

FULL STACK WEB DEVELOPER

FERNANDO LIRA

FLAG

JAVASCRIPT

APRESENTAÇÃO – FERNANDO LIRA

3



it.fernandolira@gmail.com



<https://www.linkedin.com/in/fernandolira74/>



+351 93 317 99 21



@fernandolira74



HTML + CSS + JS – Como aplicar

```
<body onload="alert('Olá!!');">
```

```
<body>  
  <script>  
    alert("Olá!!!");  
  </script>  
</body>
```

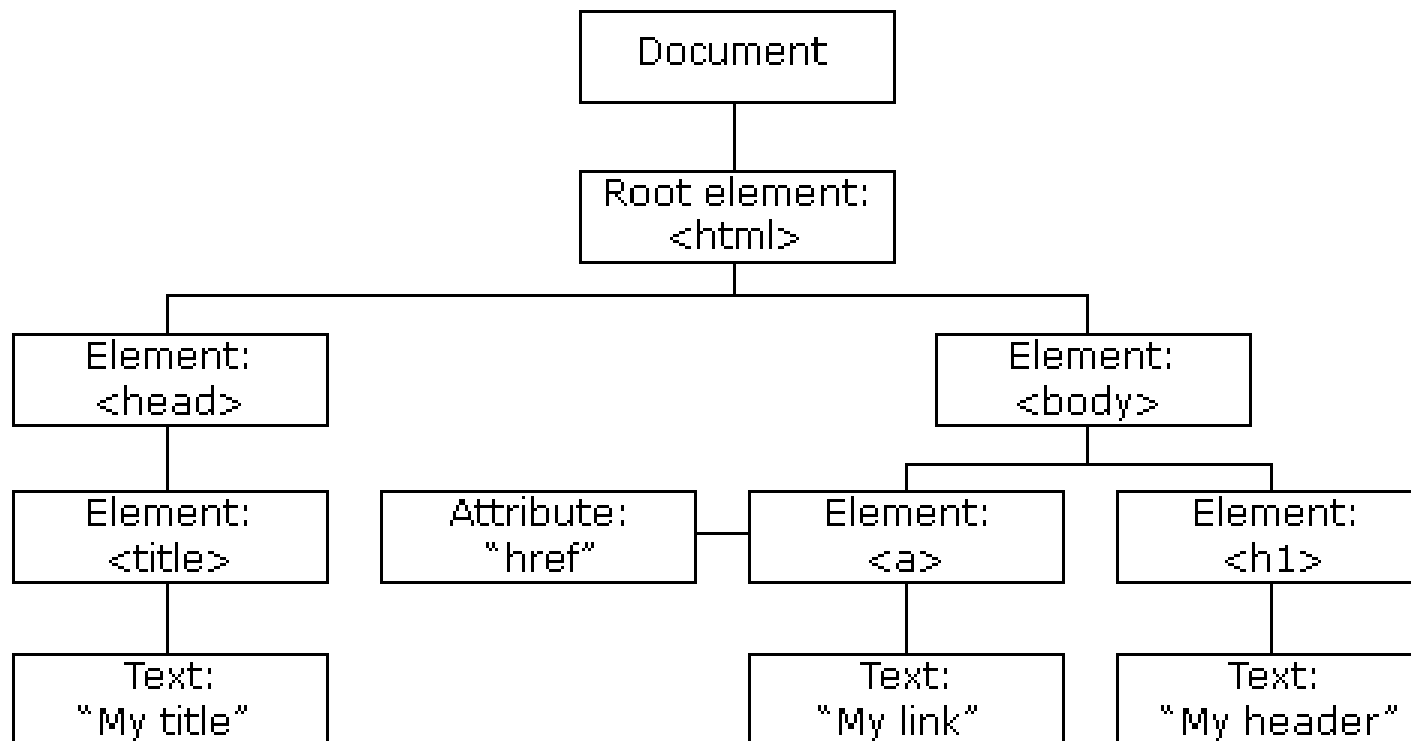
```
<body>  
  <script src="index.js"></script>  
</body>
```

HTML + CSS + JS – Onde aplicar

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>

</head>
<body>
  <h1>Olá, JSCRIPT!!</h1>
  <script>
    document.querySelector('h1').innerHTML="Bem-vindo,
Javascript!";
  </script>
</body>
```

O modelo DOM



HTML Tree Generator

<https://chrome.google.com/webstore/detail/html-tree-generator/dlbbmhhaadfnbbdnjalilhdakfmiffeg>

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Básico</title>
</head>
<body>
  <h1>Documento HTML simples</h1>
  <input type="checkbox">
  <button>Clicar</button>
  <ul>
    <li><a href="http://www.google.com">GOOGLE</a></li>
    <li>Segundo</li>
    <li>Terceiro</li>
  </ul>
</body>
</html>
```

DOM – Jscript Console

- document
- firstElementChild
- lastElementChild
- innerText e innerHtml
- style

DOM – Jscript Console

- `querySelector`
- `click()`

Alterar o conteúdo de Terceiro para FLAG na console

Seleccionar elementos

- **getElementById**
- **getElementsByTagName**
- **getElementsByClassName**

Selecionar elementos

```
<div id="primeiro">Primeiro</div>
<div id="segundo">Segundo</div>
<div class="terceiro">Terceiro</div>
<div class="terceiro">Outro Terceiro</div>
<script>
    //Selecionar elemento por ID
    document.getElementById('primeiro').classList.add('destaque'); //Se existirem 2 elementos com o mesmo id só retorna o primeiro
</script>
```

Selecionar elementos

```
//Selecionar elementos por tag
const divs = document.getElementsByTagName('div'); //ver
na consola dir(divs)

//Como obtem uma lista, é necessário criar array e com o
foreach remover a class destaque
Array.from(divs).forEach(e =>
e.classList.remove('destaque'));

//Outra forma é aplicar o operador spread ...
const divsArray=[...divs];
divsArray.forEach(e => e.classList.add('destaque'));
```

Selecionar elementos

```
//Selecionar elementos por class
const terceiroElementos =
document.getElementsByClassName('terceiro');
for (let e of terceiroElementos)
    e.classList.remove('destaque');
</script>
```

Seleccionar elementos

- **querySelector**
- **querySelectorAll**

Selecionar elementos

```
//Genérico daí necessitar do #
document.querySelector('#primeiro').classList.add('destaque');

//Executar na consola document.querySelectorAll('div')
//Não é uma Collection mas uma nodeList onde se pode
aplicar o forEach
//Não estão disponíveis outros métodos dos Arrays
document.querySelectorAll('div').forEach(e =>
e.classList.add('destaque'));
```

Selecionar elementos

```
//Criar funções genéricas para aplicar o add ou o remove por
//argumento
const selecionar = seletor =>
document.querySelectorAll(seletor).forEach(e =>
e.classList.add('destaque'));

const deselectar = seletor =>
document.querySelectorAll(seletor).forEach(e =>
e.classList.remove('destaque'));

//Deselecionar todos
deselectar('*');
```


Selecionar elementos

```
//Selecionar todos os divs
selecionar('div');

deselecionar('*');

//Selecionar class terceiro
selecionar('.terceiro');

deselecionar('*');

//Selecionar class terceiro e primeiro
selecionar('.terceiro.primeiro');

deselecionar('*');
```

Selecionar elementos

```
//ver o web site can i use e consultar o querySelect
//Todos exceto :not
selecionar(':not(html):not(body):not(head):not(#primeiro)');

deselecionar('*');

//Primeira child
selecionar(':nth-child(1):not(html)');

deselecionar('*');

//Primeiro elemento do tipo div
selecionar('div:nth-of-type(1)');

deselecionar('*');

//Selecionar a partir de um atributo
selecionar('[qualquerAtrb]');
```

:nth-child Properties



:nth-child(3)



:nth-child(odd)



:nth-child(-n+3)



:nth-child(even)



:last-child



:nth-last-child(n-3)



:nth-child(2n+1)



:nth-child(n+3)

Code Architects
@code.architects

Selecionar elementos - form

```
<div class="formulario">
  <h1>Formulário</h1>
  <form name="pesquisa"
action="http://www.google.com/search">
    <input type="text" name="q" placeholder="Palavras a
pesquisar..." />
    <select name="hl">
      <option value="pt-pt">Português</option>
      <option value="en-us">Inglês</option>
    </select>
    <button>Pesquisar</button>
  </form>
</div>
```

Selecionar elementos - form

```
<script>
    console.log(document.forms); //HTML Collection
    console.log(document.forms[0]); //Aceder ao elemento
    console.log(document.getElementsByName('pesquisa'));
//nodeList
    console.log(document.getElementsByTagName('form'));
//HTML Collection
    console.log(document.pesquisa); //Igual ao Aceder ao
elemento
    console.log(document.pesquisa.q); //Aceder ao elemento
    document.pesquisa.q.value = "dom jscript";
    document.pesquisa.hl.value = "en-us";
    console.log(document.pesquisa.hl.selectedIndex); //
elemento 1 - en-us
</script>
```

Selecionar elementos – child e parent

```
<div  
id="primeiro">Primeiro</div>  
  <div id="segundo">  
    <ul>  
      <li>Pedro</li>  
      <li>Angela</li>  
      <li>Rui</li>  
    </ul>  
  </div>  
  <div  
class="terceiro">Terceiro</div>
```

Selecionar elementos – child e parent

```
const lista = document.querySelector('div ul');  
console.log(lista.parentNode); //Seleciona o pai do ul, o  
div  
console.log(lista.childNodes); // nodeList com 7 elementos!!  
Para retirar o primeiro text, é colocar tudo na mesma linha no HTML  
  
const primeiro = lista.firstChild;  
console.log(primeiro); //Texto por causa do espaço  
console.log(primeiro.nextSibling); //O próximo "irmão"  
  
const ultimo = lista.lastChild;  
console.log(ultimo); //O último também é texto  
console.log(ultimo.previousSibling); //O "irmão" anterior
```

Selecionar elementos – child e parent

```
console.log(lista.children); // HTML Collection de todos os LI
(sem o text)
    console.log(lista.firstChild); //Seleciona o
primeiro LI, diferente do firstChild
    console.log(lista.lastElementChild); // Seleciona o
último LI

    console.log(lista.parentNode.previousElementSibling); //
o elemento anterior o parente é o primeiro div
    console.log(lista.parentNode.nextElementSibling); //
Seleciona o terceiro div

    console.log(document.querySelectorAll('div ul li')); //
nodeList com os 3 LI - outra alternativa
```


Aceder a atributos

```
<div id="primeiro">  
  <a href="http://www.flag.pt">FLAG</a>  
</div>  
<div id="segundo">  
    
</div>
```

Aceder a atributos

```
const img = document.querySelector('img'); // selecionar uma tag
// .img selecionar uma class
// #img selecionar um id
// [img] selecionar uma propriedade ou atributo
console.log(img.getAttribute('src')); //obter o atributo
src

console.log(img.src); // Alternativa
console.log(img['src']); //Alternativa
console.log(img.alt);
console.log(img.nodeType); //Atributo implicito nodeType
- ver os diferentes tipos
console.log(img.nodeName); //Atributo implicito nodeName
```

Aceder a atributos

```
const src = img.getAttributeNode('src'); // Dá o nó do atributo
    console.log(src.nodeType);
    console.log(src.nodeName);
    console.log(src.nodeValue);

const link = document.querySelector('a');
    console.log(link.href);
    console.log(link.firstChild.nodeType); // Text node 3
primeira child
    console.log(link.firstChild.nodeValue); //Texto escrito
    link.firstChild.nodeValue = "Academia FLAG";
    link.innerHTML = "F L A G"; //Alternativa mais comum a
detalhar mais à frente
```

Aceder a atributos

```
<div id="primeiro">
  <label for="compras">Compras</label>
  <ul id="compras" destino="escritorio" data-
urgencia="5">
    <li>Café</li>
    <li>Água</li>
    <li>Copos</li>
  </ul>
</div>
```

Aceder a atributos

```
const lista = document.querySelector('div
ul[destino="escritorio"');
    console.log(lista);
    console.log(document.querySelector('#compras'));
//Instrução equivalente e mais simples
    console.log(lista.childNodes);
    console.log(lista.id); //compras
```

Aceder a atributos

```
    console.log(lista.destino); //undefined porque é um
atributo próprio
    console.log(lista.getAttribute('destino')); //obter o
atributo específico
    console.log(lista.hasAttribute('destino')); //true
    lista.setAttribute('destino', 'empresa'); // define um novo
valor para o atributo
    console.log(lista.getAttribute('destino')); //empresa
    lista.setAttribute('status', 'aberto'); //define um novo
atributo caso não exista
    lista.removeAttribute('destino'); //remove um atributo
```

Aceder a atributos

```
    console.log(lista.dataset); //os atributos que começam por
data- são tratados de forma diferente!
    console.log(lista.dataset.urgencia);
    lista.dataset.urgencia = 4;
    console.log(lista.getAttribute('data-urgencia'));

    //cuidado
    const label = document.querySelector('label');
    console.log(label);
    console.log(label.for); //o for é uma palavra reservada da
linguagem
    console.log(label.htmlFor); //alternativa
    console.log(label.getAttribute('for')); //alternativa
```

Aceder a atributos

```
<div id="primeiro">  
  <ul id="compras" destino="escritorio" data-  
urgencia="5">  
    <li>Café</li>  
    <li>Água</li>  
    <li>Copos</li>  
  </ul>  
</div>
```


Aceder a atributos

```
const lista = document.querySelector('#compras');

//Somente leitura
console.log(lista.attributes);
console.log(lista.attributes[0]); //Aceder ao atributo
console.log(lista.attributes.destino); //Alternativa
console.log(lista.attributes['data-urgencia']);
//Alternativa
console.log(lista.attributes.data-urgencia); //ERRO!!! O
- é um operador no javascript
```

Aceder a classes

```
<div>  
  Classes  
</div>
```

Aceder a classes

```
const classes =  
document.querySelector('div').classList;  
console.log(classes);  
classes.add('destaque'); //adiciona a classe  
classes.remove('destaque'); //remove a classe  
classes.toggle('destaque'); //se existir, retira. Caso  
contrário, adiciona.  
console.log(classes.contains('destaque'));  
console.log(classes.length);  
  
classes.toggle('destaque');  
console.log(classes.contains('destaque'));  
console.log(classes.length);
```

Modificar o HTML

```
const body=document.querySelector('body');

const div1 = document.createElement('div');
div1.setAttribute('id','primeiro');
div1.innerHTML="Primeiro";
div1.classList.add('destaque');

const p = document.createElement('p');
p.innerHTML = "Segundo";
const div2 = document.createElement('div');
div2.setAttribute('id','segundo');
div2.appendChild(p);

body.appendChild(div1);
body.appendChild(div2);
```

Modificar o HTML

```
const texto = document.createTextNode('Antes...');
div2.insertBefore(texto, div2.childNodes[0]);

const div3 = div2.cloneNode(true) //se true copia
também os filhos
body.appendChild(div3);

const div4 = div2.cloneNode(false)
body.appendChild(div4);

body.removeChild(div3);
```

Modificar o HTML

```
<div id="primeiro">
  <p>Texto inicial</p>
</div>
<script>
  const div=document.querySelector('div');
  div.innerHTML = "Texto alterado"; //substitui todo o conteúdo da
div e o p desaparece

  //div.outerHTML = "Texto alterado"; //substitui o próprio elemento

  div.insertAdjacentHTML('beforebegin','<p>Antes do início</p>');
  div.insertAdjacentHTML('afterbegin','<p>Após o início</p>');
  div.insertAdjacentHTML('beforeend','<p>Antes do fim</p>');
  div.insertAdjacentHTML('afterend','<p>Após o fim</p>');

  console.log(div.innerHTML); //todo o html
  console.log(div.innerText); //apenas o texto
</script>
```

Eventos

```
<div id="primeiro">
  <a href="http://www.google.com"
onclick="navegarEm5s(event)">GOOGLE</a>
</div>
<script>
  function navegarEm5s(e) {
    e.preventDefault(); //ignora a ação normal do clique
e de navegar
    console.log('A navegar em 5 segundos....');
    setTimeout(() => {
      const link = e.target;
      //console.log(e);
      //console.log(link);
      window.location.href = link.href;
    },5000);
  }
</script>
```

Eventos – alternativa 1

```
<div id="primeiro">
  <!-- onclick="navegarEm5s(event)" -->
  <a href="http://www.google.com">GOOGLE</a>
</div>
<script>
  function navegarEm5s(e) {
    e.preventDefault(); //ignora a ação normal do clique e de navegar
    console.log('A navegar em 5 segundos....');
    setTimeout(() => {
      const link = e.target;
      //console.log(e);
      //console.log(link);
      window.location.href = link.href;
    },5000);
  }

  const a = document.querySelector('a');
  a.onclick = navegarEm5s;
</script>
```


Eventos – alternativa 2

```
<div id="primeiro">
  <!-- onclick="navegarEm5s(event)" -->
  <a esperar href="http://www.google.com">GOOGLE</a>
</div>
<script>
  function navegarEm5s(e) {
    e.preventDefault(); //ignora a ação normal do clique e de navegar
    console.log('A navegar em 5 segundos....');
    setTimeout(() => {
      const link = e.target;
      //console.log(e);
      //console.log(link);
      window.location.href = link.href;
    },5000);
  }

  const a = document.querySelector('[esperar]');
  a.onclick = navegarEm5s;
</script>
```

Eventos – mover

```
<div mover id="primeiro" class="destaque">
  Mover
</div>
<script>
  const item = document.querySelector('[mover]');
  item.style.position = 'absolute';

  item.onmousemove = e => {
    const item = e.target;
    item.style.cursor = 'move';
    if (e.buttons) {
      item.style.top = `${e.clientY - (item.clientHeight /
2)}}px`;
      item.style.left = `${e.clientX - (item.clientWidth /
2)}}px`;
    }
  };
</script>
```

Eventos – drag and drop

```
<div class="container">  
  <div class="esquerda" dropzone>  
    <div class="item">1</div>  
    <div class="item">2</div>  
    <div class="item">3</div>  
    <div class="item">4</div>  
  </div>  
  <div class="direita" dropzone></div>  
</div>
```

Eventos – drag and drop

```
const items = document.querySelectorAll('[dropzone]
.item');
items.forEach((item, index) => {
  item.draggable = true;
  item.id = item.id || `draggable-item-${index}`;
  item.ondragstart = e => e.dataTransfer.setData('item-
id',e.target.id);
});
```

Eventos – drag and drop

```
const dropzones =
document.querySelectorAll('[dropzone]');
dropzones.forEach( dropzone => {
  dropzone.ondragover = e => e.preventDefault();
  dropzone.ondrop = function (e) {
    const id = e.dataTransfer.getData('item-id');
    const item = document.getElementById(id);
    dropzone.appendChild(item);
    //e.target.appendChild(item);
  }
});
```

Eventos – forms

```
<div class="formulario">
  <h1>Formulário</h1>
  <form name="pesquisa"
action="http://www.google.com/search">
    <input type="text" name="q" placeholder="Palavras a
pesquisar..." />
    <select name="h1">
      <option value="pt-pt">Português</option>
      <option value="en-us">Inglês</option>
    </select>
    <button submit>Pesquisar</button>
  </form>
</div>
```

Eventos – forms – Alternativa 1

```
<script>
  const submit = document.querySelector('[submit]');
  submit.onclick = function (e) {
    e.preventDefault();

    const form = e.target.parentNode;
    const formData = new FormData(form);
    console.log(formData.get('q'));
    console.log(formData.has('h1'));

    for (let dado of formData)
      console.log(dado);
  }
</script>
```

Eventos – forms – Alternativa 2

```
<script>
  const form = document.forms[0];
  form.onsubmit = function (e) {
    e.preventDefault();

    const form = e.target;
    const formData = new FormData(form);
    console.log(formData.get('q'));
    console.log(formData.has('h1'));

    for (let dado of formData)
      console.log(dado);
  }
</script>
```


EXERCÍCIOS

Desafios

JOGO DE DADOS

JOGO DO GALO