

Programação Visual e Autoria Web

Media Query

Universidade Federal do Rio Grande do Norte

Introdução

- Atualmente páginas web devem ser desenvolvidas partindo do pressuposto que elas poderão ser exibidas em qualquer tipo de dispositivo.



FONTE: <https://www.makeuseof.com/tag/best-browser-for-android-tv/>



Introdução

- Uma alternativa que pode ser utilizada é criar páginas web diferentes para cada tamanho de tela.
- Existem vários problemas dessa abordagem:
 - Surgem diversos tamanhos de tela o tempo todo;
 - Uma versão com menos recursos pode frustrar o usuário;
 - Necessidade de manutenção para cada página;
 - Desajustes nos indexadores de pesquisa;
 - Necessidade de lidar com redirects de links criados por um usuário em um tamanho de tela para outro.

Media Query

- Media Queries são usadas no CSS quando se deseja modificar a exibição do site de acordo com o dispositivo utilizado ou parâmetros específicos (como largura do navegador)

```
@media screen {  
  body {  
    background-color: red;  
  }  
}
```

```
@media only screen and (max-width: 450px) {  
  body {  
    background-color: yellow;  
  }  
}
```

Media Query

- Considere o seguinte HTML que utiliza a media query do slide anterior.
 - O que acontece quando reduzimos o tamanho da janela do browser?
 - O que acontece se trocar o display por uma tela de smartphone?

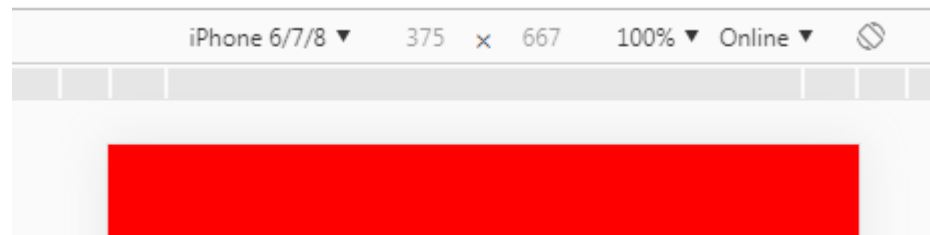
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body></body>
</html>
```

Media Query

- No exemplo anterior, se redimensionarmos a janela para um valor abaixo de 450px a tela fica amarela.
- No entanto, se utilizamos o tamanho de tela de um dispositivo móvel isso não ocorre.
 - Esse comportamento se dá pela diferença entre as resoluções de diferentes telas.
 - 1400px de resolução de um monitor é diferente de uma resolução de 1400px em um smartphone.
 - No smartphone a tela é menor.

Viewport

- Considerar a tela como 1400px em um smartphone para exibir o mesmo conteúdo de um desktop não faz sentido.
- Foi criado então o conceito de device-width que representa um número em pixels que o fabricante do aparelho considera como mais próximo da sensação que o usuário tem ao visualizar a tela.
- No iphone, por exemplo o device-width é de 375 mesmo com a tela tendo uma resolução maior que isso.



Viewport

- Assim, surge a necessidade do conceito de Viewport.
- Viewport, ou “janela de visualização” é a área visível do usuário de uma página da web.
- A Viewport varia de acordo com o dispositivo e será menor em um telefone celular do que na tela do computador.
- O HTML5 introduziu um método para permitir que os web designers assumam o controle sobre a Viewport, por meio da tag `<meta>`.

Viewport

- Devemos incluir o seguinte elemento de viewport `<meta>` em todas as suas páginas da web:

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

- O elemento `<meta>` do tipo **viewport** fornece ao navegador instruções sobre como controlar as dimensões e a escala da página.
- O atributo `width=device-width` define a largura da página para seguir a largura da tela do dispositivo (que variará dependendo do dispositivo).
- O atributo `initial-scale = 1.0` define o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador.

Viewport

- Adicione o Viewport no código de exemplo anterior para analisar seu comportamento ao abrir a página utilizando um dispositivo de tela pequena.

Media Query - Sintaxe

- Uma Media Query é composta por um tipo de mídia opcional e qualquer número de expressões de recurso de mídia.
- Várias consultas podem ser combinadas de várias maneiras, usando operadores lógicos.
- As consultas de mídia não diferenciam maiúsculas de minúsculas.
- Uma consulta de mídia é calculada como true quando o tipo de mídia (se especificado) corresponde ao dispositivo no qual um documento está sendo exibido e todas as expressões de recursos de mídia são computadas como verdadeiras.
- Consultas que envolvem tipos de mídia desconhecidos são sempre falsas.

Media Query – Tipo de mídia

- Os tipos de mídia descrevem a categoria geral de um dispositivo.
 - all: Adequado para todos os dispositivos.
 - print: Destinado a material paginado e documentos visualizados em uma tela no modo de visualização de impressão.
 - screen: Destinado a telas.
 - speech: Destinado a sintetizadores de fala.
- Exceto ao usar operadores lógicos “not” ou “only” lógicos, o tipo de mídia é opcional e o tipo “all” será assumido.

Media Query – Media features

- Media features, ou recursos de mídia, descrevem características específicas do navegador, dispositivo de saída ou ambiente.
- Por exemplo,
 - any-hover testa se o dispositivo de exibição existe algum mecanismo de entrada que possa passar sobre os elementos.
 - orientation testa a orientação (vertical ou horizontal do dispositivo)

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Media Query – Media features

```
@media screen {  
  body {  
    background-color: red;  
  }  
}
```

```
@media only screen and (max-width: 450px) {  
  body {  
    background-color: yellow;  
  }  
}
```

```
@media (orientation: landscape) {  
  body {  
    background-color: green;  
  }  
}
```

Media Query – Operadores Lógicos

- Os operadores lógicos `not`, `and` e `only` podem ser usados para compor uma consulta de mídia complexa
- É possível também combinar várias consultas de mídia em uma única regra, separando-as com vírgulas.
 - `and`: usado para combinar vários recursos de mídia em uma única consulta de mídia, exigindo que cada recurso encadeado retorne `true` para que a consulta seja verdadeira.
 - `not`: usado para negar uma consulta de mídia, retornando `true` se a consulta retornaria `false`. Se você usar o operador `not`, também deverá especificar um tipo de mídia.

Media Query – Operadores Lógicos

- **only**: necessário o uso para impedir que navegadores mais antigos apliquem estilos de maneira inadequada.
- Sem usar o **only**, navegadores mais antigos podem interpretar **screen and (largura máxima: 500px)** simplesmente como **screen**, ignorando o restante da consulta e aplicando seus estilos em todas as telas.

CSS Boxsizing

- Por padrão, a largura e a altura de um elemento HTML são calculadas da seguinte forma:

`width + padding + border = largura real de um elemento`

`height + padding + border = altura real de um elemento`

- Quando você define a largura / altura de um elemento, o elemento geralmente aparece maior do que o definido (porque a borda e o padding do elemento são adicionados à largura / altura especificada do elemento).

CSS Boxsizing

- Esse comportamento do CSS faz com que o desenvolvedor tenha que recalcular sempre o tamanho real de um elemento.
- Como alternativa a essa situação podemos alterar esse comportamento.
- A propriedade de box-sizing permite incluir o padding e a borda na largura e altura total de um elemento.
 - border-box; padding e a borda estão incluídos na largura e altura.

```
* {  
    box-sizing: border-box;  
}
```

CSS Boxsizing

- Como essa propriedade facilita muito a vida dos desenvolvedores, é comum observar em arquivos CSS o uso de `border-box` para todos os elementos.

```
* {  
    box-sizing: border-box;  
}
```

Mobile first

- Mobile First significa projetar para celular antes de projetar para desktop ou qualquer outro dispositivo.
 - Tornará a exibição da página mais rápida em dispositivos menores.
- Em vez de alterar os estilos quando a largura for menor que 768 px, devemos alterar o design quando a largura for maior que 768 px.
- Isso tornará a página Mobile First.

Atividade

- Trabalho 2
- Converta a página HTML do trabalho 1 para Mobile First
- 3,0 Pontos unidade 2
- Enviar vídeo de apresentação no SIGAA
- Entrega 11/11/2022

Bibliografia Básica

- DUCKETT, J. **Javascript e JQuery**: desenvolvimento de interfaces web interativas. Rio de Janeiro: Alta Bokks, 2016
- FLANAGAN, D. **JavaScript**: O Guia Definitivo. Bookman, 2012.
- ROGERS, Y.; SHARP, H.; PREECE, J. **Design de Interação: além da interação homem-computador**. Bookman, 2013.
- MEYER
- Web Content Accessibility Guidelines (WCAG) 2.1. Disponível em: <<https://www.w3.org/TR/WCAG21/>>
- https://www.w3schools.com/css/css_rwd_mediaqueries.asp
- https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Bibliografia Complementar

- AMARAL, L. G. **CSS Cascading Style Sheets**: guia de consulta rápida. 2. ed. São Paulo: Novatec, c2006.
- KAWANO, W. **Crie aplicativos Web com HTML, CSS, JavaScript, PHP, PostgreSQL, Bootstrap, AngularJS e Laravel**. Rio de Janeiro: Ciência Moderna, 2016.
- PUREWAL, S. **Aprendendo a Desenvolver Aplicações Web**. Novatec, 2014.
- TONSIG, S. L. **Aplicações na nuvem**: como construir com HTML5, javaScript, CSS, PHP e MYSQL. Rio de Janeiro: Ciência moderna, 2012.
- USABILIDADE.com. Disponível em <<http://www.usabilidade.com/>>
- TASK-Centered User Interface Design: A Practical Introduction. Disponível em <<http://hcibib.org/tcuid/>>