
Buscar

post favorito comentários

Utilizando CSS Media Queries

Veja neste artigo como fazer com que páginas HTML tenham diferentes comportamentos e formatações, de acordo com as características do dispositivo no qual estão sendo visualizadas. Isso é possível utilizando as CSS Media Queries.



Neste artigo vamos falar sobre Media Queries, que é a utilização de Media Types com uma ou mais expressões envolvendo características de uma media para definir formatações para diversos dispositivos. Os browsers ou as aplicações lêem as expressões definidas na Query, caso o dispositivo em questão se encaixe nessas requisições o

CSS será aplicado.

Explicando o que é Media types

Os media types definem para que tipo de media um certo código CSS é direcionado. 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. Supondo que vistamos um site através de um desktop, ele não terá a mesma reação de que se visitarmos por um dispositivo móvel. Isso ocorre porque são dispositivos diferentes e formas diferentes de navegação.

Para que possamos manter nossas páginas sempre adequadas a cada tipo de visualização, utilizamos os media types, que são listados abaixo de acordo com sua utilização.

- **All:** Para todos os dispositivos.
- **Braille:** Para dispositivos táteis.
- **Embossed:** Para dispositivos que imprimem em braile.
- **Handheld:** Para dispositivos portáteis, geralmente com telas pequenas e banda limitada.
- **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 tem 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.

Observação: Os nomes das medias são todas case-sensitive, ou seja, letras maiúsculas e minúsculas são diferenciadas.

Media Queries - Problemas

Os problemas das media queries são que cada vez mais surgem dispositivos com diversos tamanhos e também

com hardwares bem parecidos aos dos desktops, fazendo com que esses dispositivos tenha uma navegação quase igual aos dos desktops. Podemos citar como exemplo o iPhone, sua tela tem excelente qualidade e seu navegador reproduz as páginas como um navegador comum de desktop). Assim não há a necessidade de preparar um layout e um CSS com media type Handheld para o iPhone, pois mesmo ele sendo um Handheld, ele não trabalha como um, porém, é importante frisar que ele também não trabalha exatamente como um desktop. Mesmo a renderização do MobileSafari sendo idêntica ao de um desktop, existe a questão do comportamento do usuário e a forma de navegação que são diferentes. Assim ficamos em um meio termo, nem podendo disponibilizar um CSS específico para Handheld e nem um CSS totalmente Screen.

Media Queries – Solução

A solução é que as media queries definem condições de um CSS específico, isso significa que se essas condições forem aprovadas, ou seja, se o dispositivo se adequar a todas as condições, o CSS será aplicado.

Listagem 1: Solução de problema com Media Queries.

```
<link rel="styleSheet" href="estilo.css" media="screen and (color)" />
```

Nesse código determinamos que o CSS só seja aplicado aos dispositivos de media Screen que possuam uma característica color. Dessa forma o CSS não será aplicado em dispositivos monocromáticos. Assim conseguiremos corrigir esse problema específico.

Operadores Lógicos

Nas media queries são utilizados três operadores lógicos: **not**, **or** e **only**.

Not: Gera uma sentença de negação.

Listagem 2: Usando o operador logico Not

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

Dessa maneira deixamos disponível também para dispositivos monocromáticos, pois estamos negando o atributo color.

Only: Não mostra os estilos do Browser que não reconhecem media queries, para ser utilizado basta utilizar a palavra only antes media desejada. Vejamos um exemplo.

Listagem 3: Usando o operador logico Only

```
<link rel="styleSheet" href="estilo.css" media="Only screen and (color)" />
```

Or: Serve para usar varias medias juntas, podendo ser substituído por uma vírgula “,”.

Listagem 4: Trocando o operador or por vírgula

```
<link rel="styleSheet" href="estilo.css" media="All and (color), projection and
```

Se qualquer uma das medias definidas forem verdadeiras, o CSS será aplicado.

Media Feature

Para diferenciarmos um dispositivo do outro, precisamos utilizar as características de cada um, para isso utilizamos as media features, que reconhecem certos recursos do dispositivo no qual está sendo visualizada a página. Abaixo temos um exemplo.

Listagem 5: Utilizando media feature

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

No exemplo acima determinamos que o CSS apenas seja aplicado aos dispositivos de media screen e com resolução máxima de 480px.

Há uma variedade de características que podemos utilizar para identificar os dispositivos que desejamos. Veja a lista com alguma dessas características:

- width
- height
- device-width
- device-height
- orientation
- aspect-ratio
- device-aspect-ratio
- color
- color-index
- monochrome
- resolution
- scan
- grid

Agora vamos por o que aprendemos até aqui em pratica. Para iniciar vamos começar criando uma folha de estilos CSS com nossas media queries definidas.

Listagem 6: Folha de estilo CSS

```
<style type="text/css">
body { padding: 15px; background:#eee; font-family: Arial, Helvetica, sans-serif; }

.caixa {
    border: solid 1px #666;
    padding: 5px 10px;
    margin: 40px;
}

.area-visivel span {
    color: #666;
    display: none;
}

/* max-width */
```

```
@media screen and (max-width: 600px) {  
    .um {  
        background: #F9C;  
    }  
    span.lt600 {  
        display: inline-block;  
    }  
}  
  
/* min-width */  
@media screen and (min-width: 900px) {  
    .dois {  
        background: #F90;  
    }  
    span.gt900 {  
        display: inline-block;  
    }  
}  
  
/* min-width & max-width */  
@media screen and (min-width: 600px) and (max-width: 900px) {  
    .tres {  
        background: #9CF;  
    }  
    span.bt600-900 {  
        display: inline-block;  
    }  
}  
  
/* max device width */  
@media screen and (max-device-width: 480px) {  
    .iphone {  
        background: #ccc;  
    }  
}  
</style>
```

Agora que já criamos nossa folha de estilos CSS, vamos salvar como estilo.css. Vamos agora criar a página HTML e carregar a nossa folha de estilos que acabamos de criar.

Listagem 7: Criando index.html.

```
<head>
<title>Media Query</title>

<link rel="stylesheet" href="estilo.css" />
<!--Folha de estilo criada anteriormente carregada -->

</head>

<body>

    <h1>TESTANDO MEDIA QUERIE - RESOLUÇÃO DIFERENTE. </h1>
    <p>Mude o tamanho do seu navegador para ver as mudanças.</p>

    <div class="caixa um">Esta caixa ficará rosa se a área de visão for info
    <div class="caixa dois">Esta caixa ficará laranja se a área de visão foi
    <div class="caixa tres">Esta caixa ficará azul se a área de visão estivi
    <div class="wrapper iphone">Esta caixa só será aplicável a dispositivos
    <p class="area-visivel"><strong>Sua área de visão atual é: </strong> <sp

</body>
</html>
```

Salvamos esse arquivo HTML como index.html e abrimos com um navegador de nossa preferência para testarmos. O resultado é o seguinte:

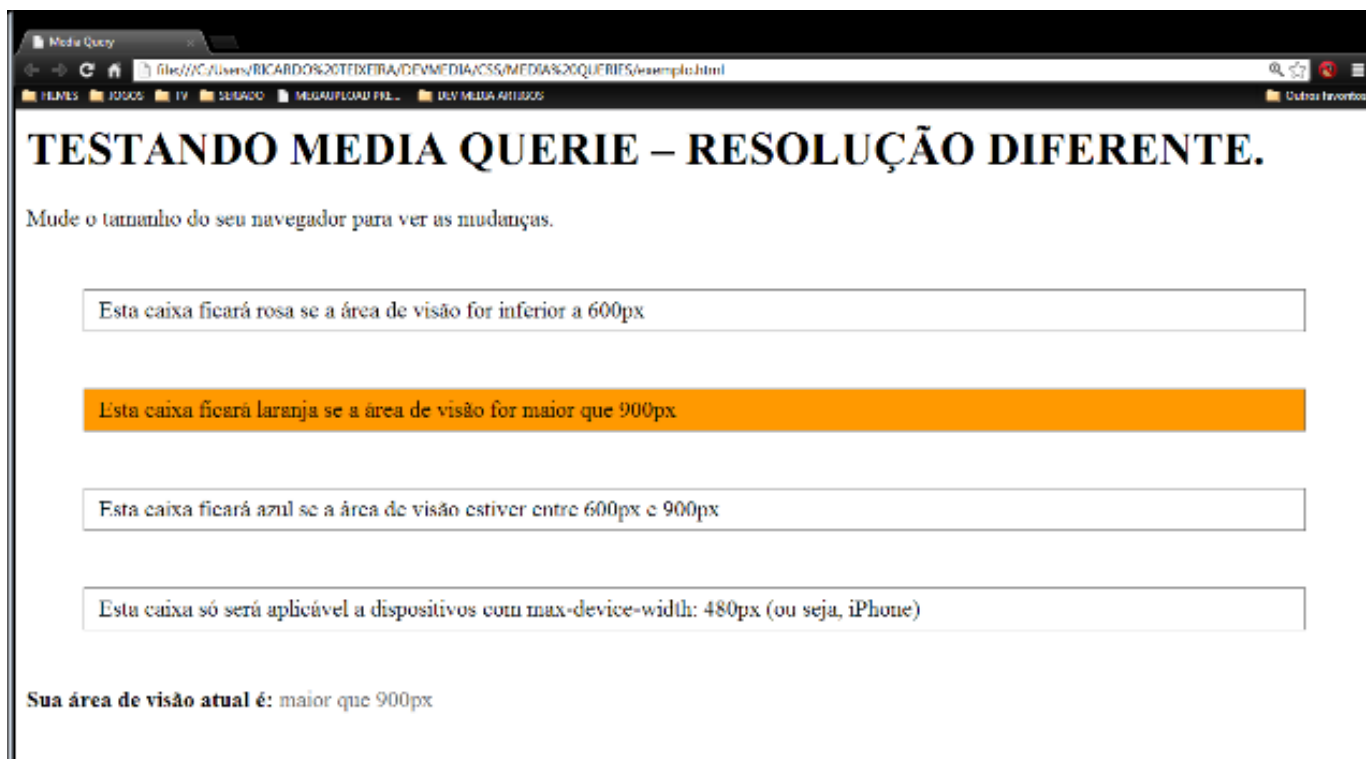


Figura 1: Navegador com resolução maior que 900px

Resultado com navegador em tamanho menor.

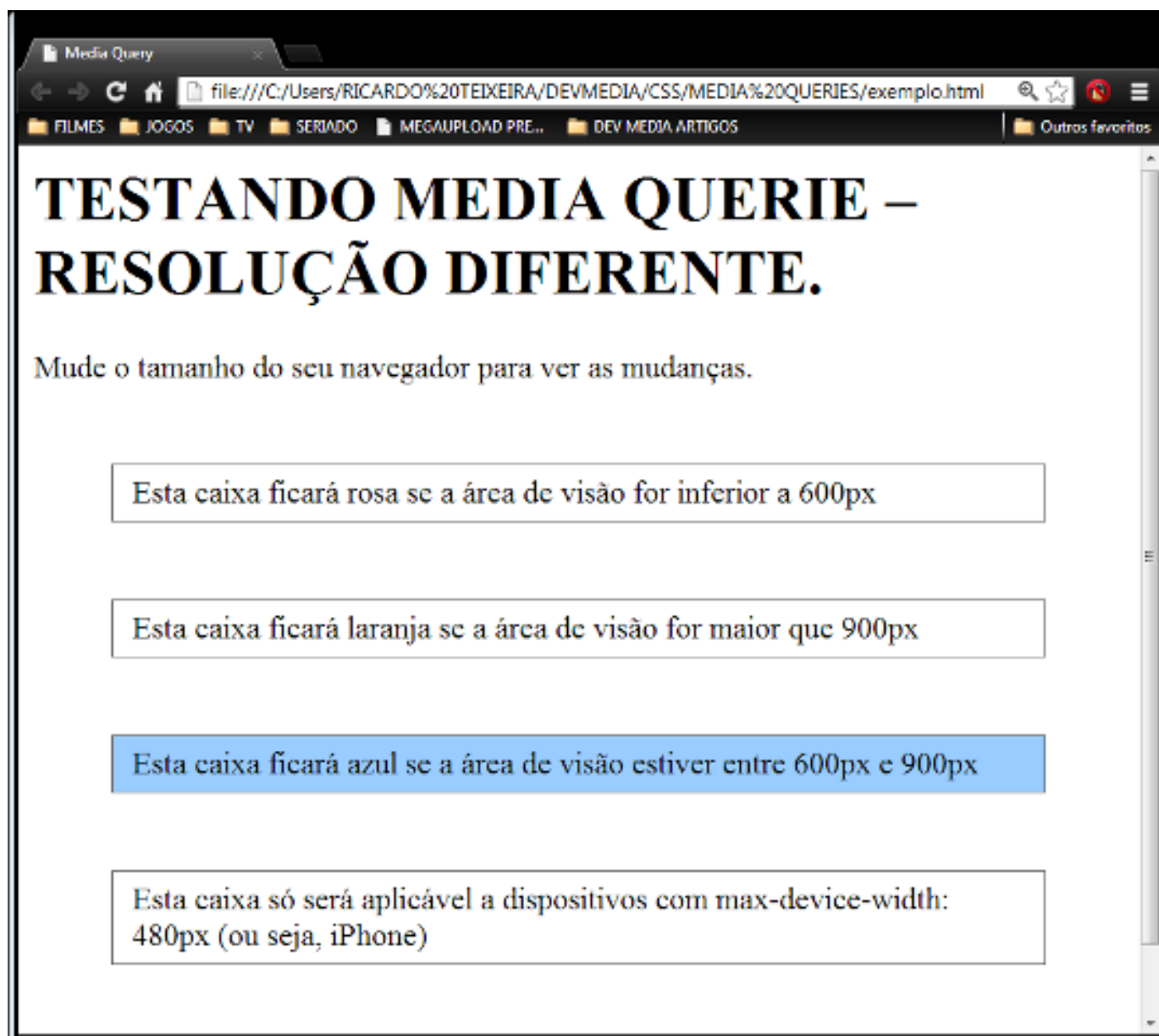


Figura 2: Navegador com resolução entre 600px e 900px

E para finalizar, o navegador com uma resolução abaixo de 600px.



QUERIE – RESOLUÇÃO DIFERENTE.

Mude o tamanho do seu navegador para ver as mudanças.

Esta caixa ficará rosa se a área de visão for inferior a 600px

Esta caixa ficará laranja se a área de visão for maior que 900px

Esta caixa ficará azul se a área de

Figura 3: Navegador com resolução abaixo de 600px

Vemos que a página, com a simples aplicação de media queries, se comportou de formas diferentes, de acordo com a resolução da tela.

Dessa forma terminamos mais um artigo, o qual espero que seja do agrado de todos os leitores. Me coloco à

disposição de todos para tirar duvidas e receber sugestões.

Até ao próximo artigo e um abraço a todos.



José Ricardo Teixeira

Colunista do portal DevMedia, formado em Web Designer , Manutenção de SoftWare e HardWare , Tecnico de programação em SQL, Delphi e PHP.

O que você achou deste post?



Gostei (1)



(0)

Comentário | Tire sua dúvida



Edson Rodrigues Pereira

Cara, muito interessante seu artigo, pena que os códigos não aparecem...

[há +1 mês] - Responder



[autor] José Ricardo Teixeira

Que bom que gostou, a falta de código foi um erro de publicação. Estarei avisando ao suporte para corrigirem. E obrigado por avisar.

[há +1 mês] - Responder



Ricardo Arrigoni

Aqui estão aparecendo normalmente os códigos

[há +1 mês] - Responder



Serviços

Inclua um comentário

Adicionar aos Favoritos

Marcar como lido/assistido

Incluir anotação pessoal

Versão para impressão

+Front-end web

Mais posts

Pocket Video

Desenvolvendo um chat usando NodeJS, Socket.io e jQuery

Artigo

Interface Metro: Criando um Menu Metro com HTML e CSS

Artigo

Programando em HTML5

Artigo

HTML5 Audio Tag: Crie um player de Áudio com HTML5

Artigo

NodeJS: Criando uma Rede Social

Artigo

JavaScript Blog: Criando serviço de microblog com NodeJS

Artigo

Nashorn: Desenvolvendo JavaScript na JVM

Artigo

Padronização com IIFE, AMD e RequireJS

Revista

Revista Front-end Magazine 1

Video aula

Contagem de caracteres - Curso de jQuery Básico - Aula 32

Listar mais conteúdo



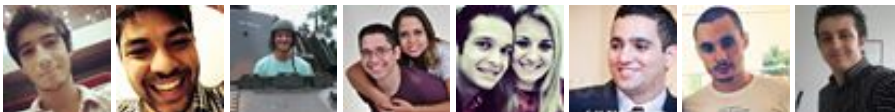
Anuncie | Loja | Publique | Assine | Fale conosco



DevMedia

Curtir

27.700 pessoas curtiram [DevMedia](#).



Plug-in social do Facebook

Hospedagem web por [Porta 80 Web Hosting](#)

Todos os Direitos Reservados a [Web-03](#)