

## CODE EXERCISE

# XML: EXTENSIBLE MARKUP LANGUAGE

XML (eXtensible Markup Language) is a markup language that is similar to HTML, but is designed to store and transport data rather than display it. It uses tags and attributes, like HTML, to define elements and their properties.

Here's an example of some XML code that represents a bookstore:

```
<bookstore>
  <book>
    <title>The Shawshank Redemption</title>
    <author>Stephen King</author>
    <year>1982</year>
    <price>10.99</price>
  </book>
  <book>
    <title>The Godfather</title>
    <author>Mario Puzo</author>
    <year>1969</year>
    <price>9.99</price>
  </book>
</bookstore>
```

This XML code defines a bookstore element that contains two book elements. Each book element has child elements of title, author, year, and price, which store information about the book.

The code is well structured, using a parent-child relationship. XML allows to use own names for the tags and attributes, which can help to structure the data and make it more readable for both humans and machines. The XML code can also be used to transport data between systems and applications, and can be parsed and read by different programming languages, including JavaScript.

It's important to note that XML requires that the tags are properly nested and closed, otherwise the data will not be parsed correctly.

The `fetch()` method can also be used to retrieve XML data from a server and handle the response in JavaScript.

```
fetch('path-to-file.xml')
  .then(response => response.text())
```

## SUBMITTING EXERCISES

To earn certification all exercises must be submitted and accepted.

1) When you've completed all of the HTML exercises, please zip them into a single file and submit to our Dropbox at <http://bit.ly/CWDP2324>.

2) Next, fill out the certification completion form at <https://forms.gle/5EiUGCM6dGdIF2Py6>.

Remember, that all of your exercises for each module should be included in a separate zip file.

## GETTING HELP

We always want to ensure that your questions are answered. There are a number of ways to get in touch.

1) We operate a lively Discord server. Join us at <https://discord.gg/tgxX2fCrv5> and you can ask your question on Discord. Mark and our team of instructional assistants monitor this Discord and answer questions ASAP.

2) This certification program is offered on several platforms. Most platforms have a Q & A section where you can post questions. We monitor these and respond as quickly as we can.

3) You may use our question hotline email at [questions@dollardesignschool.com](mailto:questions@dollardesignschool.com).

**Remember to always to include which section your question is from and send any code you're working on!**



## CODE EXERCISE

# XML: EXTENSIBLE MARKUP LANGUAGE

```
.then(data => {  
    var parser = new DOMParser();  
    var xmlDoc = parser.parseFromString(data, "text/xml");  
    // your code here to handle the xmlDoc  
})  
.catch(err => {  
    console.log(err);  
});
```

The `fetch()` method is used to initiate a GET request to the xml file, the data is returned as text, then it uses the `DOMParser()` to parse the xml string so it.

## LAB 1

1) To simplify parsing, we'll bring in the XML as a string.

```
var xmlString = `  
<bookstore>  
  <book>  
    <title>The Shawshank Redemption</title>  
    <author>Stephen King</author>  
    <year>1982</year>  
    <price>10.99</price>  
  </book>  
  <book>  
    <title>The Godfather</title>  
    <author>Mario Puzo</author>  
    <year>1969</year>  
    <price>9.99</price>  
  </book>  
</bookstore>`;
```

2) Create a `DOMParser` object to parse the XML string and convert it into a Document object.

```
var parser = new DOMParser();  
var xmlDoc = parser.parseFromString(xmlString, "text/xml");
```

3) Once the XML is parsed, you can use JavaScript to access and manipulate the data in the XML document. For example, you can loop through the book elements and access the title, author, year, and

## SUBMITTING EXERCISES

To earn certification all exercises must be submitted and accepted.

1) When you've completed all of the HTML exercises, please zip them into a single file and submit to our Dropbox at <http://bit.ly/CWDP2324>.

2) Next, fill out the certification completion form at <https://forms.gle/5EiUGCM6dGdIF2Py6>.

Remember, that all of your exercises for each module should be included in a separate zip file.

## GETTING HELP

We always want to ensure that your questions are answered. There are a number of ways to get in touch.

1) We operate a lively Discord server. Join us at <https://discord.gg/tgxX2fCrv5> and you can ask your question on Discord. Mark and our team of instructional assistants monitor this Discord and answer questions ASAP.

2) This certification program is offered on several platforms. Most platforms have a Q & A section where you can post questions. We monitor these and respond as quickly as we can.

3) You may use our question hotline email at [questions@dollardesignschool.com](mailto:questions@dollardesignschool.com).

**Remember to always to include which section your question is from and send any code you're working on!**



## CODE EXERCISE

# XML: EXTENSIBLE MARKUP LANGUAGE

price child elements.

```
var books = xmlDoc.getElementsByTagName("book");
for (var i = 0; i < books.length; i++) {
    var title = books[i].getElementsByTagName("title")[0].textContent;
    var author = books[i].getElementsByTagName("author")[0].textContent;
    var year = books[i].getElementsByTagName("year")[0].textContent;
    var price = books[i].getElementsByTagName("price")[0].textContent;
    console.log("Title: " + title);
    console.log("Author: " + author);
    console.log("Year: " + year);
    console.log("Price: $" + price);
}
```

4) You can also access specific elements by id or class, this works similar to the way it works in HTML.

```
var bookstore = xmlDoc.getElementsByTagName("bookstore")[0];
var bookstoreName = bookstore.getAttribute("name");
```

## LAB 2

Use the following XML for this lab, saved in a file called products.xml.

```
<products>
  <product id="1">
    <name>Computer</name>
    <price>1000</price>
    <quantity>10</quantity>
    <description>A powerful computer with 16GB of RAM and a 1TB hard drive.</description>
  </product>
  <product id="2">
    <name>Monitor</name>
    <price>300</price>
    <quantity>20</quantity>
    <description>A 27-inch, 1080p monitor with a fast refresh rate.</description>
  </product>
  <product id="3">
    <name>Keyboard</name>
```

## SUBMITTING EXERCISES

To earn certification all exercises must be submitted and accepted.

1) When you've completed all of the HTML exercises, please zip them into a single file and submit to our Dropbox at <http://bit.ly/CWDP2324>.

2) Next, fill out the certification completion form at <https://forms.gle/5EiUGCM6dGdIF2Py6>.

Remember, that all of your exercises for each module should be included in a separate zip file.

## GETTING HELP

We always want to ensure that your questions are answered. There are a number of ways to get in touch.

1) We operate a lively Discord server. Join us at <https://discord.gg/tgxX2fCrv5> and you can ask your question on Discord. Mark and our team of instructional assistants monitor this Discord and answer questions ASAP.

2) This certification program is offered on several platforms. Most platforms have a Q & A section where you can post questions. We monitor these and respond as quickly as we can.

3) You may use our question hotline email at [questions@dollar.designschool.com](mailto:questions@dollar.designschool.com).

**Remember to always to include which section your question is from and send any code you're working on!**



## CODE EXERCISE

# XML: EXTENSIBLE MARKUP LANGUAGE

```
<price>50</price>
<quantity>30</quantity>
<description>A wireless keyboard with a built-in touchpad.</description>
</product>
</products>
```

This XML file contains a `<products>` element that contains several `<product>` elements, each representing a product. The `<product>` elements have child elements of `<name>`, `<price>`, `<quantity>`, and `<description>`, which store information about the product. The product also have a unique id attribute which can be used to search for a specific product by id.

Create a JavaScript program that parses an XML file containing information about products and displays the data in a table on a web page.

- Use the `DOMParser()` object to parse the XML data and convert it into a Document object.
- Use JavaScript to loop through the products and access their properties (product name, price, quantity, etc.).
- Create a table and use JavaScript to add rows to the table, displaying the product information.
- Create a function that takes the name of a product and returns the information of that product
- Add a "search bar" that allows users to search for products by name

Deliverable:

- A single HTML file called `product.html`
- A single JavaScript file called `product.js`
- A single XML file called `products.xml`
- A readme file that explains the usage of your program.

## SUBMITTING EXERCISES

To earn certification all exercises must be submitted and accepted.

1) When you've completed all of the HTML exercises, please zip them into a single file and submit to our Dropbox at <http://bit.ly/CWDP2324>.

2) Next, fill out the certification completion form at <https://forms.gle/5EiUGCM6dGdIF2Py6>.

Remember, that all of your exercises for each module should be included in a separate zip file.

## GETTING HELP

We always want to ensure that your questions are answered. There are a number of ways to get in touch.

1) We operate a lively Discord server. Join us at <https://discord.gg/tgxX2fCrv5> and you can ask your question on Discord. Mark and our team of instructional assistants monitor this Discord and answer questions ASAP.

2) This certification program is offered on several platforms. Most platforms have a Q & A section where you can post questions. We monitor these and respond as quickly as we can.

3) You may use our question hotline email at [questions@dollardesignschool.com](mailto:questions@dollardesignschool.com).

**Remember to always to include which section your question is from and send any code you're working on!**

