

HTML5, CSS3 E JAVASCRIPT

JEMERSON FERNANDO MAIA

**PÓS-GRADUAÇÃO EM DESENVOLVIMENTO DE
APLICAÇÕES WEB E MOBILE**

UNIPAR – UNIVERSIDADE PARANAENSE

EMENTA

HTML5 e CSS3; Fundamentos de HTML5 e CSS3; Responsividade com Media Queries; Seletores; SASS; UX; Javascript; ES6;

AGENDA

27/03: HTML5, Modernizr, HTML5 Boilerplate, Formulários e Validações, CSS3, Normalizer.css, Media Query, Framework Fron-End Bootstrap

10/04: SASS, JS, ES6, 2016, 2017, 2018, JQuery

24/04: API HTML5, UX, Acessibilidade, Testes de Responsividade, Performance, Admin Templates

Ferramentas

- Sass
 - <https://sass-lang.com/>

Referências

<https://www.w3schools.com>

<https://sass-lang.com/documentation>

https://www.w3schools.com/js/js_es6.asp

Sass

Syntactically Awesome StyleSheets

Sass

É um pré-processador CSS.

É uma extensão do CSS que adiciona potência e elegância ao CSS nativo.

[Less](#) e [Stylus](#)

SCSS

```
section {  
  height: 100px;  
  width: 100px;  
  
  .class-one {  
    height: 50px;  
    width: 50px;  
  
    .button {  
      color: #074e68;  
    }  
  }  
}
```

CSS

```
1 section {  
2     height: 100px;  
3     width: 100px;  
4 }  
5  
6 section .class-one {  
7     height: 50px;  
8     width: 50px;  
9 }  
10  
11 section .class-one .button {  
12     color: #074e68;  
13 }
```



```
$font-stack: Helvetica, sans-serif
$primary-color: #333
```

```
body
  font: 100% $font-stack
  color: $primary-color
```

sass

```
$font-stack: Helvetica, sans-serif;
$primary-color: #333;
```

```
body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

SCSS

Arquivo .sass vs .scss

Instalação

- <https://github.com/sass/dart-sass/releases>, (add PATH)
- npm install -g sass (node.js)
- choco install sass (chocolatey – Windows)
- brew install sass/sass/sass (mac)
- <https://scout-app.io/> (App)

- [Live Sass Compiler](#) (VSCode)
- Ruby (gem install sass)

<https://sass-lang.com/install>

sass --version

Execução

```
sass source/stylesheets/index.scss build/stylesheets/index.css
```

```
sass --watch source/stylesheets/index.scss: build/stylesheets/index.css
```

```
sass --watch app/sass:public/stylesheets
```

```
$ sass --style=nested
```

```
h1 {  
  font-size: 40px; }  
h1 code {  
  font-face: Roboto Mono; }
```

```
$ sass --style=expanded style.scss
```

```
h1 {  
  font-size: 40px;  
}  
h1 code {  
  font-face: Roboto Mono;  
}
```

```
$ sass --style=compact style.scss
```

```
h1 { font-size: 40px; }  
h1 code { font-face: Roboto Mono; }
```

```
$ sass --style=compressed style.scss
```

```
h1{font-size:40px}h1 code{font-face:Roboto Mono}
```

Saídas

Variáveis

Permite definir valores de propriedades de forma dinâmica.

Variáveis

```
$myFont: Helvetica, sans-serif;
$myColor: red;
$myFontSize: 18px;
$myWidth: 680px;

body {
  font-family: $myFont;
  font-size: $myFontSize;
  color: $myColor;
}

#container {
  width: $myWidth;
}
```

```
body {
  font-family: Helvetica, sans-serif;
  font-size: 18px;
  color: red;
}

#container {
  width: 680px;
}
```

Variáveis- Escopos

```
$myColor: red;
```

```
h1 {  
  $myColor: green;  
  color: $myColor;  
}
```

```
p {  
  color: $myColor;  
}
```

```
h1 {  
  color: green;  
}
```

```
p {  
  color: red;  
}
```

Aninhamentos

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  li {  
    display: inline-block;  
  }  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```


Aninhamentos em propriedades

```
font: {  
  family: Helvetica, sans-serif;  
  size: 18px;  
  weight: bold;  
}
```

```
text: {  
  align: center;  
  transform: lowercase;  
  overflow: hidden;  
}
```

```
font-family: Helvetica, sans-serif;  
font-size: 18px;  
font-weight: bold;
```

```
text-align: center;  
text-transform: lowercase;  
text-overflow: hidden;
```

Aninhamentos em propriedades

```
span{
  text:{
    align:center;
    indent: 20px;
  }

  font:{
    family: 'Helvetica';
    size: 30px;
  }

  border:{
    top:{
      style:dashed;
      left:{
        radius: 10px;
      }
    }
  }
}
```

```
span {
  text-align: center;
  text-indent: 20px;
  font-family: 'Helvetica';
  font-size: 30px;
  border-top-style: dashed;
  border-top-left-radius: 10px; }
```

@Mixin

Permitem a construção de blocos de código independentes, permitindo misturá-los para criar uma grande folha de estilo

@Mixin

```
@mixin important-text {  
  color: red;  
  font-size: 25px;  
  font-weight: bold;  
  border: 1px solid blue;  
}  
  
.danger {  
  @include important-text;  
  background-color: green;  
}
```

```
.danger {  
  color: red;  
  font-size: 25px;  
  font-weight: bold;  
  border: 1px solid blue;  
  background-color: green;  
}
```

@Mixin – Passando Parâmetros

```
@mixin bordered($color, $width) {  
  border: $width solid $color;  
}
```

```
.myArticle {  
  @include bordered(blue, 1px); // Call mixin with two values  
}
```

```
.myNotes {  
  @include bordered(red, 2px); // Call mixin with two values  
}
```

```
.myArticle {  
  border: 1px solid blue;  
}
```

```
.myNotes {  
  border: 2px solid red;  
}
```

@Mixin – Uso para Prefixos

```
@mixin transform($property) {  
  -webkit-transform: $property;  
  -ms-transform: $property;  
  transform: $property;  
}  
  
.myBox {  
  @include transform(rotate(20deg));  
}
```

```
.myBox {  
  -webkit-transform: rotate(20deg);  
  -ms-transform: rotate(20deg);  
  transform: rotate(20deg);  
}
```

@extend

Permite que um seletor herde propriedades e valores declarados em outro

@extend

```
.button-basic {
  border: none;
  padding: 15px 30px;
  text-align: center;
  font-size: 16px;
  cursor: pointer;
}

.button-report {
  @extend .button-basic;
  background-color: red;
}

.button-submit {
  @extend .button-basic;
  background-color: green;
  color: white;
}
```

```
.button-basic, .button-report, .button-submit {
  border: none;
  padding: 15px 30px;
  text-align: center;
  font-size: 16px;
  cursor: pointer;
}

.button-report {
  background-color: red;
}

.button-submit {
  background-color: green;
  color: white;
}
```


@import

Permite particionar a formatação CSS em arquivos para reaproveita-los em vários arquivos.

@import

_reset.scss

```
html,  
body,  
ul,  
ol {  
  margin: 0;  
  padding: 0;  
}
```

```
@import "reset";  
  
body {  
  font-family: Helvetica, sans-serif;  
  font-size: 18px;  
  color: red;  
}
```

```
html, body, ul, ol {  
  margin: 0;  
  padding: 0;  
}  
  
body {  
  font-family: Helvetica, sans-serif;  
  font-size: 18px;  
  color: red;  
}
```

@import

Reset/reboot

variáveis

cores

fontes

Componentes

Módulos

Vendors

Páginas



Exemplos

Atividade



Sass- Módulos Integrados

Sass provê muitos módulos integrados que contém funções úteis.

<https://sass-lang.com/documentation/modules>

Sass- Regras de controle de Fluxo

Com Sass é possível controlar se estilos são gerados ou gera-los várias vezes com variações. Permite criar pequenos algoritmos para gerar o estilo css.

<https://sass-lang.com/documentation/at-rules/control>

Sass- Compass

O Compass fornece funções extras de cores e uma série de mixins como reset, clear-fix, truncate e montagem de gradientes.

<http://compass-style.org/>

JavaScript

Permite interatividade com conteúdo.

Como usar

- Interna
 - Tag script dentro do documento
 - Geralmente dentro da tag head ou como último elemento dentro da tag body

Como usar

```
<!DOCTYPE html>
<html>

<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</body>
</html>
```

Como usar

- Externa
 - Arquivo externo ao html
 - Arquivo com extensão js

```
<script src="myScript1.js"></script>  
<script src="myScript2.js"></script>
```

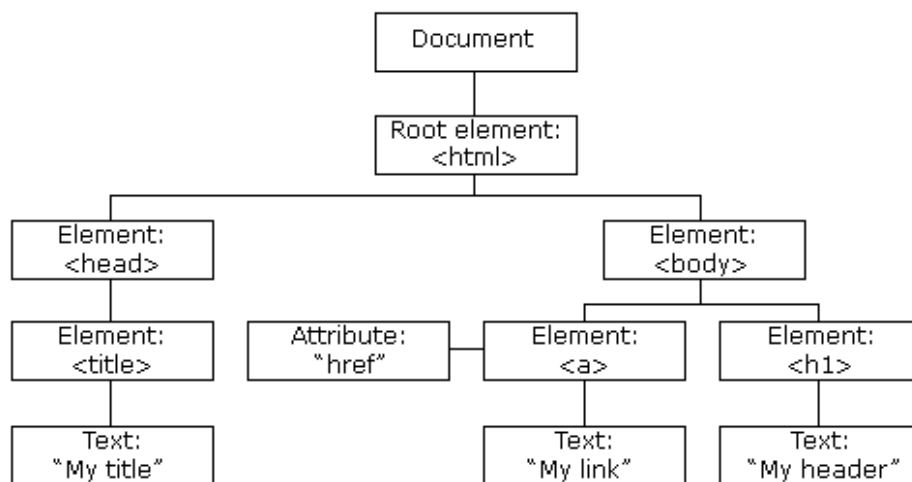
```
<script src="/js/myScript1.js"></script>
```

```
<script src="https://www.w3schools.com/js/myScript1.js"></script>
```

```
function myFunction() {  
  document.getElementById("demo").innerHTML = "Paragraph changed."  
}
```

Manipulação de HTML

The HTML DOM Tree of Objects



- Elementos HTML como objetos
- Propriedades de todos os elementos
- Métodos para acessar os elementos
- Eventos para todos os elementos

Manipulação de HTML

Finding HTML Elements

| Method | Description |
|---|-------------------------------|
| <code>document.getElementById(<i>id</i>)</code> | Find an element by element id |
| <code>document.getElementsByTagName(<i>name</i>)</code> | Find elements by tag name |
| <code>document.getElementsByClassName(<i>name</i>)</code> | Find elements by class name |

Manipulação de HTML

```
var x = document.getElementById("main");  
var y = x.getElementsByTagName("p");
```

```
var x = document.getElementsByClassName("intro");
```

es5

```
var x = document.querySelectorAll("p.intro");
```

Manipulação de HTML

Changing HTML Elements

| Property | Description |
|---|---|
| <code>element.innerHTML = new html content</code> | Change the inner HTML of an element |
| <code>element.attribute = new value</code> | Change the attribute value of an HTML element |
| <code>element.style.property = new style</code> | Change the style of an HTML element |
| Method | Description |
| <code>element.setAttribute(attribute, value)</code> | Change the attribute value of an HTML element |

Manipulação de HTML

Adding and Deleting Elements

| Method | Description |
|--|-----------------------------------|
| <code>document.createElement(<i>element</i>)</code> | Create an HTML element |
| <code>document.removeChild(<i>element</i>)</code> | Remove an HTML element |
| <code>document.appendChild(<i>element</i>)</code> | Add an HTML element |
| <code>document.replaceChild(<i>new</i>, <i>old</i>)</code> | Replace an HTML element |
| <code>document.write(<i>text</i>)</code> | Write into the HTML output stream |

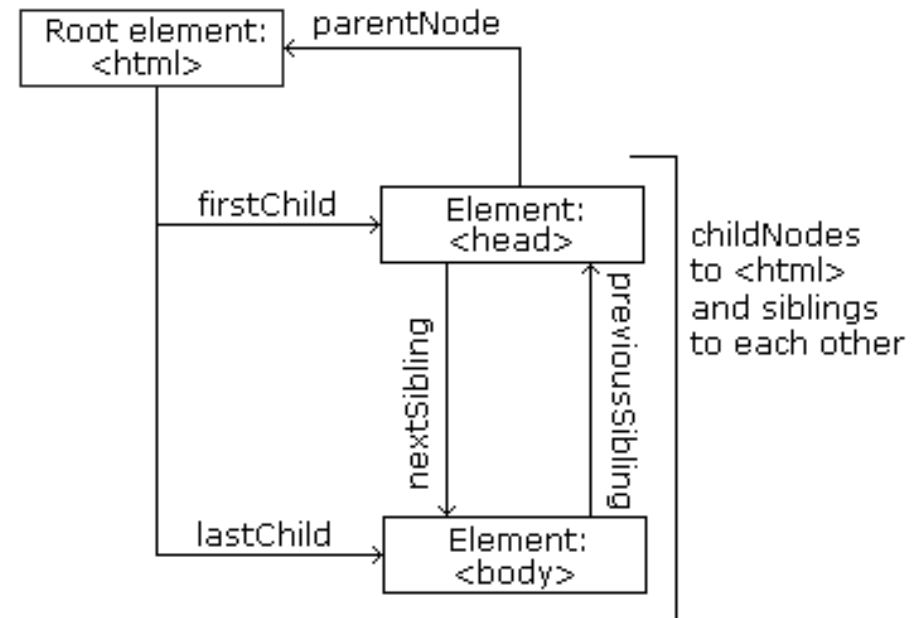
Manipulação de HTML

```
<html>

<head>
  <title>DOM Tutorial</title>
</head>

<body>
  <h1>DOM Lesson one</h1>
  <p>Hello world!</p>
</body>

</html>
```



Manipulação de HTML

Navigating Between Nodes

You can use the following node properties to navigate between nodes with JavaScript:

- `parentNode`
- `childNodes[nodenum]`
- `firstChild`
- `lastChild`
- `nextSibling`
- `previousSibling`

Manipulação de HTML- Eventos

- JavaScript podem ser executados quando eventos acontecem, exemplos:
 - Quando usuário clica com mouse
 - Quando uma página é carregada
 - Quando uma imagem é carregada
 - Quando mouse é movido sobre elementos
 - Quando campos de entrada em formulários são alterados
 - Quando formulários tem seus dados submetidos
 - Quando usuário pressiona uma tecla

Variáveis e Saídas

```
var x = 5;  
var y = 6;  
var z = x + y;
```

```
<script>  
document.getElementById("demo").innerHTML = 5 + 6;  
</script>
```

```
<script>  
document.write(5 + 6);  
</script>
```

```
<script>  
window.alert(5 + 6);  
</script>
```

```
<script>  
console.log(5 + 6);  
</script>
```



Exemplos

JQuery

É uma biblioteca que simplifica a programação em JS.

“Escrever menos, fazer mais”

Como usar

<https://jquery.com/>

```
<head>  
<script src="jquery-3.5.1.min.js"></script>  
</head>
```

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
</head>
```

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
<script src="my_jquery_functions.js"></script>  
</head>
```


Sintaxe

\$(selector).action()

```
$(document).ready(function() {  
    $("p").click(function() {  
        $(this).hide();  
    });  
});
```

Sintaxe- Seletores

```
$ ("p")
```

```
$ ("#test")
```

```
$ (".test")
```

Eventos

```
$("#p").click(function() {  
    $(this).hide();  
});
```

| Mouse Events | Keyboard Events | Form Events | Document/Window Events |
|--------------|-----------------|-------------|------------------------|
| click | keypress | submit | load |
| dblclick | keydown | change | resize |
| mouseenter | keyup | focus | scroll |
| mouseleave | | blur | unload |

text(), html(), val() e attr()

- text() – define ou retorna os texto contido nos elementos selecionados
- html() – define ou retorna o conteúdo, incluindo marcações html, dos elementos selecionados
- val() – define ou retorna valor de campos selecionados de formulários
- attr() – define ou retorna valor de atributos dos elementos selecionados

```
alert("Text: " + $("#test").text());
```

```
alert("HTML: " + $("#test").html());
```

```
alert("Value: " + $("#test").val());
```

```
alert($("#w3s").attr("href"));
```

Manipulação HTML- Adicionar

- `append()` - Insere conteúdo no fim do elemento selecionado
- `prepend()` - Insere conteúdo no começo do elemento selecionado
- `after()` - Insere conteúdo depois do elemento selecionado
- `before()` - Insere conteúdo antes do elemento selecionado

```
function appendText() {  
    var txt1 = "<p>Text.</p>"; // Create element with HTML  
    var txt2 = $("<p></p>").text("Text."); // Create with jQuery  
    var txt3 = document.createElement("p"); // Create with DOM  
    txt3.innerHTML = "Text.";  
    $("body").append(txt1, txt2, txt3); // Append the new elements  
}
```

Manipulação HTML- Remove

- `remove()` - Remove o elemento selecionado (e seus filhos)
- `empty()` - Remove os filhos do elemento selecionado

```
$("p").remove(".test, .demo");
```

Manipulação HTML- CSS

- `addClass()` - Adiciona uma ou mais classes no elemento selecionado
- `removeClass()` - Remove uma ou mais classes do elemento selecionado
- `toggleClass()` - Alterna entre adicionar e remover classes no elemento selecionado
- `css()` - Adiciona ou recupera atributos de estilo

```
$("#p").css("background-color");
```

```
$("#p").css("background-color", "yellow");
```

```
$("#button").click(function() {  
    $("#h1, h2, p").addClass("blue");  
    $("#div").addClass("important");  
});
```

Manipulação HTML- Ancestrais

- `parent()` – Retorna o elemento pai do elemento selecionado
- `parents()` – Retorna todos os elementos ancestrais do elemento selecionado
- `parentsUntil()` – Retorna todos os elementos ancestrais entre dois elementos (selecionado e outro)

```
$(document).ready(function(){  
    $("span").parentsUntil("div");  
});
```


Manipulação HTML- Descendentes

- `children()` – Retorna todos os filhos diretos do elemento selecionado
- `find()` – Retorna todos os elementos descendentes do elemento selecionado

```
$(document).ready(function() {  
    $("div").children("p.first");  
});
```

```
$(document).ready(function() {  
    $("div").find("span");  
});
```

Manipulação HTML- Irmãos

- `siblings()` – Retorna todos os irmãos do elemento selecionado
- `next()` – Retorna o próximo elemento irmão do elemento selecionado
- `nextAll()` – Retorna todos os próximos elementos irmãos do elemento selecionado
- `nextUntil()` – Retorna todos os próximos elementos irmãos entre dois elementos
- `prev()` – Retorna o elemento anterior irmão do elemento selecionado
- `prevAll()` – Retorna todos os elementos anteriores irmãos do elemento selecionado
- `prevUntil()` – Retorna todos os elementos anteriores irmãos entre dois elementos

Manipulação HTML- Filtrar

- first() – Filtra retornando o primeiro elemento informado
- last() – Filtra retornando o último elemento informado
- eq() – Filtra retornando o elemento do índice informado
- filter() – Filtra retornando os elementos encontrados de acordo com um critério de busca
- not() – Filtra retornando os elementos que não satisfazem o critério de busca

```
$(document).ready(function(){  
    $("p").filter(".intro");  
});
```



Exemplos

Atividade

- Adicionar coluna na tabela contendo link em cada linha que ao clicar remove a linha toda da tabela;

ES6 – ECMAScript 6 ou ES2015

O ECMAScript (ES) é a especificação da linguagem de script que o JavaScript implementa.

<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>