Programação Cliente-servidor Aula 3: JavaScript – Parte 2

Interação com a página HTML

Precisamos ter em mente que o principal objetivo do JavaScript é expandir as funcionalidades básicas de uma página HTML. Para isso, deve ser capaz de interagir com seus componentes e com o utilizador da página.

Neste contexto, temos dois caminhos que funcionam em sentidos inversos:

O uso de DOM permitirá ao JavaScript acessar os componentes da página;

Através de eventos, a página poderá acionar rotinas em JavaScript.

Métodos de acesso DOM

Através de document, acessamos os elementos DOM de diferentes formas, sendo a mais tradicional através do atributo Id

Alguns dos métodos de document para acesso aos componentes da página podem ser observados no quadro:

|  |  |
| --- | --- |
| Método | Retorno |
| getElementById | Retorna um objeto como referência a um componente unicamente identificado através de seu atributo id |
| getElementsByClassName | Retorna uma coleção de objetos referenciando todos os componentes que apresentem o atributo class desejado |
| getElementsByTagName | Retorna uma coleção de objetos referenciando todos os componentes do tipo desejado, como button, div ou h1 |
| querySelector | Retorna um objeto com o primeiro elemento que utilize o seletor CSS |
| querySelectorAll | Retorna uma coleção de objetos com todos os elementos que utilizem o seletor CSS |

Alguns dos atributos que podem ser utilizados pelo nó para a navegação na árvore podem ser observados no quadro:

|  |  |
| --- | --- |
| Atributo | Conteúdo |
| firstChild | Retorna o primeiro filho do nó corrente |
| childNodes | Retorna uma coleção de nós contendo os filhos do nó corrente |
| parentNode | Retorna o nó que ascende ao nó corrente (pai) |
| firstElementChild | Retorna o primeiro filho do tipo Element para o nó corrente |
| lastElementChild | Retorna o último filho do tipo Element para o nó corrente |
| children | Retorna todos os filhos do tipo Element para o nó corrente |

Eventos

Podemos definir evento como uma ação pré-determinada que, ao ocorrer, permite que seja iniciada uma ação personalizada, o que certamente será feito através de programação

document.getElementById("btn1").addEventListener("click",

  function(event) {

   somar();

  });

O método addEventListener recebe, como parâmetros, o nome do evento a ser utilizado e uma função callback para resposta ao evento.

Modificações dinâmicas

Com o uso de DOM, podemos acessar, alterar ou criar elementos em uma página HTML de forma dinâmica, durante a sua visualização no navegador.

A tabela seguinte mostra alguns métodos existentes em document para a criação de nós DOM.

|  |  |
| --- | --- |
| Método |  |
| createElement | Cria um nó do tipo Elemento (tag) |
| createTextNode | Cria um nó de texto |
| createAttribute | Cria um atributo. O nó pode utilizar com setAttributeNode |
| createComment | Cria um comentário |

Outra forma muito comum de modificação de conteúdo é com o uso do atributo innerHTML dos nós DOM. Com ele, podemos colocar qualquer conteúdo HTML dentro da tag representada pelo nó de elemento.

document.getElementById(“minhaDiv”).innerHTML = "<h1>TESTE</h1>";

Um formulário é uma entrada de dados simples, que considera apenas texto e seleções, mas sem grandes críticas acerca de formato e validade destes dados.

Entre diversas outras ações, devemos nos preocupar com a necessidade de:

Definir campos obrigatórios;

Utilizar tipos de dados específicos;

Controlar a visibilidade de campos alternativos.

Por que validar no cliente se temos que validar no servidor novamente?

A resposta tem a ver com usabilidade e fluxo de rede, pois a resposta da validação de formato no cliente é mais rápida, além de diminuir o fluxo de rede com chamadas desnecessárias ao servidor, já que os dados serão criticados antes do envio.

Assim como em todos os demais elementos de interatividade da página com as rotinas JavaScript, também na validação contamos com os eventos para definir o momento de acionamento da crítica ou formatação.

Outras validações e formatações podem ser efetuadas no momento da perda do foco pela caixa de texto, ou quando selecionamos o elemento de uma lista de valores, entre diversas outras opções.

Observe, no quadro seguinte, alguns eventos do HTML e suas respectivas aplicações no processo de validação.

|  |  |
| --- | --- |
| Evento | Aplicação |
| onsubmit | Efetua a validação do formulário imediatamente antes do envio para o servidor. Necessita o retorno booleano, indicando se os valores podem ser enviados ou não |
| onclick | Normalmente uma chamada explicita de validação. Muito utilizado em botões de rádio e caixas de marcação |
| onchange | Ocorre quando o valor (value) sofre uma alteração |
| onfocus | Ocorre quando o componente ganha o foco. Pode ser utilizado, por exemplo, para apagar o valor do campo |
| onblur | Ocorre na perda do foco pelo componente. É comum a aplicação de máscaras em valores numéricos como CEP e CPF |
| onsearch | Este evento é iniciado quando um usuário digita algo em um campo de pesquisa (type=”search”) |
| onselect | Utilizado quando algum texto é selecionado no campo |

Orientação a objetos

Abstração

refere à definição de um modelo simplificado de algo maior.

Protótipo

O uso de protótipo é uma peculiaridade do JavaScript e não uma característica própria da orientação a objetos.

Na verdade, o JavaScript não apresenta um mecanismo específico para herança, que seria outro dos pilares da orientação a objetos, mas permite a utilização de prototype para expandir a funcionalidade de classes já existentes.

function Pessoa(nome, sobrenome){

this.nome = nome;

this.telefone = sobrenome;

}

Com o uso de prototype, podemos adicionar a nacionalidade para esta classe já existente. Claro, que exigirá uma inicialização prévia, já que o construtor não tem como prever o novo atributo.

Pessoa.prototype.nacionalidade = "Brasileiro(a)";

Da mesma forma que podemos adicionar um atributo, podemos adicionar um método a esta classe com o uso de prototype.

Pessoa.prototype.nomeCompleto = function( ) {

return this.nome + " "+this.sobrenome;

};

Observando o trecho de HTML abaixo, qual seria a instrução JavaScript para obter acesso ao elemento DIV e colocar nele a frase “EXERCICIO DOM”?

<div id="XPTO">ALVO</div>

a) document.getElementById("XPTO").value = "EXERCICIO DOM";

b) document.querySelector("#XPTO").value = "EXERCICIO DOM";

X c) document.getElementById("XPTO").innerHTML = "EXERCICIO DOM";

d) document.querySelector(".XPTO").innerHTML = "EXERCICIO DOM";

e) document.querySelector("DIV").value = "EXERCICIO DOM";

2. Você está criando um formulário para cadastro de leitores de um jornal na Web, onde devem constar os dados residenciais do leitor. A empresa pediu que dados como rua, bairro, cidade e estado sejam preenchidos automaticamente após o leitor digitar o CEP e sair da caixa de texto. Qual evento deve ser utilizado para efetuar este preenchimento?

X a) onblur

b) onclick

c) onexit

d) onenter

e) onchange

3. As linguagens da atualidade buscam metodologias mais organizadas para a programação, e a orientação a objetos acaba sendo amplamente adotada com este objetivo. O JavaScript também permite o uso desta metodologia, e, para definir atributos de uma classe, é utilizada uma palavra reservada específica. Qual a palavra utilizada?

a) function

b) new

c) inherited

X d) this

e) super

|  |  |
| --- | --- |
| |  | | --- | | Quanto à validação de formulários em Java Script, é correto afirmar que: | |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | As críticas relativas à formatação são realizadas com sucessivas chamadas ao servidor. |
| Certo |  | Efetua críticas relativas à formatação e regras simples de interface, ao mesmo tempo em que diminui o fluxo de chamadas desnecessárias ao servidor. |
|  |  | Permite se utilizar de recursos locais, entre eles o acesso a bancos de dados e arquivos presentes no computador cliente. |
|  |  | Quando associada ao evento de envio do formulário necessita de uma função sem retorno de valor. |
|  |  | Pode ser utilizada para substituir os testes de consistência referentes aos relacionamentos do banco de dados, normalmente efetuados no servidor. |
| |  | | --- | | A afirmativa apresentada na questão diz respeito a que evento?  "Efetua a validação do formulário imediatamente antes do envio para o servidor. Necessita o retorno booleano, indicando se os valores podem ser enviados ou não". | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
|  |  | onclick |
|  |  | onsearch |
|  |  | onselect |
| Certo |  | onsubmit |
|  |  | onblur |
| |  | | --- | | Marque a resposta correta em relação a afirmativa.  "Retorna uma coleção de objetos referenciando todos os componentes que apresentem o atributo class desejado." A afirmativa faz referência a: | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
|  |  | querySelectorAll |
|  |  | querySelector |
|  |  | getElementById |
| Certo |  | getElementsByClassName |
|  |  | getElementsByTagName |
| |  | | --- | | Diversos eventos podem ser utilizados no processo de validação de formulários. Qual deles deve ser utilizado para efetuar críticas quando o valor de um campo é alterado? | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
|  |  | onsubmit |
|  |  | onblur |
|  |  | onclick |
| Certo |  | onchange |
|  |  | onfocus |
| |  | | --- | | Caso fosse necessário validar um campo para CPF, de acordo com a regra de formação do número, logo que o usuário saísse da caixa de texto, o evento correto seria: | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
|  |  | onclick |
| Certo |  | onblur |
|  |  | onchange |
|  |  | onsubmit |
|  |  | onfocus |
| |  | | --- | | Quando falamos das ações que um objeto pode realizar estamos nos referindo a: | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
|  |  | Nível de Acesso |
|  |  | Agrupamentos |
| Certo |  | Métodos |
|  |  | Atributos |
|  |  | Herança |