XML
Extensible Markup Language

Professor Vinícius Costa

CONTINUAÇÃO - DOM

Alterando o Valor de um Elemento

  - <mensagem id="m">Ola Mundo</mensagem>
- Não é possível mudar o valor do elemento "mensagem"
- É preciso pegar o nó texto, filho de "mensagem" e então alterar sua propriedade nodeValue
- Exemplo
  - document.getElementById("m").firstChild.nodeValue="Oi Planeta";

Alterando o Valor de um Atributo

- Atributos são nós
- Diferente dos nós elementos os atributos tem nodeValue
- É possível alterar o valor de um atributo de duas formas
  - Usando o método setAttribute( ) do elemento
  - Atribuindo um valor para a propriedade nodeValue do atributo

Método setAttribute

- <mensagem id="m">Ola Mundo</mensagem>
- document.getElementById("m").setAttribute("id","n");
  - Se o atributo não existir um novo atributo será criado

Atributo nodeValue

- A propriedade "nodeValue" é o valor do atributo
- Mudar o valor de "nodeValue" muda o valor do atributo
  - <mensagem id="m">Ola Mundo</mensagem>
  - document.getElementById("m").getAttributeNode("id").nodeValue="n";
- A propriedade nodeValue muda o valor de um atributo já existente

Remover um Elemento Filho

- O método removeChild( ) remove um nó especifico
- Quando um nó é removido todos os seus filhos são removidos juntos
- Exemplo
  - <mensagem id="m">Ola Mundo</mensagem>
  - mens = document.getElementById("m");
  - pai.removeChild(mens);

Remover o Próprio Elemento

- <mensagem id="m">Ola Mundo</mensagem>
- mens = document.getElementById("m");
- mens.parentNode.removeChild(mens);

Romover um Nó Texto

- O metodo "removeChild" pode ser utilizado para remover um nó texto
  - <mensagem id="m">Ola Mundo</mensagem>
  - mens = document.getElementById("m");
  - texto = mens.firstChild;
  - mens.removeChild(texto);
- Outra alternativa é limpar o nó texto
  - document.getElementById("m").firstChild.nodeValue="";

Remover Atributo pelo Nome

- O metodo removeAttribute remove um atributo utilizando seu nome
  - <mensagem id="m">Ola Mundo</mensagem>
  - mens = document.getElementById("m");
  - mens.removeAttribute("id")

Remover Atributo pelo Nó Atributo

- O metodo removeAttributeNode remove um atributo utilizando o próprio atributo
  - <mensagem id="m">Ola Mundo</mensagem>
  - mens = document.getElementById("m");
  - atributo = mens.getAttributeNode("id");
  - mens.removeAttributeNode(atributo);

Criar Novo Elemento

- O método createElement cria um novo nó
  - novoElemento = xmlDoc.createElement(“novo”);

Criar um Nó Texto

- O método createTextNode cria um nó texto
  - noTexto=elemento.createTextNode(“Ola Mundo”);

Criar Novo Atributo

- O método createAttribute cria um novo atributo
  - novoAtrib = xmlDoc.createAttribute(“novoAtrib);
  - novoAtrib.nodeValue=“novo valor”;
  - elemento.setAttributeNode(novoAtrib)
- Se o atributo já existir ele será substituído
- É possível criar um novo atributo com setAttribute
  - elemento.setAttribute(“novoAtrib”,”novo valor”);

Criar um Nó CDATA

- O método createCDATASection cria uma nova seção CDATA
  - novoCDATA=xmlDoc.createCDATASection(“texto com & e com <“)

Criar um Nó Comentário

- O método createComment cria um novo comentário
  - novoCom=xmlDoc.createComment(“comentário”);

Adicionando Nó a Árvore de Nós

- Criar um elemento ou atributo não significa que ele faz parte da árvore de documento
- Após criar o elemento ou atributo é preciso inseri-lo na árvore de documentos

Adicionando um Nó Filho

- O método appendChild
  - adiciona um novo nó como filho de um nó já existente
  - O novo nó será adicionado como último filho
- Exemplo
  - novo = xmlDoc.createElement(“novo”);
  - pai.appendChild(novo);

Adicionando um Novo Nó com Texto

- Exemplo
  - novoEle = xmlDoc.createElement(“novo”);
  - novoTex=xmlDoc.createTextNode(“Ola Mundo”);
  - novoEle.appendChild(novoTex);
  - pai.appendChild(novoEle);

Adicionando Novo Nó Antes do Irmão

- O método insertBefore adiciona um novo elemento antes de um elemento especificado
- Exemplo
  - novoEle = xmlDoc.createElement(“novo”);
  - antigoEle=xmlDoc.getElementById(“antigo”);
  - pai.insertBefore(novoEle,antigoEle);

Adicionando Texto a um Nó Texto

- O método insertData insere texto a um nó texto já existente
- Parâmetros
  - Offset – onde iniciará a colocar o novo texto
  - String – o novo texto que será inserido
- Exemplo
  - elemento.firstChild.insertData(0,”novo texto”);

Trocar Elementos

- O método replaceChild é usado para trocar um nó por outro
  - pai.replaceChild(novoFilho,antigoFilho);

Trocar Texto de um Nó Texto

- O método replaceData é utilizado para trocar texto
- Parâmetros
  - Offset - início para começar a troca de caracteres
  - Length – quantidade de caracteres trocados
  - String – texto que será inserido
- Exemplo
  - <mensagem id="m">Ola Mundo</mensagem>
  - noTexto = document.getElementById("m").firstChild;
  - noTexto.replaceData(0,3,"Oi");

Clonar um Nó

- O método cloneNode faz uma copia de um nó especificado
- O método cloneNode espera um parâmetro booleano. Se o parâmetro for true todos os atributos e nós filhos serão também clonados
- Exemplo
  - copia = elemento.cloneNode(true);