**Assignment-2**

1. **Write a Short note on jQuery?**

* **JQuery** is a fast, small, cross-platform and feature-rich JavaScript library. It is designed to simplify the client-side scripting of HTML. It makes things like HTML document traversal and manipulation, animation, event handling, and AJAX very simple with an easy-to-use API that works on a lot of different type of browsers. The main purpose of jQuery is to provide an easy way to use JavaScript on your website to make it more interactive and attractive. It is also used to add animation.
* **Following are the important features of jQuery:**
* HTML manipulation, DOM manipulation, DOM element selection, CSS manipulation, Effects and Animations, Utilities, AJAX, HTML event methods, JSON Parsing, Extensibility through plug-ins.

1. **Describe about jQuery DOM Manipulation?**

* **JQuery DOM Manipulation:** jQuery provides methods such as attr(), html(), text() and val() which act as getters and setters to manipulate the content from HTML documents. Document Object Model (DOM) - is a W3C (World Wide Web Consortium) standard that allows us to create, change, or remove elements from the HTML or XML documents.
* **Here are some basic operations which you can perform on DOM elements with the help of jQuery standard library methods:**
* Extract the content of an element
* Change the content of an element
* Adding a child element under an existing element
* Adding a parent element above an existing element
* Adding an element before or after an existing element
* Replace an existing element with another element
* Delete an existing element
* Wrapping content with-in an element

1. **How to Call a jQuery Library Functions?**

* If you want an event to work on your page, you should call it inside the $(document).ready() function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded. To do this, we register a ready event for the document as follows:

|  |
| --- |
| $(document).ready(function() {  // do stuff when DOM is ready  }); |

* **To call upon any jQuery library function, use HTML script tags as shown below:**

|  |
| --- |
| <!doctype html>  <html>  <head>  <title>The jQuery Example</title>  <script src="https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>  <script>  $(document).ready(function() {  $("div").click(function() {alert("Hello, world!");});  });  </script> </head>  <body>  <div>Click on this to see a dialogue box.</div>  </body> </html> |

1. **Explain jQuery DOM Manipulation methods?**

* **JQuery DOM Manipulation:** jQuery provides various methods to add, edit or delete DOM element(s) in the HTML page. The following table lists some important methods to add/remove new DOM elements.

|  |  |
| --- | --- |
| **Method** | **Description** |
| **append()** | Inserts content to the end of element(s) which is specified by a selector. |
| **before()** | Inserts content (new or existing DOM elements) before an element(s) which is specified by a selector. |
| **after()** | Inserts content (new or existing DOM elements) after an element(s) which is specified by a selector. |
| **prepend()** | Insert content at the beginning of an element(s) specified by a selector. |
| **remove()** | Removes element(s) from DOM which is specified by selector. |
| **replaceAll()** | Replace target element(s) with specified element. |
| **wrap()** | Wrap an HTML structure around each element which is specified by selector. |

1. **Explain CSS Manipulation using jQuery library methods?**

* **CSS Manipulation using jQuery:** CSS manipulation using jQuery library methods allows you to dynamically modify the styles of HTML elements on a web page. jQuery provides several methods for working with CSS properties, including css(), addClass(), hasClass(), removeClass(), and toggleClass(). These methods make it easy to change the appearance and behavior of elements in response to user interactions or other events.
* **css():** The css() method allows you to get or set CSS properties for one or more elements. You can use it to both retrieve and change the values of CSS properties. Here's an example of how to use it:

|  |
| --- |
| // Get the value of a CSS property  var fontSize = $('#myElement').css('font-size');  // Set the value of a CSS property  $('#myElement').css('color', 'red'); |

* **addClass():** The addClass() method adds one or more CSS classes to selected elements. This can be used to apply predefined styles to elements or add new styles dynamically. For example:

|  |
| --- |
| $('#myElement').addClass('highlighted'); |

* **hasClass():** The hasClass() method checks if a selected element has a specific CSS class. It returns true if the class exists on the element and false otherwise. This is useful for conditionally applying styles based on whether an element has a certain class:

|  |
| --- |
| if ($('#myElement').hasClass('highlighted')) {  // Do something if the element has the 'highlighted' class  } |

* **removeClass():** The removeClass() method removes one or more CSS classes from selected elements. This is helpful for removing styles or toggling the appearance of elements. For example:

|  |
| --- |
| $('#myElement').removeClass('highlighted'); |

* **toggleClass():** The toggleClass() method toggles the presence of a CSS class on selected elements. If the class is present, it is removed; if it's not present, it's added. This is often used for creating interactive elements like buttons that change appearance when clicked:

|  |
| --- |
| // Toggle a CSS class on an element$('#myButton').click(function() {  $('#myElement').toggleClass('active');  }); |

* These jQuery CSS manipulation methods are powerful tools for creating dynamic and interactive web pages. They allow you to easily change styles, apply animations, and respond to user actions without the need for complex JavaScript and CSS code.

1. **Short note on jQuery dimensions with diagram?**

* **JQuery dimensions:** jQuery has several important methods for working with dimensions: width(), height(), innerWidth(), innerHeight(), outerWidth(), outerHeight().
* **JQuery width() and height() Methods:** The width() method sets or returns the width of an element (excludes padding, border and margin). The height() method sets or returns the height of an element (excludes padding, border and margin). The following example returns the width and height of a specified <div> element:

|  |
| --- |
| $("button").click(function(){   var txt = "";   txt += "Width: " + $("#div1").width() + "</br>";   txt += "Height: " + $("#div1").height();   $("#div1").html(txt); }); |

|  |
| --- |
|  |

* **JQuery innerWidth() and innerHeight() Methods:** The innerWidth() method returns the width of an element (includes padding). The innerHeight() method returns the height of an element (includes padding). The following example returns the inner-width/height of a specified <div> element:

|  |
| --- |
| $("button").click(function(){   var txt = "";   txt += "Inner width: " + $("#div1").innerWidth() + "</br>";   txt += "Inner height: " + $("#div1").innerHeight();   $("#div1").html(txt); }); |

* **JQuery outerWidth() and outerHeight() Methods:** The outerWidth() method returns the width of an element (includes padding and border). The outerHeight() method returns the height of an element (includes padding and border). The following example returns the outer-width/height of a specified <div> element:

|  |
| --- |
| $("button").click(function() {    var txt = "";    txt += "Outer width: " + $("#div1").outerWidth() + "</br>";    txt += "Outer height: " + $("#div1").outerHeight();    $("#div1").html(txt);  }); |

* **JQuery More width() and height():** The following example returns the width and height of the document (the HTML document) and window (the browser viewport):

|  |
| --- |
| $("button").click(function(){   var txt = "";   txt += "Document width/height: " + $(document).width();   txt += "x" + $(document).height() + "\n";   txt += "Window width/height: " + $(window).width();   txt += "x" + $(window).height();   alert(txt); }); |

1. **What is Jquery Traversing? Explain why it is necessary?**

* **JQuery Traversing:** JQuery is a very powerful tool that provides a variety of DOM traversal methods to assist us in selecting elements in an HTML or XML document in both a random and sequential manner.
* The DOM organizes elements into a tree-like data structure that can be traversed to navigate and locate content within an HTML or XML document.
* The DOM tree can be thought of as a collection of nodes connected by parent-child and sibling-sibling relationships, with the root starting from the top parent, which is an HTML element in an HTML document.
* **DOM traversing using jQuery:** It is used to find (or select) HTML elements based on their relationship to other elements. Begin with one option and work your way through it until you reach the desired elements.
* An HTML page is depicted as a tree in the image below (DOM tree). You can easily move up(ancestors), down(descendants), and sideways(siblings) in the tree using jQuery traversing, starting from the selected (current) element. This is known as traversing - or moving through the DOM tree.
* jQuery provides a number of methods for traversing the DOM. Tree traversal is the most common type of traversal method. An ancestor is a parent, grandparent, great-grandparent, and so on in logical relationships.
* jQuery provides useful methods such as parent(), parents(), and parentsUntil() that can be used to traverse the DOM tree on single or multiple levels to quickly get the parent or other ancestors of an element in the hierarchy.
* The majority of DOM Traversal Methods do not modify the jQuery DOM object and are used to filter out elements from a document based on specified conditions.
* **jQuery provides methods for traversing in three directions:**
* **Traversing Upwards** - This path leads to the ancestors (Parent, Grandparent, Great-grandparent).
* **Traversing Downwards-** entails traversing the descendants in this direction (Child, Grandchild, Great-grandchild).
* **Sideways** - Traveling in this direction means passing through the ancestors and siblings (For example, brother and sisters are at the same level).

1. **Explain jQuery methods for traversing in Upwards?**

* **Traversing Up the DOM Tree:** Three useful jQuery methods for traversing up the DOM tree are: parent(), parents(), parentsUntil().
* **jQuery parent() Method:** The parent() method returns the direct parent element of the selected element. This method only traverse a single level up the DOM tree. The following example returns the direct parent element of each <span> elements:

|  |
| --- |
| $(document).ready(function(){   $("span").parent(); }); |

* **jQuery parents() Method:** The parents() method returns all ancestor elements of the selected element, all the way up to the document's root element (<html>). The following example returns all ancestors of all <span> elements:

|  |
| --- |
| $(document).ready(function(){    $("span").parents();  }); |

* **jQuery parentsUntil() Method:** The parentsUntil() method returns all ancestor elements between two given arguments. The following example returns all ancestor elements between a <span> and a <div> element:

|  |
| --- |
| $(document).ready(function(){   $("span").parentsUntil("div");  }); |

1. **Explain jQuery methods for traversing in Downwards?**

* **Traversing Down the DOM Tree:** Two useful jQuery methods for traversing down the DOM tree are: children() & find()
* **jQuery children() Method:** The children() method returns all direct children of the selected element. This method only traverses a single level down the DOM tree. The following example returns all elements that are direct children of each <div> elements:

|  |
| --- |
| **Example1:**  $(document).ready(function(){  $("div").children();  }); |
| **Example2:**  $(document).ready(function(){    $("div").children("p.first");  }); |

* **jQuery find() Method:** The find() method returns descendant elements of the selected element, all the way down to the last descendant. The following example returns all <span> elements that are descendants of <div>:

|  |
| --- |
| **Example1:**  $(document).ready(function(){   $("div").find("span"); }); |
| **Example2:**  $(document).ready(function(){   $("div").find("\*"); }); |

1. **Explain jQuery methods for traversing in sideways?**

* **JQuery methods for traversing in sideways**: Traveling in this direction means passing through the ancestors and siblings (For example, brother and sisters are at the same level).With jQuery you can traverse sideways in the DOM tree to find siblings of an element. Siblings share the same parent. Traversing Sideways in The DOM Tree. There are many useful jQuery methods for traversing sideways in the DOM tree: siblings(), next(), nextAll(), nextUntil(), prev(), prevAll(), prevUntil().
* **jQuery siblings() Method:** The siblings() method returns all sibling elements of the selected element. The following example returns all sibling elements of <h2>:

|  |
| --- |
| $(document).ready(function(){  $("h2").siblings(); }) |

* **jQuery next() Method:** The next() method returns the next sibling element of the selected element. The following example returns the next sibling of <h2>:

|  |
| --- |
| $(document).ready(function(){ $("h2").next();}); |

* **jQuery nextAll() Method:** The nextAll() method returns all next sibling elements of the selected element. The following example returns all next sibling elements of <h2>:

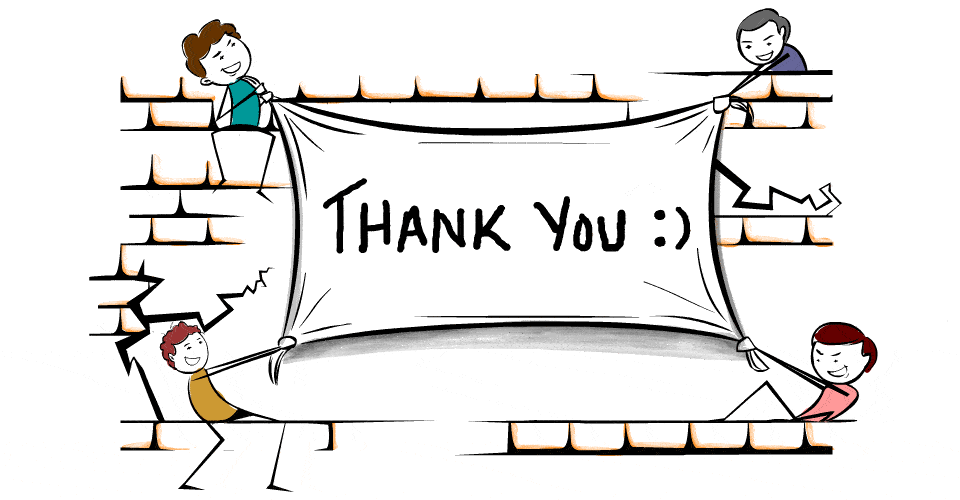
|  |
| --- |
| $(document).ready(function(){ $("h2").nextAll(); }); |

* **jQuery nextUntil() Method:** The nextUntil() method returns all next sibling elements between two given arguments.The following example returns all sibling elements between a <h2> and a <h6> element:

|  |
| --- |
| $(document).ready(function(){  $("h2").nextUntil("h6"); }); |

* **jQuery prev(), prevAll() & prevUntil() Methods:**The prev(), prevAll() and prevUntil() methods work just like the methods above but with reverse functionality: they return previous sibling elements (traverse backwards along sibling elements in the DOM tree, instead of forward).

**(PPT Give you more understanding than PDF)** The material for the PDF has been compiled from various sources such as books, tutorials (offline and online), lecture notes, several resources available on Internet. The information contained in this PDF is for general information and education purpose only. While we endeavor to keep the information up to date and correct, **we make no representation of any kind about the completeness and accuracy of the material.** The information shared through this PDF material should be used for educational purpose only.

****