JavaScript

DOCUMENT OBJECT MODEL (DOM)

Document Object Model (DOM)

- Encontrando elementos por tag
- Encontrando elementos por classe
- Encontrando elementos por seletor CSS
- Criando elementos no documento

Eventos

- addEventListener()

Object Window

Biblioteca JQuery

Document Object Model (DOM)

Qual a Diferença entre DOM e BOM?

DOM (Document Object Model)

BOM (Browser Object Model) referencia window

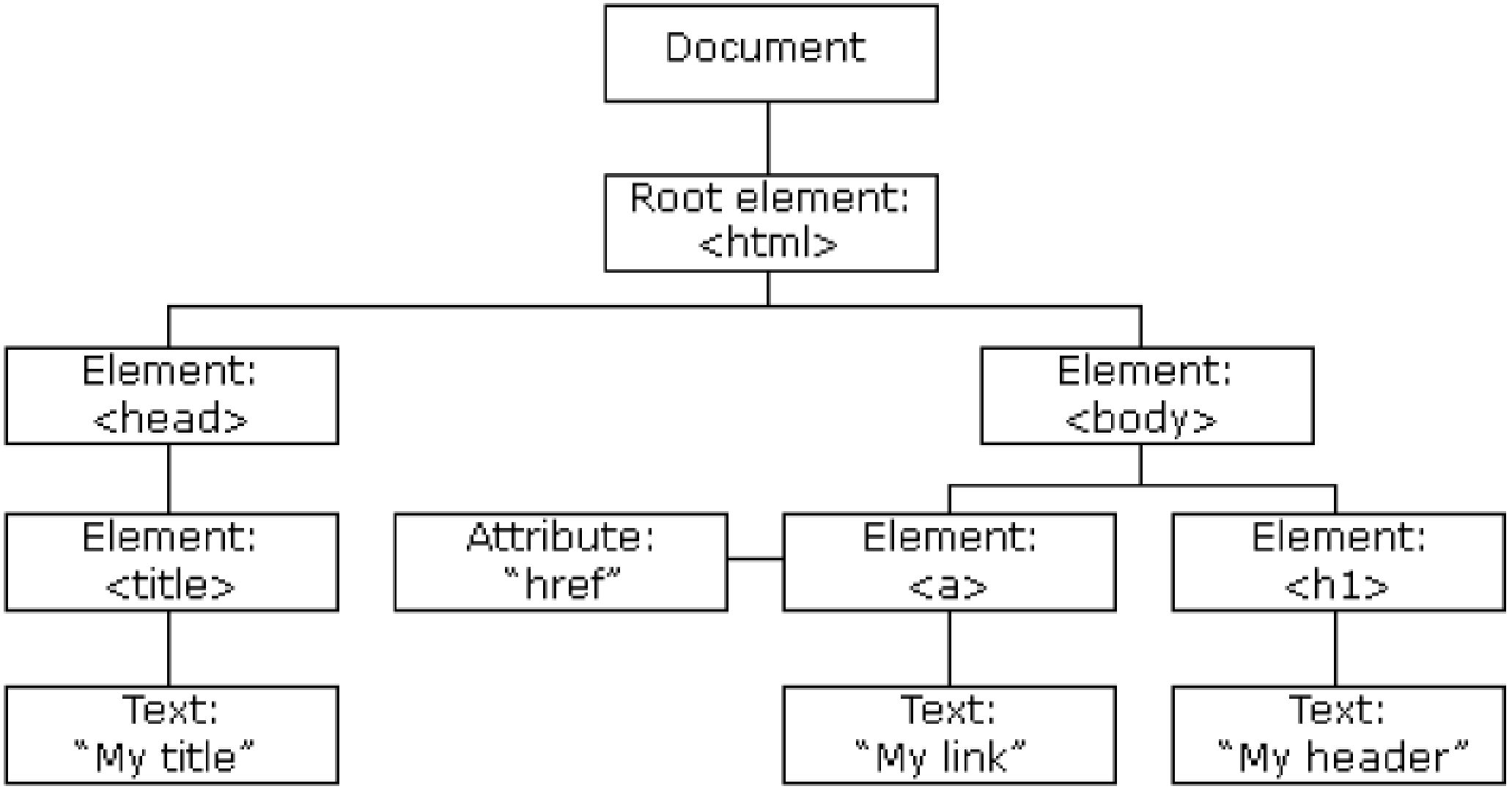
DOM objetos do HTML (links, ids, head, p, H1)

BOM objetos do Navegador (History, Navigator, Location, o próprio DOM é um objeto do Navegador).

Document Object Model (DOM)

- O DOM é a representação de dados dos objetos que compõem a estrutura e o conteúdo de um documento na Web

- É uma interface de programação para os documentos HTML. Permite que o JS possa alterar a estrutura o estilo e conteúdo da página.


Document
Root element:
<html>
Element:
<head>
Element:
<body>
Element:
<title>
Attribute:
"href"
Element:
<a>
Element:
<h1>
Text:
"My title"
Text:
"My link"
Text:
"My header"

Document Object Model (DOM)

JS pode alterar todos os elementos HTML na página
JS pode alterar todos os atributos dos elementos HTML na página
JS pode alterar todos os estilos CSS
JS pode remover um elemento HMTL e seus atributos
JS pode adicionar um elemento HMTL e seus atributos
JS pode reagir a todos os eventos que ocorrerem em uma página
JS pode criar novos eventos na page

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>DOM</title>
        <script src="meuscript.js"></script>
    </head>

    <body>
        <h1 id="id_h1" class="classe_h1">Sou um cabeçalho!</h1>
        <p id="id_p1" class="classe_p">
            Um texto qualquer dentro de uma tag de parágrafo. Aqui também
            temos outras tags, como <a href="#">um link<a>, ou um texto
            <b>em negrito</b>.
        </p>
        <img src="#" id="id_imagem" class="classe_imagem">
        <p id="id_p2" class="classe_p">
            Este é outro parágrafo.
        </p>
    </body>
</html>
```

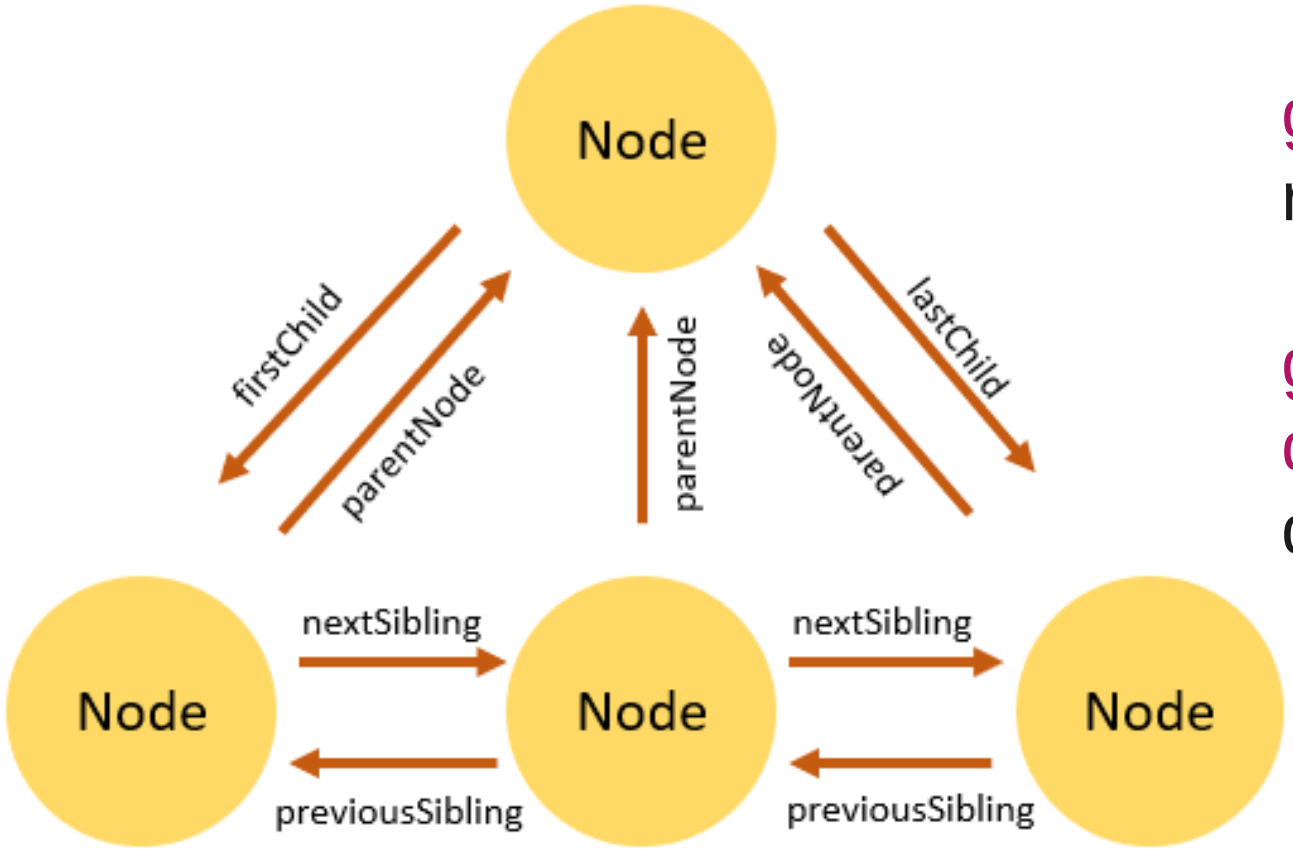
Document Object Model (DOM)

Existem elementos pai (parent), filhos (childs) e irmãos (siblings).

Estes elementos são caracterizados na forma como estão na árvore

nodeName	id	class
#document		
html		
HTML		
HEAD		
#text		
META		
#text		
TITLE		
#text		
#text		
SCRIPT		
#text		
#text		
BODY		
#text		
H1	id_h1	classe_h1
#text		
#text		
P	id_p	classe_p
#text		
A		
#text		
#text		
B		
#text		
#text		
#text		
P	id_p2	classe_p2
#text		
#text		

Um nó (node) é um nome genérico de qualquer objeto na árvore DOM (tag HTML como ou <p>)



getElementById () e querySelector () retornam um objeto tipo Element,

getElementsByTagName() ou querySelectorAll() retorna NodeList, que é uma coleção de nós.

Document Object Model (DOM)

Objeto possui métodos (funções) e propriedades (atributos)

Funções do objeto document: referencia

element.addEventListener()
document.getElementById(id)

Propriedades do objeto document:

value
style

Document Object Model (DOM)

Encontrando elementos na página

Através do ID
Através da tag
Através da classe
Por seletores CSS
Por coleções de objetos

Encontrando elementos através do ID

getElementById(id)

Retorna a referência do elemento através do seu ID ou null se o elemento especificado com o ID não existir. ID é uma string que diferencia maiúsculas e minúsculas.

Para o parágrafo de nome ***para1*** passa-se:

document.getElementById('para1');

exemplo

Encontrando elementos através da tag

getElementsByTagName

O parâmetro a ser passado é o **nome da tag** que se deseja buscar, o método sempre irá retornar um **array** contendo t**odos** elementos daquela determinada tag **existentes na página**

Para parágrafos, passa-se o parâmetro p

document.getElementsByTagName("p")

Para divs, passa-se o parâmetro div

document.geteElementsByTagName("div")

exemplo

Encontrando elementos através da classe

getElementsByClassName

O parâmetro a ser passado é o **nome da classe** que se deseja buscar, o método sempre irá retornar um **array** contendo **todos** os elementos daquela tag **existentes na página**

Exemplo: uma classe CSS chamada oculto, que faz com que os objetos fiquem ocultos (display: none). Deseja-se buscar todos esses elementos e transformá-los em visíveis.

getElementsByClassName('oculto')

Exemplo exemplo remove exemplo add

Encontrando elementos através seletores CSS

querySelector.

O método retorna **o primeiro element** que combine com o padrão. Deve-se especificar um ou mais seletores CSS. Para múltiplos seletores, separe através de vírgulas.

Exemplos:

document.querySelector("p")
document.querySelector("p.oculto")
document.querySelector("[type=text]")

Exemplo

Seletores CSS: http://www.w3schools.com/cssref/css_selectors.asp

Encontrando elementos através seletores CSS

querySelectorAll.

O método sempre irá retornar um **array** contendo **todos** os elements que combinam com o padrão.

Exemplo:

```
const divs = document.querySelectorAll("div.note, div.alert");
```

Seletores CSS: http://www.w3schools.com/cssref/css_selectors.asp

Alterando elementos no documento

createElement(element) - Cria um elemento HTML

element.removeChild(elementToRemove) - Remove o filho de um elemento

element.appendChild(newElement) – Adiciona um novo elemento como último filho do elemento da chamada do método

replaceChild(newElement, oldElement) – Substitui um elemento por outro

Exemplo create

Outras funções: https://developer.mozilla.org/pt-BR/docs/Web/API/Document

Encontrando elementos através de coleções

document.anchors : todos os elementos <a> com atributo name
document.body: o elemento body
document.documentElement : o próprio elemento <html>
document.embeds : todos os elementos incorporados <embed>
document.forms : todos os formulários na página (array de objetos)
document.head : o elemento head
document.images : todas as imagens <img> existentes na página
document.links : todos os elementos <a> e <area> com atributo href
document.scripts : todos os elementos <script>
document.title : o elemento <title>

Eventos

Eventos são ações ou ocorrências que são disparados dentro da janela do navegador (window)

Exemplo: usuário **clica em um botão** numa pagina web (window), você pode responder a esta ação

Event reference

Exemplo

addEventListener() e removeEventListener()

O mais novo tipo de mecanismo de evento é definido na Especificação de Eventos Nível 2 do Document Object Model (DOM)

Semelhante às propriedades do manipulador de eventos, mas a sintaxe é obviamente diferente

Event reference

Exemplo

Window

O objeto window representa uma janela que contém um elemento DOM

Em um navegador com suporte a abas, cada aba contém seu próprio objeto window

window reference

Exemplo

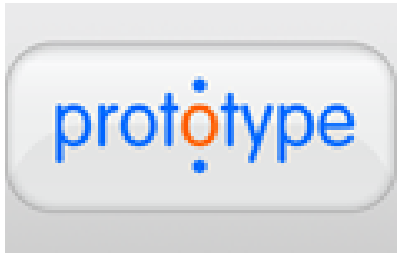
Biblioteca JQuery

Uma biblioteca é um conjunto de funções para executar diversas tarefas

Um framework normalmente é um conjunto de bibliotecas

Aprender a usar bibliotecas e frameworks antes de aprender a linguagem nativa não resolve


jQuery
write less, do more.


prototype


script.aculo.us
it's about the user interfaces, baby!


mootools
an ultra compact javascript framework


Modernizr

Jquery: write less, do more.

- ✓ É fácil, é produtivo!
- ✓ É compatível
- ✓ Resolução da incompatibilidade entre os navegadores.
- ✓ Redução de código.
- ✓ Reutilização do código através de plugins.
- ✓ Vasta quantidade de plugins criados por outros desenvolvedores.
- ✓ Trabalha com DOM.
- ✓ Recursos CSS3

Para aprender mais Jquery

https://www.codecademy.com/learn/learn-jquery

https://www.w3schools.com/jquery/

http://vitorfs.github.io/jquery-na-pratica/#seletores-jquery