

1 Responsividade

A responsividade é uma abordagem de design que permite que um site ou aplicativo se ajuste e adapte a diferentes tamanhos de tela e dispositivos. O CSS oferece recursos poderosos para criar layouts responsivos, e uma das técnicas mais comuns usadas para isso é o uso de media queries.

1.1 Media Queries

As media queries são uma funcionalidade do CSS que permitem aplicar estilos diferentes com base nas características do dispositivo ou nas características do ambiente de exibição.

Com as media queries, você pode definir pontos de quebra ou limites em que o layout do seu site ou aplicativo pode ser ajustado automaticamente. A sintaxe básica de uma media query é a seguinte:

Listagem 1: Sintaxe da media query

```
1 @media (condicao) {  
2 /* estilos a serem aplicados quando a condicao for atendida */ 3 }
```

A condição em uma media query pode ser baseada em várias características, como largura e altura da janela, orientação do dispositivo, resolução, tipo de dispositivo e muito mais.

Aqui está um exemplo de uma media query que aplica estilos específicos quando a largura da janela é menor que 600px:

Listagem 2: Uso da media query

```
1 @media (max-width: 600px) {  
2 /* estilos a serem aplicados quando a largura for menor ou igual a 600px */ 3 }  
4  
5 @media (min-width: 600px) {  
6 /* estilos a serem aplicados quando a largura for maior ou igual a 600px */ 7 }
```

Dentro de uma media query, você pode aplicar qualquer estilo CSS válido. Isso inclui alterar tamanhos de fonte, larguras, alturas, margens, alinhamentos e praticamente qualquer outra propriedade de estilo.

Usando media queries, você pode criar um layout flexível que se adapte a diferentes dispositivos e tamanhos de tela. Por exemplo, você pode definir regras de estilo diferentes para dispositivos móveis, tablets e desktops, garantindo uma experiência de usuário otimizada em cada caso.

Aqui está um exemplo de uso das media queries:

Listagem 3: Exemplo de responsividade

```
1 <div class="container">  
2 <h1>Meu Site Responsivo</h1>  
3 <p>Bem-vindo ao meu site!</p>
```

```

4 
5 </div>
6
7 ...
8
9 .container {
10 max-width: 800px;
11 margin: 0 auto;
12 padding: 20px;
13 background-color: #f0f0f0;
14 }
15
16 h1 {
17 font-size: 28px;
18 color: #333;
19 }
20
21 p {
22 font-size: 16px;
23 color: #777;
24 }
25
26 img {
27 width: 100%;
28 height: auto;
29 margin-top: 20px;
30 }
31
32 /* Media Query para telas menores que 600px */
33 @media (max-width: 600px) {
34 h1 {
35 font-size: 24px;
36 }
37
38 p {
39 font-size: 14px;
40 }
41 }

```

Neste exemplo, o cabeçalho (h1) e o parágrafo (p) se adaptam para telas menores que 600px e adquirem seu visual padrão fora desse valor.

1.2 Resolução dos dispositivos

A resolução dos dispositivos desempenha um papel importante na criação de designs responsivos e na utilização de media queries.

A resolução de um dispositivo se refere ao número de pixels que compõem a tela em termos de largura e altura. É um fator-chave a ser considerado ao desenvolver um site ou aplicativo que se adapte a diferentes tamanhos de tela.

Ao projetar para a responsividade, é necessário levar em conta a ampla variedade de resoluções de dispositivos disponíveis atualmente.

Alguns dispositivos têm resoluções muito altas, como monitores de desktop com telas 4K, enquanto outros têm resoluções mais baixas, como dispositivos móveis com telas menores.

A resolução influencia diretamente a forma como o conteúdo é exibido em um dispositivo. Quanto maior a resolução, mais espaço disponível na tela e mais detalhes podem ser mostrados.

Por outro lado, dispositivos com resoluções mais baixas têm menos espaço disponível, o que pode levar a um conteúdo mais compacto. Para lidar com as diferentes resoluções dos dispositivos, as *media queries* são uma ferramenta valiosa. Elas permitem que os desenvolvedores ajustem os estilos com base em pontos de quebra específicos (**breakpoints**), onde a aparência do site ou aplicativo é adaptada para se adequar a uma determinada resolução.

Por exemplo, você pode usar *media queries* para definir diferentes tamanhos de fonte, layouts de grade, tamanhos de imagem e outras propriedades com base nas resoluções do dispositivo.

Isso permite que você otimize a experiência do usuário e garanta que seu conteúdo seja apresentado da melhor maneira possível, independentemente da resolução do dispositivo.

Ao combinar a resolução do dispositivo com *media queries*, você pode criar designs responsivos que se ajustam perfeitamente a diferentes tamanhos de tela. Essa abordagem permite que o conteúdo seja exibido de forma ideal, seja em um grande monitor de desktop, em um tablet ou em um smartphone.

No entanto, é importante notar que a resolução do dispositivo não é o único fator a ser considerado na criação de designs responsivos. Outros aspectos, como a largura e a altura da janela do navegador, também devem ser levados em conta ao aplicar *media queries* para garantir uma experiência responsiva consistente em diferentes cenários de exibição.

Em resumo, a resolução dos dispositivos desempenha um papel crucial na responsividade dos designs. Ao utilizar *media queries* de forma inteligente, os desenvolvedores podem adaptar o layout e os estilos com base na resolução do dispositivo, garantindo uma experiência de usuário otimizada em diferentes tamanhos de tela.

1.3 Outros recursos

Você pode entender mais sobre responsividade e técnicas comuns na [documentação](#) do MDN Web Docs sobre este assunto.

O assunto de responsividade está relacionada ao assunto de layouts, então é muito importante conhecer a tecnologia de layout que você está utilizando, seja flexbox, grid, floats ou outras formas.

2 Bootstrap

O Bootstrap é um framework front-end de código aberto amplamente utilizado para desenvolvimento web responsivo. Ele foi criado pela equipe do Twitter e lançado em 2011. O objetivo principal do Bootstrap é fornecer uma estrutura consistente e eficiente para a criação de interfaces de usuário modernas e responsivas.

O Bootstrap é baseado em HTML, CSS e JavaScript e oferece uma coleção extensa de estilos CSS pré-definidos, componentes reutilizáveis e utilitários que facilitam a criação de layouts responsivos e a implementação de funcionalidades interativas.

2.1 Aplicando Bootstrap no site

Existem várias formas de aplicar o bootstrap em nossos sites, a forma mais comum é usando CDN (Content Delivery Network, ou Rede de Fornecimento de Conteúdo) que são serviços especializados em disponibilizar recursos para os websites.

São estáveis e garantem que o seu site não ficará sem estilização caso não seja interessante para o seu website hospedar localmente o bootstrap. Para aplicar o Bootstrap de maneira rápida ao seu site, basta usar o seguinte código:

Listagem 4: Tags link e script para o Bootstrap

```

1 <!doctype html>
2 <html lang="pt-br">
3 <head>
4 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
    rel="stylesheet" integrity="sha384-9
    ndCyUalbAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
    crossorigin="anonymous">
5 </head>
6 <body>
7 ...
8 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
    geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68Sly3Te
    4Bkz" crossorigin="anonymous"></script>
9 </body>
10 </html>

```

Na linha 4 temos todo o CSS que o Bootstrap possui, é um arquivo num formato minificado e de certa ilegível, porém continua sendo uma folha de estilos válida e com diversos recursos úteis para o site.

Imagine como se você tivesse escrito CSS, hospedado em outro site, agora está linkando ele a partir de uma página HTML, o CSS não está na sua máquina, mas ele existe em outro serviço, aqui é a mesma ideia.

Na linha 8 temos a tag `<script>` que é essencial para o bom funcionamento de alguns componentes que possuem comportamento dinâmico, como drop downs, modais e etc.

4

Unidade VI: Responsividade e Bootstrap soulcode.com

Apenas estas duas linhas são fundamentais para que o site apresente uma estilização bem diferente da habitual. Para mais informações sobre instalações, consulte a [documentação oficial](#) do framework.

2.2 Reboot

Para que o Bootstrap funcione tão bem e seja consistente, ele possui diversas regras CSS padrões aplicadas aos mais diversos elementos. Por exemplo, por padrão *box-sizing* é definido como *border-box*.

Chamamos esta técnica de definir regras padrões iniciais de **normalização**. O Bootstrap tem por base esta técnica, por isso ele consegue ser muito consistente entre os diversos navegadores existentes.

2.3 Componentes

Esta seção tem por objetivo mostrar o uso de alguns componentes básicos do Bootstrap. Porém é consenso que a documentação do próprio framework é didática e demonstrativa o bastante.

Para conhecer todos os componentes e possibilidades do Bootstrap, consulte a [documentação](#).

2.3.1 Buttons

Os [botões](#) customizados do Bootstrap são muito úteis e ricos visualmente.

Figura 1: Buttons



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Fonte: Bootstrap Docs.

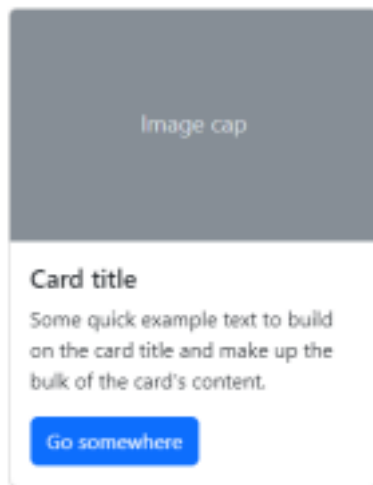
5

Unidade VI: Responsividade e Bootstrap soulcode.com

2.3.2 Cards

Os [cards](#) oferecem uma maneira muito bem visual e estruturada de mostrar itens.

Figura 2: Cards



```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

Fonte: Bootstrap Docs.

2.3.3 Dropdown

Os **dropdowns** são úteis para dispôr várias ações relacionadas.

Figura 3: Dropdown

Dropdown button ▾

```
<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle" type="button" data-bs-toggle="dropdown" aria-expanded="false">
    Dropdown button
  </button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</div>
```

Fonte: Bootstrap Docs.

2.4 Como aprender Bootstrap?

Você pode ficar surpreso com a riqueza de recursos do framework, e é normal se sentir intimidado quando se está começando. Poderíamos propor cada um dos recursos do Bootstrap nesse material, porém as documentações sempre estão em constantes mudanças e melhoria, o que tornaria em questão de meses este material desatualizado.

Recomendamos que favorite a [documentação](#) do Bootstrap e consulte sempre que quiser aplicar algum novo componente. Ela oferece exemplos e explicações muito didáticas que podem ajudar bastante no uso diário da ferramenta.