

Tecnologias para Internet I

Aula 9: Web Responsiva I

INTRODUÇÃO



A nova versão do HTML 5 apresenta uma série de mudanças, sendo a mais importante de todas a transformação de layouts dependendo das mídias. Podemos criar um layout para um determinado tamanho e se mudar de mídia o layout se adapta para esse novo tamanho, mantendo a base original do trabalho e suas definições.

Vamos lá!

OBJETIVOS



Determinar o uso de Media Queries no HTML 5;

Examinar tipos de Layouts no html 5.

UTILIZANDO CSS *MEDIA QUERIES*

Podemos definir media queries como a maneira que o navegador define que formatação de página vai apresentar a cada tipo de dispositivo.

Os *media types* são definidos para cada tipo de media recebendo um certo código CSS, fazendo com que o navegador direcione a abertura da página de uma certa forma. O HTML foi gerado para ser portátil, ou seja, lido e interpretado por qualquer tipo de dispositivo. Só que cada dispositivo exibe o HTML de uma maneira, devido a fatores como, por exemplo, as dimensões da tela.

Imagine que visitamos um site em um desktop; ele não terá o mesmo aspecto se visitarmos o mesmo site em um dispositivo móvel. Isso ocorre porque são dispositivos diferentes e formas diferentes de navegação. Por isso, a importância de usar essa nova forma de definição de CSS. Isso facilita a leitura do site em mídias diferentes.

Para que possamos manter nossas páginas sempre adequadas a cada tipo de visualização, utilizamos *media types* com a seguinte formatação:

All

Para todos os dispositivos.

Braille

Para dispositivos táteis.

Embossed

Para dispositivos que imprimem
em braille.

Print

Para impressão em papel.

Projection

Para apresentações como PPS.

Screen

Para monitores ou dispositivos com telas coloridas e resolução adequada.

Speech

Para sintetizadores de voz. As CSS 2 têm uma especificação de CSS chamada Aural, onde podemos formatar a voz dos sintetizadores.

Tty

Para dispositivos que possuem uma grade fixa para exibição de caracteres, tais como: Teletypes, Terminais e também dispositivos portáteis com display limitados.

TV

Para dispositivos como televisores, ou seja, com baixa resolução, quantidade de cores e scroll limitados.

Handheld

Para dispositivos portáteis, geralmente com telas pequenas e banda limitada.

Observe, abaixo, um exemplo de utilização:



Com esse arquivo definimos que ele será usado quando for acessado pela mídia correspondente. Dessa maneira, o CSS não será aplicado em **dispositivos monocromáticos**.

Nas *media queries* são utilizados três operadores lógicos:

not

or

only

Veja um exemplo de utilização do operador lógico not:

```
<link rel="styleSheet" href="estilo.css" media="All and (not color)" />
```

Usamos para dispositivos monocromáticos, pois estamos negando o atributo color.

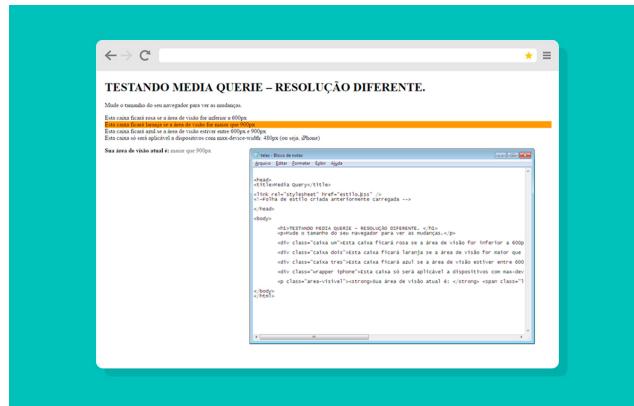
No próximo exemplo, podemos definir a resolução de vídeo que queremos modificar.

```
<link rel="stylesheet" href="estilo.css" media="screen and (max-width: 480px)" />
```

As modificações serão feitas em dispositivos de media screen e com resolução máxima de 480px.

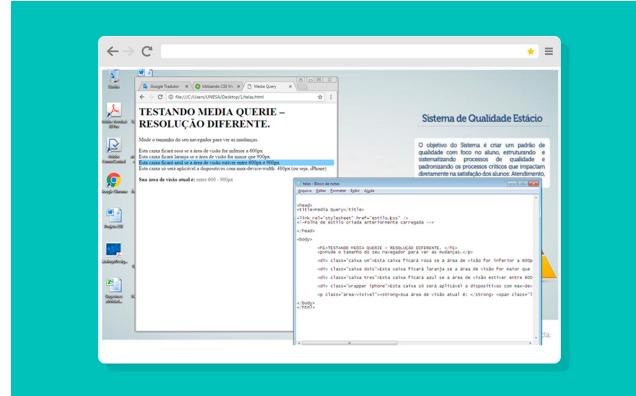
Agora, vamos a um exemplo completo.

A seguir temos um arquivo HTML criado sem nenhuma formatação.

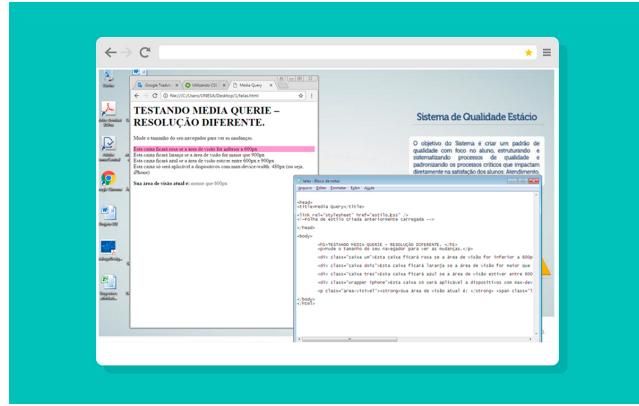


Se diminuirmos o navegador, trocaremos de cor sem chamar novamente a página:

Entre 600px e 900px



Menor que 600px



Com isso, podemos definir comandos para tamanhos de tela diferentes.

Saiba Mais

Clique aqui ([galeria/aula9/docs/a09_t03.pdf](#)) e analise os códigos utilizados no exemplo.

LAYOUT MULTICOLUNAS

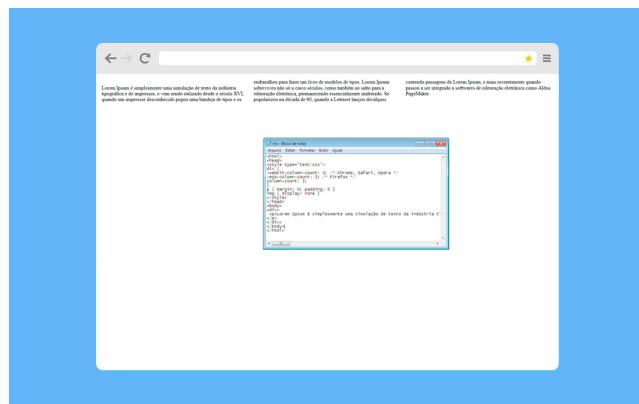
Sua principal vantagem é a flexibilidade, pois independe de dispositivo.

Esta especificação permite que o conteúdo da página se adeque de uma coluna para outra e o número de colunas pode variar dependendo do tamanho da janela.

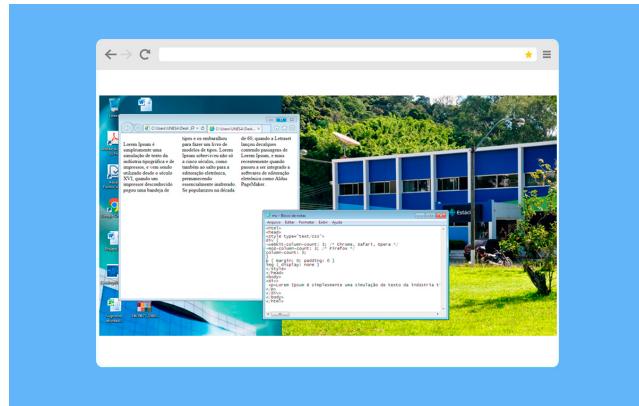
Para facilitar o seu entendimento, vamos direto a um exemplo:

No **código CSS (glossário)** definimos em quantas colunas apresentaremos o texto, fazendo distinção de navegador (existem comandos específicos para cada navegador).

No **HTML (glossário)** temos um código simples de texto.



Se diminuirmos o navegador, o layout irá se ajustar:



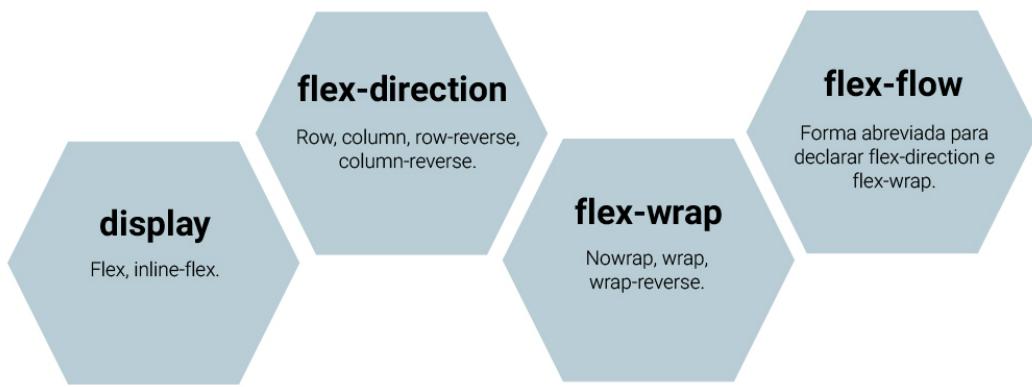
Podemos ainda dividir as colunas com uma linha. Para isso, utilizaremos o [CSS \(glossário\)](#).



LAYOUT DE BOX FLEXÍVEL

É um novo algoritmo para disposição de boxes dentro do HTML. Este modelo permite um controle mais apurado do Box permitindo “dobrar” suas dimensões para preencher espaços não utilizados.

Veja as propriedades a seguir:



Pode-se usar as seguintes propriedades:

row, column, row-reverse, column-reverse, nowrap, wrap, wrap-reverse.

VAMOS AO EXEMPLO!

Renderização dos boxes (`.box1` e `.box2`) contidos no `div#container` renderizados segundo o Box Model CSS 2.1.

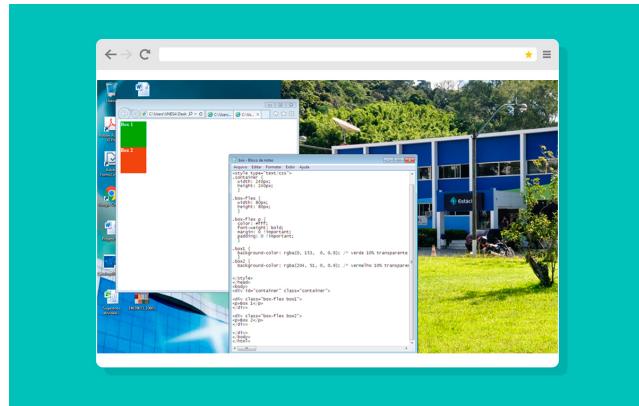
Observe agora a estilização padrão da marcação e sua respectiva renderização.

CSS

```
.container {  
    width: 240px;  
    height: 240px;  
}  
  
.box-flex {  
    width: 80px;  
    height: 80px;  
}  
  
.box-flex p {  
    color: #fff;  
    font-weight: bold;  
    margin: 0 !important;  
    padding: 0 !important;  
}  
  
.box1 {  
    background-color: rgba(0, 153, 0, 0.9); /* verde 10% transparente */  
}  
.box2 {  
    background-color: rgba(204, 51, 0, 0.9); /* vermelho 10% transparente */  
}
```

HTML

```
<div id="containe" class="container">  
  
    <div class="box-flex box1">  
        <p>Box 1</p>  
    </div>  
    <div class="box-flex box2">  
        <p>Box 2</p>  
    </div>  
</div>
```



Agora, iremos criar um identificador container e configurar como queremos colocar as caixas.

Criar container

```
#container {
  display: box;
  box-align: stretch;
  box-orient: horizontal;
}
```

Definir a largura das caixas

```
.container {
  /* width: 240px; */
  height: 240px;
}
```

Tornar o box flexível

```
.box-flex {
  box-flex: 1;
}
```

Definir as medias do box

```
.box-flex {
  /* width: 80px; */
  height: 80px;
}
```

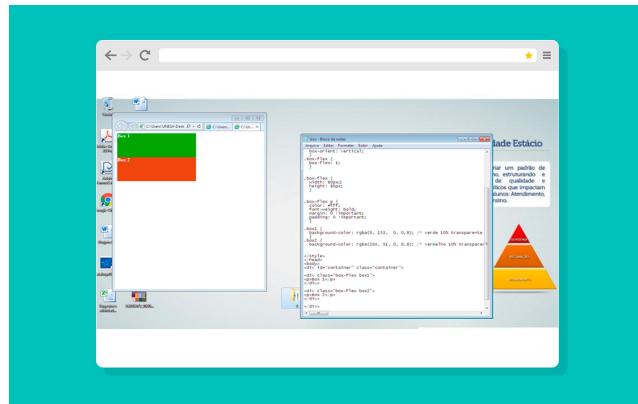
```
.box-flex p {
```

```

color: #fff;
font-weight: bold;
margin: 0 !important;
padding: 0 !important;
}

.box1 {
background-color: rgba(0, 153, 0, 0.9); /* verde 10% transparente */
}
.box2 {
background-color: rgba(204, 51, 0, 0.9); /* vermelho 10% transparente */
}

```



Podemos definir diversas formas de apresentação dessas caixas, cada uma para um tipo de apresentação.

Saiba Mais

, Clique aqui (<http://www.maujor.com/tutorial/css3-modulo-box-flexivel-layout.php>) e veja mais sobre construção de layout.

ATIVIDADES

Para finalizarmos esta aula, responda as questões a seguir:

Questão 1: São propriedades de divisão de colunas:

webkit-column-count.
moz-column-count.
column-color.

Marque a alternativa correta:

- a) Somente a I.
- b) Somente a III.
- c) Somente a I e a II.

- d) Somente a I e a III.
- e) Somente a I, II e III.

Justificativa

Questão 2: Qual a função do código a seguir? `<link rel="styleSheet" href="estilo.css" media="All and (not color)" />`:

- a) Chama um arquivo CSS para definição dos comando da página.
- b) Chama um arquivo CSS e define configurações de um arquivo monocromático.
- c) Chama um comando específico dentro da Tag style.
- d) Chama um arquivo interno do HTML para definir as propriedades de cor.
- e) Chama uma comando diretamente na linha de instrução.

Justificativa

Questão 3: Qual a função do atributo media de um comando link?

Resposta Correta

Questão 4: O que é Layout de Box Flexível?

Resposta Correta

Glossário

CÓDIGO CSS

```
<style type="text/css">
div {
-webkit-column-count: 3; /* Chrome, Safari, Opera */
-moz-column-count: 3; /* Firefox */
column-count: 3;
}
p { margin: 0; padding: 0 }
</style>
```

HTML

```
<html>
<head>
</head>
<body>
<div>
<p>Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o séc
```

úculo XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Popularizou-se na década de 1960, quando a Letraset lançou decalques contendo passagens de Lorem

Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.
</p>
</div>
</body>
</html>

CÓDIGO CSS

```
<style type="text/css">

div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    /* Define o espaçamento entre as colunas */
    -webkit-column-gap: 30px; /* Chrome, Safari, Opera */
    -moz-column-gap: 30px; /* Firefox */
    column-gap: 30px;

    /* Define uma linha solida entre as colunas */
    -webkit-column-rule-style: solid; /* Chrome, Safari, Opera */
    -moz-column-rule-style: solid; /* Firefox */
    column-rule-style: solid;
    p {
        margin: 0;
        padding: 0;
    }
}

</style>
```