



JavaScript

THASSIO RAMON VIEIRA LUCENA

Roteiro

Objetos

- Objetos e Propriedades
- Acesso

Formulários

- Introdução
- Validando Formulários com JS

DOM

- O que é?
- Modelo
- Praticando



JavaScript

OBJETOS

Objetos e Propriedades

Na vida real, um carro é um objeto.

Um carro tem propriedades como peso e cor, e métodos como iniciar e parar



Propriedades

```
car.name = Fiat  
car.model = 500  
car.weight = 850kg  
car.color = white
```

Métodos

```
car.start ()  
car.drive ()  
car.brake ()  
car.stop ()
```

Objetos e Propriedades

Na vida real, um carro é um objeto.

Um carro tem propriedades como peso e cor, e métodos como iniciar e parar



- Todos os carros têm as mesmas **propriedades**, mas os valores das propriedades diferem de carro para carro.
- Todos os carros têm os mesmos **métodos**, mas os métodos são realizados em momentos diferentes.

Objetos JavaScript

- Você já aprendeu que as variáveis JavaScript são contêineres para valores de dados.
- Este código atribui um **valor simples** (Fiat) a uma **variável** chamada carro:

```
var car = "Fiat";
```

Objetos JavaScript

- Os objetos também são variáveis. Mas objetos podem conter muitos valores.
- Este código atribui **muitos valores** (Fiat, 500, branco) a uma **variável** chamada carro:

```
var car = {type:"Fiat", model:"500", color:"white"};
```

Objetos JavaScript

- Em JavaScript objetos guardam um conjunto de duplas (chave-valor).

Por exemplo, um objeto
vôo pode ter a
seguinte chave:
horário, com valor: "18:30"



Cia	Voo	Prev	Conf	Origem/Escalas
TAM	3726	00:07		Brasília
GOL	1772	00:20		São Paulo
E-TRIP	5348	00:30		Goiânia
Azul	4142	02:10		Goiânia
SEIE	6412	08:35		Goiânia
PASSAREDO	2344	12:10		Uberlândia
GOL	1202	12:20		São Paulo
E-TRIP	5601	12:30		Brasília
Azul	4080	14:02		Campinas
E-TRIP	5608	15:15		Rio de Janeiro
PASSAREDO	2345	15:40		Belém
E-TRIP	5207	16:25		Carajas

Acessando propriedades do objeto

- Você pode acessar as propriedades do objeto de duas maneiras:

objectName.propertyName

ou

objectName["propertyName"]

Exemplos

objectName.propertyName

<p>

There are two different ways to access an object property:

</p>

<p>You can use person.property or person["property"].</p>

<p id="demo"></p>

<script>

```
var person = {  
    firstName: "John",  
    lastName : "Doe",  
    id       : 5566
```

```
};
```

```
document.getElementById("demo").innerHTML =  
person.firstName + " " + person.lastName;
```

</script>

</body>

</html>

Exemplos

objectName["propertyName"]

< p >

Existem duas maneiras diferentes de acessar uma propriedade de objeto:

</ p >

< p > Você pode usar person.property ou person ["property"]. </ p >

< p id = "demo" > </ p >

< script >

```
var person = {  
    primeiro nome : "John" ,  
    último nome : "Doe" ,  
    ID           : 5566
```

```
};
```

```
documento . getElementById ( "demo" ). innerHTML =  
pessoa [ "primeiro nome" ] + "" + pessoa [ "último nome" ];
```

</ script >

</ body >

</ html >

Acessando métodos do objeto

- Você pode acessar um método de objeto com a seguinte sintaxe:

objectName.methodName()

Acessando métodos do objeto

Exemplo:

```
<! DOCTYPE html>
< html >
< corpo >

< p > Criando um objeto JavaScript. </ p >

< p id = "demo" > </ p >

< script >
var person = {
    primeiro nome : "John" ,
    último nome   : "Doe" ,
    idade         : 50 ,
    olhoColor     : "azul"
};

documento . getElementById ( "demo" ). innerHTML =
pessoa . firstName + "is" + pessoa . idade + "anos de idade". ;
</ script >

</ body >
</ html >
```



JavaScript

FORMULÁRIOS

Introdução

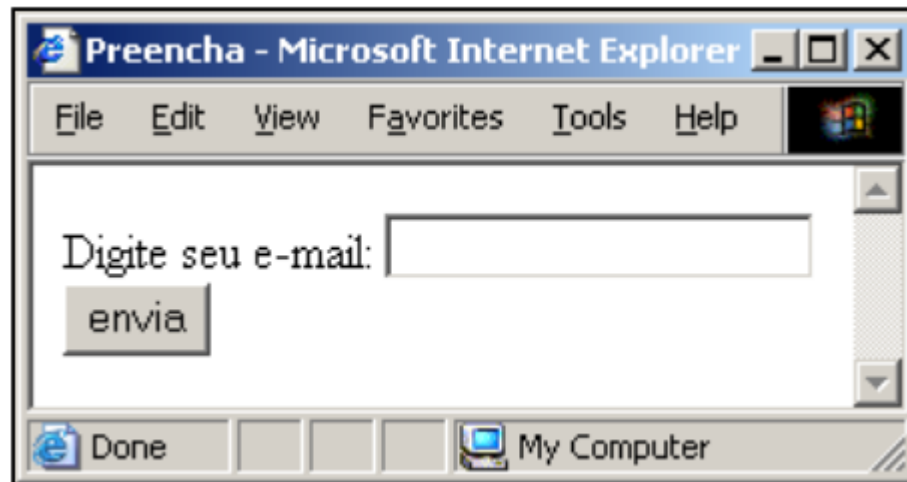
- Peças de composição de uma página presentes em praticamente todos os web sites;
- Interface de coleta de dados, mesmo apenas para busca;
- São interativos;
- O desenvolvedor não deve supor que o usuário digitará apenas o que se espera.

Formulários

- Primeiro fundamento do uso de JavaScript com formulários: validar campos;
- Mas ações não devem apenas no cliente, tem que ser desenvolvida no lado do servidor;
- O elemento `<form></form>` é o container de todos os controles(campos) de um formulário.

Formulários

O objeto Form é o mais alto da hierarquia dos componentes de formulários. Através dele se tem acesso aos componentes existentes dentro de um bloco HTML , que podem ser botões, caixas de texto, caixas de seleção, etc.



Formulários

- O elemento form admite os atributos:
 - Action: aponta para o endereço em que se encontra o programa destinado a processar os dados enviados pelo formulário;
 - Method: define o método de envio de dados para o servidor (get/post);
 - Enctype: tipo de codificação de caracteres. Por default usa a mesma codificação do documento.
 - Name: nome para o formulário. Esse atributo está em desuso e, em seu lugar, deve ser usado o atributo id.

Outros atributos do form

- id
- class
- lang
- style
- title
- target
- onsubmit
- onreset
- onclick
- ondblclick
- onmousedown
- onmouseup
- onmouseover
- onmousemove
- onmouseout
- onkeypress
- onkeydown
- onkeyup

Tipos de Campo

Objeto	Descrição e tipo de dados
Campo de Texto	Entrada de dados com texto e números.
Senha	Campo de texto do tipo Senha não exibem os caracteres digitados, apenas asteriscos.
Botão	Botões que executam determinadas funções, como enviar os dados do formulário ou apagar os dados.
Caixa de seleção	Permite escolher uma única opção entre um grupo de diferentes opções.
Botão de opções	Cria um menu ou uma lista com várias opções a serem escolhidas.
Menu ou listas	Cria um menu ou uma lista com várias opções a serem escolhidas.
Campos de arquivo	Insere uma caixa de texto com um botão de seleção que permite escolher um arquivo no sistema.
Campos de imagem	Define um campo de imagem que pode ser usado com o botão de envio do formulário.

Validando Formulários

1. Criando Formulário

– Primeiramente, vamos fazer nosso formulário:

```
1 <html>
2 <head>
3 <title>Validando Formulários com Javascript</title>
4 </head>
5 <body>
6 <form name="form1" action="enviar.php" method="post">
7 Nome:
8 <input name="nome" type="text"><br /><br />
9 Email:
10 <input name="email" type="text"><br /><br />
11 Senha:
12 <input name="senha" type="password"><br /><br />
13 Repitir Senha:
14 <input name="rep_senha" type="password"><br /><br />
15 <input type="submit" onclick="return validar()>
16 </form>
17 </body>
18 </html>
```

Validando Formulários

2. Criando a função validar()

Agora, vamos criar nosso código javascript para fazer validação, lembrando que esse código deve estar, de preferência, entre as tags head (cabeçalho):

```
1 <script language="javascript" type="text/javascript">
2 function validar() {
3 }
4 </script>
```

Validando Formulários

2. Criando a função validar()

Agora, vamos criar nosso código javascript para fazer validação, lembrando que esse código deve estar, de preferência, entre as tags head (cabeçalho):

```
1 <script language="javascript" type="text/javascript">
2 function validar() {
3   var nome = form1.nome.value;
4   var email = form1.email.value;
5   var senha = form1.senha.value;
6   var rep_senha = form1.rep_senha.value;
7 }
8 </script>
```

Validando Formulários

- Resumindo, caso o campo nome esteja em branco será exibido uma mensagem de alerta dizendo ao usuário que preencha o campo com seu nome;
- O mesmo pode e será feito com os outros campos;

Validando Formulários

- Podemos verificar também se um campo tem o número de caracteres mínimo/máximo especificado.

```
1 if (nome.length < 5) {  
2   alert('Digite seu nome completo');  
3   form1.nome.focus();  
4   return false;  
5 }
```

Validando Formulários

- Podemos também ver se dois campos coincidem, no caso a senha e sua repetição;

```
1 if (senha != rep_senha) {  
2   alert('Senhas diferentes');  
3   form1.senha.focus();  
4   return false;  
5 }
```



JavaScript

O DOM e JAVASCRIPT

O que é o DOM?

- O DOM é um padrão do W3C (World Wide Web Consortium).
- O DOM define um padrão para acessar documentos:

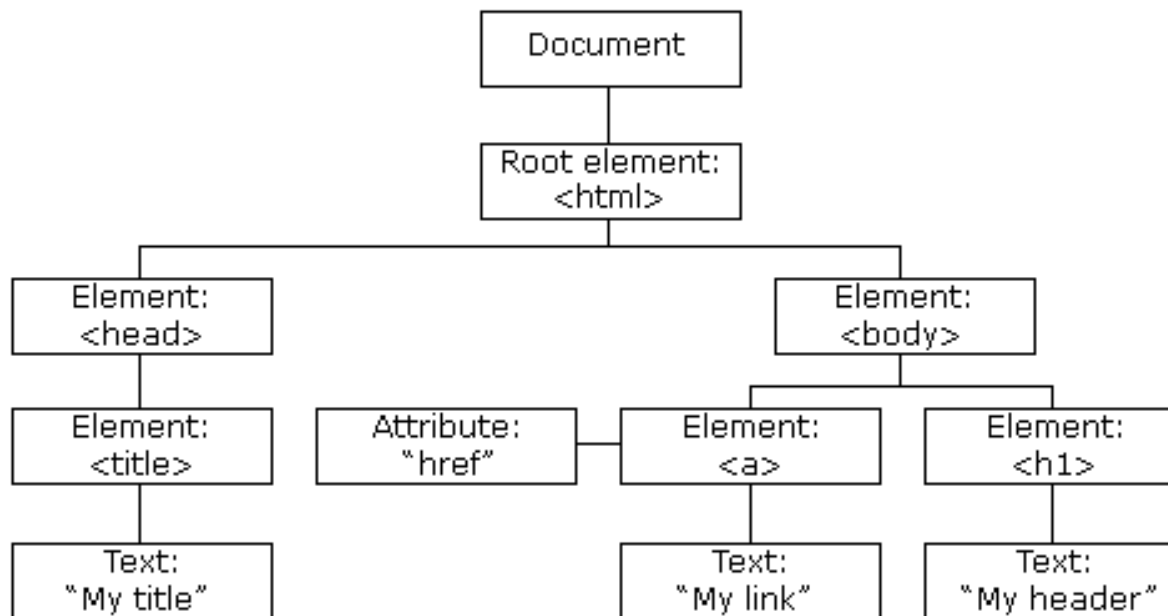
"O modelo de objeto de documento (DOM) do W3C é uma plataforma e interface neutra do idioma que permite que programas e scripts acessem e atualizem dinamicamente o conteúdo, estrutura e estilo de um documento".

Modelo de objetos do HTML - DOM

- Dentro de um documento, a hierarquia de objetos e propriedades JS reflete a hierarquia do HTML, através de um modelo de objetos do documento (Document Object Model - DOM);
- O DOM relaciona cada elemento HTML, respeitando sua hierarquia, a um objeto JS;
- Por exemplo, em HTML um bloco está relacionado a um objeto do tipo Text. Contudo, o elemento deve estar dentro de um bloco , representado por um objeto do tipo Form, que por sua vez, deve estar dentro de um bloco , representado por um objeto do tipo Document;

Modelo de objetos do HTML - DOM

Com o HTML DOM, o JavaScript pode acessar e alterar todos os elementos de um documento HTML.



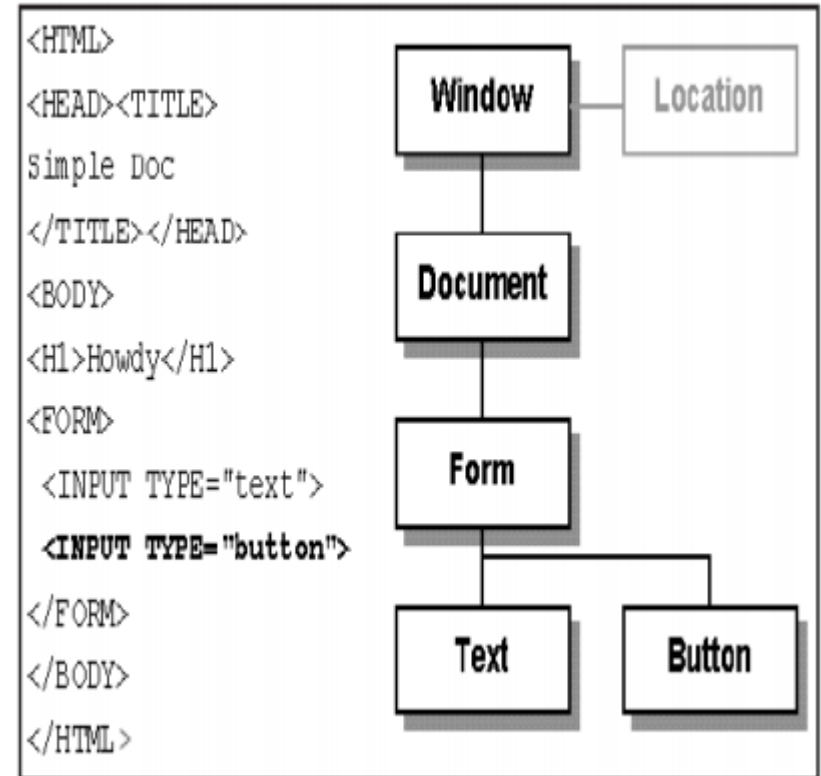
Modelo de objetos do HTML - DOM

Com o modelo de objeto, o JavaScript obtém todo o poder que precisa para criar HTML dinâmico:

- O JavaScript pode Alterar todos os elementos HTML na página
- Alterar todos os atributos HTML na página
- Alterar todos os estilos CSS na página
- Remover elementos e atributos HTML existentes
- Adicionar novos elementos e atributos HTML
- Reagir a todos os eventos HTML existentes na página
- Criar novos eventos HTML na página

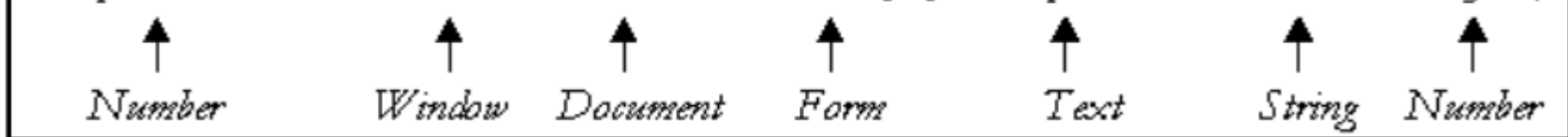
Modelo de objetos do HTML - DOM

- Ao lado vemos exemplos de instâncias de um campo e um botão no DOM, seguindo a hierarquia do HTML.
- OBS: DOM significa Document Object Model, e não tem nada a ver com Domino Object Model.



Acesso a objetos do browser e da página

```
comprimento = window.document.forms[2].campotexto.value.length;
```



Number *Window* *Document* *Form* *Text* *String* *Number*

- A expressão acima mostra como a hierarquia de elementos do HTML se reflete em JS através de propriedades.
- Para ler propriedades ou invocar métodos de um objeto é necessário citar toda a hierarquia de objetos acima dele.

Acesso a objetos do browser e da página

```
comprimento = window.document.forms[2].campotexto.value.length;
```

Diagram illustrating the object hierarchy for the above code:

↑	↑	↑	↑	↑	↑	↑
<i>Number</i>	<i>Window</i>	<i>Document</i>	<i>Form</i>	<i>Text</i>	<i>String</i>	<i>Number</i>

- A única exceção à regra é a referência `window`, que sempre pode ser omitida.
- As instruções abaixo fazem a mesma coisa:

```
window.document.write("Tigres");  
document.write("Tigres");
```

Manipulação de Objetos

- Todos os objetos criados em HTML estão automaticamente disponíveis em JS, mesmo que um nome não seja atribuído a eles.
- Por exemplo, se há três blocos <form> em uma página, há três objetos do tipo Form no JS.
- Se eles não tem nome, pode-se ter acesso através da propriedade 'forms' definida em Document, que fica armazenada em um vetor, que sempre se inicia em 0 no JS.

```
frm1 = document.forms[0];  
/* mesma coisa que window.document.forms[0], ou  
document.meuForm, caso forms[0] seja meuForm */  
frm2 = document.forms[1];
```

Alguns Exemplos

- JavaScript HTML DOM - Alterando CSS

O HTML DOM permite que o JavaScript altere o estilo dos elementos HTML.

```
<! DOCTYPE html>
< html >
< corpo >

< h1 id = "id1" > Meu título 1 </ h1 >

< botão tipo = "botão"
onclick = "document.getElementById ('id1'). style.color = 'red'" >
Clique-me! </ button >

</ body >
</ html >
```

Alguns Exemplos

- JavaScript HTML DOM - Alterando CSS

O HTML DOM permite que o JavaScript altere o estilo dos elementos HTML.

Meu título 1

Clique-me!



Meu título 1

Clique-me!

Alguns Exemplos

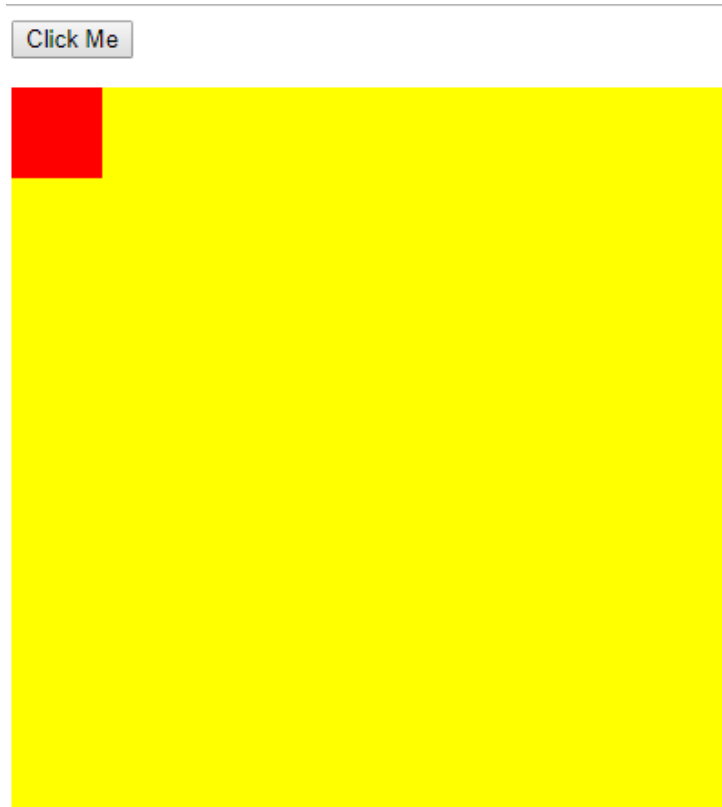
- JavaScript HTML DOM Animation

```
function myMove() {  
    var elem = document.getElementById("animate");  
    var pos = 0;  
    var id = setInterval(frame, 5);  
    function frame() {  
        if (pos == 350) {  
            clearInterval(id);  
        } else {  
            pos++;  
            elem.style.top = pos + 'px';  
            elem.style.left = pos + 'px';  
        }  
    }  
}
```

As animações de JavaScript são feitas pela programação de mudanças graduais no estilo de um elemento.

Alguns Exemplos

- JavaScript HTML DOM Animation



Considerações Finais

A linguagem JavaScript é extremamente grande para conseguirmos cobrir todo o assunto em apenas 3 aulas, portanto, é imprescindível o estudo detalhado de componentes que não foram abordados e sempre estar se atualizando.

Obrigado pela atenção!

Referências

- Disponível em <https://becode.com.br/javascript-para-iniciantes-origens-o-que-e-para-que-serve/> Acesso em Dezembro de 2017.
- MORRISON, Michael. **Use a Cabeça:JavaScript**. Rio de Janeiro: Alta Books, 2008.
- W3Schools. **JavaScript Tutorial**. Disponível online em [<http://www.w3schools.com/js/default.asp>]