

UNIVERSIDADE ESTÁCIO DE SÁ

Web – Estácio Comparações

Manaus - AM
2020

Universidade Estácio de Sá

Desenvolvedores

ERIC NEVES RODRIGUES

DIEGO FRANCO DA SILVA

NATANAEL XAVIER DE SOUZA

LEONARDO DA SILVA XISTO

THIAGO DANIEL CASTRO DA FONSECA

SUZANA COSTA DE MOURA

Manaus - AM
2020

DESIGN RESPONSIVO

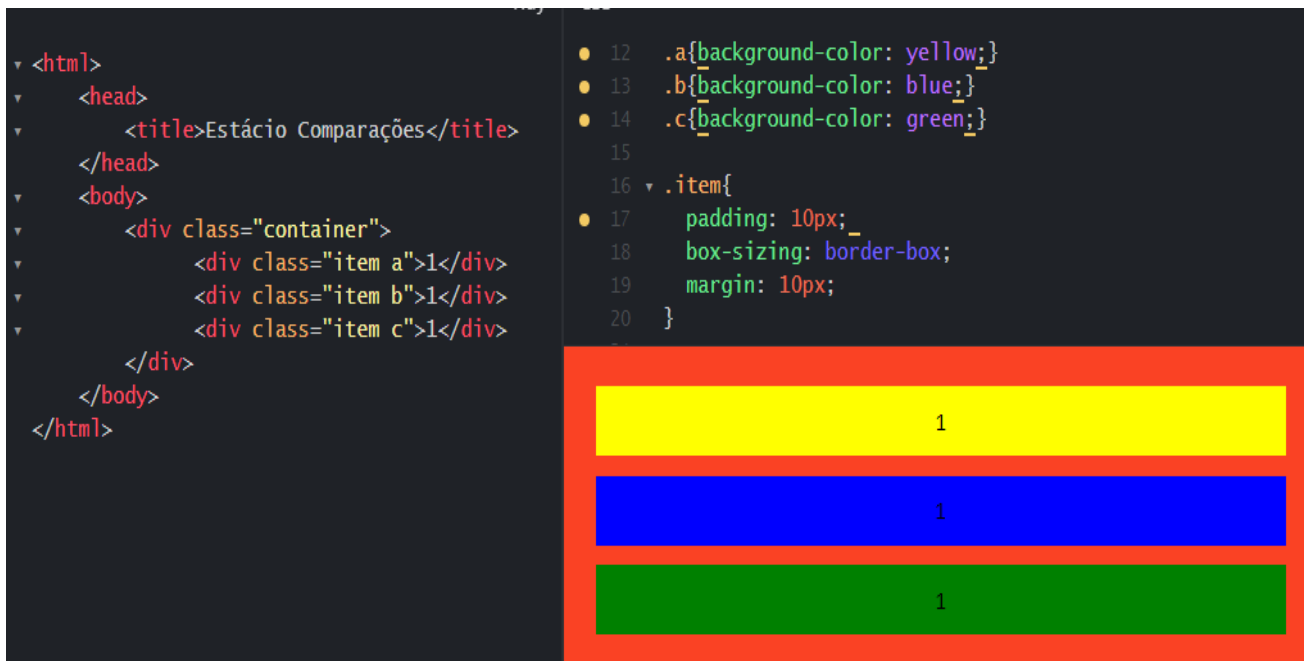
Sabe-se, que, atualmente, para desenvolver um web site, é preciso listar alguns requisitos para a criação de tal, dentre elas, o público alvo, dispositivos que irão acessar o site, tecnologias de desenvolvimento e entre outros. Durante o desenvolvimento do site [Estácio Comparações](#), foram vistos alguns desafios, como por exemplo: “Como deixar o site responsivo?”, após algumas pesquisas, foram listados duas tecnologias do **CSS3**, o FlexBox e o Media Query, com base nisso, o projeto foi todo alterado, trazendo novas funcionalidades e o principal, a responsividade.

O QUE É FLEXBOX?

CSS Flexible Box Layout ou simplesmente FlexBox, é modelo de layout flexível do CSS3, esse recurso faz com que, os elementos da **DOM** (Document Object Model), sejam organizados automaticamente dentro de seus containers, dependendo do tamanho da tela do dispositivo, reorganizando-se conforme a necessidade.

COMO USAR O FLEXBOX?

Usar o FlexBox é bastante fácil, vamos de um exemplo simples, primeiro, será criado um “**container**”, sendo representado por uma tag do HTML5 chamada “**div**”, depois definiremos uma classe para essa **div**, chamada de **container**, após isso, dentro desse **container** ficarão todos os componentes flexíveis, como no exemplo abaixo.



Três tags “**div**” foram adicionadas dentro do **container**, nesse exemplo, serão os componentes flexíveis do **container**, após isso, é preciso chamar a classe da **div** principal, nesse caso “**.container**” e definir as configurações.



Com esses exemplos, é possível entender o funcionamento básico do FlexBox, um recurso muito poderoso do CSS3.

O QUE É MEDIA QUERY?

O **Media Query** é um recurso muito utilizado do CSS3, seu uso está alicerçado no desenvolvimento de aplicações responsivas para web, esta técnica, consiste na adaptação do conteúdo da **DOM** (Document Object Model), em várias condições, como, por exemplo, no tamanho da resolução. É a tecnologia fundamental para o desenvolvimento de design responsivo.

O exemplo abaixo, consiste em trocar a cor de fundo da **DOM**, quando a tela estiver numa resolução **menor** do que **550px**.



A cor de fundo foi definida como hexadecimal “**#FA4224**”, após isso, o **Media Query** entra em ação, vamos utilizá-lo com uma estrutura condicional.



Com a resolução de **558px** a **DOM** continua com a cor de fundo “**#FA4224**”, agora, veremos numa resolução inferior.



Após a resolução ser definida menor do que **550px**, a mágica simplesmente acontece, e voa lá, a interface entra dentro da condição e altera as configurações definidas. Esses e muitos outros conceitos são aplicados no dia a dia para o desenvolvimento de Design Responsivo.

Validações em Javascript

Javascript está em cada canto da web, é uma das tecnologias mais populares entre os programadores, no projeto [Estácio Comparações](#), de início, a linguagem foi usada para fazer a validação de formulários somente no front-end, não há mistério em validar dados no front, há diversas tecnologias e frameworks modernos para isso, mas, nesse caso foi usado JS puro, validando alguns dados.

HTML ▼

```
1 <html>
2   <head>
3     <title>Estácio Comparações</title>
4   </head>
5   <body>
6     <input type="text" id="email"><br><br>
7     <button id="btn">Validar</button><br>
8     <div id="result"></div>
9   </body>
10 </html>
```

CSS ▼

```
1 #email, #btn, #result{
2   font-size: 22pt;
3 }
4
```

JavaScript + No-Library (pure JS) ▼

```
1 window.onload = function(){
2   let btn = document.getElementById('btn')
3   let result = document.getElementById('result')
4
5   btn.onclick = function(){
6     let email = document.getElementById('email').value
7     // verificando se há o '@'
8     if (email.indexOf('@') !== -1){
9       result.innerHTML = 'E-mail válido'
10    }else {
11      result.innerHTML = 'E-mail Inválido'
12    }
13  }
14 }
```

E-mail Inválido

HTML ▼

```
1 <html>
2   <head>
3     <title>Estácio Comparações</title>
4   </head>
5   <body>
6     <input type="text" id="email"><br><br>
7     <button id="btn">Validar</button><br>
8     <div id="result"></div>
9   </body>
10 </html>
```

CSS ▼

```
1 #email, #btn, #result{
2   font-size: 22pt;
3 }
4
```

JavaScript + No-Library (pure JS) ▼

```
1 window.onload = function(){
2   let btn = document.getElementById('btn')
3   let result = document.getElementById('result')
4
5   btn.onclick = function(){
6     let email = document.getElementById('email').value
7     // verificando se há o '@'
8     if (email.indexOf('@') !== -1){
9       result.innerHTML = 'E-mail válido'
10    }else {
11      result.innerHTML = 'E-mail Inválido'
12    }
13  }
14 }
```

E-mail Válido

No exemplo acima, validamos o formulário simples, verificando se há um e-mail válido, não é um filtro muito avançado, por não haver interação com o back-end da aplicação, já que ambos os lados devem filtrar muito bem os dados inseridos pelo o usuário.

Link do projeto: <http://appestacio.herokuapp.com/>

Referências

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<http://devfuria.com.br/html-css/media-queries/>