



ESCOLA DE TECNOLOGIA DA INFORMAÇÃO
DESENVOLVIMENTO WEB BÁSICO

Visão geral do Document Object Model (DOM)

Geucimar Briatore
geucimar@up.edu.br

Atualizado em 05/2022

Vanilla.js

Introduction

The *Vanilla JS* team maintains every byte of code in the framework and works hard each day to make sure it is small and intuitive. Who's using *Vanilla JS*? Glad you asked! Here are a few:

Facebook Google YouTube Yahoo Wikipedia Windows Live Twitter Amazon LinkedIn MSN
eBay Microsoft Tumblr Apple Pinterest PayPal Reddit Netflix Stack Overflow

In fact, *Vanilla JS* is already used on more websites than jQuery, Prototype JS, MooTools, YUI, and Google Web Toolkit - *combined*.

Download

Ready to try *Vanilla JS*? Choose exactly what you need!

<input checked="" type="checkbox"/> Core Functionality	<input type="checkbox"/> DOM (Traversal / Selectors)
<input type="checkbox"/> Prototype-based Object System	<input checked="" type="checkbox"/> AJAX
<input type="checkbox"/> Animations	<input type="checkbox"/> Event System
<input type="checkbox"/> Regular Expressions	<input type="checkbox"/> Functions as first-class objects
<input type="checkbox"/> Closures	<input type="checkbox"/> Math Library
<input type="checkbox"/> Array Library	<input type="checkbox"/> String Library

Options

<input type="checkbox"/> Minify Source Code	<input type="checkbox"/> Produce UTF8 Output
<input type="checkbox"/> Use "CRLF" line breaks (Windows)	

Final size: 0.00 KBytes uncompressed, 0.02 KBytes gzipped. ☒ Show human-readable sizes

Download

<http://vanilla-js.com>

Tópicos da linguagem de programação Javascript

- ~~Introdução ao Javascript;~~
- Visão geral do Document Object Model (DOM);
- Ajax e consumo de webservices;
- SPAs e frameworks.

Diferentes linguagens do desenvolvimento web

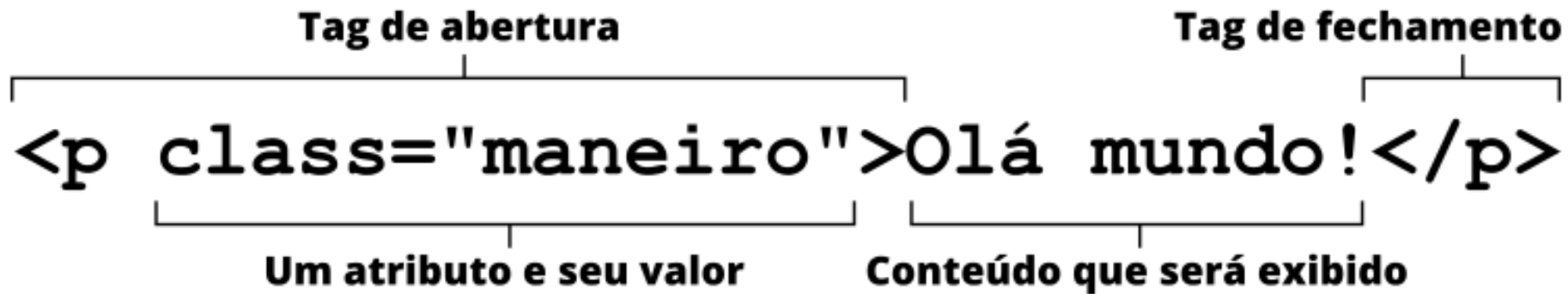
RECURSOS USADOS NA CRIAÇÃO DE PÁGINAS WEB

HTML: Linguagem de marcação utilizada para organizar o conteúdo das páginas.

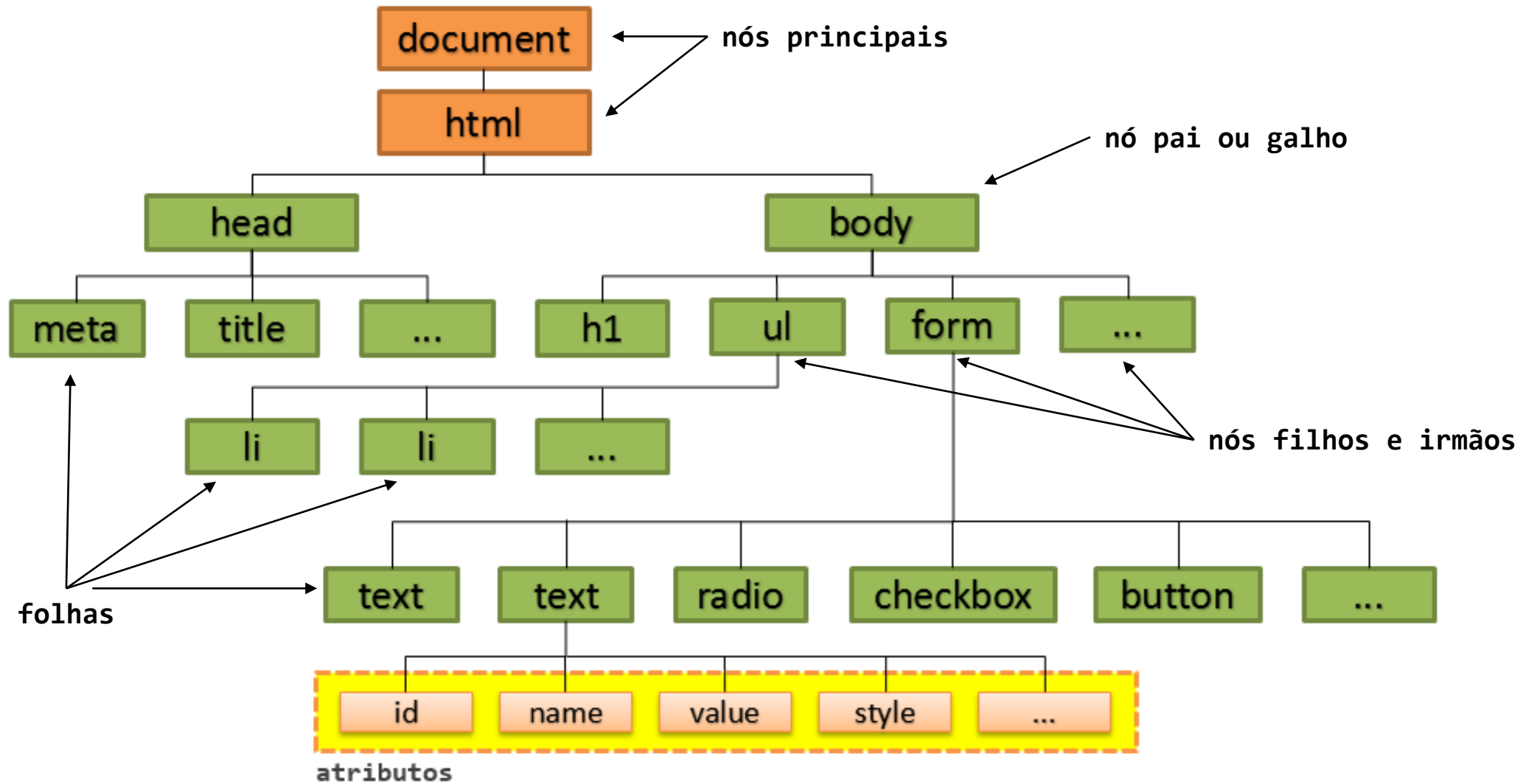
CSS: Linguagem de formatação utilizada para cuidar da apresentação do conteúdo.

JavaScript: Linguagem de programação utilizada para manipular os elementos e eventos das páginas.

Anatomia de um elemento HTML

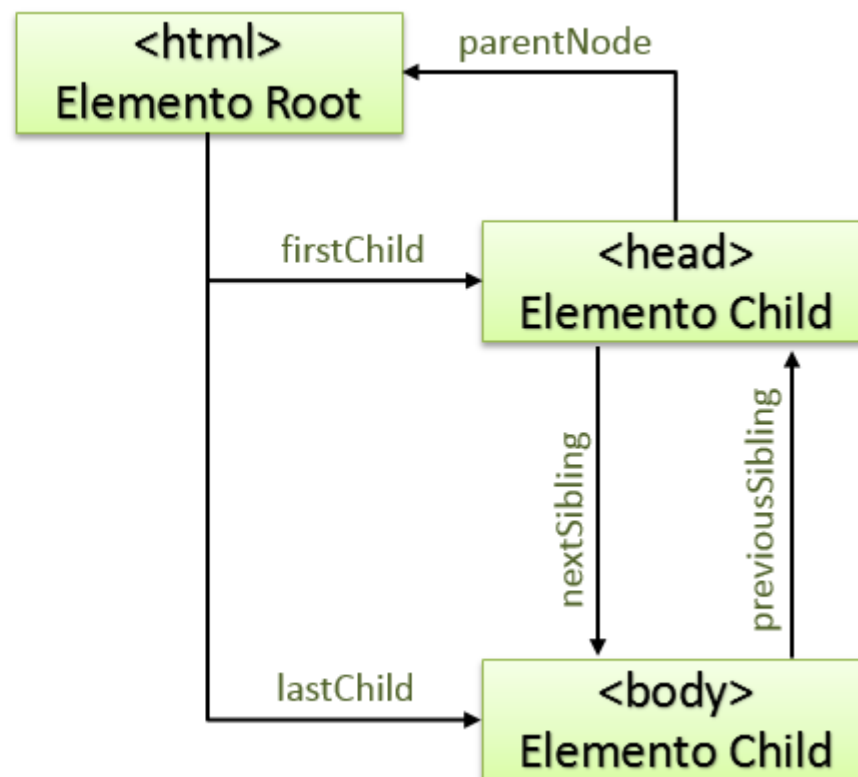


Document Object Model (DOM)



Relação entre os elementos (nós) no DOM

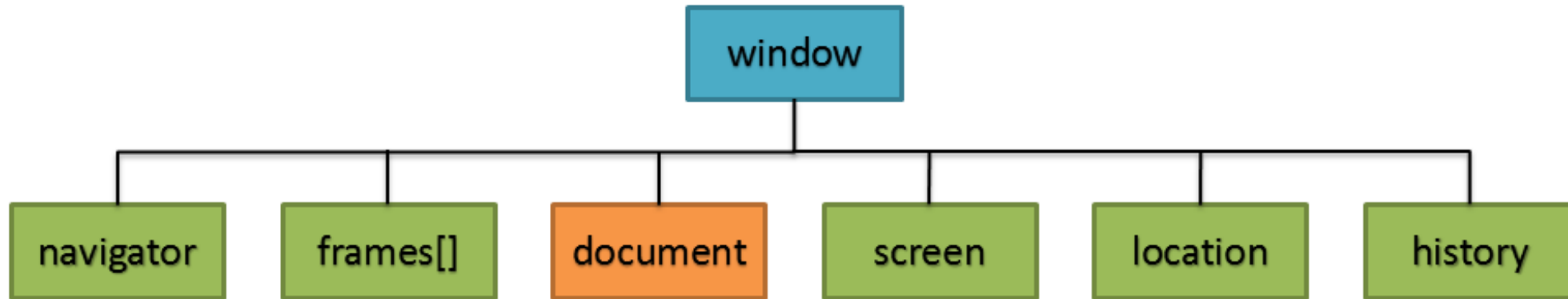
```
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body>
    <h1>Título</h1>
    <p>Texto</p>
  </body>
</html>
```



Funções básicas do DOM

```
document.writeln();           // Escreve no documento.
document.getElementById();    // Busca pelo ID.
document.getElementsByTagName(); // Busca pela Tag.
document.getElementsByClassName(); // Busca pela Class.
document.getElementsByName(); // Busca pelo Name.
document.createElement();     // Cria um elemento HTML.
document.removeChild();       // Remove um elemento.
document.appendChild();       // Adiciona um elemento.
document.replaceChild();      // Substitui um elemento.
document.querySelector();     // Primeiro elemento.
document.querySelectorAll();  // Lista de elementos.
```


Browser Object Model (BOM)



```
window.navigator; // Informações sobre o navegador.  
window.frames;    // Quantidade de frames na página.  
window.document;  // Document Object Model (DOM).  
window.screen;    // Informações sobre a tela do usuário.  
window.location;  // Define a página de navegação.  
window.history;    // Controla o histórico de navegação.
```

Funções mais comuns do objeto window

<code>window.alert();</code>	<code>// Informações ao usuário.</code>
<code>window.confirm();</code>	<code>// Confirmação Sim ou Não.</code>
<code>window.prompt();</code>	<code>// Permite entrada ao usuário.</code>
<code>window.open();</code>	<code>// Abre uma nova janela.</code>
<code>window.close();</code>	<code>// Fecha a janela corrente.</code>
<code>window.moveTo();</code>	<code>// Move a janela corrente.</code>
<code>window.resizeTo();</code>	<code>// Dimensiona a janela corrente.</code>
<code>window.innerHeight;</code>	<code>// Altura interna da janela.</code>
<code>window.innerWidth;</code>	<code>// Largura interna da janela.</code>
<code>window.setInterval();</code>	<code>// Chama a cada intervalo.</code>
<code>window.clearInterval();</code>	<code>// Limpa intervalo.</code>
<code>window.setTimeout(minhaFunc,10);</code>	<code>// Chama apenas uma vez.</code>
<code>window.clearTimeout();</code>	<code>// Limpa timeout.</code>

Javascript inline e não-obtrusivo

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Não vincular Javascript no head -->
  </head>
  <body>
    <h1>Conteúdo HTML</h1>
    <button onclick="alert('Olá! ')">Clique-me</button>

    <!-- Colocar como último elemento do body -->
    <script src="arquivo_de_scripts.js"></script>
  </body>
</html>
```

Javascript (Event Listener)

```
<button id="btn">Clique-me</button>
```

```
var button = document.getElementById("btn");  
button.onclick = () => {  
    console.log("Botão clicado!");  
});
```

https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Building_blocks/Events

Javascript (Event Listener)

```
<button id="btn">Clique-me</button>
```

```
var button = document.getElementById("btn");  
button.addEventListener('click', () => {  
    var cor = 'rgb(' + random(255) + ',  
                ' + random(255) + ',  
                ' + random(255) + ')';  
    document.body.style.backgroundColor = cor;  
});
```

https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Building_blocks/Events

Javascript assíncrono (json)

```
var produtos;  
fetch('produtos.json').then(function(response) {  
    return response.json();  
}).then(function(json) {  
    produtos = json;  
}).catch(function(err) {  
    console.log('Falha Fetch: ' + err.message);  
});
```

Javascript assíncrono (text)

```
var html;  
fetch('home.html').then((response) => {  
    return response.text();  
}).then((texto) => {  
    document.getElementById('home').innerHTML = texto;  
}).catch((error) => {  
    console.log('Falha: ' + error.message);  
});
```

Javascript assíncrono (blob)

```
var imagem;  
fetch('minha_imagem.jpg').then((response) => {  
    return response.blob();  
}).then((meuBlob) => {  
    imagem = document.createElement('img');  
    imagem.src = URL.createObjectURL(meuBlob);  
    document.body.appendChild(imagem);  
}).catch((error) => {  
    console.log('Falha: ' + error.message);  
});
```


Javascript (window.onload)

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      window.onload = () => {
        console.log("Evento de carregamento concluído!");
      }
    </script>
  </head>
  <body>
    <p>O evento (onload) dispara após o carregamento da página</p>
  </body>
</html>
```