



Programação WEB

Linguagem JavaScript

Douglas Nassif Roma Junior

douglas.junior@grupointegrado.br

Conteúdo

- JavaScript
 - Hierarquia
 - Objetos html
 - Form.submit()



JavaScript

- Quando se carrega uma página, o navegador cria uma hierarquia de objetos na memória, que servem para controlar os distintos elementos de tal página.
- Com JavaScript podemos trabalhar com essa hierarquia de objetos, acessar às suas propriedades e invocar seus métodos.

JavaScript

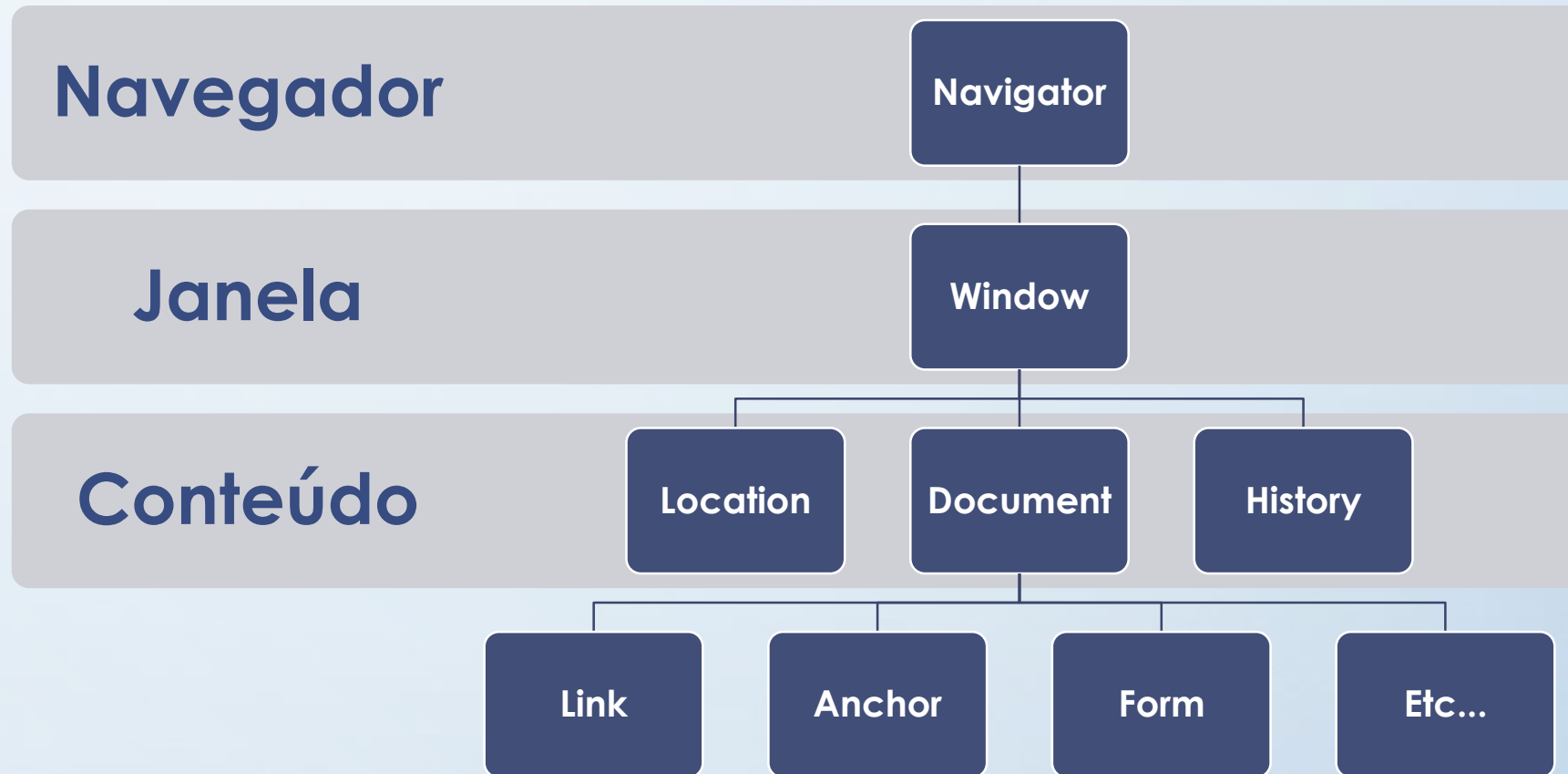
- Qualquer elemento da página pode ser controlado de uma maneira ou de outra acessando a essa hierarquia.
- É crucial conhecê-la bem para poder controlar perfeitamente as páginas web com JavaScript ou qualquer outra linguagem de programação do lado do cliente.

JavaScript

```
<html>
  <head>
    <title>bgColor</title>
  </head>
  <body style="color:white">
    <script type="text/javascript">
      document.body.style.color = "red";
    </script>
  </body>
</html>
```



JavaScript



JavaScript

- Javascript considera HTML uma linguagem orientada a objetos, na qual os diversos tags HTML correspondem a diferentes tipos de objetos JavaScript.
- **Exemplo:**

```
<html>
  <head><title>Minha pagina</title></head>
  <body>
    <form name="formulario1">
      <input type="button" name="botao1">
    </form>
  </body>
</html>
```

JavaScript

- A partir do código anterior, obtêm-se os seguintes objetos JavaScript:

`document.title` : título da página

`document.formulario1` : formulário da página

`document.formulario1.botaol` : botão do formulário

Objeto navigator

Este objeto fornece informações sobre o navegador. Ele apresenta as seguintes propriedades:

- `navigator.appName` : retorna o nome do browser do usuário
- `navigator.appVersion`: retorna a versão do browser e a versão do sistema operacional em que ele está rodando. [formato: número da versão (plataforma; país)]
- `navigator.appCodeName`: retorna o nome do código de desenvolvimento interno do desenvolvedor de um browser específico.
- `navigator.userAgent`: usada em cabeçalhos HTTP para fins de identificação, é a combinação das propriedades `appCodeName` e `appVersion`. Servidores Web usam esta informação para identificar os recursos que o navegador dispõe.

Objeto location

- Este objeto é utilizado para identificar o documento corrente. Ele consiste em uma URL completa no formato: **protocolo//servidor:porta/caminho seguidos de ?search OU #hash.**

Suas propriedades são: **protocol**, **hostname**, **port**, **pathname**, **search**, **hash**.

Cada uma de suas propriedades representa uma parte da URL total.

- **protocol**: retorna o protocolo de transporte do documento.
- **hostname**: identifica o nome do computador hospedeiro.
- **port**: especifica a porta para o endereço. Esta informação é utilizada apenas se uma porta não-padrão estiver sendo usada.

JavaScript

- **pathname:** define o caminho e o nome do arquivo.
- **search:** retorna quaisquer comandos de consulta que possam estar embutidos na URL corrente. Valores de `search` são separados do resto da URL por um sinal de interrogação (“?”).
- **hash:** retorna quaisquer âncoras que possam ter sido passadas na URL. Valores de hash são separados do resto da URL por um sinal de cerquilha (“#”).

Objeto window

- O objeto `window` representa a janela do navegador ou um `frame`. É criado um objeto `window` sempre que o navegador encontra uma tag **<body>** ou **<frameset>**. Também são criados objetos para cada `frame` definido.
- **Propriedades mais utilizadas:**
 - **height:** esta propriedade contém a altura, em pixels, da janela do navegador;
 - **width:** semelhante à propriedade anterior, porém trabalha com a largura;
 - **name:** representa o nome da janela;

Objeto window

- Métodos mais utilizados:

- **back ()** : é equivalente a apertar o botão **voltar** do navegador. Ou seja, volta atrás na última navegação feita pelo usuário;
- **forward ()** : tem o mesmo efeito do botão **avancar** do navegador. Ou seja, se o usuário tiver apertado o botão back para voltar à última página visitada, o forward avança novamente para a página mais recente;
- **open ()** : abre uma nova janela. O método recebe como parâmetros uma URL (o endereço da página que vai ficar na nova janela), o nome da janela e uma string com suas características;
- **close ()** : fecha a janela especificada. O Javascript somente pode fechar automaticamente janelas abertas por ele. Caso contrário, aparece uma caixa de confirmação para o usuário;

Objeto document

- Com o objeto `document` controla-se a página web e todos os elementos que contem.
- O objeto `document` é a página atual que está sendo visualizada nesse momento. Depende do objeto `window`, mas também pode depender do objeto `frame` no caso de que a página esteja sendo mostrada em um frame.

Objeto document

- **alinkColor**
Cor dos links ativos
- **anchor**
Uma âncora da página. Consegue-se com a etiqueta . Acessa-se pelo seu nome.
- **anchors (array)**
Um array das âncoras do documento.
- **applet**
Um applet da página. Acessa-se pelo seu nome. (Javascript 1.1)
- **applets (array)**
Um array com todos os applets da página. (Javascript 1.1)

Objeto document

- **bgColor**
A cor de fundo do documento.
- **classes**
As classes definidas na declaração de estilos CSS. (Javascript 1.2)
- **cookie**
Um cookie
- **domain**
Nome do domínio do servidor da página.
- **Embed**
Um elemento da página rotulado com a etiqueta <EMBED>. Acessa-se pelo seu nome. (Javascript 1.1)
- **embeds (array)**
Todos os elementos da página rotulados com <EMBED>. (Javascript 1.1)

Objeto document

- **fgColor**
A cor do texto. Para ver as mudanças há que reescrever a página.
- **Form**
Um formulário da página. Acessa-se pelo seu nome.
- **forms (array)**
Um array com todos os formulários da página.
- **ids**
Para acessar a estilos CSS. (Javascript 1.2)
- **Image**
Uma imagem da página web. Acessa-se pelo seu nome. (Javascript 1.1)
- **images (array)**
Cada uma das imagens da página introduzidas em um array. (Javascript 1.1)

Objeto document

- **lastModified**
A data de última modificação do documento.
- **linkColor**
A cor dos links.
- **Link**
Um link da página. Acessa-se pelo seu nome.
- **links (array)**
Um array com cada um dos links da página.
- **location**
A URL do documento que se está visualizando. É somente de leitura.

Objeto document

- **referrer**
A página da qual vem o usuário.
- **tags**
Estilos definidos às etiquetas de HTML na página web. (Javascript 1.2)
- **title**
O título da página.
- **URL**
O mesmo que location, mas é aconselhável utilizar location já que URL não existe em todos os navegadores.
- **vlinkColor**
A cor dos links visitados

```
...
<script>
  function mudaCor(colorin) {
    document.bgColor = colorin
  }
</script>
...
<form>
  <input type="Button" value="Vermelho"
        onclick="mudaCor('ff0000')" >
  <input type="Button" value="Verde"
        onclick="mudaCor('00ff00')" >
  <input type="Button" value="Azul"
        onclick="mudaCor('0000ff')" >
</form>
```

```
...
<script type="text/javascript">
  function mudaTitulo(texto) {
    document.title = texto
  }
</script>
...
<form>
  <input type="Button" value="Titulo = Titulo 1"
        onclick="mudaTitulo('Titulo 1')">
  <input type="Button" value="Titulo = Bem-vindos a pagina"
        onclick="mudaTitulo('Bem-vindos a página')">
  <input type="Button" value="Titulo = Outro Titulo"
        onclick="mudaTitulo('Outro Titulo')">
</form>
...
```

JavaScript Exemplos

- Vamos ver um exemplo muito simples sobre como validar um formulário para submetê-lo no caso de que esteja preenchido. Para isso, vamos utilizar o método `submit()` do formulário.
- O mecanismo é o seguinte: em vez de colocar um botão de `submit` colocamos um botão normal (`<INPUT type="button">`) e fazemos que ao clicar esse botão se chame a uma função que é a encarregada de validar o formulário e, no caso de que esteja correto, submete-lo.

JavaScript Exemplos

- O formulário ficaria assim:

```
<form name="meuForm" action="mailto:endereco@servidor.com" >  
  <input type="Text" name="campo1" value="" size="12">  
  <input type="button" value="Enviar" onclick="validaSubmete()">  
</form>
```

- Observamos que não há um botão de `submit`, e sim, um botão normal com uma chamada a uma função que podemos ver a seguir.

```
function validaSubmete() {  
    if (document.meuForm.campo1.value == "")  
        alert("Deve preencher o formulário")  
    else  
        document.meuForm.submit()  
}
```

Dúvidas

