

Instituto Federal Goiano - Campus Ceres
Bacharelado em Sistemas de Informação
Prof. Me. Ronneesley Moura Teles

Andrey Silva Ribeiro
Jeferson Rossini Ferreira Lourenço

Layout responsivo / adaptivo

Outubro
2017

Sumário

| | | |
|----------|---|----------|
| 1 | Introdução | 2 |
| 2 | Responsive Web Design | 2 |
| 2.1 | Tag viewport | 2 |
| 3 | Já adicionei a meta no meu html, más o layout não está visivelmente elegante, e agora? | 3 |
| 3.1 | Media Query | 3 |
| 3.2 | Media Query | 4 |
| 4 | Conclusão | 5 |
| 5 | Referências | 5 |

Layout responsivo / adaptivo

1 Introdução

Com o crescimento acelerado dos dispositivos móveis que navegam na internet, grande parte da Web ainda não otimizada para esses dispositivos que são limitados pelo tamanho da tela e necessitam de uma abordagem diferente para o layout.

Os tamanhos de tela mudam entre celulares, tablets, desktops e consoles de jogos, logo é importante que seu site possa se adaptar a todos os tamanhos de telas que existem ou possam vir a existir.

2 Responsive Web Design

Originalmente definido por Ethan Marcotte, o Web design responsivo reage às necessidades dos usuários e seus dispositivos, mudando de acordo com o tamanho da tela e seus recursos.

Um exemplo disso é que enquanto um tablet poderia mostrar um conteúdo em duas colunas, em um celular os usuários veriam o mesmo conteúdo em uma só coluna.

Páginas que são otimizadas para reagir a diferentes dispositivos devem incluir uma tag meta viewport no cabeçalho do documento, essa tag instrui o navegador como controlar o tamanho e o dimensionamento da página.

2.1 Tag viewport

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1,
6       user-scalable=no">
7   </head>
8   <body>
9     </body>
10 </html>
```

recursos/exel.html

Use a tag meta viewport para controlar a largura e o dimensionamento da janela de visualização dos navegadores.

Enumerando então fica:

- width=device-width para corresponder à largura da tela em pixels independentes de dispositivo.
- initial-scale=1 para estabelecer uma relação 1:1 entre pixels CSS e pixels independentes de dispositivo.
- user-scalable=no esta opção impede que o usuário consiga ativar o zoom da página em smartphones.

Tentando oferecer uma melhor experiência, navegadores móveis renderizarão a página à largura de uma tela de desktop (geralmente cerca de 980 pixels, mas isso varia de acordo com os dispositivos) e tentarão melhorar a aparência do conteúdo aumentando os tamanhos das fontes e dimensionando o conteúdo para que ele caiba na tela. Isso significa que os tamanhos das fontes podem parecer inconsistentes para os usuários, que precisarão tocar duas vezes ou controlar o zoom com gestos de pinça para ver e interagir com o conteúdo.

Usar o valor `meta viewport width=device-width` instrui a página a acompanhar a largura da tela em pixels independentes de dispositivos. Isso permite que a página ajuste o fluxo do conteúdo para diferentes tamanhos de telas, seja para renderização em pequenos celulares ou para um grande monitor de desktop.

3 Já adicionei a meta no meu html, más o layout não está visivelmente elegante, e agora?

Consultas de mídia nos permitem criar uma experiência responsiva, onde estilos específicos são aplicados a telas pequenas, telas grades e qualquer tela intermediária. A sintaxe da consulta de mídia permite a criação de regras que podem ser aplicadas dependendo das características do dispositivo.

3.1 Media Query

```
@media (query) {  
  /* CSS Rules used when query matches */  
}
```

recursos/exe2.html

Embora existam vários itens para os quais possamos fazer consultas, os mais usados para um Web design responsivo são `min-width`, `max-width`, `min-height` e `max-height`. Além disso, o uso do `min-device-width` pode impedir que o conteúdo se adapte em desktops ou outros dispositivos que permitam que as janelas sejam redimensionadas, pois a consulta é baseada no tamanho do dispositivo, não no da janela do navegador.


| | |
|------------------------|---|
| <code>min-width</code> | Regras aplicadas para qualquer largura de navegador acima do valor definido na consulta. |
| <code>max-width</code> | Regras aplicadas para qualquer largura de navegador abaixo do valor definido na consulta.  |

Figura 1: Media Query. Fonte: <https://developers.google.com/web/fundamentals/design-and-ux/responsive/?hl=pt-br> em 27/10/2017

Vejamos um exemplo:

| | |
|-----------------------|--|
| min-height | Regras aplicadas para qualquer altura de navegador acima do valor definido na consulta. |
| max-height | Regras aplicadas para qualquer altura de navegador abaixo do valor definido na consulta. |
| orientation=portrait | Regras aplicadas a qualquer navegador cuja altura é maior ou igual à largura. |
| orientation=landscape | Regras aplicadas a qualquer navegador cuja largura é maior do que a altura. |

Figura 2: Media Query. Fonte: <https://developers.google.com/web/fundamentals/design-and-ux/responsive/?hl=pt-br> em 27/10/2017



Figura 3: Media Query. Fonte: <https://googlesamples.github.io/web-fundamentals/fundamentals/design-and-ux/responsive/media-queries.html> em 27/10/2017

3.2 Media Query

```

1 <link rel="stylesheet" media="(max-width: 640px)" href="max-640px.css">
2 <link rel="stylesheet" media="(min-width: 640px)" href="min-640px.css">
3 <link rel="stylesheet" media="(orientation: portrait)" href="portrait.
  css">

```

```

<link rel="stylesheet" media="(orientation: landscape)" href="landscape
5  .css">
<style>
  @media (min-width: 500px) and (max-width: 600px) {
7    h1 {
      color: fuchsia;
9    }
    .desc:after {
11     content:" In fact , it 's between 500px and 600px wide.";
    }
13  }
</style>

```

recursos/exe3.html

- Quando o navegador estiver entre 0 e 640 pixels de largura, max-640px.css será aplicado.
- Quando o navegador tiver entre 500 e 600 de largura, os estilos em @media serão aplicados.
- Quando o navegador tiver 640 ou mais pixels de largura, min-640px.css será aplicado.
- Quando a largura do navegador for maior do que sua altura, landscape.css será aplicado.
- Quando a altura do navegador for maior do que sua largura, portrait.css será aplicado.

4 Conclusão

No desenvolvimento de aplicações web atualmente é imprescindível a utilização de layouts somente para navegador, visto que hoje o grande fluxo de acesso provém de dispositivos móveis. A utilização da meta tag viewport e de mídias queries em layouts tornam-os mais flexíveis e adaptativos com fácil implementação e com ótimos resultados.

5 Referências

1. Google. **Google developers** Disponível em: <<https://developers.google.com/web/fundamentals/and-ux/responsive/?hl=pt-br>>. Acesso em: 27 de outubro de 2017.