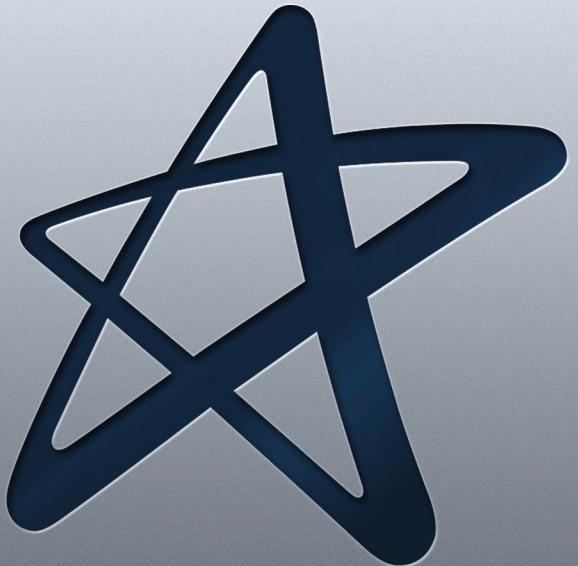


Aplicações para Internet



Material Teórico



Layouts

Responsável pelo Conteúdo:

Prof. Ms. Alexander Gobbato Paulino Albuquerque

Revisão Textual:

Profa. Ms. Fátima Furlan



- *Layouts*
- Construindo *Layout* e Mesclando Estilos
- *Frameworks de Grid*
- A Regra CSS para Mídia: @media



OBJETIVO DE APRENDIZADO

- *Web Standards*;
- *Wireframe*;
- Modelagem de aplicação WEB;
- Criação de *layouts*;
- *Layouts* responsivos e adaptativos.



Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja uma maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:



Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como o seu “momento do estudo”.
- ✓ Procure se alimentar e se hidratar quando for estudar, lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo.
- ✓ No material de cada Unidade, há leituras indicadas. Entre elas: artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados.
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e aprendizagem.

Layouts

Layout é um termo da língua inglesa que faz parte dos dicionários de língua portuguesa também utilizada como leiaute. O conceito pode traduzir-se como plano, arranjo, esquema, design, projeto e é bastante usado no âmbito das tecnologias.

Layout Líquido

Trata-se de uma organização de conteúdo que se adapta ou se molda ao tamanho da janela.

Veja as telas abaixo e conforme o dispositivo elas devem se comportar de forma diferente, apresentando o mesmo conteúdo.



Figura 1 – Janela cheia



Figura 2 – Tela Ipad



Figura 3 – Tela Iphone



Figura 4 – Tela responsiva

Layout Congelado

Trata-se de uma organização de conteúdo que permanece a mesma, mesmo que o tamanho da janela mude.



Figura 5



Figura 6

Mesmo a janela aumentando o conteúdo permanece ocupando o mesmo espaço.

Flutuação – *Float*

A propriedade *float* foi criada para fazer com que os elementos flutuem à direita ou à esquerda de um conteúdo.



Figura 7

Construindo Layout e Mesclando Estilos

Layout Líquido com Float

- Todos os blocos (divs) tem seu tamanho definidos com porcentagem
- Margens e “*Paddings*” devem ser definidos com porcentagens.
- Todos os blocos flutam para a esquerda
- O bloco que inicia uma nova linha deve ter a propriedade *clear: left;*

Problema:

- não é desejável que as margens, os ‘*paddings*’ sejam de tamanho variável e as bordas somem.

Código:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
5  <title>Layout com float</title>
6  </head>
7  <body>
8
9  <div id="a"><h1>Título</h1><h2>Sub-título</h2></div>
10
11 <div id="b">
12 <h3>Bloco Esquerdo</h3>
13 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium,
14 totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt
15 explicabo.
16 </div>
17
18 <div id="c">
19 <h3>Bloco Central</h3>
20 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium,
21 totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae
22 dicta sunt explicabo.
23 </div>

```

Figura 8

```

24 <div id="d">
25   <h3>Bloco Direito</h3>
26   Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium,
27   totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta
28   sunt explicabo.
29 </div>
30
31 <div id="e">&copy; Rodapé - 2017</div>
32 </body>
33 </html>

```

Figura 9

Aplicando css as camadas divs:

```

1  div {
2    border:15 solid gray;
3    margin: 0.5%;
4    padding: 0.5%;
5  }

```

Figura 10

Explicando:

Para cada elemento div na página será aplicada uma borda de espessura 15 pixels, sólida e cinza e as margens estão separadas por 0,5% e espaçamento entre os elementos também estão separadas por 0,5%.

```

7  #a { background-color:aquamarine;}
8
9  #b {
10   background-color:coral;
11   float:left;
12   width: 25%;
13 }
14
15 #c {
16   background-color:darkcyan;
17   float:left;
18   width: 44%;
19 }
20
21 #d {
22   background-color:darksalmon;
23   float:left;
24   width: 25%;
25 }
26
27 #e {
28   background-color:darkseagreen;
29   text-align:center;
30   clear:left;
31 }

```

Figura 11 – Aplicando estilo

Explicando:

No html temos vários elementos marcados com o id **a,b,c,d e e**.

Para o identificador **a** está sendo preenchida a cor de fundo com *aquamarine*, no identificador **b** está sendo aplicada a cor de fundo coral, elemento deslocado e móvel a esquerda com do tamanho de 25%. No identificador **c** está sendo aplicado

a cor de fundo *darkcyan*, elemento deslocado e móvel a esquerda com do tamanho de 44%, no identificador **d** a cor de fundo é *darksalmon*, elemento deslocado e móvel a esquerda e o tamanho do elemento em 25% e por último o identificador **e** com cor de fundo *darkseagreen*, texto centralizado e o utilizando o elemento *clear:left*, estamos criando uma nova sessão, ou seja, o próximo elemento que será inserido estará abaixo das divs que foram criadas.

Resultado:



Figura 12



Figura 13

Layout Parcialmente Líquido com Float

- Todos os blocos (divs) têm seu tamanho definidos em *pixels*, menos o bloco central;
- Margens, “*Paddings*” e bordas devem ser definidos em *pixels*;
- Há blocos que flutuam para a esquerda e outros para direita;
- O bloco que inicia uma nova linha deve ter a propriedade *clear: both*.

Problemas:

- Os blocos à direita têm que vir antes do bloco central no HTML;
- Definir a margem do bloco central pode ser confuso.

Código:

```

1  <!DOCTYPE html>
2  <html>
3  |<head>
4  |<meta content="text/html; charset=utf-8" http-equiv="Content-Type">
5  |<title>Layout com float</title>
6  |</head>
7  <body>
8
9  <div id="a"><h1>Título</h1><h2>Sub-título</h2></div>
10
11 <div id="b">
12 |<h3>Bloco Esquerdo</h3>
13 |<p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque
14 |</p></div>

```

Figura 14

```

15 <div id="d">
16 |<h3>Bloco Direito</h3>
17 |<p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque
18 |<p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque
19 |</div>
20
21 <div id="c">
22 |<h3>Bloco Central</h3>
23 |<p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque
24 |</div>
25
26 <div id="e">© Rodapé - 2016</div>
27
28 </body>
29 </html>

```

Figura 15

Explicando:

Observem que nesse documento temos a seguinte disposição **a**, **b**, **d**, **c** e **e**.

```


Figura 16 – Aplicando css as camadas divs



Explicando:



Todo elemento div no documento html será configurado para ter uma borda de 1 pixel, sólida e cinza. A margem de 5 pixel e espaçamento entre os elementos de 5 pixel.



14


```

Aplicando estilo:

```

#a { background-color:aquamarine; }

#b {
  background-color:coral;
  float:left;
  width: 230px;
  margin-top:0;
}

#c {
  background-color:darkcyan;
  margin-left: 252px;
  margin-right: 252px;
}

#d {
  background-color:darksalmon;
  float:right;
  width: 230px;
  margin-top:0;
}

#e {
  background-color:darkseagreen;
  text-align:center;
  clear:both;
}
  
```

Figura 17

Explicando:

No html temos vários elementos marcados com o id **a,b,c,d e e**.

Para o identificador **a** está sendo preenchida a cor de fundo com *aquamarine*, no identificador **b** está sendo aplicado a cor de fundo coral, elemento deslocado e móvel a esquerda com do tamanho fixo de 230 *pixel* e a margem de cima com top 0 (zero) ou seja, o elemento começará bem no topo da tela, No identificador **c** está sendo aplicada a cor de fundo *darkcyan* com a margem esquerda definida em 252 *pixel* e a margem direita em 252 *pixel*, no identificador **d** a cor de fundo é *darksalmon*, elemento deslocado e móvel a direita e o tamanho do elemento em 230 *pixel* fixo e a margem superior em 0 *pixel* e por último o identificador **e** com cor de fundo *darkseagreen*, texto centralizado e o utilizando o elemento *clear:both*, estamos criando uma nova sessão, ou seja, o próximo elemento o conteúdo permanece abaixo dos elementos flutuantes independentemente para que lado flutuem.

Resultado:

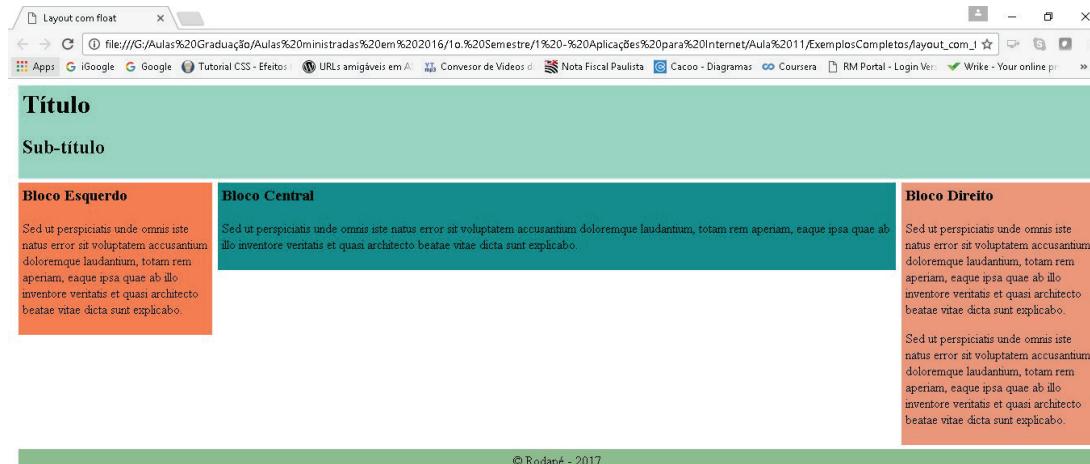


Figura 18



Figura 19

Layout Congelado com *Float*

- Todos os blocos de *layout* devem estar dentro de um bloco que os contém, comumente chamado “*wrapper*” ou “*container*”
- Este bloco deve ter uma largura (*width*) determinada e ser centralizado. Para isso se define uma margem automática, o que faz o sistema colocar a mesma margem do lado direito e esquerdo: *margin: 0 auto;*

Problema:

- como dentro do “wrapper” se utiliza uma das duas técnicas descritas, os mesmos problemas aparecem aqui.

Código:

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
6  <title>Layout com float</title>
7
8  </head>
9
10 <body>
11
12 <div id="wrapper">
13
14   <div id="a"><h1>Titulo</h1><h2>Sub-título</h2></div>
15
16   <div id="b">
17     <h3>Bloco Esquerdo</h3>
18     Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium,
19     totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta
20     sunt explicabo.
21 </div>

```

Figura 20

```

22
23   <div id="d">
24     <h3>Bloco Direito</h3>
25     Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium,
26     totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta
27     sunt explicabo.
28 </div>
29
30
31   <div id="e">
32     <h3>Bloco Central</h3>
33     Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium,
34     totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta
35     sunt explicabo.
36 </div>
37
38   <div id="e">&copy; Rodapé - 2016</div>
39
40 </div>
41 </body>
42 </html>

```

Figura 21

Explicando:

Observem que nesse documento temos a seguinte disposição **a**, **b**, **d**, **c** e **e**.

```

div {
  border: 1px solid gray;
  margin: 5px;
  padding: 5px;
}

```

Figura 22 – Aplicando css as camadas divs

Explicando:

Todo elemento div no documento html será configurado para ter uma borda de 1 *pixel*, sólida e cinza. A margem de 5 *pixel* e espaçamento entre os elementos de 5 *pixel*.

Aplicando estilo:

```
#a { background-color:aquamarine; }

#b {
    background-color:coral;
    float:left;
    width: 230px;
    margin-top:0;
}

#c {
    background-color:darkcyan;
    margin-left: 252px;
    margin-right: 252px;
}

#d {
    background-color:darksalmon;
    float:right;
    width: 230px;
    margin-top:0;
}

#e {
    background-color:darkseagreen;
    text-align:center;
    clear:both;
}
```

Figura 23

Explicando:

No html temos vários elementos marcados com o id **a,b,c,d e e**.

Para o identificador **a** está sendo preenchida cor de fundo com *aquamarine*, no identificador **b** está sendo aplicada a cor de fundo coral, elemento deslocado e móvel a esquerda com do tamanho fixo de 230 *pixel* e a margem de cima com top 0 (zero) ou seja, o elemento começará bem no topo da tela, No identificador **c** está sendo aplicado a cor de fundo *darkcyan* com a margem esquerda definida em 252 *pixel* e a margem direita em 252 *pixel*, no identificador **d** a cor de fundo é *darksalmon*, elemento deslocado e móvel a direita e o tamanho do elemento em 230 *pixel* fixo e a margem superior em 0 *pixel* e por último o identificador **e** com cor de fundo *darkseagreen*, texto centralizado e o utilizando o elemento *clear:both*, estamos criando uma nova sessão, ou seja, o próximo elemento o conteúdo permanece abaixo dos elementos flutuantes independentemente para que lado flutuem.

Aplicando estilo:

```
#wrapper {
    width: 960px; /* esta é uma largura usada em muitos sites */
    margin: auto;
}
```

Figura 24

Explicando:

O elemento *id wrapper* terá o tamanho fixo de 960 *pixel* e margin:auto para centralizar o elemento horizontalmente onde ele está contido.

Resultado:



Figura 25



Figura 26

Problemas de *Layout* usando *Float*

- Dependência da ordem dos blocos no HTML (quando se usa *float: right;* a ordem fica estranha)
- Margem e borda de tamanho variável ou de cálculo complicado
- Fica no controle do programador manter as larguras das colunas coerentes em linhas diferentes.
- A altura dos blocos em uma mesma linha ou devem ter tamanho fixado (o que pode ser ruim se o conteúdo for alterável) ou ficam de tamanhos diferentes.

Frameworks de Grid

Framework é um conjunto de códigos (ou biblioteca) que estendem uma linguagem, ou seja, que ‘criam’ novos termos com novas funcionalidades.

Os *Frameworks* de *Grid* normalmente são criados em um arquivo CSS. Este arquivo deve ser anexado à página onde o *Framework* deve ser usado.

Na sua maioria, eles oferecem classes preparadas para criar o *wrapper* (que pode ser de tamanho fixo ou não) e blocos cuja largura ocupa uma certa quantidade de ‘colunas’ do *grid*.



Figura 27 – Framework 960.css

O usual é que hajam 12 ou 16 colunas

Usando o 960.gs

- Baixe o *framework* (<http://960.gs/>)
- Crie a página HTML com divs para *layout* dentro de uma div *wrapper* e dispostos na ordem correta, mas sem nenhuma informação de *layout* no CSS
- Coloque o arquivo 960.css na pasta e faça o link para ele no arquivo HTML:
`<link rel="stylesheet" href="960.css" />`
- No *wrapper*, adicione a classe container_12

Em cada div, adicione uma classe de acordo com o espaço que o bloco irá ocupar no *layout*:

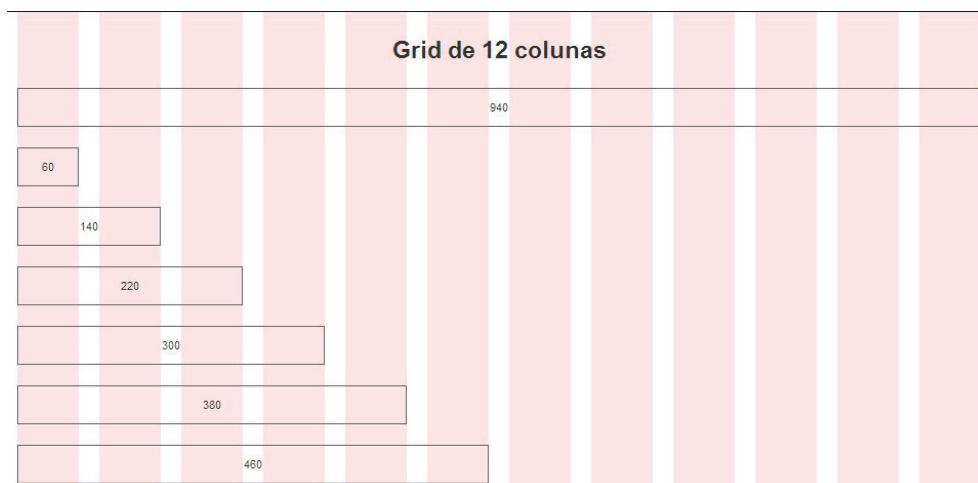


Figura 28

Assegure-se de em cada linha ocupar exatamente doze colunas.

Código:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
5  <title>Layout com Framework</title>
6  <link rel="stylesheet" href="css/960.css" />
7  <style>
8  body > div > div {
9    margin-bottom: 10px;
10   padding: 15px;
11   box-sizing: border-box;
12   border: 1px solid black;
13 }
14
15 #a, #f {background-color:aquamarine;}
16
17 #b, #g {background-color:coral;}
18
19 #c, #h {background-color:darkcyan;}
20
21 #d,#i {background-color:darksalmon;}
22
23 #e, #j, #k {
24   background-color:darkseagreen;
25   text-align:center;
26 }

```

Figura 29

```

27 h1,h3, h2{ margin-top: 0;}
28
29 </style>
30 </head>
31 <body>
32 <div id="wrapper" class="container_12">
33 <div id="a" class="grid_12"><h1>Título</h1><h2>Sub-título</h2></div>
34
35 <div id="b" class="grid_6">
36 <h3>Bloco 1 - Linha 1</h3>
37 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
38 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
39 </div>
40
41 <div id="c" class="grid_6">
42 <h3>Bloco 2 - Linha 1</h3>
43 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
44 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
45 </div>
46
47 <div id="d" class="grid_4">
48 <h3>Bloco 3 - Linha 2</h3>
49 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
50 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
51 </div>
52

```

Figura 30

```

53 <div id="e" class="grid_4">
54 <h3>Bloco 4 - Linha 2</h3>
55 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
56 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
57 </div>
58
59 <div id="f" class="grid_4">
60 <h3>Bloco 5 - Linha 2</h3>
61 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
62 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
63 </div>
64
65 <div class="clear"></div>
66
67 <div id="g" class="grid_3">
68 <h3>Bloco 6 - Linha 3</h3>
69 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
70 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
71 </div>
72
73 <div id="h" class="grid_3">
74 <h3>Bloco 7 - Linha 3</h3>
75 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
76 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
77 </div>

```

Figura 31

```

78
79 <div id="i" class="grid_3">
80 <h3>Bloco 8 - Linha 3</h3>
81 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
82 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
83 </div>
84
85 <div id="j" class="grid_3">
86 <h3>Bloco 9 - Linha 3</h3>
87 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem
88 aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
89 </div>
90
91 <div id="k" class="grid_12">&copy; Rodapé - 2017</div>
92
93 </div>
94 </body>
95 </html>

```

Figura 32

Entendendo o código:

Vamos aos pontos principais, aqui não comentaremos os estilos e sim a estrutura do documento para que o *framework* tenha o melhor proveito.

Observem que na linha 6 temos o *import* do *framework*:

```
<link rel="stylesheet" href="css/960.css" />
```

Na linha 38 foi criada uma “caixa organizadora” de identificador “wrapper” mas o importante aqui não é o nome e sim a *class*, observem que a *class* é a “container_12”, ou seja, estamos estruturando o documento para que recebe 12 colunas.

Dentre dessa “caixa organizadora” temos várias outras “caixas”, essas caixas devem contabilizar um total de 12 *grids* por linha, ou seja, cada caixa receberá uma *class* chamada de “grid_x”, onde o x será representado pela quantidade de coluna na linha.

Com base nessas informações temos o seguinte resultado:



Figura 33

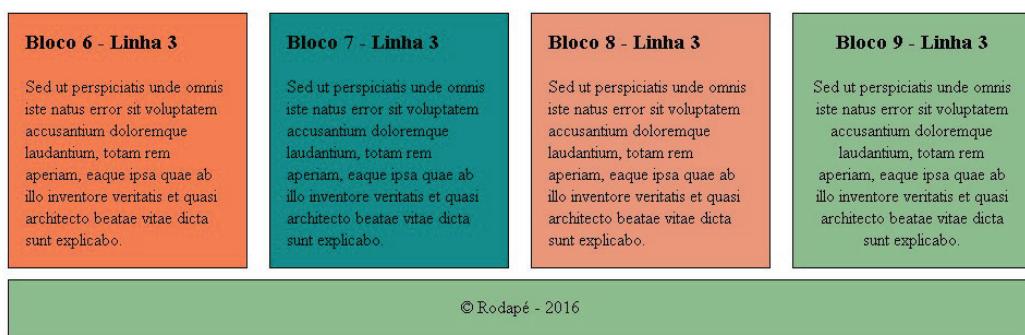


Figura 34

O resultado é o seguinte:



Figura 35 – 1 class grid_12



Figura 36 – 2 class grid_6

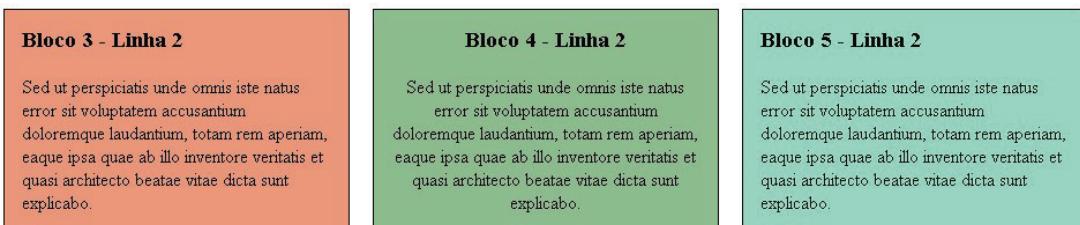


Figura 37 – 3 class grid_4

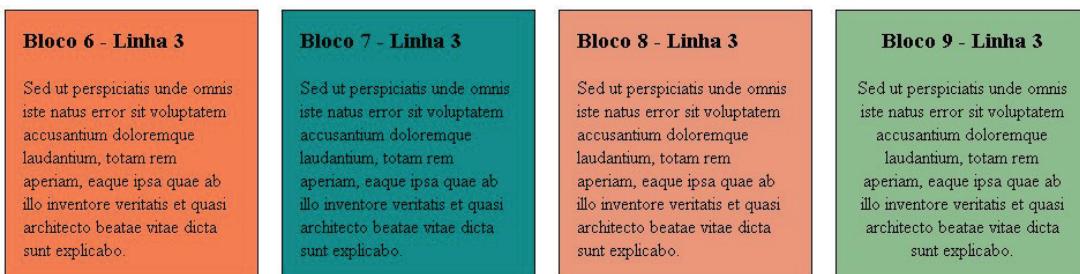


Figura 38 – 4 class grid_3

© Rodapé - 2016

Figura 39 – 1 class grid_12

Layout adaptados para Contextos específicos

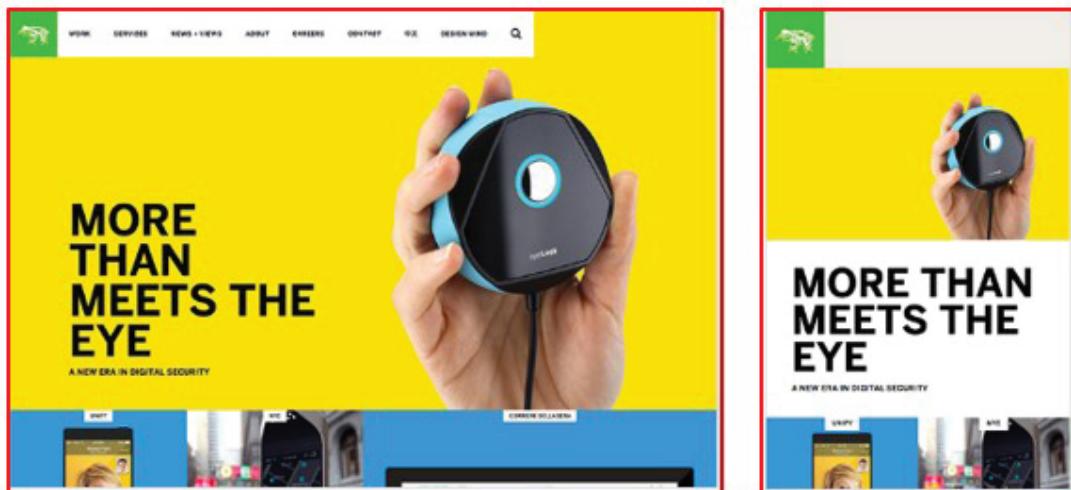


Figura 40 - Mesmo HTML – Apresentações diferentes

- Acessibilidade de sites em vários tipos de telas/dispositivos.
- Devemos pensar na mobilidade (celular, tablets), além de outras telas como em geladeiras, máquinas de lavar, TVs etc.
- **Layout responsivo**
 - Forte uso de CSS3, consulta de mídia, regras @media e uso de porcentagens, sua definição básica é para caber em todo tamanho de tela ou dispositivo.
- **Layout adaptativo**
 - Usando um conjunto pré-definido de tamanhos de layout com base no tamanho da tela do dispositivo, juntamente com CSS e JavaScript.
- **Layout responsivo vs adaptativo:**
 - **Semelhança:** A semelhança entre os dois é que ambos permitem que sites sejam visualizados em dispositivos móveis e diferentes tamanho de tela.
 - **Diferenças:** a diferença está na criação da estrutura, o responsivo conta com grades flexíveis e layout fluído, e o adaptativo conta com tamanhos de tela pré-definidos.
 - O responsivo pode ter mais código, já o adaptativo tem uma abordagem mais simplificada, porém este exige scripts para auxiliar na adaptação.

Fonte: <https://goo.gl/3nSYpm>

A Regra CSS para Mídia: @media

É possível no CSS fazer especificações que só são consideradas em condições determinadas. Isso é feito envolvendo as regras de especificação pela diretiva.

```
@media <condição> { ... }
```

Entre as chaves vão as regras CSS, que só serão usadas se a mídia corresponder à <condição>. Exemplos de <condição> são:

- *only screen and (max-width: 768px) and (min-width: 481px)* (tela com largura entre 481px e 768px);
- *only screen and (max-width: 480px)* (tela de largura menor ou igual a que 480px).

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:

Leitura

Media Queries

<https://goo.gl/QVaXya>

Criar um site responsivo com HTML5 e CSS3 - Parte 1/3

<http://bit.ly/2Uuk15E>

Criar um site responsivo com HTML5 e CSS3 - Parte 2/3

<http://bit.ly/2UrZUVd>

Criar um site responsivo com HTML5 e CSS3 - Parte 3/3

<http://bit.ly/3bdZXuf>

Visão geral do Flexible Box Model

<https://goo.gl/mvSlkC>

Usando CSS flexible boxes (Caixas Flexíveis)

<https://goo.gl/Pyl6Xx>

Referências

Clark, R.; Murphy, C.; Studholme, O.; Manian, D. **Introdução ao HTML5 e ao CSS3**. Rio de Janeiro: Editora Alta Books, 2014.

SILVA, M. S. **HTML 5: a linguagem de marcação que revolucionou a web**. São Paulo: Novatec, 2011.

DEITEL, PAUL J.; DEITEL, HARVEY M. Ajax, **Rich Internet Applications e Desenvolvimento Web para Programadores** Pearson 776 ISBN: 9788576051619. (e-book).



Cruzeiro do Sul Virtual
Educação a Distância

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo - SP - Brasil
Tel: (55 11) 3385-3000



Cruzeiro do Sul
Educacional