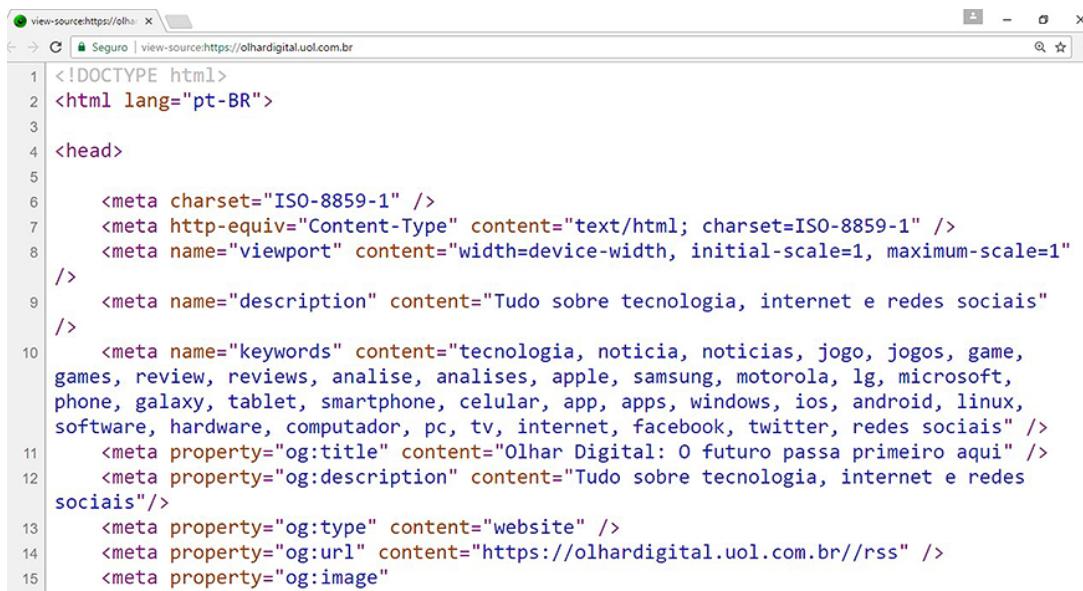
Juntando todos as possibilidades do viewport teríamos a seguinte linha de comando meta que deve ser inserido na tag head da página:

```
1. <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Nesta linha de comando, temos então que:

- O valor viewport define que o conteúdo deve se adaptar a tela;
- O valor width=device-width do atributo content informa que a largura da página deve seguir a largura da tela do dispositivo em que a página for aberta;
- O valor initial-scale=1.0 define o zoom inicial, ou seja, o tamanho que o conteúdo deve ter quando a página for aberta.

Veja um exemplo desta utilização no site do olhar digital disponível em
<https://olhardigital.uol.com.br/>
[\(https://olhardigital.uol.com.br/\)](https://olhardigital.uol.com.br/) (lembre-se de utilizar o atalho Ctrl+u para visualizar o código).



```

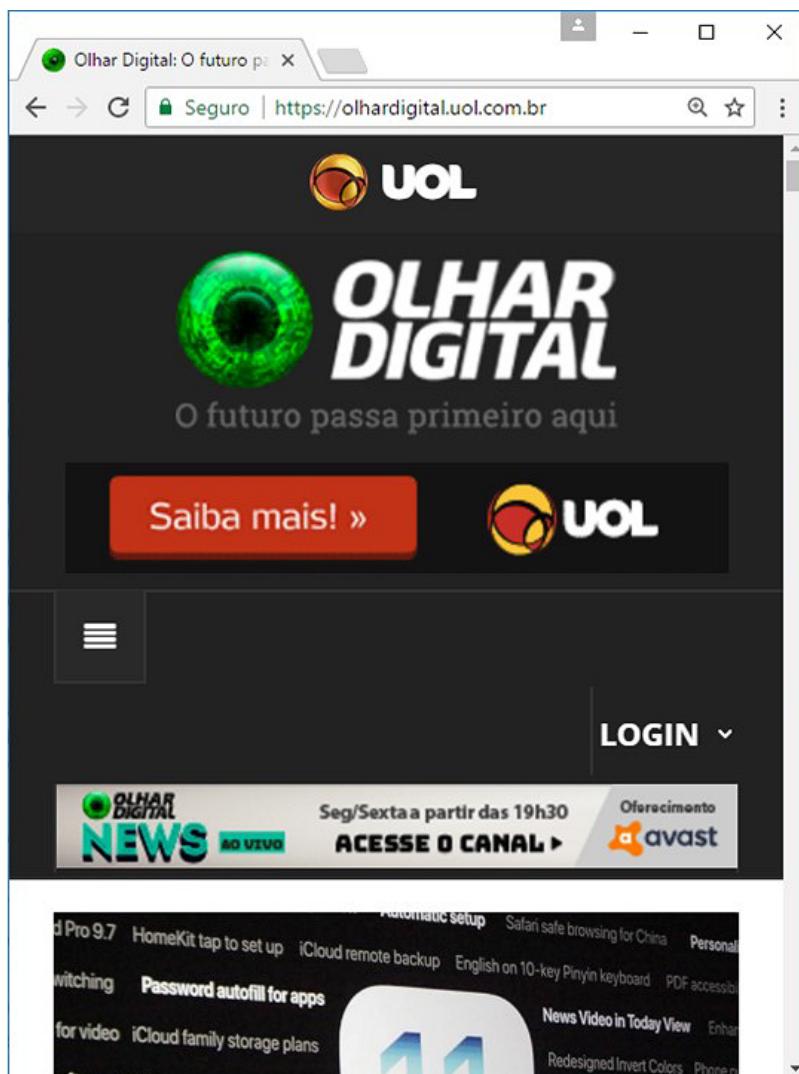
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5
6   <meta charset="ISO-8859-1" />
7   <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
8   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />
9
10  <meta name="description" content="Todo sobre tecnologia, internet e redes sociais" />
11
12  <meta name="keywords" content="tecnologia, noticia, noticias, jogo, jogos, game, games, review, reviews, analise, analises, apple, samsung, motorola, lg, microsoft, phone, galaxy, tablet, smartphone, celular, app, apps, windows, ios, android, linux, software, hardware, computador, pc, tv, internet, facebook, twitter, redes sociais" />
13  <meta property="og:title" content="Olhar Digital: O futuro passa primeiro aqui" />
14  <meta property="og:description" content="Todo sobre tecnologia, internet e redes sociais" />
15  <meta property="og:type" content="website" />
16  <meta property="og:url" content="https://olhardigital.uol.com.br//rss" />
17  <meta property="og:image" />

```

A visualização do site no navegador é:



O site em uma janela redimensionada fica da seguinte forma:



DICA LEGAL:

Veja uma lista de tamanhos comuns da viewport para dispositivos móveis e tablets [clicando aqui.](http://viewportsizes.mattstow.com/) (<http://viewportsizes.mattstow.com/>)

Viewport: Trabalhando bem os conteúdos

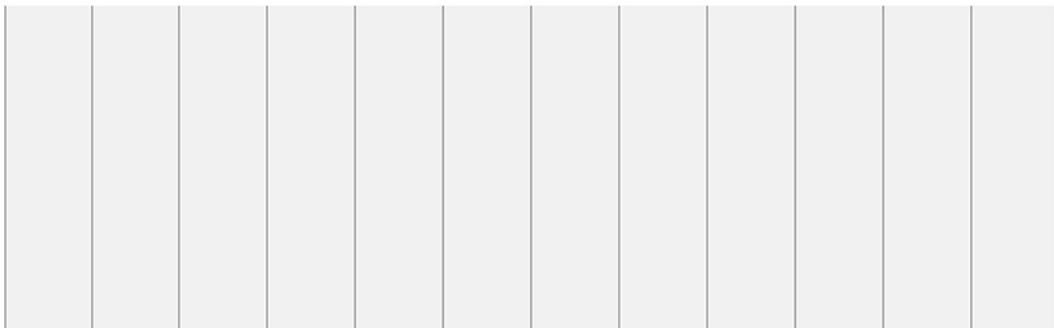
Segundo do site do W3schools (ONLINE), considerando que os usuários tem o hábito de utilizar os seus dispositivos móveis na vertical, temos de tomar alguns cuidados quando da inserção em sites que utilizam o viewport como, por exemplo:

- Não insira figuras que tenham uma largura fixa para que, quando o site for adaptado, a barra de rolagem da horizontal não apareça;
- Não deixe que o conteúdo dependa de uma largura fixa para ser bem renderizado corretamente;
- Utilize tamanhos relativos (em exemplo seria uma largura em %).



Trabalhando com Grid-View

Muitas páginas na internet são construídas com uma exibição em grade ou seja, a página é dividida em colunas elaboradas com tags div, conforme a imagem abaixo:

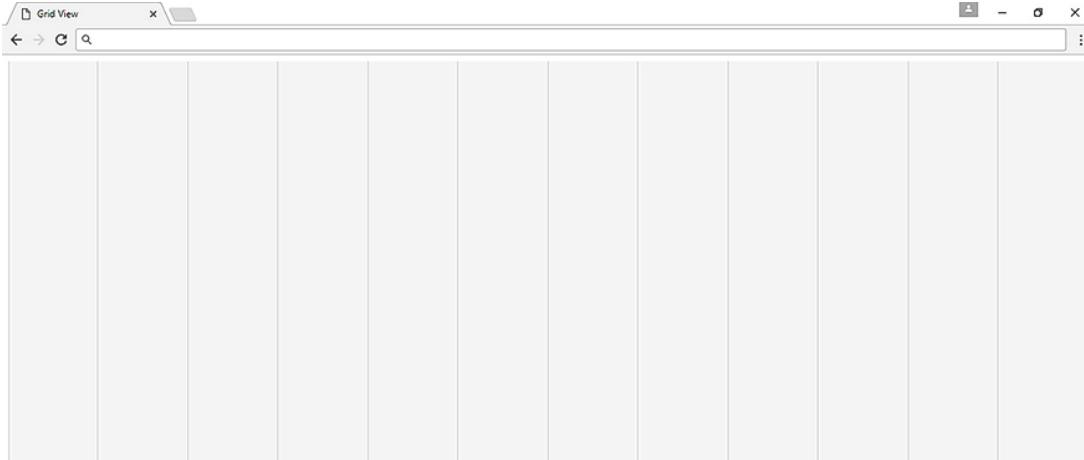


Um exemplo de código para a construção destes grids (já responsivos) seria:

```
1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.  <head>
4.      <meta charset="utf-8"/>
5.      <title>Grid View</title>
6.      <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
7.      <style>
8.          * {
9.              box-sizing:border-box;
10.         }
11.        .grid1 {
12.            overflow:auto;
13.            position:relative;
14.        }
15.        .grid2 {
16.            width:8.33%;
17.            margin:0;
18.            border-right:1px solid grey;
19.            height:100%;
20.            float:left;
21.        }
22.    </style>
23.  </head>
24.  <body>
25.
26.
27.  <div class="grid0" style="opacity:0.2;">
28.  <div class="grid1" style="height:500px;">
29.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000; border-left:1px solid #000000;"></div>
30.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
31.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
32.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
33.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
34.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
35.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
36.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
37.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
38.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
39.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
40.      <div class="grid2" style="background-color:#ccc; border-right:1px solid #000000;"></div>
41.  </div>
42.  </div>
43.
44.  </body>
45. </html>
```



Este exemplo visualizado no navegador ficaria da seguinte forma:



VAMOS FAZER UM EXERCÍCIO?

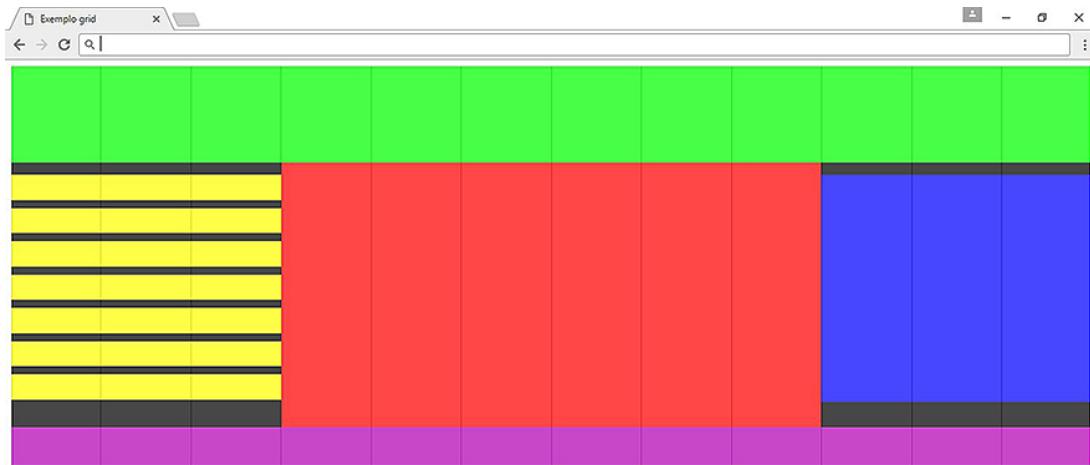
Faça um exercício de fixação seguindo os seguintes passos:

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Salve o arquivo com a extensão .html;
5. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

Para ver o layout responsivo em ação, altere o tamanho da janela para ver as colunas se adaptando.

Com o desenho de nosso grid fica mais fácil desenhar nossa página com as próximas divs que acomodarão o conteúdo. Um exemplo deste planejamento pode ser visto na figura abaixo:



Redimensione a janela para ver o resultado

O código para a criação desta página seria:

```
1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.  <head>
4.  <title>Exemplo grid</title>
5.  <meta charset="utf-8"/>
6.  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
7.  <style>
8.  * {
9.    box-sizing:border-box;
10. }
11. .organiza {
12.   overflow:auto;
13.   position:relative;
14. }
15. .grid {
16.   width:8.33%;
17.   margin:0;
18.   border-right:1px solid #666;
19.   height:100%;
20.   float:left;
21. }
22. </style>
23. </head>
24. <body>
25.
26. <div style="position:absolute;opacity:0.9;width:auto;left:8px;right:10px;"><!--inici
o da div principal que contém o layout -->
27.   <div class="layout"><!--inicio do layout -->
28.
29.     <div class="organiza" style="height:120px;"><!-- inicio do topo -->
30.       <div class="grid" style="width:100%;background:#0F0;border-right-color:t
ransparent;"></div><!-- fim do topo -->
31.
32.
33.     <div class="organiza" style="height:330px;"><!-- inicio do meio -->
34.       <div class="grid" style="background-color:#000;border:none;width:25%;pad
ding-top:15px;"><!-- inicio da div preta embaixo das divs amarelas da esquerda - men
u esquerda -->
35.         <div style="background-color:#FF0;border:none;width:100%;height:10%
margin-bottom:10px;"></div><!--div amarela-->
36.         <div style="background-color:#FF0;border:none;width:100%;height:10%
margin-bottom:10px;"></div><!--div amarela-->
37.         <div style="background-color:#FF0;border:none;width:100%;height:10%
margin-bottom:10px;"></div><!--div amarela-->
38.         <div style="background-color:#FF0;border:none;width:100%;height:10%
margin-bottom:10px;"></div><!--div amarela-->
39.         <div style="background-color:#FF0;border:none;width:100%;height:10%
margin-bottom:10px;"></div><!--div amarela-->
40.         <div style="background-color:#FF0;border:none;width:100%;height:10%
margin-bottom:10px;"></div><!--div amarela-->
41.         <div style="background-color:#FF0;border:none;width:100%;height:10%
"></div><!--div amarela-->
42.         </div><!-- fim inicio da div preta embaixo das divs amarelas da esquerd
a - menu esquerda-->
43.         <div class="grid" style="background-color:#F00;border:none;"></div><!--
div vermelha do meio-->
44.         <div class="grid" style="background-color:#F00;border:none;"></div><!--
div vermelha do meio-->
45.         <div class="grid" style="background-color:#F00;border:none;"></div><!--
div vermelha do meio-->
46.         <div class="grid" style="background-color:#F00;border:none;"></div><!--
div vermelha do meio-->
47.         <div class="grid" style="background-color:#F00;border:none;"></div><!--
div vermelha do meio-->
48.         <div class="grid" style="background-color:#F00;border:none;"></div><!--
div vermelha do meio-->
49.         <div class="grid" style="background-color:#000;border:none;width:25%;pad
ding-top:15px;"><!--inicio da div preta embaixo da div da direita azul-->
50.           <div style="background-color:#00F;border:none;width:100%;height:90%
"></div><!--div da direita azul-->
51.           </div><!-- fim da div preta embaixo da div da direita azul-->
```



```
52.         </div><!-- fim do meio -->
53.
54.         <div class="organiza" style="height:50px;"><!-- inicio do rodapé -->
55.             <div class="grid" style="width:100%;background:#C0C;border-right-color:transparent;"></div><!--div roxa-->
56.         </div><!-- fim do rodapé -->
57.
58.         </div><!--fim do layout -->
59.     </div><!--fim da div principal que contém o layout -->
60.
61.     <div class="grid_principal" style="opacity:0.1;"><!--inicio 12 grids principais -->
62.         <div class="organiza" style="height:500px;">
63.             <div class="grid" style="background-color:#fff;border-right:1px solid #000; border-left:1px solid #000;"></div>
64.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
65.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
66.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
67.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
68.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
69.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
70.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
71.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
72.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
73.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
74.             <div class="grid" style="background-color:#fff;border-right:1px solid #000;"></div>
75.         </div>
76.     </div><!--fim 12 grids principais -->
77.
78.     <p>Redimensione a janela para ver o resultado</p>
79.
80. </body>
81. </html>
```



VAMOS FAZER UM EXERCÍCIO?

Faça um exercício de fixação seguindo os seguintes passos:

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Salve o arquivo com a extensão .html;
5. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

Para ver o layout responsivo em ação, altere o tamanho da janela para ver o layout se adaptando.

VAMOS TREINAR?

Treine a utilização de um layout responsivo [clicando aqui.](https://www.w3schools.com/css/trystyle_grid.htm)
(https://www.w3schools.com/css/trystyle_grid.htm)



Criando o CSS de um grid-view

Vamos entender como fazer com que nosso layout funcione de forma responsiva à partir do código exemplificado acima.

Além da linha de comando do viewport que já conhecemos, começaremos com a linha de comando do CSS abaixo:

```
1. * {  
2.     box-sizing:border-box;  
3. }
```

A propriedade box-sizing

Esta linha de comando no CSS avisa ao navegador que os elementos que possuírem valores de padding e de border não devem ter estes valores adicionados em sua largura.

Quando adicionamos padding e border a um elemento, o que ocorre é que o tamanho deste elemento se altera, pois o navegador irá "somar" os valores de padding e border ao tamanho do elemento, então se este elemento tiver 200px de largura, 10px de padding e 1px de borda, apareceria na tela com uma largura de 222px. Com a aplicação desta linha de comando no CSS, todos os elementos manterão apenas a largura original, sem ser alterados pela inclusão dos valores de padding e border.

Se tivéssemos o código abaixo, por exemplo:

```
1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.  <head>
4.  <title>Exemplo box-sizing</title>
5.  <meta charset="utf-8"/>
6.  <style>
7.  .teste1 {
8.      width: 300px;
9.      height: 100px;
10.     border: 1px solid blue;
11.    box-sizing: border-box;
12. }
13.
14. .teste2 {
15.     width: 300px;
16.     height: 100px;
17.     padding: 50px;
18.     border: 1px solid red;
19.     box-sizing: border-box;
20. }
21. </style>
22. </head>
23. <body>
24.
25.     <div class="teste1">Eu tenho um tamanho</div>
26.     <br>
27.     <div class="teste2">Apesar de ter padding e border eu tenho o mesmo tamanho da di
v teste 1</div>
28.
29. </body>
30. </html>
```



Ambas as divs mantém o mesmo tamanho, já que estão com a propriedade box-sizing: border-box, com o código acima visualizado no navegador:



VAMOS TREINAR?

Treine a utilização do box-sizing clicando aqui (https://www.w3schools.com/css/tryit.asp?filename=trycss3_box-sizing_new).

IMPORTANTE

Quando utilizamos o * como seletor no CSS., estamos avisando ao navegador que a propriedade deve ser aplicada a todos os elementos, ok?

Treine sua utilização clicando [aqui.](https://www.w3schools.com/cssref/tryit.asp?filename=trycss_sel_all)
(https://www.w3schools.com/cssref/tryit.asp?filename=trycss_sel_all)



Utilizando o que já aprendemos e a propriedade box-sizing vamos construir uma página HTML com um layout simples como o apresentado no código abaixo:

```
1.  <!DOCTYPE HTML>
2.  <html lang="pt-br">
3.  <head>
4.  <title>layout simples</title>
5.  <meta charset="utf-8"/>
6.  <meta name="viewport" content="width=device-width, initial-scale=1.0">
7.  <style>
8.  * {
9.    box-sizing: border-box;
10. }
11.
12. .topo {
13.   border: 1px solid #F00;
14.   background: #CCC;
15.   padding: 5px;
16.   height:50px;
17. }
18. .menu_esquerda {
19.   width: 15%;
20.   height:400px;
21.   float: left;
22.   padding: 5px;
23.   border: 1px solid #00F;
24. }
25. .meio{
26.   width: 70%;
27.   height:400px;
28.   float: left;
29.   padding: 5px;
30.   border: 1px solid #0F0;
31. }
32. .menu_direita {
33.   width: 15%;
34.   height:400px;
35.   float: right;
36.   padding: 5px;
37.   border: 1px solid #00F;
38.
39. }
40. .rodape {
41.   height:50px;
42.   clear: both;
43.   border: 1px solid #F00;
44.   background: #FF0;
45.   padding: 5px;
46.
47. }
48. </style>
49. </head>
50. <body>
51.
52.   <div class="topo"> <p>topo</p> </div>
53.   <div class="menu_esquerda"> <p>menu esquerda</p> </div>
54.   <div class="meio"> <p>meio</p> </div>
55.   <div class="menu_direita"> <p>menu direita</p> </div>
56.   <div class="rodape"> <p>rodapé</p> </div>
57.
58. </body>
59. </html>
```



No navegador este código ficaria da seguinte maneira:



VAMOS FAZER UM EXERCÍCIO?

Faça um exercício de fixação seguindo os seguintes passos:

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Salve o arquivo com a extensão .html;
5. Altere o código para treinar e entender sua execução;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

Para ver o layout responsivo em ação, altere o tamanho da janela para ver o layout se adaptando.

VAMOS TREINAR?

Treine a configuração de um layout simples responsivo [clicando aqui](https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_webpage) (https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_webpage).

Construindo o layout

No exemplo acima temos poucas colunas em nossa página, mas no exemplo completo temos 12 grids perfeitamente distribuídos. Para criar estes grids utilizamos o seguinte código em nosso CSS:

```

1. .grid {
2.   width:8.33%;
3.   margin:0;
4.   border-right:1px solid #666;
5.   height:100%;
6.   float:left;
7. }

```

Observe que para que seja possível seu ajuste, determinamos na classe .grid do CSS para cada grid uma largura de 8.33% ou seja, 100% dividido por 12, que é a quantidade de grids que queremos ter em nosso layout.

Esta classe também informa que cada grid terá 100% de altura e uma borda de 1px sólida e cinza.

Para sua aplicação em nossa página utilizamos, no corpo (body) o seguinte código HTML:

```

1. <div class="grid_principal" style="opacity:0.1;"><!--inicio 12 grids principais -->
2.   <div class="organiza" style="height:500px;">
3.     <div class="grid" style="background-color:#fff; border-right:1px solid #000; border-left:1px solid #000;"></div>
4.     <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
5.     <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
6.     <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
7.     <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
8.     <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
9.     <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
10.    <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
11.    <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
12.    <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
13.    <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
14.    <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
15.  </div><!--fim 12 grids principais -->
16. </div>

```

Observe que no código acima a div class=organiza (que contém nossos 12 grids) tem uma altura de 500px, que a altura total de nosso layout e que será dividida da seguinte forma:

- 120px para a div class=organiza do topo;
- 330px para a div class=organiza do meio;
- 50px para a div class=organiza do rodape;

O CSS desta div organiza é:



```
1. .organiza {  
2.     overflow:auto;  
3.     position:relative;  
4. }
```

Neste css temos que:

Propriedade overflow:auto - informa ao navegador que, se necessário, devem aparecer barras de rolagem quando o conteúdo "transbordar";

position:relative - informa ao navegador que as divs que estiverem posicionadas dentro dela devem ter suas posições relativas a esta div e não ao view (corpo da página);

Após a configuração do CSS e das grids no corpo, montamos então as "partes" de nosso layout distribuídas em:

1. Divs que estruturam e posicionam nosso layout:

```
1. <div style="position:absolute;opacity:0.9;width:auto;left:8px;right:10px;"><!--inici  
o da div principal que contém o layout -->  
2.     <div class="layout"><!--inicio do layout -->  
3.  
4.         O layout vai aqui...  
5.  
6.     </div><!--fim do layout -->  
7. </div><!--fim da div principal que contém o layout -->
```

2. Inclusão do topo do site com o código:

```
1. <div class="organiza" style="height:120px;"><!-- inicio do topo -->  
2.     <div class="grid" style="width:100%;background:#0F0;border-right-color:t  
ransparent;"></div>  
3.     </div><!-- fim do topo -->
```

3. Inclusão da parte do meio do site com o código (dividido entre parte da esquerda, do meio e da direita do site):



```

1. <div class="organiza" style="height:330px;"><!-- inicio do meio -->
2.         <div class="grid" style="background-color:#000; border:none; width:25%; padding-top:15px;"><!-- inicio da div preta embaixo das divs amarelas da esquerda - menu esquerda -->
3.             <div style="background-color:#FF0; border:none; width:100%; height:10%; margin-bottom:10px;"></div><!--div amarela-->
4.                 <div style="background-color:#FF0; border:none; width:100%; height:10%; margin-bottom:10px;"></div><!--div amarela-->
5.                     <div style="background-color:#FF0; border:none; width:100%; height:10%; margin-bottom:10px;"></div><!--div amarela-->
6.                         <div style="background-color:#FF0; border:none; width:100%; height:10%; margin-bottom:10px;"></div><!--div amarela-->
7.                             <div style="background-color:#FF0; border:none; width:100%; height:10%; margin-bottom:10px;"></div><!--div amarela-->
8.                                 <div style="background-color:#FF0; border:none; width:100%; height:10%; margin-bottom:10px;"></div><!--div amarela-->
9.                                     <div style="background-color:#FF0; border:none; width:100%; height:10%; margin-bottom:10px;"></div><!--div amarela-->
10.                                         <div style="background-color:#FF0; border:none; width:100%; height:10%; margin-bottom:10px;"></div><!-- fim inicio da div preta embaixo das divs amarelas da esquerda - menu esquerda-->
11.                                         <div class="grid" style="background-color:#F00; border:none;"></div><!--div vermelha do meio-->
12.                                         <div class="grid" style="background-color:#F00; border:none;"></div><!--div vermelha do meio-->
13.                                         <div class="grid" style="background-color:#F00; border:none;"></div><!--div vermelha do meio-->
14.                                         <div class="grid" style="background-color:#F00; border:none;"></div><!--div vermelha do meio-->
15.                                         <div class="grid" style="background-color:#F00; border:none;"></div><!--div vermelha do meio-->
16.                                         <div class="grid" style="background-color:#F00; border:none;"></div><!--div vermelha do meio-->
17.                                         <div class="grid" style="background-color:#000; border:none; width:25%; padding-top:15px;"><!--inicio da div preta embaixo da div da direita azul-->
18.                                             <div style="background-color:#00F; border:none; width:100%; height:90%;"></div><!--div da direita azul-->
19.                                         </div><!-- fim da div preta embaixo da div da direita azul-->
20. </div><!-- fim do meio -->
```



4. Inclusão do rodapé do site com o código:

```

1. <div class="organiza" style="height:50px;"><!-- inicio do rodapé -->
2.         <div class="grid" style="width:100%; background:#C0C; border-right-color:transparent;"></div><!--div roxa-->
3. </div><!-- fim do rodapé -->
```

Trabalhando com imagens:

Se uma imagem estiver com sua largura definida em porcentagem ela diminuirá e aumentará de acordo com o redimensionamento da janela como, por exemplo:

```

1. img {
2.     width: 100%;
3.     height: auto;
4. }
```

Ou seja, ao ser inserida em uma página ela se redimensionará de acordo com o tamanho da página.

VAMOS TREINAR?

Treine o redimensionamento de uma imagem com width=100% [clicando aqui.](#)
(https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_image2)



Mas você pode limitar este redimensionamento até apenas 100% do tamanho da imagem utilizando o comando max-width em CSS:

```
1. img {  
2.     max-width: 100%;  
3.     height: auto;  
4. }
```

VAMOS TREINAR?

Treine o redimensionamento de uma imagem com max-width: [clicando aqui.](#)
(https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_image)

Trabalhando com vídeos

O elemento video também podem ser redimensionado utilizando o width=100% ou o max-width. O funcionamento será o mesmo da imagem.

Configurando uma alteração se uma determinada condição for verdadeira

Para configurar uma alteração se uma determinada condição for verdadeira utilizamos a regra a @media.

Por exemplo, para mudar a cor de fundo se nosso layout atingir a largura de 700px utilizariamos:

VAMOS TREINAR?

Treine o redimensionamento de um vídeo com width: [clicando aqui.](#)
(https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_video3)

Treine o redimensionamento de um vídeo com max-width: [clicando aqui.](#)
(https://www.w3schools.com/css/tryit.asp?filename=tryresponsive_video)

```
1. @media only screen and (max-width: 700px) {  
2.     body {  
3.         background-color: #000;  
4.     }  
5. }
```

Nosso código com a inserção desta regra ficaria:



```
1. <!DOCTYPE HTML>
2. <html lang="pt-br">
3. <head>
4. <title>Exemplo grid</title>
5. <meta charset="utf-8"/>
6. <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
7. <style>
8. * {
9.     box-sizing:border-box;
10. }
11. .organiza {
12.     overflow:auto;
13.     position:relative;
14. }
15. .grid {
16.     width:8.33%;
17.     margin:0;
18.     border-right:1px solid #666;
19.     height:100%;
20.     float:left;
21. }
22.
23. @media only screen and (max-width: 700px) {
24.     body {
25.         background-color: #000;
26.     }
27.
28. }
29. </style>
30. </head>
31. <body>
32.
33. <div style="position:absolute;opacity:0.9;width:auto;left:8px;right:10px;"><!-- inicia o da div principal que contém o layout -->
34.     <div class="layout"><!-- inicio do layout -->
35.
36.         <div class="organiza" style="height:120px;"><!-- inicio do topo -->
37.             <div class="grid" style="width:100%;background:#0F0;border-right-color:transparent;"></div>
38.         </div><!-- fim do topo -->
39.
40.         <div class="organiza" style="height:330px;"><!-- inicio do meio -->
41.             <div class="grid" style="background-color:#000;border:none;width:25%;padding-top:15px;"><!-- inicio da div preta embaixo das divs amarelas da esquerda - menu esquerda -->
42.                 <div style="background-color:#FF0;border:none;width:100%;height:10%;margin-bottom:10px;"></div><!-- div amarela-->
43.                 <div style="background-color:#FF0;border:none;width:100%;height:10%;margin-bottom:10px;"></div><!-- div amarela-->
44.                 <div style="background-color:#FF0;border:none;width:100%;height:10%;margin-bottom:10px;"></div><!-- div amarela-->
45.                 <div style="background-color:#FF0;border:none;width:100%;height:10%;margin-bottom:10px;"></div><!-- div amarela-->
46.                 <div style="background-color:#FF0;border:none;width:100%;height:10%;margin-bottom:10px;"></div><!-- div amarela-->
47.                 <div style="background-color:#FF0;border:none;width:100%;height:10%;margin-bottom:10px;"></div><!-- div amarela-->
48.                 <div style="background-color:#FF0;border:none;width:100%;height:10%;"></div><!-- div amarela-->
49.             </div><!-- fim inicio da div preta embaixo das divs amarelas da esquerda - menu esquerda-->
50.             <div class="grid" style="background-color:#F00;border:none;"></div><!-- div vermelha do meio-->
51.                 <div class="grid" style="background-color:#F00;border:none;"></div><!-- div vermelha do meio-->
52.                 <div class="grid" style="background-color:#F00;border:none;"></div><!-- div vermelha do meio-->
53.                 <div class="grid" style="background-color:#F00;border:none;"></div><!-- div vermelha do meio-->
54.                 <div class="grid" style="background-color:#F00;border:none;"></div><!-- div vermelha do meio-->
```



```

55.         <div class="grid" style="background-color:#F00; border:none;"></div><!--
      div vermelha do meio-->
56.         <div class="grid" style="background-color:#000; border:none; width:25%; padding-top:15px;"><!-- inicio da div preta embaixo da div da direita azul-->
57.             <div style="background-color:#00F; border:none; width:100%; height:90%;"></div><!--div da direita azul-->
58.             </div><!-- fim da div preta embaixo da div da direita azul-->
59.         </div><!-- fim do meio -->
60.
61.         <div class="organiza" style="height:50px;"><!-- inicio do rodapé -->
62.             <div class="grid" style="width:100%; background:#C0C; border-right-color:transparent;"></div><!--div roxa-->
63.             </div><!-- fim do rodapé -->
64.
65.         </div><!--fim do layout -->
66.     </div><!--fim da div principal que contém o layout -->
67.
68. <div class="grid_principal" style="opacity:0.1;"><!--inicio 12 grids principais -->
69.     <div class="organiza" style="height:500px;">
70.         <div class="grid" style="background-color:#fff; border-right:1px solid #000; border-left:1px solid #000;"></div>
71.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
72.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
73.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
74.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
75.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
76.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
77.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
78.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
79.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
80.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
81.         <div class="grid" style="background-color:#fff; border-right:1px solid #000;"></div>
82.     </div>
83. </div><!--fim 12 grids principais -->
84.
85. <p>Redimensione a janela para ver o resultado</p>
86.
87. </body>
88. </html>

```



VAMOS FAZER UM EXERCÍCIO?

Faça um exercício de fixação seguindo os seguintes passos:

1. Abra o editor Brackets;
2. Crie um novo arquivo no Brackets;
3. Copie e cole o código acima no arquivo do Brackets;
4. Salve o arquivo com a extensão .html;
5. Altere o código para treinar e entender sua execução;
6. Teste no navegador (preview) para ver o resultado.

Você também pode realizar este exercício utilizando o editor online codepen [clicando aqui](https://codepen.io/pen/?editors=1100) (<https://codepen.io/pen/?editors=1100>).

Para ver o layout responsivo em ação, altere o tamanho da janela para ver o layout se adaptando.

Os frameworks (bibliotecas)



Há na internet diversos frameworks disponíveis e grátis que oferecem ótimas ferramentas de desenvolvimento. Sua utilização é de grande ajuda no desenvolvimento de sistemas pois, como vimos, criar layouts responsivos nem sempre é uma tarefa fácil.

Abaixo seguem alguns frameworks para servir de apoio e de inspiração no desenvolvimento de seus layouts responsivos:

- <http://getbootstrap.com.br/v4/> (<http://getbootstrap.com.br/v4/>)
- <https://nt1m.github.io/material-framework/#introduction> (<https://nt1m.github.io/material-framework/#introduction>)
- <http://getleaf.com/> (<http://getleaf.com/>)
- <http://materializecss.com/> (<http://materializecss.com/>)
- <https://getbootstrap.com/> (<https://getbootstrap.com/>)
- <https://semantic-ui.com/> (<https://semantic-ui.com/>)
- <http://foundation.zurb.com/> (<http://foundation.zurb.com/>)
- <http://www.cascade-framework.com/> (<http://www.cascade-framework.com/>)
- <http://basegui.de/> (<http://basegui.de/>)
- <https://siimple.juanes.xyz/> (<https://siimple.juanes.xyz/>)
- <http://concisecss.com/> (<http://concisecss.com/>)

Pesquise mais possibilidades na internet e mãos à obra!

Templates

Uma ótima forma de trabalhar com layouts responsivos são as bibliotecas de templates prontos e totalmente free. Há muitas disponíveis na internet que permitem sua utilização.

Uma sugestão para iniciar esta pesquisa e utilização é aproveitar os templates disponibilizados no site do W3schools disponíveis [clicando aqui](https://www.w3schools.com/w3css/w3css_templates.asp). (https://www.w3schools.com/w3css/w3css_templates.asp)

Nesta biblioteca do W3schools você pode editar diretamente o código e ver o resultado como neste exemplo. (https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_templates_band&stacked=h)

Você também pode utilizar os layouts disponíveis em sites na internet como os descritos abaixo:

- <http://www.free-css.com/template-categories/responsive> (<http://www.free-css.com/template-categories/responsive>)
- <https://templated.co/> (<https://templated.co/>)
- <https://dcrazed.com/free-responsive-html5-css3-templates/> (<https://dcrazed.com/free-responsive-html5-css3-templates/>)
- <https://w3layouts.com/free-responsive-html5-css3-website-templates/> (<https://w3layouts.com/free-responsive-html5-css3-website-templates/>)
- <http://www.os-templates.com/free-website-templates> (<http://www.os-templates.com/free-website-templates>)



Pesquise mais possibilidades na internet e mãos à obra!

Validadores HTML

Na internet há ferramentas de validação de seu código HTML e CSS que podem ser utilizadas de forma gratuita.

As principais são:

- Validador de HTML do W3C - <https://validator.w3.org/> (<https://validator.w3.org/>)
- Validador de CSS do W3C - <http://jigsaw.w3.org/css-validator/> (<http://jigsaw.w3.org/css-validator/>)
- Validador Unificado do W3C - Unicorn - <https://validator.w3.org/unicorn/> (<https://validator.w3.org/unicorn/>)

Nestes validadores você pode realizar sua validação por um link de site publicado na web, por um código ou mesmo por um arquivo .html.

Utilize os validadores sempre que precisar verificar o seu código!

O QUE VOCÊ DEVE TER ENTENDIDO NESTE TÓPICO:

- O que são e como trabalhar com layouts responsivos;
- O que são e como utilizar frameworks;
- O que são e como utilizar bibliotecas de templates prontos;
- O que são e como utilizar validadores de código HTML e CSS;

Quiz

Exercício Final

Layout responsivo



INICIAR ➤

Referências

DEVELOPER MOZILLA. Usando a meta tag viewport para controlar o layout em navegadores de dispositivos móveis. Disponível em <https://developer.mozilla.org/pt-BR/docs/Mozilla/Mobile/Viewport_meta_tag> (https://developer.mozilla.org/pt-BR/docs/Mozilla/Mobile/Viewport_meta_tag). Acesso em 05 jun. 2017.

DEVELOPER MOZILLA. Mobile Web Development. Disponível em <<https://developer.mozilla.org/en-US/docs/Web/Guide/Mobile>> (<https://developer.mozilla.org/en-US/docs/Web/Guide/Mobile>). Acesso em 05 jun. 2017.

SILVA, Murício Samy. *HTML5: A linguagem de marcação que revolucionou a web.* 1. ed. São Paulo: Novatec Editora, 2011.

TERUEL, Evandro Carlos. *HTML5: Guia prático.* 1. ed. São Paulo: Erica, 2011.

W3C. HTML5 - Curso W3C Escritório Brasil. Disponível em <<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>> (<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>). Acesso em 01 jun. 2017.

W3SCHOOLS. Responsive Web Design - The Viewport. Disponível em <https://www.w3schools.com/css/css_rwd_viewport.asp> (https://www.w3schools.com/css/css_rwd_viewport.asp). Acesso em 01 jun. 2017.

W3SCHOOLS. What is a Media Query?. <https://www.w3schools.com/css/css_rwd_mediaqueries.asp> (https://www.w3schools.com/css/css_rwd_mediaqueries.asp)https://www.w3schools.com/css/css_rwd_mediaqueries.asp. Acesso em 05 jun. 2017.

W3SCHOOLS. Responsive Web Design - Introduction. Disponível em <https://www.w3schools.com/css/css_rwd_intro.asp> (https://www.w3schools.com/css/css_rwd_intro.asp). Acesso em 05 jun. 2017.

W3SCHOOLS. What is a Grid-View?. Disponível em <https://www.w3schools.com/css/css_rwd_grid.asp> (https://www.w3schools.com/css/css_rwd_grid.asp). Acesso em 05 jun. 2017.



Avalie este tópico



ANTERIOR
[Som e vídeo](#)

Biblioteca
(<https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

Índice

® Todos os direitos reservados

Ajuda?
(<https://ava.uninove.br/seu/AVA/topico/topico.php?idCurso=1>)

