Xml

* **Xml** (eXtensible Markup Language) is a mark up language.
* XML is designed to store and transport data.
* Xml was released in late 90’s. it was created to provide an easy to use and store self describing data.
* XML became a W3C Recommendation on February 10, 1998.
* XML is not a replacement for HTML.
* XML is designed to be self-descriptive.
* XML is designed to carry data, not to display data.
* XML tags are not predefined. You must define your own tags.
* XML is platform independent and language independent.

XML Syntax Rules

You must follow these rules when you create XML syntax:

* All XML elements must have a closing tag.
* XML tags are case sensitive.
* All XML elements must be properly nested.
* All XML documents must have a root element.
* Attribute values must always be quoted.

# XML Attributes

XML elements can have attributes, just like HTML.

Attributes are designed to contain data related to a specific element.

## XML Attributes Must be Quoted

Attribute values must always be quoted. Either single or double quotes can be used.

For a person's gender, the <person> element can be written like this:

<person gender="female">

# XML Namespaces

XML **Namespace** is used to avoid element name conflict in XML document.

## XML Namespace Declaration

An XML namespace is declared using the reserved XML attribute. This attribute name must be started with "xmlns".

Let's see the XML namespace syntax:

1. **<element** xmlns:name = "URL"**>**

Here, namespace starts with keyword **"xmlns"**. The word **name** is a namespace prefix. The **URL** is a namespace identifier.

# XML CSS

## Purpose of CSS in XML

CSS (Cascading Style Sheets) can be used to add style and display information to an XML document. It can format the whole XML document.

## How to link XML file with CSS

To link XML files with CSS, you should use the following syntax

1. **<?xml-stylesheet** type="text/css" href="cssemployee.css"**?>**

## XML CSS Example

Let's see the css file.

cssemployee.css

1. employee
2. {
3. background-color: pink;
4. }
5. firstname,lastname,email
6. {
7. font-size:25px;
8. display:block;
9. color: blue;
10. margin-left: 50px;
11. }

# XML DTD

## What is DTD

DTD stands for **Document Type Definition**. It defines the legal building blocks of an XML document. It is used to define document structure with a list of legal elements and attributes.

## Purpose of DTD

Its main purpose is to define the structure of an XML document. It contains a list of legal elements and define the structure with the help of them.

## CDATA

CDATA: (Unparsed Character data): CDATA contains the text which is not parsed further in an XML document. Tags inside the CDATA text are not treated as markup and entities will not be expanded.

Let's take an example for CDATA:

1. **<?xml** version="1.0"**?>**
2. <!DOCTYPE employee SYSTEM "employee.dtd"**>**
3. **<employee>**
4. <![CDATA[
5. <firstname>geu</firstname>
6. <lastname>dehradun</lastname>
7. <email>info@geu.ac.in</email>
8. ]]>
9. **</employee>**

In the above CDATA example, CDATA is used just after the element employee to make the data/text unparsed, so it will give the value of employee:

<firstname>geu</firstname><lastname>dehradun</lastname><email>info@geu.ac.in</email>

## PCDATA

PCDATA: (Parsed Character Data): XML parsers are used to parse all the text in an XML document. PCDATA stands for Parsed Character data. PCDATA is the text that will be parsed by a parser. Tags inside the PCDATA will be treated as markup and entities will be expanded.

In other words you can say that a parsed character data means the XML parser examine the data and ensure that it doesn't content entity if it contains that will be replaced.

Let's take an example:

1. **<?xml** version="1.0"**?>**
2. <!DOCTYPE employee SYSTEM "employee.dtd"**>**
3. **<employee>**
4. **<firstname>**geu**</firstname>**
5. **<lastname>**dehradun**</lastname>**
6. **<email>**info@geu.ac.in**</email>**
7. **</employee>**

In the above example, the employee element contains 3 more elements 'firstname', 'lastname', and 'email', so it parses further to get the data/text of firstname, lastname and email to give the value of employee as:

geu dehradun info@geu.ac.in

**JSON**

JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange. Conventions used by JSON are known to programmers, which include C, C++, Java, Python, Perl, etc.

* JSON stands for JavaScript Object Notation.
* The format was specified by Douglas Crockford.
* It was designed for human-readable data interchange.
* It has been extended from the JavaScript scripting language.
* The filename extension is **.json**.
* JSON Internet Media type is **application/json**.
* The Uniform Type Identifier is public.json.

## Uses of JSON

* It is used while writing JavaScript based applications that includes browser extensions and websites.
* JSON format is used for serializing and transmitting structured data over network connection.
* It is primarily used to transmit data between a server and web applications.
* Web services and APIs use JSON format to provide public data.
* It can be used with modern programming languages.

## Characteristics of JSON

* JSON is easy to read and write.
* It is a lightweight text-based interchange format.
* JSON is language independent.

## Simple Example in JSON

The following example shows how to use JSON to store information related to books based on their topic and edition.

{

"book": [

{

"id":"01",

"language": "Java",

"edition": "third",

"author": "Herbert Schildt"

},

{

"id":"07",

"language": "C++",

"edition": "second",

"author": "E.Balagurusamy"

}

]

}

After understanding the above program, we will try another example. Let's save the below code as **json.htm**l

<html>

<head>

<title>JSON example</title>

<script language = "javascript" >

var object1 = { "language" : "Java", "author" : "herbert schildt" };

document.write("<h1>JSON with JavaScript example</h1>");

document.write("<br>");

document.write("<h3>Language = " + object1.language+"</h3>");

document.write("<h3>Author = " + object1.author+"</h3>");

var object2 = { "language" : "C++", "author" : "E-Balagurusamy" };

document.write("<br>");

document.write("<h3>Language = " + object2.language+"</h3>");

document.write("<h3>Author = " + object2.author+"</h3>");

document.write("<hr />");

document.write(object2.language + " programming language can be studied " + "from book written by " + object2.author);

document.write("<hr />");

</script>

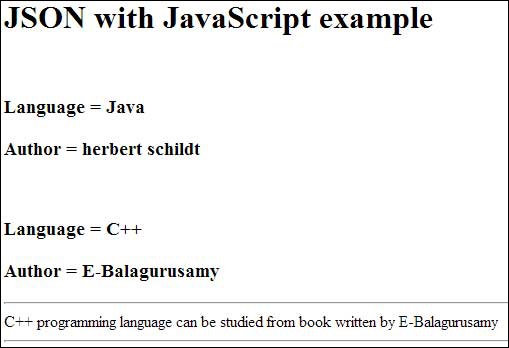
</head>

<body>

</body>

</html>

Now let's try to open json.htm using IE or any other javascript enabled browser that produces the following result −



## jQuery

A free and open-source javascript library which is basically used for designing, traversing and manipulating the HTML DOM. A DOM is a tree-like structure used to represent the elements of a webpage. jQuery helps the designer to use javascript code easily for their websites. The advanced approach to jQuery enables to create powerful dynamic webpages and web applications. The syntax of jQuery is designed to make things easy, such as:

* Navigation of a document
* Selection of DOM elements
* Creating animations
* Handling events
* Developing Ajax applications.

jQuery is one of the widely used [javascript](https://www.javatpoint.com/javascript-tutorial) library among all other libraries holding the following core features:

1. DOM elements selection
2. Traversal and manipulation which is enabled by Sizzle(the selector engine)
3. Creating a new programming style
4. Fusing DOM data structures and algorithms

On the other hand, [jQuery](https://www.javatpoint.com/jquery-tutorial) allows developers to create plug-ins on top of the JavaScript library. Developers can even create abstractions for low-level interaction and animations, too.

**Different ways to include jQuery in a Webpage**

we are going to learn different ways to include jQuery on a page. Basically, we know that jQuery comes with a lot of exciting features. So if we want to use those features, we just have to add the jQuery library to our webpage.

There are two ways of adding jQuery library to our webpage.

* Include jQuery from CDN (Content Delivery Network)
* Download the jQuery library from the official website

**1. Include jQuery from CDN Link:** **CDN** stands for Content Delivery Network which is basically a set of servers used for storing and delivering data. Basically, these jQuery library files are already uploaded to various CDNs and we can use them directly on our web page. Then, we don’t need to download any files on our local machine.

*<script src=”https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js”></script>*

We can see the CDN link inside the “src” attribute. We have successfully added jQuery to our web page. We can use all the features of jQuery on our page. While loading the page, the browser will automatically download the jQuery library files from the CDN link.

**Example 1:** In this example, we will add jQuery CDN link to execute jQuery code.

<!DOCTYPE html>

<html>

  <head>

    <!-- jQuery library -->

    <script src=

"<https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js>">

    </script>

</head>

<body>

    <p></p>

      <script>

        $('p').text('CDN Working');

    </script>

</body>

  </html>

**2. Download the jQuery library:** In this way, we will add jQuery library to our page. First, we will download the jQuery library files to our localhost from the[jQuery](https://jquery.com/download/) Website. After downloading, we will add the downloaded files to our web page in this manner.

<script src="file\_name\_with\_full\_path"></script>

**Example 2:** In this example, we will add jQuery link from downloaded path to execute jQuery code.

<!DOCTYPE html>

<html>

<head>

    <!-- jQuery library -->

    <script src="jquery-3.6.0.js"></script>

</head>

<body>

    <p></p>

    <script>

        $('p').text('Downloaded files');

    </script>

</body>

</html>

## jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: **$(*selector*).*action*()**

* A $ sign to define/access jQuery
* A (*selector*) to "query (or find)" HTML elements
* A jQuery *action*() to be performed on the element(s)

Examples:

$(this).hide() - hides the current element.

$("p").hide() - hides all <p> elements.

$(".test").hide() - hides all elements with class="test".

$("#test").hide() - hides the element with id="test".

## The Document Ready Event

$(document).ready(function(){  
  
  *// jQuery methods go here...*  
  
});

**$(document).ready()**

The $(document).ready() method allows us to execute a function when the document is fully loaded.

This is to prevent any jQuery code from running before the document is finished loading (is ready).

It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

# jQuery Event Methods

**click()**

The click() method attaches an event handler function to an HTML element.

The function is executed when the user clicks on the HTML element.

**dblclick()**

The dblclick() method attaches an event handler function to an HTML element.

**mouseenter()**

The mouseenter() method attaches an event handler function to an HTML element.

**mouseleave()**

The mouseleave() method attaches an event handler function to an HTML element.

**mousedown()**

The mousedown() method attaches an event handler function to an HTML element.

**mouseup()**

The mouseup() method attaches an event handler function to an HTML element.

**hover()**

The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.

**focus()**

The focus() method attaches an event handler function to an HTML form field.

**blur()**

The blur() method attaches an event handler function to an HTML form field.

## jQuery Effects

Hide, Show, Toggle, Slide, Fade, and Animate.

## jQuery hide() and show()

With jQuery, you can hide and show HTML elements with the hide() and show() methods:

JQuery Example-1

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#hide").click(function(){

$("p").hide();

});

$("#show").click(function(){

$("p").show();

});

});

</script>

</head>

<body>

<p>If you click on the "Hide" button, I will disappear.</p>

<button id="hide">Hide</button>

<button id="show">Show</button>

</body>

</html>

JQuery Example-2

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#p1").mouseenter(function(){

$("p").hide();

});

$("#p1").mouseleave(function(){

$("p").show();

});

});

</script>

</head>

<body>

<p id="p1">Hello Welcome to jquery program</p>

</html>

The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.

**Syntax:**

$(*selector*).hide(*speed,callback*);  
  
$(*selector*).show(*speed,callback*);

$("p").hide(“slow”);

$("p").hide(1000);

A callback function is executed after the current effect is finished.

Typical syntax: **$(*selector*).hide(*speed,callback*);**

$("button").click(function(){  
  $("p").hide(1000,function(){  
    alert("The paragraph is now hidden");  
  });  
});

## jQuery toggle()

You can also toggle between hiding and showing an element with the toggle() method.

Shown elements are hidden and hidden elements are shown:

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").toggle();

});

});

</script>

</head>

<body>

<button>Toggle between hiding and showing the paragraphs</button>

<p>This is a paragraph with little content.</p>

<p>This is another small paragraph.</p>

</body>

</html>

# jQuery Effects - Sliding

With jQuery you can create a sliding effect on elements.

jQuery has the following slide methods:

* slideDown()
* slideUp()
* slideToggle()

The jQuery slideDown() method is used to slide down an element.

**Syntax:**

$(*selector*).slideDown(*speed,callback*);

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideDown("slow");

});

});

</script>

<style>

#panel, #flip {

padding: 5px;

text-align: center;

background-color: #e5eecc;

border: solid 1px #c3c3c3;

}

#panel {

padding: 50px;

display: none;

}

</style>

</head>

<body>

<div id="flip">Click to slide down panel</div>

<div id="panel">Hello world!</div>

</body>

</html>

## jQuery slideUp() Method

The jQuery slideUp() method is used to slide up an element.

**Syntax:**

$(*selector*).slideUp(*speed,callback*);

<!DOCTYPE html>

<html>

<head>

<script

src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideUp("slow");

});

});

</script>

<style>

#panel, #flip {

padding: 5px;

text-align: center;

background-color: #e5eecc;

border: solid 1px #c3c3c3;

}

#panel {

padding: 50px;

}

</style>

</head>

<body>

<div id="flip">Click to slide up panel</div>

<div id="panel">Hello world!</div>

</body>

</html>

## jQuery slideToggle() Method

The jQuery slideToggle() method toggles between the slideDown() and slideUp() methods.

$(*selector*).slideToggle(*speed,callback*);

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideToggle("slow");

});

});

</script>

<style>

#panel, #flip {

padding: 5px;

text-align: center;

background-color: #e5eecc;

border: solid 1px #c3c3c3;

}

#panel {

padding: 50px;

display: none;

}

</style>

</head>

<body>

<div id="flip">Click to slide the panel down or up</div>

<div id="panel">Hello world!</div>

</body>

</html>

## jQuery HTML

## jQuery DOM Manipulation

One very important part of jQuery is the possibility to manipulate the DOM.

jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.

## Get Content - text(), html(), and val()

Three simple, but useful, jQuery methods for DOM manipulation are:

* text() - Sets or returns the text content of selected elements
* html() - Sets or returns the content of selected elements (including HTML markup)
* val() - Sets or returns the value of form fields

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

alert("Value: " + $("#test").val());

});

});

</script>

</head>

<body>

<p>Name: <input type="text" id="test"></p>

<button>Show Value</button>

</body>

</html>

## Remove Elements/Content

To remove elements and content, there are mainly two jQuery methods:

* remove() - Removes the selected element (and its child elements)
* empty() - Removes the child elements from the selected element

## jQuery remove() Method

The jQuery remove() method removes the selected element(s) and its child elements.

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#div1").remove();

});

});

</script>

</head>

<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

This is some text in the div.

<p>This is a paragraph in the div.</p>

<p>This is another paragraph in the div.</p>

</div>

<br>

<button>Remove div element</button>

</body>

</html>

## jQuery empty() Method

The jQuery empty() method removes the child elements of the selected element(s).

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#div1").empty();

});

});

</script>

</head>

<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

This is some text in the div.

<p>This is a paragraph in the div.</p>

<p>This is another paragraph in the div.</p>

</div>

<br>

<button>Empty the div element</button>

</body>

</html>

# jQuery - Get and Set CSS Classes

## jQuery Manipulating CSS

jQuery has several methods for CSS manipulation. We will look at the following methods:

* addClass() - Adds one or more classes to the selected elements
* removeClass() - Removes one or more classes from the selected elements
* toggleClass() - Toggles between adding/removing classes from the selected elements
* css() - Sets or returns the style attribute

## jQuery addClass() Method

The following example shows how to add class attributes to different elements. Of course you can select multiple elements, when adding classes:

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("h1, h2, p").addClass("blue");

$("div").addClass("important");

});

});

</script>

<style>

.important {

font-weight: bold;

font-size: xx-large;

}

.blue {

color: blue;

}

</style>

</head>

<body>

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

<div>This is some important text!</div><br>

<button>Add classes to elements</button>

</body>

</html>

## jQuery removeClass() Method

The following example shows how to remove a specific class attribute from different elements:

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("h1, h2, p").removeClass("blue");

});

});

</script>

<style>

.blue {

color: blue;

}

</style>

</head>

<body>

<h1 class="blue">Heading 1</h1>

<h2 class="blue">Heading 2</h2>

<p class="blue">This is a paragraph.</p>

<p>This is another paragraph.</p>

<button>Remove class from elements</button>

</body>

</html>

## jQuery toggleClass() Method

The following example will show how to use the jQuery toggleClass() method. This method toggles between adding/removing classes from the selected elements:

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("h1, h2, p").toggleClass("blue");

});

});

</script>

<style>

.blue {

color: blue;

}

</style>

</head>

<body>

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

<button>Toggle class</button>

</body>

</html>

# jQuery - css() Method

The css() method sets or returns one or more style properties for the selected elements.

## Return a CSS Property

To return the value of a specified CSS property, use the following syntax:

css("*propertyname*");

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

alert("Background color = " + $("p").css("background-color"));

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>

<button>Return background-color of p</button>

</body>

</html>

## Set a CSS Property

To set a specified CSS property, use the following syntax:

css("*propertyname*","*value*");

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").css("background-color", "yellow");

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>

<p style="background-color:#00ff00">This is a paragraph.</p>

<p style="background-color:#0000ff">This is a paragraph.</p>

<p>This is a paragraph.</p>

<button>Set background-color of p</button>

</body>

</html>

## Set Multiple CSS Properties

To set multiple CSS properties, use the following syntax:

css({"*propertyname*":"*value*","*propertyname*":"*value*",...});

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").css({"background-color": "yellow", "font-size": "200%"});

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>

<p style="background-color:#00ff00">This is a paragraph.</p>

<p style="background-color:#0000ff">This is a paragraph.</p>

<p>This is a paragraph.</p>

<button>Set multiple styles for p</button>

</body>

</html>

JQuery Example-3

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("input").focus(function(){

$(this).css("background-color", "yellow");

});

$("input").blur(function(){

$(this).css("background-color", "green");

});

});

</script>

</head>

<body>

Name: <input type="text" name="fullname"><br>

Email: <input type="text" name="email">

</body>

</html>