



Departamento de Lenguajes y
Sistemas Informáticos



Universitat d'Alacant
Universidad de Alicante

PHP y XML

Programación en Internet
Curso 2009-2010

Programación en Internet – Curso 2009-2010

Índice

- Introducción
- El DOM en PHP

Programación en Internet – Curso 2009-2010

Introducción

- ¿Cómo se trabaja con un documento XML desde el punto de vista de un programador?

Programación en Internet – Curso 2009-2010

Introducción

- DOM:
 - Especificación de W3C
 - Mecanismo para que desde cualquier lenguaje de programación se pueda manipular un documento XML o HTML
- Básicamente:
 - Crea una estructura lógica para el documento (llamada “árbol nodal”)
 - Proporciona un método para leer y manipular el documento

Introducción

- DOM trata un documento XML como una jerarquía de nodos
 - Distintos nodos y cada uno tiene permitido solo ciertos hijos de ciertos tipos de nodo
 - Existen algunos tipos de nodo que son hoja → No pueden tener hijos

Introducción

Tipo de nodo	Hijos permitidos
Document	Element (<i>uno como máximo</i>) ProcessingIntruction Comment DocumentType
DocumentFragment	Element ProcessingIntruction Comment Text CDATASection EntityReference
DocumentType	<i>Ninguno</i>
EntityReference	Element ProcessingIntruction Comment Text CDATASection EntityReference

Programación en Internet – Curso 2009-2010

Introducción

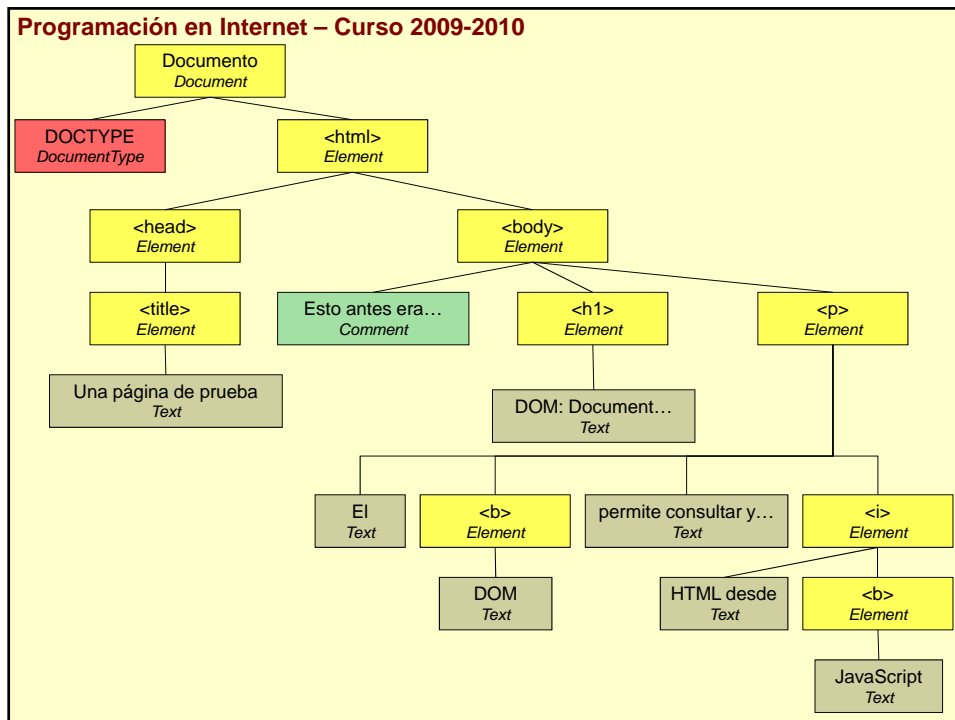
Tipo de nodo	Hijos permitidos
Element	Element ProcessingIntruction Comment Text CDATASection EntityReference
Attr	Text EntityReference
ProcessingInstruction	<i>Ninguno</i>
Comment	<i>Ninguno</i>
Text	<i>Ninguno</i>
CDATASection	<i>Ninguno</i>
Entity	Element ProcessingIntruction Comment Text CDATASection EntityReference
Notation	<i>Ninguno</i>

Programación en Internet – Curso 2009-2010

Introducción

- Ejemplo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es"
lang="es">
<head>
<title>Una página de ejemplo</title>
</head>
<body>
<!-- Esto antes era un ejemplo un poco simple, pero ahora ya no --
>
<h1>DOM: Document Object Model</h1>
<p>
El <b>DOM</b> permite consultar y modificar un documento <i>HTML
desde <b>JavaScript</b></i>
</p>
</body>
</html>
```



Programación en Internet – Curso 2009-2010

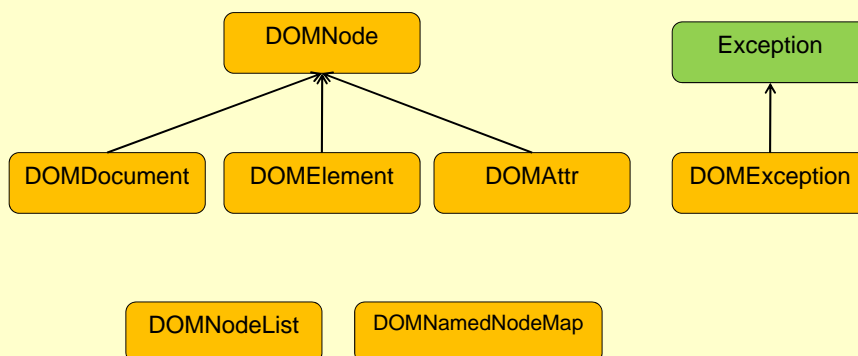
El DOM en PHP

- Desde PHP 5, está disponible como parte del núcleo de PHP la extensión DOM
- En PHP 4 hay que usar DOM XML

El DOM en PHP

- Clases principales:
 - **DOMDocument**: representa el nivel superior de un documento XML (pero no el elemento raíz)
 - **DOMException**: representa un error producido en el documento XML
 - **DOMNode**: representa un nodo en el documento XML
 - **DOMElement**: representa un elemento (etiqueta)
 - **DOMAttr**: representa un atributo
 - **DOMNodeList**: representa una lista de nodos
 - **DOMNamedNodeMap**: representa una lista de nodos que se pueden acceder por un índice o por un nombre (se emplea para representar los atributos)

Jerarquía de clases



Programación en Internet – Curso 2009-2010

DOMDocument

- Representa un documento XML (XHTML)
- Funciona como raíz del árbol del documento
- Hereda de DOMNode

Programación en Internet – Curso 2009-2010

DOMDocument

- Propiedades:
 - `doctype`: contiene el DOCTYPE
 - `documentElement`: el elemento raíz del documento
 - `documentURI`: la localización del documento
 - `formatOutput`: formatea la salida “bonita” con sangría y espacios en blanco extra
 - `textContent`: contenido textual del documento
 - `validateOnParse`: indica si se tiene que validar el documento durante la carga

DOMDocument

- Métodos:
 - load: carga un documento (local o remoto)
 - loadXML: carga un documento a partir de una cadena
 - save: guarda un documento en un fichero
 - saveXML: guarda un documento en una cadena
 - schemaValidate: valida un documento contra su esquema
 - validate: valida un documento contra su DTD

DOMDocument

- Crear nuevo contenido:
 - createAttribute: crea un atributo
 - createComment, createElement, createProcessingInstruction, createTextNode, ...: crea un nodo concreto

DOMDocument

- Seleccionar contenido de un documento:
 - `getElementById`: busca y devuelve el elemento con el ID indicado; devuelve un `DOMElement`
 - Para que funcione, se debe crear un ID con `setIdAttribute` o se debe proporcionar un DTD que defina el ID
 - `getElementsByTagName`: busca y devuelve todos los elementos con la etiqueta indicada; devuelve un `DOMNodeList`

DOMDocument

- Ejemplos:
 - Cargar y mostrar un documento XML a partir de un fichero
 - Cargar y mostrar un documento XML a partir de una URL
 - Cargar, mostrar y guardar un documento XML a partir de una cadena

DOMDocument

- A partir de un fichero:

```
<?php
$objDOM = new DOMDocument();
$objDOM->load("biblioteca.xml");

echo "<pre>\n";
echo "textContent:\n";
echo htmlspecialchars($objDOM->textContent);
echo "XML:\n";
echo htmlspecialchars($objDOM->saveXML());
echo "</pre>\n";
?>
```

DOMDocument

- A partir de una URL:

```
<?php
$objDOM = new DOMDocument();
$objDOM->load("http://www.dlsi.ua.es/~slujan/files/biblioteca.xml");

echo "<pre>\n";
echo "textContent:\n";
echo htmlspecialchars($objDOM->textContent);
echo "XML:\n";
echo htmlspecialchars($objDOM->saveXML());
echo "</pre>\n";
?>
```

Programación en Internet – Curso 2009-2010

DOMDocument

- A partir de una cadena:

```
<?php
```

```
$doc = '<?xml version="1.0" encoding="ISO-8859-1"
standalone="no"?>';
$doc .= '<!DOCTYPE BIBLIOTECA SYSTEM
"biblioteca.dtd">';
$doc .= '<BIBLIOTECA>';
$doc .= '<LIBRO COD="1">';
$doc .= '<TITULO>XML para todos</TITULO>';
$doc .= '<AUTOR>Sergio Lujan Mora</AUTOR>';
$doc .= '<ANYO>2001</ANYO>';
$doc .= '<EDITORIAL>UA Prensa</EDITORIAL>';
$doc .= '</LIBRO>';
```

Programación en Internet – Curso 2009-2010

DOMDocument

```
$doc .= '<LIBRO COD="11">';
$doc .= '<TITULO>Como aprobar una oposición</TITULO>';
$doc .= '<AUTOR>Marisa Zayas Fornieles</AUTOR>';
$doc .= '<AUTOR>Sergio Lujan Mora</AUTOR>';
$doc .= '<ANYO>1999</ANYO>';
$doc .= '<EDITORIAL>Prensa Editorial</EDITORIAL>';
$doc .= '</LIBRO>';
$doc .= '</BIBLIOTECA>';

$objDOM = new DOMDocument();
$objDOM->loadXML($doc);

echo "<pre>\n";
echo "textContent:\n";
echo htmlspecialchars($objDOM->textContent);
echo "XML:\n";
echo htmlspecialchars($objDOM->saveXML());
echo "</pre>\n";
?>
```

DOMNode

- Propiedades:
 - attributes: contiene la lista de atributos
 - nodeName: el nombre del nodo (etiqueta)
 - nodeValue: el texto asociado con el nodo
 - nodeType: especifica el tipo de un nodo
 - textContent: contenido texto del nodo y todos sus descendientes

XML constants		
Constant	Value	Description
XML_ELEMENT_NODE (integer)	1	Node is a DOMElement
XML_ATTRIBUTE_NODE (integer)	2	Node is a DOMAttr
XML_TEXT_NODE (integer)	3	Node is a DOMText
XML_CDATA_SECTION_NODE (integer)	4	Node is a DOMCharacterData
XML_ENTITY_REF_NODE (integer)	5	Node is a DOMEntityReference
XML_ENTITY_NODE (integer)	6	Node is a DOMEntity
XML_PI_NODE (integer)	7	Node is a DOMProcessingInstruction
XML_COMMENT_NODE (integer)	8	Node is a DOMComment
XML_DOCUMENT_NODE (integer)	9	Node is a DOMDocument
XML_DOCUMENT_TYPE_NODE (integer)	10	Node is a DOMDocumentType
XML_DOCUMENT_FRAG_NODE (integer)	11	Node is a DOMDocumentFragment
XML_NOTATION_NODE (integer)	12	Node is a DOMNotation

DOMNode

- **¡Importante!**: DOM no trata los atributos como nodos
- Se tiene que emplear la propiedad `attributes`, que es de tipo `DOMNamedNodeMap`

DOMNode

- Propiedades de `DOMDocument` y `DOMNode` para recorrer el árbol:
 - `childNodes`: contiene una lista con los nodos hijo
 - `firstChild`: contiene el primer nodo hijo
 - `lastChild`: contiene el último nodo hijo
 - `nextSibling`: contiene el siguiente nodo hermano
 - `parentNode`: contiene el nodo padre
 - `previousSibling`: contiene el nodo hermano anterior
- Métodos:
 - `hasAttributes`: indica si un nodo tiene atributos
 - `hasChildNodes`: indica si un nodo contiene nodos hijo

DOMNode

- **¡Importante!**: No todos los diferentes tipos de nodos pueden contener la misma información
- Consultar la propiedad `nodeType`
- Ejemplo:
 - Atributos (`attributes`): pueden tener Element, Entity y Notation
 - Nodos hijo (`childNodes`): NO pueden tener CDATASection, Comment, Notation, ProcessingInstruction, Text

DOMNodeList

- Algunas propiedades o métodos devuelven una lista de nodos: `childNodes`, `getElementsByTagName`
- Propiedades:
 - `length`: indica el número de nodos en la lista (colección)
- Métodos:
 - `item`: permite un acceso aleatorio a nodos de la lista (desde 0 hasta `length - 1`)

DOMNodeList

```
<?php
$objDOM = new DOMDocument();
$objDOM->load("biblioteca.xml");

$listaNodos = $objDOM->childNodes;
for($i = 0; $i < $listaNodos->length; $i++) {
    $nodo = $listaNodos->item($i);
    echo "<p>nodeName: " . $nodo->nodeName . "<br
/>nodeType: " . $nodo->nodeType . "<br />text: " .
    $nodo->textContent . "</p>";
}
?>
```

Construcción y manipulación de documentos

- Métodos comunes de DOMDocument y DOMNode:
 - appendChild: añade un hijo
 - cloneNode: duplica un nodo
 - insertBefore: añade un hijo antes del nodo indicado
 - removeChild, replaceChild: elimina o sustituye un nodo hijo

Programación en Internet – Curso 2009-2010

DOMElement

- Los mismos que DOMNode más algunos específicos
- Propiedades:
 - tagName: la etiqueta del elemento
- Métodos:
 - getAttribute: devuelve el valor de un atributo
 - hasAttribute: comprueba si el atributo existe
 - removeAttribute: elimina el atributo
 - setAttribute: modifica el valor de un atributo (si no existe, lo crea)

Programación en Internet – Curso 2009-2010

DOMNamedNodeMap

- Se emplea para representar los atributos de un nodo
 - DOMNode->attributes
- Propiedades:
 - length: indica el número de nodos en la colección
- Métodos:
 - getNamedItem: devuelve el valor del atributo con el nombre indicado
 - item: permite un acceso aleatorio a nodos de la lista

Programación en Internet – Curso 2009-2010

DOMNamedNodeMap

```
<?php
function atributos($nodo) {
    if($nodo->nodeType == XML_ELEMENT_NODE) {
        $cadena = $nodo->nodeName . "<br />";
        for($i = 0; $i < $nodo->attributes->length; $i++)
            $cadena .= $nodo->attributes->item($i)->nodeName . " = " .
            $nodo->attributes->item($i)->nodeValue . "<br />";
        if($nodo->hasChildNodes())
            for($i = 0; $i < $nodo->childNodes->length; $i++)
                $cadena .= atributos($nodo->childNodes->item($i));
    }

    return $cadena;
}
```

Programación en Internet – Curso 2009-2010

DOMNamedNodeMap

```
$objDOM = new DOMDocument();
$objDOM->load("biblioteca.xml");

echo atributos($objDOM->documentElement);
?>
```

Cómo recorrer un documento

- Diversas formas:
 1. Obtener un elemento concreto identificado `getElementById`
 2. Obtener todos los elementos con una misma etiqueta:
`getElementsByTagName`
 3. Utilizando las relaciones entre los nodos:
`childNodes`, `firstChild`,
`nextSibling`, etc.