



XML Exercises

Antes de comenzar:

1. Lee los siguientes apartados de w3school:
<https://www.w3schools.com/xml/default.asp>
https://www.w3schools.com/xml/xml_what_is.asp
https://www.w3schools.com/xml/xml_used_for.asp
https://www.w3schools.com/xml/xml_tree.asp
https://www.w3schools.com/xml/xml_elements.asp
https://www.w3schools.com/xml/xml_attributes.asp
https://www.w3schools.com/xml/xml_display.asp
2. Lee el siguiente artículo:
<https://www.mclibre.org/consultar/xml/lecciones/xml-conceptos-basicos.html>
3. Lee el siguiente artículo e instala en tu ordenador XML Copy Editor
<https://www.mclibre.org/consultar/xml/otros/xmlcopyeditor.html>
4. Recopila las respuestas a los siguientes ejercicios.

Exercises

1. Simple questions:

- a. How should an element with the name `test` element and the content "This is our first element!" look?
- b. Please show (in xml-syntax) the hierarchy of the element `book` with the subelements `chapters` and `author`. `author` has the further subelements `name` and `address`.
- c. How does an element with the name `entry` which has got 2 attributes: `no` with the value 24 and `date` with the value 27.10.2004 look?

2. XML tree structures

Analyse the XML file below.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<publication>
  <newsitem>
    <source>
      <agency>STT</agency>
      <editor>Miller</editor>
    </source>
    <class>local</class>
    <article>
      <heading>Snowstorms in Lapland</heading>
      <text>snow.html</text>
      <summary>snowx.html</summary>
    </article>
  </newsitem>
</publication>
```



Create the file and save as Ex2.xml. Open the file with a web browser.

3. *XML structure*

Analyse the following XML file. Correct errors in design.

```
<?xml version='1.0' encoding='UTF-8'?>
<phonebook>
  <company>
    <cname>Microsoft</cname>
    <exchange>09-999000</exchange>
  </company>
  <president>Bill Gates
  <extension>09-9990011</president></extension>
  <secretary>Katharine Finch <extension>09-9990012</secretary></extension>
  <company>
    <cname>Oracle</cname>
    <exchange>09-888000</exchange>
  </company>
  <president>Larry Ellison
  <extension>09-8880011</president></extension>
  <secretary>Helen Calhoun
  <extension>09-8880012</secretary></extension>
</phonebook>
```

Create a file called ex3-err.txt and write down every mistake and correction.

4. *Mark up the following email message to identify its information content:*

From: Simon North <north@synopsys.com>
To: Nick <sintac@xs4all.nl>
Subject: Hi
Hi Nick, this is just a quick message
to say I got the material. Thanks.

Create an xml file (ex4.xml) to store the necessary information.

5. *XML file, CD information*

Plan an XML description for a music CD. You could use a root element called <CDList>. Information about the contents should include at least name of the record, artists, songs, publishing year, and music category. You can add more information e.g. about each track (number, duration, name, etc.) Plan first the tree structure of your XML data, and then find some data from the Web. Save as Ex5a.xml and test with your browser .

Next, add some information as attributes, like below <CD serial=B6B41B disc-length='36:55'>
Save as Ex5b.xml and test. Fix errors. Add the following comment: <!-- This CD has more tracks



than in this file -->

6. *XML: a letter template*

Create a letter structure in XML, including sender and receiver information, date, text body and a signature. Make a document instance with contents. Save your file as Ex6.xml.

7. *The XML standard*

How XML should look is described in the [XML 1.0 W3C Recommendation](#). The language of the standard is somewhat technical and formal, but being able to read the standard yourself is quite useful. Look into the standard and answer the following questions:

- a. What do comments look like in XML? (section 2.5)
- b. Can an attribute be repeated in the same element? (section 3.1)
- c. Where is it stated that the element names in opening and closing tags must match?



XML DOM Exercises

Para estos ejercicios usaremos la consola del navegador y el fichero xml:

<https://www.w3schools.com/xml/books.xml>

Nota: si queremos mostrar un mensaje dentro de un bucle (for, while ...) haremos uso de "console.log()"

Ejercicio 1. Escribe el código necesario para obtener los siguientes datos:

1. Título del primer libro

```
document.getElementsByTagName("title")[0].childNodes[0].nodeValue;
```

2. Año del primer libro

```
document.getElementsByTagName("year")[0].childNodes[0].nodeValue;
```

3. Precio del segundo libro

```
document.getElementsByTagName("price")[1].childNodes[0].nodeValue;
```

4. Autor del primer libro

```
document.getElementsByTagName("author")[0].childNodes[0].nodeValue;  
// otra posibilidad es...  
document.getElementsByTagName("book")[0].getElementsByTagName('author')[0].childNodes[0].nodeValue;
```

5. Primer autor del tercer libro

```
document.getElementsByTagName("book")[2].getElementsByTagName('author')[0].childNodes[0].nodeValue;
```

6. Segundo autor del tercer libro

```
document.getElementsByTagName("book")[2].getElementsByTagName('author')[1].childNodes[0].nodeValue;
```

7. Nombre del primer atributo del primer libro

```
document.getElementsByTagName("book")[0].attributes[0].nodeName;
```

8. Valor del primer atributo del primer libro

```
document.getElementsByTagName("book")[0].attributes[0].nodeValue;
```

9. Número de autores del tercer libro

```
document.getElementsByTagName("book")[2].getElementsByTagName('author').length;
```

10. Mostrar todos los autores del tercer libro

```
l = document.getElementsByTagName("book")[2].getElementsByTagName('author').length;
```



```
a = document.getElementsByTagName("book")[2].getElementsByTagName('author');
for(i = 0; i < 1; i++)
    console.log(a[i].childNodes[0].nodeValue);
```

11. Mostrar el número de atributos del cuarto libro

```
document.getElementsByTagName("book")[3].attributes.length;
```

12. Mostrar el valor de los atributos del cuarto libro

```
l = document.getElementsByTagName("book")[3].attributes.length;
a = document.getElementsByTagName("book")[3].attributes
for(i = 0; i < l; i++)
    console.log(a[i].nodeValue);
```

Ejercicio 2. Escribe el código necesario para obtener los siguientes datos:

Puedes hacer uso del editor de w3schools siguiente, que muestra el nombre del artista del archivo https://www.w3schools.com/js/cd_catalog.xml

https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_responsexml

Haz las modificaciones necesarias para mostrar lo que se pide:

1. Muestra el título de los CD

```
titulos = "";
n = document.getElementsByTagName("TITLE").length;
for(i = 0; i < n; i++)
    titulos += document.getElementsByTagName("TITLE")[i].childNodes[0].nodeValue + " <br> ";
```

2. Muestra el precio

3. Muestra la compañía

Ejemplo en el que se muestra título, artista y precio mediante Ajax

```
<!DOCTYPE html>
<html>
<body>
<h2>The XMLHttpRequest Object</h2>
<p id="demo"></p>

<script>
```



```
const xhttp = new XMLHttpRequest();
xhttp.onload = function() {
  const xmlDoc = this.responseXML;
  const x = xmlDoc.getElementsByTagName("CD");
  let txt = "", title= "", artist = "";
  for (let i = 0; i < x.length; i++) {
    title = x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue;
    artist = x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue;
    price = x[i].getElementsByTagName("PRICE")[0].childNodes[0].nodeValue;
    txt = txt + title + " - (" + artist + ") " + price + "€<br>";
  }
  document.getElementById("demo").innerHTML = txt;
}
xhttp.open("GET", "cd_catalog.xml");
xhttp.send();
</script>
</body>
</html>
```

Ejercicio 3. Escribe el código necesario para obtener los siguientes datos:

Ahora usando el mismo editor y el fichero "<https://www.w3schools.com/xml/books.xml>"

1. Título del primer libro
2. Todos los títulos
3. Número de atributos del cuarto libro
4. Valor de los atributos del cuarto libro
5. Número de autores del tercer libro
6. Autores del tercer libro
7. Muestra una tabla que muestre el título, primer autor, precio y año



AJAX Exercises

Exercise 1.

Using the cd_catalog.xml file and the following code:

cd_catalog.xml

https://www.w3schools.com/js/cd_catalog.xml

AJAX example

https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_xml2

1. The code must include information about country, company, price and year.
2. Using a div element, show the same information.
Hint: In order to create a table using “div” tags, you can use a table generator like this: <https://divtable.com/generator/>

Exercise 2.

Escribe un programa que muestre información meteorológica de una localidad de Huelva (o cualquier otra provincia de tu elección): temperatura máxima y mínima, estado del cielo por tramo horario, dirección y velocidad del viento y humedad relativa).

Para acceder a datos meteorológicas vamos a usar la siguiente url:

- http://www.aemet.es/xml/municipios/localidad_21041.xml (Huelva capital)
- http://www.aemet.es/xml/municipios/localidad_21002.xml (Aljaraque)
- http://www.aemet.es/xml/municipios/localidad_21021.xml (Cartaya) ...

Para mostrar datos de un municipio debe incluir en la url el código INE de localidad.

Nota: El código INE está accesible en la siguiente dirección:

- <https://www.ine.es/daco/daco42/codmun/codmunmapa.htm>

Nota adicional: La Aemet cuenta con un servicio de datos abiertos que puedes consultar en la siguiente dirección:

opendata aemet

http://www.aemet.es/es/datos_abiertos/AEMET_OpenData



Exercise 3. JSON

En este ejercicio usaremos el archivo "json_demo_array" y el editor de w3schools. Responde a los siguientes apartados.

https://www.w3schools.com/js/json_demo_array.txt

https://www.w3schools.com/js/tryit.asp?filename=tryjson_ajax

1. Muestra la longitud del array
2. Muestra los elementos usando una tabla.

Exercise 4. JSON

En este ejercicio usaremos el archivo "json_demo" y el editor de w3schools. Responde a los siguientes apartados.

https://www.w3schools.com/js/json_demo.txt

https://www.w3schools.com/js/tryit.asp?filename=tryjson_ajax

Nota: Para mostrar la longitud de un array usamos la propiedad length.

https://www.w3schools.com/js/js_arrays.asp

Nota: Para iterar los elementos de un array podemos usar

1- Bucle for(i=0; i < arr.length; i++)

2- Bucle for(x of arr)

https://www.w3schools.com/js/js_loop_for.asp

Apartados

1. Muestra edad
2. Muestra del primer elemento pets, el tipo de animal
3. Muestra del primer elemento pets, el nombre



4. Muestra el número de mascotas (pets);
5. Muestra el nombre de todas las mascotas
6. Muestra todos los datos en un párrafo
7. Muestra los datos usando una tabla.

```
var myObj = JSON.parse(this.responseText);  
  
for( i = 0; i < myObj.pets.length; i++)  
  
    txt = txt + myObj.pets[i].name + "<br>";  
  
document.getElementById("demo").innerHTML = txt;
```

Exercise 5. JSON

En este ejercicio usaremos un servicio de datos de la NASA a través de una API.

https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY&date=2020-05-01

https://www.w3schools.com/js/tryit.asp?filename=tryjson_ajax

1. Muestra el valor de "explanation"
2. Muestra el valor de "date"
3. Muestra el valor de "url"
4. Añade una imagen vacía y muestra la imagen.

Enlaces:

NASA

<https://api.nasa.gov/>



APOD - Astronomy Photo Of the Day

https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY&date=2020-05-01

MRP - Mars Rover Photos

https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&camera=fhaz&api_key=DEMO_KEY

NASA Image and Video Library

<https://images-api.nasa.gov/search?q=moon>



Nota final: Truco para atender múltiples peticiones Ajax simultáneas:

<https://www.codeproject.com/Questions/764646/Send-multiple-AJAX-request-concurrently>

https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_display_table

```
function myAjaxRequest(url, callback) {
    var ajax = new XMLHttpRequest();
    ajax.onreadystatechange = function() {
        if (this.readyState==4 && this.status==200)
            callback(this);
    }
    ajax.onerror = function() {
        console.log("AJAX request failed to: " + url);
    }
    ajax.open("GET", url, true);
    ajax.send();
}

function loadXMLDoc2() {
    myAjaxRequest('cd_catalog.xml', onCdXmlLoaded);
    myAjaxRequest('cd_catalog.xml', onCdXmlLoaded2);
}

function onCdXmlLoaded(ajax) {
    var xmlDoc = ajax.responseXML;
    var txt = '';
    x=xmlDoc.getElementsByTagName("ARTIST");
    for (i=0;i<x.length;i++) {
        txt=txt + x[i].childNodes[0].nodeValue + "<br>";
    }
    document.getElementById("myDiv").innerHTML=txt;
}

function onCdXmlLoaded2(ajax) {
    var xmlDoc = ajax.responseXML;
    var txt = '';

    x=xmlDoc.getElementsByTagName("TITLE");
    for (i=0;i<x.length;i++) {
        txt=txt + x[i].childNodes[0].nodeValue + "<br>";
    }
}
```



```
document.getElementById("myDiv2").innerHTML=txt;  
}
```



RSS

Ejercicio 1

Crea una página web que lea un archivo RSS mediante AJAX. Elige un archivo de RSS de tu elección y muestra apropiadamente la información contenida en dicho archivo.