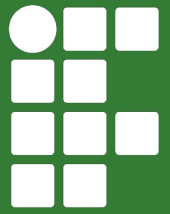


INSTITUTO FEDERAL
Sul-rio-grandense | Câmpus Pelotas

Java Script

Aula 2

Professor Sérgio Rodrigues

tsi tecnologia em sistemas para internet

Trabalhando com Objetos

A linguagem JavaScript é projetada com base no paradigma orientado a objeto.

Um objeto em JavaScript é uma coleção de propriedades, portanto um objeto tem propriedades associadas a ele. Uma propriedade de um objeto pode ser explicada como uma variável que é ligada ao objeto. As propriedades de um objeto definem as características, as informações do objeto. Você acessa as propriedades de um objeto com uma simples notação de ponto.

nomeDoObjeto.nomeDaPropriedade

Um método é uma função associada a um objeto, é uma sub-rotina que é executada por um objeto ao receber uma mensagem. Os métodos determinam o comportamento dos objetos, e podem alterar o estado de um objeto.

nomeDoObjeto.nomeDoMetodo(parâmetros)

Exemplos de Objetos, Propriedades e Métodos

Alguns objetos são pré-definidos no browser, e possuem as suas propriedades e métodos próprios.

O objeto **window** por exemplo faz a manipulação das janelas do navegador. Sempre que abrirmos o navegador automaticamente esse objeto é gerado, isso acontece porque esse objeto representa exatamente essa janela que foi aberta.

window.open() – open é um método do objeto window para o qual você pode passar uma URL para a qual deseja abrir em uma nova janela.
window.open('http://www.google.com’); uma chamada do método open que abre uma nova janela com o google

window.location - location é uma propriedade do objeto window que contem todas as informações sobre o local atual do documento (host, href, port, protocol, etc.)
window.location.href é um atributo de uma propriedade que informa a localização atual do URL do navegador.

DOM (Document Object Model)

O **Modelo de Objeto de Documento (*DOM*)** é uma interface de programação para documentos HTML.

Ele fornece uma representação estruturada do documento como uma árvore.

O DOM representa a página de forma que os programas possam alterar a estrutura do documento, alterar o estilo e conteúdo. O DOM representa o documento com nós e objetos, dessa forma, as linguagens de programação podem se conectar à página.

Embora o DOM seja frequentemente acessado usando JavaScript, não é uma parte da linguagem JavaScript. Ele também pode ser acessado por outras linguagens.

DOM (Document Object Model)

Uma página da Web é um documento que pode ser exibido na janela do navegador. O DOM (Document Object Model) representa o mesmo documento para que possa ser manipulado.

Por exemplo, o DOM padrão especifica que o método getElementsByTagName no código abaixo deve retornar uma lista de todos os elementos <p> no documento:

```
var paragraphs = document.getElementsByTagName("p");
// paragraphs[0] is the first <p> element
// paragraphs[1] is the second <p> element, etc.
```

Todas as propriedades, métodos e eventos disponíveis para manipular e criar páginas da Web são organizados em objetos (por exemplo, o objeto de document que representa o próprio documento, o objeto de table que implementa uma interface para acessar tabelas HTML e assim por diante).

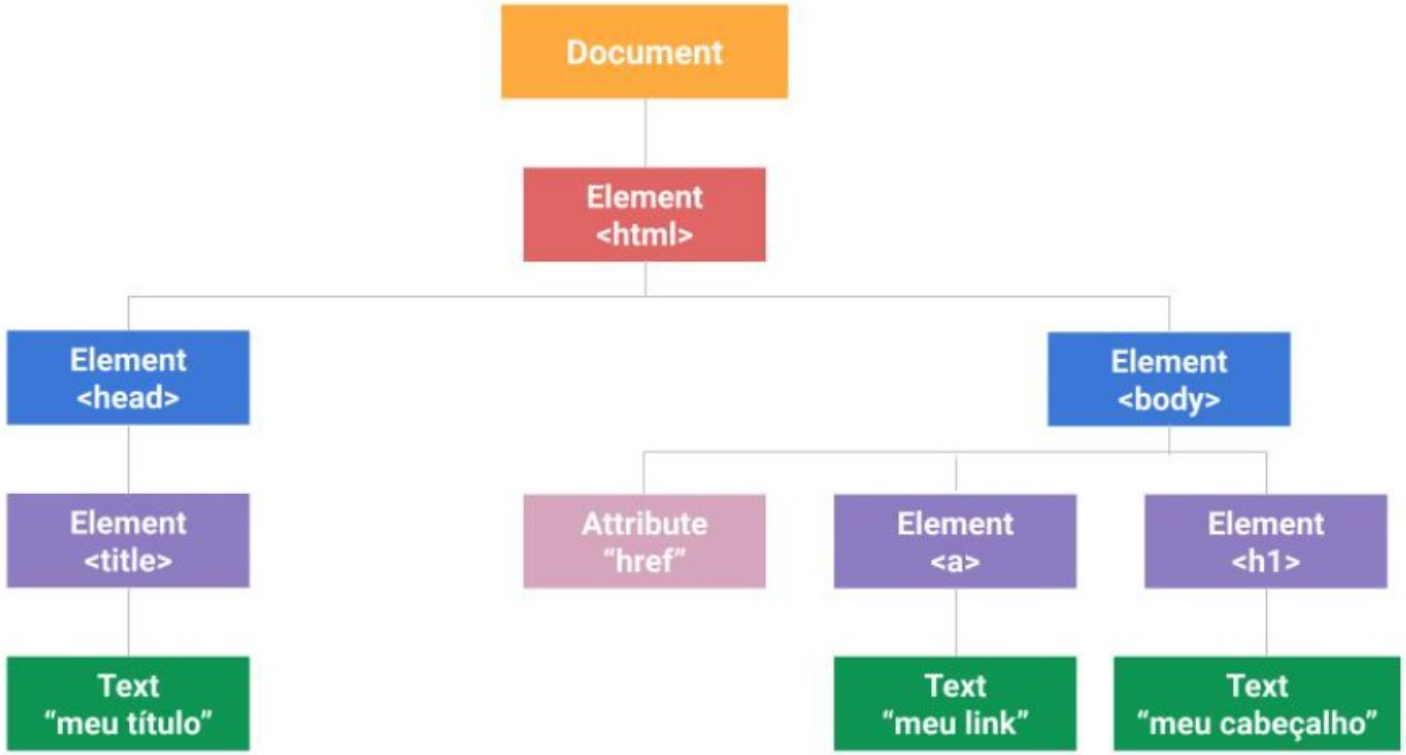
DOM (Document Object Model)

O DOM não é uma linguagem de programação, mas sem ele, a linguagem JavaScript não teria nenhum modelo ou noção de páginas da web, documentos HTML, documentos XML e suas partes componentes (por exemplo, elementos).

Cada elemento de um documento - o documento como um todo, o cabeçalho, as tabelas do documento, os cabeçalhos da tabela, o texto nas células da tabela - fazem parte do modelo de objetos do documento.

Quando você cria um script - seja embutido em um elemento ou incluído na página da web por meio de uma instrução de carregamento de script - você pode começar imediatamente a manipular os elementos na página da web.

Exemplo de uma árvore DOM de uma página Web



DOM (Document Object Model)

Agora vamos conferir alguns dos dados e objetos fundamentais do modelo de documento por objetos (DOM):

Document: é o objeto raiz, representa o próprio documento HTML.

Node: é um nó, e representa uma nomenclatura mesmo: todo objeto em um documento é algum tipo de nó.

Element: o nó do tipo elemento representa as tags do documento HTML, isto é, o objeto de elemento pode ter “nós-filhos”, como de texto e de atributos.

Attribute: este tipo de objeto representa um atributo contido num elemento HTML.

Text: é o texto, ele fica localizado entre os elementos e representa o conteúdo das tags, ou elementos. Por exemplo: aqui está um texto .

Eventos

Eventos são ações ou ocorrências que acontecem no sistema. Por exemplo, se o usuário clica em um botão numa pagina web, pode obter uma resposta a esta ação mostrando na tela uma caixa de informações.

O sistema irá disparar algum tipo de sinal quando o evento acontecer, além de prover um mecanismo pelo qual alguma ação automática possa ser executada (ou seja, rodar algum código) quando o evento ocorrer.

No caso da web, eventos são disparados dentro da janela do navegador, e tende a estarem anexados a algum item especifico nele — pode ser um único elemento, um conjunto de elementos.

Eventos

São ações capazes de disparar uma reação:

- Clicar num link
- Colocar o ponteiro do mouse sobre um elemento
- Pressionar uma tecla
- Uma página web terminando de carregar
- Um formulário sendo enviado
- Um erro ocorrendo
- Fechar ou redimensionar a janela do navegador

Eventos podem ser disparados a partir de uma ação do usuário (evento onclick) ou ocorrer independentemente da interferência dele (evento onload)

Eventos viabilizam a interatividade de uma página web.

Eventos

Existem diversos eventos que podem ser utilizados em diversos elementos para que a interação do usuário dispare alguma função:

oninput: quando um elemento input tem seu valor modificado
onclick: quando ocorre um click com o mouse
ondblclick: quando ocorre dois clicks com o mouse
onmousemove: quando mexe o mouse
onmousedown: quando aperta o botão do mouse
onmouseup: quando solta o botão do mouse
onkeypress: quando pressionar e soltar uma tecla
onkeydown: quando pressionar uma tecla
onkeyup: quando soltar uma tecla
onblur: quando um elemento perde foco
onfocus: quando um elemento ganha foco
onchange: quando um input, select ou textarea tem seu valor alterado
onload: quando a página é carregada
onunload: quando a página é fechada
onsubmit: disparado antes de submeter o formulário (útil para realizar validações)

Eventos onclick + Caixas + diretamente na propriedade

Os exemplos deste material encontram-se na pasta compactada "exemplos JavaScript - 2"

```
1  <html>
2  <head>
3  <title>JavaScript</title>
4  <meta charset="utf-8"/>
5  </head>
6  <body>
7
8  <button type="button" onclick="alert('Caixa de alerta');"> Alert</button>
9
10 <button type="button" onclick="confirm('Caixa de confirmação');"> Confirm</button>
11
12 <button type="button" onclick="prompt('Caixa de dados');"> Prompt</button>
13
14 </body>
15 </html>
```

exemplo2_1.html

Objeto document

- O objeto document representa um documento HTML
- Este objeto possibilita o acesso, via JS, a todos os elementos HTML de uma página.

Método para trabalhar com elementos da página:

getElementById()

método getElementById()

O método getElementById() acessa o elemento do DOM cujo atributo id foi definido como parâmetro.

```
1  <html>
2  <head>
3  <script>
4  window.onload = function(){
5  var elemento = document.getElementById("conteudo");
6
7      elemento.innerHTML = "<p>novo texto</p>";
8  }
9  </script>
10 </head>
11 <body>
12 <div id="conteudo">texto antigo</div>
13 </body>
14 </html>
15
```

exemplo2_2.html

- A propriedade innerHTML permite o acesso ou a definição do conteúdo HTML de um elemento do DOM.
- Este conteúdo HTML pode ser simplesmente texto ou uma estrutura mais complexa, contendo mais elementos de marcação com textos.

Tratando eventos com função

```html
<html>
<head>
<script>

 function adiciona(){
     var elemento = document.getElementById("conteudo");
     elemento.innerHTML = "<p>novo texto</p>";
}

</script>
</head>
<body>
<div id="conteudo"></div>
<button id="btn" onclick="adiciona()">Adicione o texto!</button>
</body>
</html>
```

exemplo2_3.html

Utilizando Eventos

EVENTO ONBLUR - Com o evento onBlur o usuário irá controlar o conteúdo de um elemento em um formulário select, text ou textarea quando o mesmo remove o foco.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>JavaScript 4</title>
<script src="exemplo_4.js"> </script>
</head>
<body>
Digite seu Nome :
<input type= "text" id="nome" onblur = "alerta()">
<div id="texto"> Clique fora do campo após a digitação </div>
</body>
</html>
```

exemplo2_4.html

Utilizando Eventos

```
function alerta(){
        elemento_nome=document.getElementById("nome");
        var conteudo_nome= elemento_nome.value;
        if(conteudo_nome==""){

            alert("Favor preencher o Nome");

            elemento_nome.focus();
        }
        else
        {
            conteudo_div="Bem Vindo(a) "+conteudo_nome;
            conteudo_div+=" <br> faça também o teste deixando o campo em branco";
            document.getElementById("texto").innerHTML = conteudo_div;

        }
}
```

codigo_4.js

Um campo de formulário é um elemento que possui a propriedade “value”

Utilizando Eventos

EVENTO ONCHANGE - Com o evento onChange o usuário poderá acionar alguma função sempre que for alterado o conteúdo dos objetos textarea, text ou select. Este evento é bem similar com o evento onBlur, porém ele verifica se o conteúdo do elemento foi alterado.

```
<p>Modifique o texto no campo Input e de um clique fora do campo para disparar o
evento onchange.</p>

Digite algum texto: <input type="text" name="txt" value="Hello" onchange="
    myFunction(this.value)">

<script>
function myFunction(val) {
  alert("O campo de entrada foi alterado. O novo valor é: " + val);
}
</script>
```

exemplo2_5.html

Exemplo com onclick

```
<p>Use o HTML DOM para atribuir um evento "onclick" a um elemento p:</

<p id="demo">Click me.</p>

<script>
document.getElementById("demo").onclick = function() {myFunction()};

function myFunction() {
  document.getElementById("demo").innerHTML = "YOU CLICKED ME!";
}
</script>
```

exemplo2_6.html

Exemplo Validação de Formulário

```
<h1>Login Usuário</h1>

<form name="myForm" action="#" onsubmit="return validateForm()" method="post"
>
Usuario: <input type="text" name="fname">
Senha: <input type="password" name="pass">
<button type="submit" id="btn">Logar</button>
</form>
```

exemplo2_7.html

Exemplo Validação de Formulário

```
function validateForm() {
  let x = document.forms["myForm"]["fname"].value;
  let y = document.forms["myForm"]["pass"].value;
  if (x == "" || y == "") {
    alert("Preencher todos os campos");
    return false;
  }
}
```

codigo_7.js

Exemplo Validação de Formulário

```
<script>
function myFunction() {

  var x = document.getElementById("numb").value;

  var text;
  if (isNaN(x) || x < 1 || x > 10) {  //A função isNaN() determina se o valor é
  vazio.
    text = "Número Invalido";
  } else {
    text = "Número OK";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>
```


exemplo2_8.js

Exemplo Validação de Formulário

```
    <title>Exemplo - Formulario HTML</title>

     <link href="css/estilosform.css" rel="stylesheet">
     <script src="js/codigo_9.js"> </script>
</head>
<body>

<form name="myForm" action="#" onsubmit="return validateForm()" method="post">
    <fieldset>
        <fieldset class="grupo">
            <div class="campo">
                <label for="nome">Nome</label>
                <input type="text" id="nome" name="nome" style="width: 20em"
                value="">
            </div>
            <div class="campo">
                <label for="snome">Sobrenome</label>
```


exemplo2_9.js

Exemplo Validação de Formulário

```
function validateForm() {
  let x = document.forms["myForm"]["nome"].value;
  let y = document.forms["myForm"]["snome"].value;
  if (x == "" || y == "") {
    alert("Preencher Nome ou Sobrenome");
    return false;
  }
  let s = document.forms["myForm"]["sexo"].value;
   if (s == "" ) {
    alert("Preencher Sexo");
    return false;
  }
}
```


codigo_9.js

Atividade

1) O window.open() no JavaScript o que representa ?
2) O window.location no JavaScript o que representa ?
3) Qual a utilidade do Modelo de Objeto de Documento (DOM) ?
4) Quais são os dados e objetos fundamentais do modelo de documento por objetos (DOM):
5) Eventos são ações capazes de disparar uma reação ao ? (cite 5):
6) Qual a utilidade do getElementById() ?
7) Qual a utilidade do innerHTML ?
8) O JavaScript possui três métodos (ou funções), para o objeto Window, para criar diferentes caixas de diálogo que são ?