

XML

Extensible Markup Language

Professor Vinícius Costa

DOM

- Document Object Model
- É uma plataforma e interface de linguagem neutra que permite aos programas e scripts acessar e atualizar de forma dinâmica o conteúdo, estrutura e estilo de um documento
- O DOM define um padrão para acessar e manipular documentos

DOM

- HTML DOM
 - Define uma forma padronizada de acessar e manipular documentos HTML. Representa um documento HTML como uma estrutura em arvore.
- XML DOM
 - Define uma forma padronizada de acessar e manipular documentos XML. Representa um documento XML como uma estrutura em arvore.

XML DOM

- Todo elemento XML pode ser acessado através do XML DOM
- O XML DOM é:
 - Um modelo de objeto padrão para XML
 - Uma interface de programação padrão para XML
 - Plataforma e independente de linguagem
 - Um padrão W3C
- Resumindo: XML DOM é um padrão para pegar, modificar, adicionar ou deletar elementos XML

Exemplo XML DOM

```
<body>
  <p id="demo"></p>
  <script>
    var text, parser, xmlDoc;
    text = "<bookstore><book>" +
    "<title>Everyday Italian</title>" +
    "<author>Giada De Laurentiis</author>" +
    "<year>2005</year>" +
    "</book></bookstore>";
    parser = new DOMParser();
    xmlDoc = parser.parseFromString(text,"text/xml");
    document.getElementById("demo").innerHTML =
    xmlDoc.getElementsByTagName("author")[0].childNodes[0].nodeValue;
  </script>
</body>
```

Exemplo XML DOM

```
<body>
  <p id="demo"></p>
  <script src="codigo.js"></script>
  <script>
    var parser, xmlDoc;

    parser = new DOMParser();
    xmlDoc = parser.parseFromString(text,"text/xml");
    document.getElementById("demo").innerHTML =
      xmlDoc.getElementsByTagName("year")[0].childNodes[0].nodeValue;
  </script>
</body>
```

```
var text = "<bookstore><book>" +
  "<title>Everyday Italian</title>" +
  "<author>Giada De Laurentiis</author>" +
  "<year>2005</year>" +
  "</book></bookstore>";
```

Interface de Programação

- DOM enxerga o XML como um conjunto de nós.
- Os nós podem ser acessados com JavaScript ou outra linguagem de programação
- A Interface DOM define um conjunto de propriedades e métodos

Nós DOM

- Tudo em um documento XML é um nó (node)
 - O documento inteiro é um nó documento
 - Cada elemento XML é um nó elemento
 - O texto em um documento XML é um nó texto
 - Atributos são nó atributo
 - Comentários são nó comentário

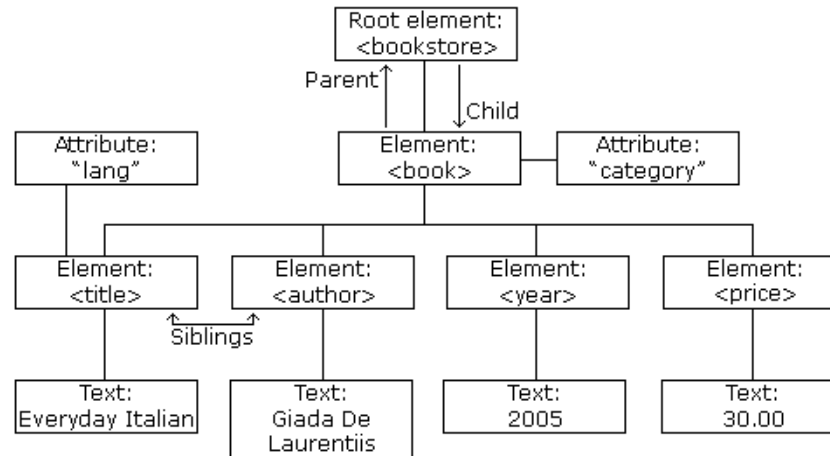
Nó Texto

- Texto são sempre armazenados dentro de nó texto.
- Exemplo
 - `<ano>2018</ano>`
 - Dentro da tag ano existe um nó do tipo texto e o valor deste nó é 2018.

Árvore de Nó XML DOM

- Um documento XML é visto pelo DOM como um estrutura em árvore
- Esta estrutura é chamada de árvore de nós (**node-tree**)
- Todos os nós podem ser acessados através da árvore
- O conteúdo dos nós pode ser modificado ou deletado e novos elementos podem ser criados
- A árvore de nós mostra os nós e suas conexões
- A árvore inicia na raiz e ramifica até chegar aos nós do tipo texto

Árvore de Nó XML DOM

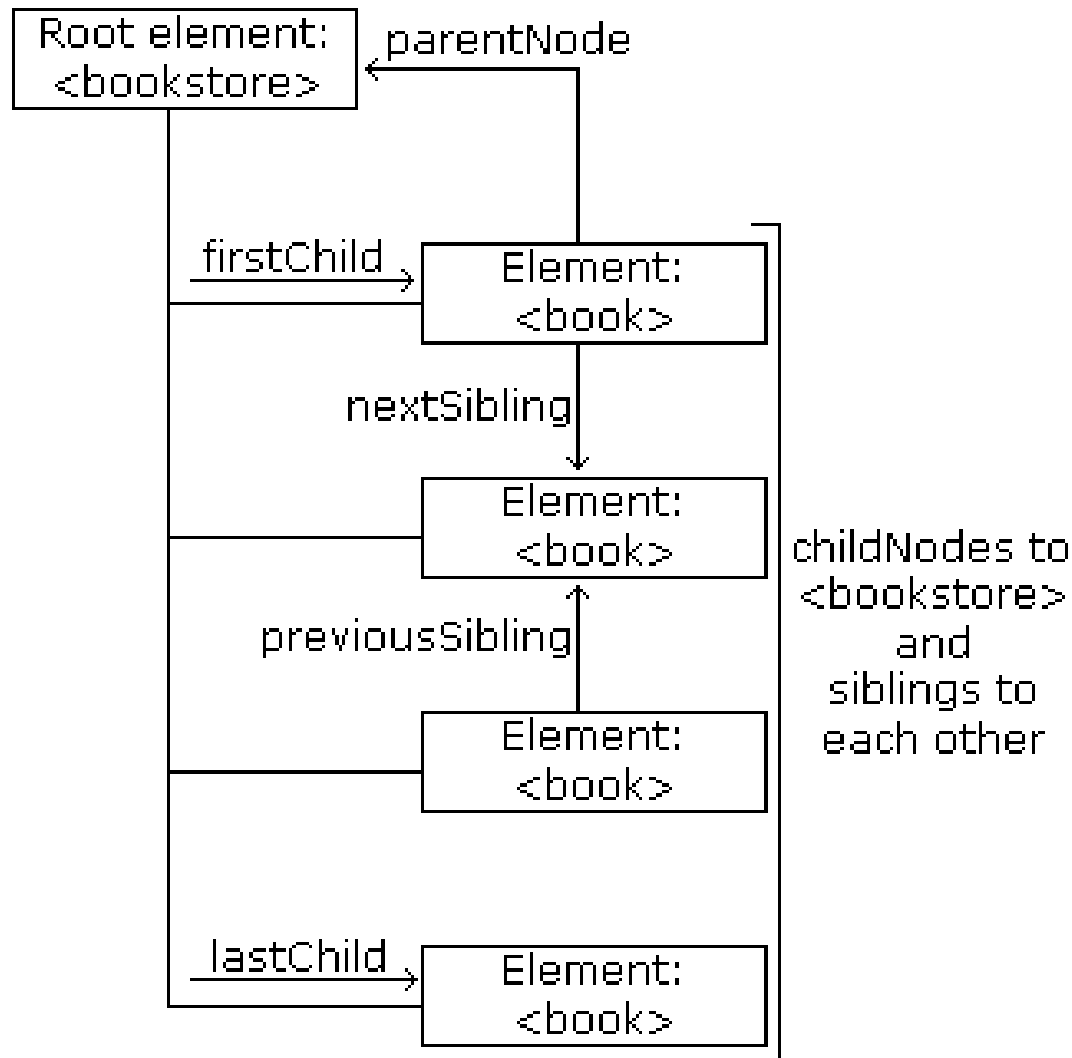


```
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>
```

Nó Pai, Filho e Irmão

- Os nós em uma árvore de nós apresentam um relacionamento hierárquico
- Nós pais tem filhos
- Filhos no mesmo nível (mesmo pai) são irmãos
- O nó mais ao topo é o nó raiz
- Cada nó, exceto o nó raiz, tem um único pai
- Um nó pode ter zero ou mais filhos
- Um nó folha é um nó sem filhos

Árvore de Nós e Seus Relacionamentos



Acessando Nós

- É possível acessar os nós das seguintes formas
 - Usando o método `getElementByTagName()`
 - Usando um laço para percorrer a árvore de nós
 - Navegando na árvore de nós pelos relacionamentos entre os nós

O Método `getElementsByTagName()`

- Retorna todos os elementos de um determinado tipo de tag
- Sitnaxe
 - `Node.getElementsByTagName("nomeDaTag")`
- Exemplo
 - `body.getElementsByTagName("p")`
- Para pegar todos os elementos de uma determinada tag em um documento XML
 - `xmlDoc.getElementsByTagName("elemento")`

Nó Lista do DOM

- O método `getElementsByTagName` retorna uma lista de nós (um array de nós)
- Exemplo
 - `lista = pai.getElementsByTagName("p")`
 - A variável `lista` é um array de parágrafos onde o índice começa em 0
 - `lista[0]` é o primeiro elemento da lista
 - `lista.length` é o tamanho do array `lista`

Nó Lista do DOM

```
var x = xmlDoc.getElementsByTagName("title");  
for (i = 0; i < x.length; i++) {  
    // faz algo com cada nó  
}
```

Tipos de Nós

- `documentElement`
 - Propriedade do documento XML que indica o nó raiz
- `nodeName`
 - Propriedade de um nó que indica seu nome
- `nodeType`
 - Propriedade de um nó que indica seu tipo

Laço para Percorrer a Árvore de Nós

- Um laço que percorre a árvore de nós passando por todos os nós filhos do nó raiz
- Exemplo

```
txt = "";  
x = xmlDoc.documentElement.childNodes;  
for (i = 0; i < x.length; i++) {  
    // Process only element nodes (type 1)  
    if (x[i].nodeType == 1) {  
        txt += x[i].nodeName + "<br>";  
    }  
}
```

Navegando pelo Relacionamento dos Nós

- Navega na árvore de nós utilizando o relacionamento que os nós tem
- Exemplo

```
x = xmlDoc.getElementsByTagName("body")[0];
xlen = x.childNodes.length;
y = x.firstChild;
txt = "";
for (i = 0; i < xlen; i++) {
    // Process only element nodes (type 1)
    if (y.nodeType == 1) {
        txt += y.nodeName + "<br>";
    }
    y = y.nextSibling;
}
```

Propriedade dos Nós

- Cada nó do XML DOM é um objeto (object)
- Objetos tem métodos e propriedades que podem ser manipulados via JavaScript
- Três importantes propriedades são:
 - nodeName
 - nodeValue
 - nodeType

Propriedade nodeName

- Especifica o nome do nó
- É somente leitura
- nodeName de um elemento nó é o mesmo nome da tag
- nodeName de um nó atributo é o nome do atributo
- nodeName de um nó texto é sempre #text
- nodeName de um nó documento é sempre #document

Propriedade nodeValue

- Especifica o valor do nó
- nodeValue de um nó elemento é indefinido (undefined)
- nodeValue de um nó texto é o próprio texto
- nodeValue de um nó atributo é o valor do atributo

Propriedade nodeValue

- Pegando o valor de um elemento
`var texto=pai.getElementsByTagName("p")[0].
firstChild.nodeValue;`
- Alterando o valor de um elemento
`Pai.getElementsByTagName("p")[0].
firstChild.nodeValue = "Ola Mundo";`

Propriedade nodeType

- Especifica o tipo de nó
- É somente leitura
- Os principais tipos de nós são:

Tipo de Nó	nodeType
Elemento	1
Atributo	2
Texto	3
Comentário	8
Documento	9

Lista de Nó XML DOM

- Uma lista de nó é retornada pelo método `getElementsByTagName` e pela propriedade `childNodes`
- Os nós da lista são acessados pelo índice numérico que começa em 0
- Exemplo
 - `lista = pai.getElementsByTagName("p")`
 - `var t= lista[0].childNodes[0].nodeValue`
 - Guarda na variável “t” o texto do primeiro elemento “p” da “lista”

Tamanho da Lista de Nós

- A propriedade length mantém a quantidade de itens em uma lista
- A propriedade “length” é atualizada sempre que a lista é modificada (elemento é adicionado ou removido)

Lista de Atributos DOM

- A propriedade “attributes” retorna uma lista de atributos de um nó
- Exemplo
 - listaAtributo = e.attributes;
 - Pega uma lista de atributos do elemento “e”
 - t = listaAtributo.getItem(“id”).nodeValue;
 - Pega o valor da propriedade “id” do elemento “e”
 - tamanho = listaAtributo.length;
 - Pega a quantidade de atributos que o elemento “e” contém

Atravessando a Árvore de Nós

- Percorrer toda a árvore de nós utilizando um loop
 - Por exemplo, para pegar o nome e valor de cada elemento
 - Construir uma string XML
 - Transformar a string em um objeto XML DOM
 - Pegar todos os filhos do elemento raiz
 - Para cada elemento mostrar o nome do nó e seu respectivo valor

```
<body>
  <p id="demo"></p>
  <script>
    var x, i, xmlDoc;
    var txt = "";
    var text = "<book>" +
      "<title>Everyday Italian</title>" +
      "<author>Giada De Laurentiis</author>" +
      "<year>2005</year>" +
      "</book>";
    parser = new DOMParser();
    xmlDoc = parser.parseFromString(text,"text/xml");
    x = xmlDoc.documentElement.childNodes;
    for (i = 0; i < x.length ;i++) {
      txt += x[i].nodeName + ": " + x[i].childNodes[0].nodeValue + "<br>";
    }
    document.getElementById("demo").innerHTML = txt;
  </script>
</body>
```

Parsing em Diferentes Navegadores

- Todos os navegadores modernos suportam a especificação W3C DOM
- Diferença entre navegadores
 - A forma em que os navegadores manipulam espaços em branco e novas linhas
 - Internet Explorer 9 e anteriores não colocavam espaços em branco ou caractere de nova linha como nó do tipo texto na árvore de nós, diferente dos outros navegadores

PCDATA – Parsed Charset Data

- O texto dentro de um elemento é analisado pelo parser XML
 - `<mensagem>este texto será analisado</mensagem>`
- O parser analisa o conteúdo dentro de um elemento pois pode haver outros elementos dentro dele
 - `<nome><primeiro>José</primeiro><ultimo>Carlos</ultimo></nome>`

CDATA – (Unparsed) Character Data

- CDATA é utilizado quando não se quer analisar seu conteúdo
- Por exemplo, quando se utiliza os seguintes caracteres
 - “<” que indica o começo de um novo elemento
 - “&” que indica o começo de uma nova entidade
- Para que o parser não analise o texto o conteúdo deve ser do tipo CDATA

CDATA – (Unparsed) Character Data

- Uma seção CDATA inicia em “<![CDATA[“ e termina em “]]>”

- Exemplo

```
<codigo>
```

```
<![CDATA[
```

```
    if (a < b && a < 0 )
```

```
        return 1;
```

```
]]>
```

```
</codigo>
```

Navegando nos Nós XML DOM

- Os nós podem ser navegados utilizando o relacionamento entre eles
- O relacionamento é definido pelas propriedades dos nós
 - parentNode
 - childNodes
 - firstChild
 - lastChild
 - nextSibling
 - previousSibling

Propriedade nodeValue

- Usada para pegar o texto de um elemento do tipo texto
 - `<p id="parag">Texto do parágrafo</p>`
 - O primeiro filho da tag “p” é um nó do tipo texto e seu `nodeValue` é o texto propriamente dito
 - `texto = document.getElementById("parag").firstChild.nodeValue;`

Método `getAttribute`

- Retorna o valor do atributo
 - `valorId = document.getElementById("parag").getAttribute("id");`

Método `getAttributeNode`

- Retorna o nó atributo
 - `valorId =`
`document.getElementById("parag").getAttributeNode("id").nodeValue;`