



APRENDERAPROGRAMAR.COM

GETELEMENTBYID CON  
XML ¿NO FUNCIONA?  
DOM HASATTRIBUTE. DTD  
O DOCUMENT TYPE.  
SIMILAR FUNCIONES  
(CU01211F)

Sección: Cursos

Categoría: Tutorial básico del programador web: Ajax desde cero

Fecha revisión: 2031

**Resumen:** Entrega nº11 del Tutorial básico "Ajax desde cero".

Autor: Alex Rodríguez

## COMPRENDIENDO LOS CONCEPTOS

En la anterior entrega del curso hemos visto un ejemplo de código con el que recuperábamos información desde un archivo XML usando Ajax. Vamos a explicar dicho código y ampliar algunos conocimientos sobre el uso de XML, Ajax y JavaScript.



### Explicación para el código de la entrega CU01210F

Partimos de un formulario (no hemos definido acción para el formulario porque nos resulta irrelevante) con un select combobox que nos da opción a elegir entre varios países. Definimos que ante el evento onchange del select se debe ejecutar la función JavaScript mostrarSugerencia, pasándole como parámetro el atributo value del item elegido en el select y que será uno de estos: none, spain, mexico, argentina ó colombia.

La función recibe el parámetro y determina la cadena a asociar a value, que puede ser España, México, Argentina, Colombia o none. Nos podemos preguntar por qué no usar directamente los nombres de los países como atributos value. El motivo para no hacerlo es evitar usar atributos que contengan tildes o eñes para evitar problemas. En los atributos sólo hemos usado letras minúsculas excluyendo tildes y eñes.

Si no se recibe cadena o se recibe "none", se muestra el mensaje "no hay datos" y se limpia la lista de ciudades.

Si se recibe cadena se crea un objeto XMLHttpRequest y se define la función de respuesta a la recuperación del contenido del archivo listadoPaíses.xml mediante GET.

Dentro de la función de respuesta se guarda el documento nodo DOM recuperado en la variable xmlDoc, y se recupera la lista de países en el documento invocando obtenerArrayNodosPorTag(xmlDoc, 'pais');

obtenerArrayNodosPorTag es una función auxiliar que hemos definido para recuperar un array de nodos a partir de un nodo raíz y de un nombre de etiqueta (tag).

Una vez recuperada la lista de países del xml, se recorre. Si un país del xml coincide con el país elegido por el usuario en el select, se procede a mostrar un mensaje informativo sobre cuál ha sido el país recibido y su índice, y se recupera la lista de ciudades para ese país con la invocación obtenerArrayNodosPorTag(paisesRecibidos[i], 'ciudadImportante');

Una vez disponemos del array de ciudades las recorremos con un bucle for y creamos el código HTML para mostrar dichas ciudades, lo que hacemos finalmente con nodoMostrarResultados.innerHTML = contenidosAMostrar;

## DIFICULTADES A TENER EN CUENTA CON XML

Para recuperar los contenidos del archivo xml hemos tenido que obtener nodos, recorrerlos y recuperar su contenido usando sentencias como esta:

```
var nombrePais = obtenerArrayNodosPorTag(paisesRecibidos[i], 'nombre')[0].innerHTML;
```

También podríamos haber usado:

```
var nombrePais = obtenerArrayNodosPorTag(paisesRecibidos[i], 'nombre')[0].textContent;
```

Sin embargo no debemos usar `nodeValue` u otras opciones que podríamos pensar que funcionarían. En realidad sí podemos usarlo si lo hacemos bien, pero el nodo de texto posiblemente es un nodo de texto interno al nodo con el que estamos trabajando y necesitaríamos una expresión más complicada aún como:

```
obtenerArrayNodosPorTag(paisesRecibidos[i], 'nombre')[0].firstChild.nodeValue;
```

Una dificultad con que nos encontramos es la necesidad de manipular y recorrer el DOM para extraer los contenidos del fichero XML, lo cual a veces puede resultar un tanto engorroso, tanto más cuanto más compleja sea la estructura del fichero xml.

Otra dificultad es que un documento XML no tiene igual definición que un documento HTML. Cada documento lleva un DTD o Document Type Definition. No vamos a detenernos ahora a estudiar el DTD porque no es objeto de este curso. Sin embargo, hemos de tener en cuenta una cosa: un documento XML no es lo mismo que un documento HTML, aunque compartan algunos aspectos. Debido a esto, algunas funciones JavaScript que funcionan sobre documentos HTML como `getElementById` pueden no responder bien sobre ciertos documentos XML o en ciertos navegadores.

A veces se definen documentos XML pensando que funcionarán como si fueran HTML y se intentan manipular como si fueran HTML, pero esto induce a confusiones porque no es lo mismo una cosa que la otra ni podemos manipularlos de la misma manera.

Ejemplo:

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101" href="http://aprenderaprogramar.com" name="libro10">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
  </book>
```

Esto sería un documento xml donde aparecen algunos atributos que “parecen” html, pero no se pueden considerar ni tratar como atributos html convencionales porque no es un documento html.

Una sintaxis como `xml_document.getElementById('myid')` no está garantizado que funcione en estos casos.

## SIMULAR EL GETELEMENTBYID EN DOCUMENTOS XML

Dado que getElementById sólo tenemos seguridad de que responda con documentos HTML, podemos valernos de un pequeño truco para conseguir la misma funcionalidad con documentos XML. En concreto se trata de aprovechar métodos disponibles gracias a la especificación DOM, y en concreto:

Método	Finalidad	Ejemplo
<b>setAttribute</b>	Establecer un atributo asociado a un nodo DOM	nodo.setAttribute('nombreAtributoHTML', 'valorAtributoHTML');
<b>getAttribute</b>	Recuperar un atributo asociado a un nodo DOM	nodo.getAttribute('nombreAtributoHTML');
<b>hasAttribute</b>	Obtener un booleano que nos indique si el nodo DOM tiene cierto atributo	nodo.hasAttribute('id');
<b>removeAttribute</b>	Eliminar un atributo de un nodo DOM	nodo.removeAttribute('id');

Estos métodos sí están disponibles sobre nodos de documentos XML. Podemos usar una función como esta para simular la recuperación por id:

```
function getElementByIdXML(the_node,the_id) {
    //Nos traemos todos los nodos con cualquier tag del documento xml
    node_tags = the_node.getElementsByTagName('*');
    for (i=0;i<node_tags.length;i++) {
        // Comprobamos si existe un atributo id
        if (node_tags[i].hasAttribute('id')) {
            // Si existe el atributo id comprobamos si coincide con el buscado
            if (node_tags[i].getAttribute('id') == the_id) {
                // Devolvemos el nodo que tiene el id buscado, solo el primero ya que debe ser único
                return node_tags[i];
            }
        }
    }
}
```

Aquí getElementsByTagName('\*') supone recuperar todos los nodos del documento (escribir como nombre de etiqueta \* supone indicar "cualquier nombre").

Con esta función y suponiendo que tenemos un documento XML donde existen atributos id, por ejemplo:

```
<pais id="spain">
    <nombre >España</nombre>
    <capital>Madrid</capital>
</pais>
```

Podemos recuperar el nodo con el id "spain" invocando algo como: getElementByIdXML(xmlDoc, 'spain'), que simula la función getElementById.

Y el contenido interior de un nodo podría recuperarse invocando getElementByIdXML(xmlDoc, 'spain').innerHTML o similar.

Crea un documento XML con atributos y haz pruebas de esta función.

### SIMULAR EL GETELEMENTSBYNAME Y GETELEMENTSBYCLASSNAME EN DOCUMENTOS XML

Dado que métodos como getElementByName y getElementByClassName pueden no funcionar sobre documentos XML, podemos simularlos siguiendo la misma idea anterior.

Esta función permitiría recuperar el array de nodos de un documento XML que tengan el atributo especificado.

```
function getElementsByAttribute(the_attribute, the_value, the_node) {
    if ( the_node == null ) {the_node = document;}
    var node_tags = the_node.getElementsByTagName('*');
    var results = new Array();
    for (i=0, j=0; i<node_tags.length; i++){
        if (node_tags[i].hasAttribute(the_attribute)) {
            if (node_tags[i].getAttribute(the_attribute) == the_value) {
                results[j] = node_tags[i]; j++; }
        }
    }
    return results;
}
```

Crea un documento XML con atributos y haz pruebas de esta función.

### EJERCICIO

Lee este texto que hemos extraído del artículo: ¿Qué es y para qué sirve XML? disponible en [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=102:i-que-es-y-para-que-sirve-el-lenguaje-de-etiquetas-xml-extensible-markup-language&catid=46:lenguajes-y-entornos&Itemid=163](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=102:i-que-es-y-para-que-sirve-el-lenguaje-de-etiquetas-xml-extensible-markup-language&catid=46:lenguajes-y-entornos&Itemid=163)

Las etiquetas XML pueden tener atributos, que son una manera de incorporar características o propiedades a las etiquetas de un documento. El atributo consta de dos partes: La propiedad del elemento y el valor de la propiedad, que siempre va entre comillas doble (") o simple ('). Por ejemplo: modelo y color serian atributos de la etiqueta Vehiculo:

```
<Vehiculo marca="Toyota" modelo="45 TC" color="plomo">En venta</Vehiculo>
```

Considera el siguiente documento xml al que denominamos ejemplo.xml:

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <price type="high">44.95</price>
    <publish_date>2000-10-01</publish_date>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <price type="low">5.95</price>
    <publish_date>2000-12-16</publish_date>
  </book>
  <book id="bk103">
    <author>Corets, Eva</author>
    <title>Maevae Ascendant</title>
    <price type="low">5.95</price>
    <publish_date>2000-11-17</publish_date>
  </book>
</catalog>
```

A priori podríamos pensar en usar getElementById para recuperar información del documento xml con Ajax. Pero esta función podría no responder con ciertos documentos XML o con ciertos navegadores, como explica este texto en inglés:

Non-HTML documents. The DOM implementation must have information that says which attributes are of type ID. Attributes with the name "id" are not of type ID unless so defined in the document's DTD. The id attribute is defined to be of ID type in the common cases of XHTML, XUL, and other. Implementations that do not know whether attributes are of type ID or not are expected to return null.

- Usa el archivo ejemplo.xml y coloca un botón que intente recuperar usando ajax y getElementById el contenido del elemento con id="bk101" ¿Qué código has empleado? ¿Qué resultado obtienes?
- Usa el archivo ejemplo.xml y coloca un botón que intente recuperar usando ajax y la función que hemos visto anteriormente getElementByIdMXL(the\_node,the\_id) el contenido del elemento con id="bk101" ¿Qué código has empleado? ¿Qué resultado obtienes?
- Usa el archivo ejemplo.xml y coloca un botón que intente recuperar usando ajax y la función que hemos visto anteriormente getElementsByAttribute(the\_attribute, the\_value, the\_node) todos los precios que lleven como atributo type "low" ¿Qué código has empleado? ¿Qué resultado obtienes?

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

**Próxima entrega:** CU01212F

**Acceso al curso completo** en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:

[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=83&Itemid=212](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=83&Itemid=212)