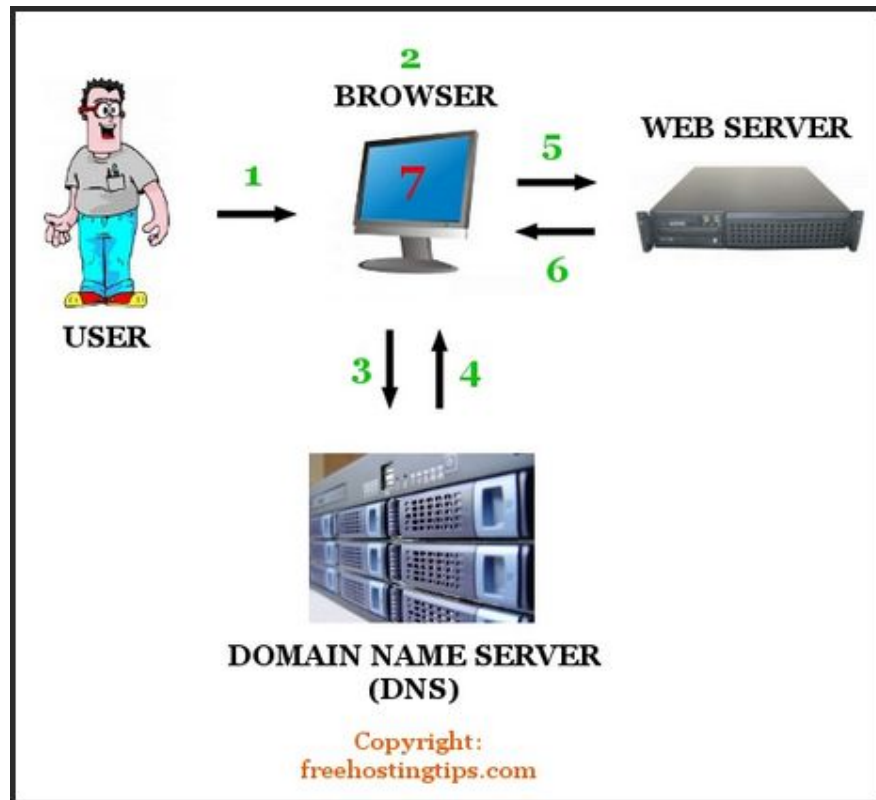


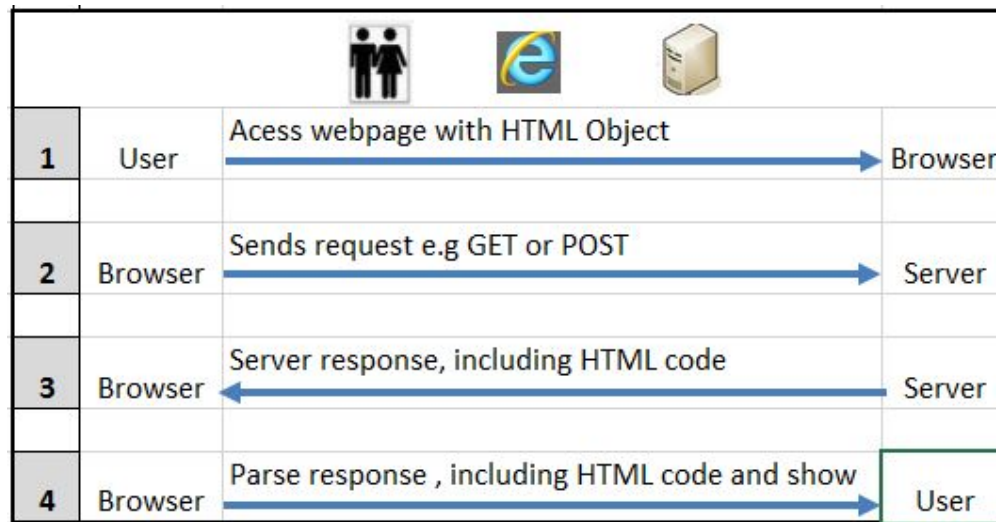
## HTML DOM and UTF



“Take the first step in faith. You don't have to see the whole staircase, just take the first step.”  
- Martin Luther King, Jr. quotes

Thanks to UFT add-ins, UFT can work with various technology. Web applications are the most popular and easy way to provide services and information to the end-user. It is true that initially QTP, UFT's predecessor, became very popular due to the easiness and vast supported features for web technologies. Even as today, it is one of the most saleable point for UFT. There are other famous tools also became very popular on web technology automation now. The central theme to do web automation is to work with HTML objects and communication between automation tool and HTML objects.

First let's understand the basic of HTML. Browser is a software that can understand many languages including HTML, JavaScript etc. When we browse on the browser, we effectively send request to server. Server returns response with HTML code. Browser parses the return response including HTML code and shows on the browser web page.



So browser is intermediary that understands the user request in form of HTML and other languages and is able to communicate with server in bi-directional manner.

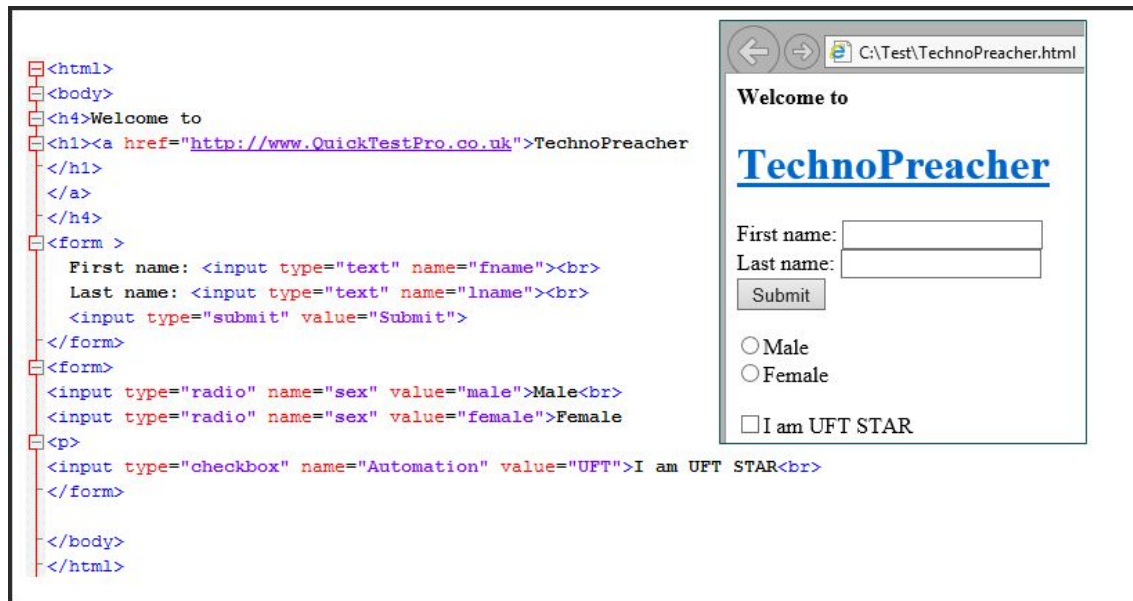
HTML is HyperText Markup Language (HTML) for creating web pages. HTML comprises "HTML elements" (or loosely refer as HTML tag). HTML tags give meaning to the content e.g. <h1> heading, <img> image, <p> paragraph etc. Usually tags come in pairs <h1>This is Heading </h1> but there are many tags that do not require closing tag e.g. <br> Line break.

Exercise – Create a webpage using HTML programming

STEP1: Create "TechnoPreacher" webpage using HTML



*Open Notepad (or Notepad++) | write following code | Save file "C:\Test\TechnoPreacher.htm" | Double click*



Concept of Exercise – Create a webpage using HTML programming

HTML Tag is listed below in the given table for reference:

HTML Tag (Alphabetically)	TAG Description
<!--...-->	This tag used for a comment
<a>	This tag used for a hyperlink
<b>	This tag used for bold text
<body>	This tag used for the document's body
 	This tag used for a single line break
<button>	This tag used for a clickable button
<dialog>New	This tag used for a dialog box or window
<div>	This tag used for a section in a document
<font>	Not supported in HTML5. Depreciated in HTML 4.01. This tag used for font, color, and size for text
<footer>New	This tag used for a footer for a document or section
<form>	This tag used for an HTML form for user input
<frame>	Not supported in HTML5. This tag used for a window (a frame) in a frameset
<h1> to <h6>	This tag used for HTML headings
<head>	This tag used for information about the document
<html>	This tag used for the root of an HTML document
<img>	This tag used for an image
<input>	This tag used for an input control
<ins>	This tag used for a text that has been inserted into a document
<label>	This tag used for a label for an <input> element
<li>	This tag used for a list item

<b>&lt;link&gt;</b>	This tag used for the relationship between a document and an external resource
<b>&lt;object&gt;</b>	This tag used for an embedded object
<b>&lt;ol&gt;</b>	This tag used for an ordered list
<b>&lt;option&gt;</b>	This tag used for an option in a drop-down list
<b>&lt;p&gt;</b>	This tag used for a paragraph
<b>&lt;ruby&gt;New</b>	This tag used for a ruby annotation
<b>&lt;script&gt;</b>	This tag used for a client-side script
<b>&lt;select&gt;</b>	This tag used for a drop-down list
<b>&lt;span&gt;</b>	This tag used for a section in a document
<b>&lt;style&gt;</b>	This tag used for style information for a document
<b>&lt;table&gt;</b>	This tag used for a table
<b>&lt;tbody&gt;</b>	Groups the body content in a table
<b>&lt;td&gt;</b>	This tag used for a cell in a table
<b>&lt;textarea&gt;</b>	This tag used for a multiline input control (text area)
<b>&lt;th&gt;</b>	This tag used for a header cell in a table
<b>&lt;title&gt;</b>	This tag used for a title for the document
<b>&lt;tr&gt;</b>	This tag used for a row in a table

## So, what is HTML DOM?

If you notice the html code there is a tree structure. So, can we traverse to the element? Or can we parse the whole html code (i.e. document) in such a way that we can access the html element directly. In previous chapters, we witnessed the “Objectization” of MS Office applications. In same way, if we can get some kind of object by which we can manipulate html element (loosely tags) then we can programmatically work along with HTML web page.

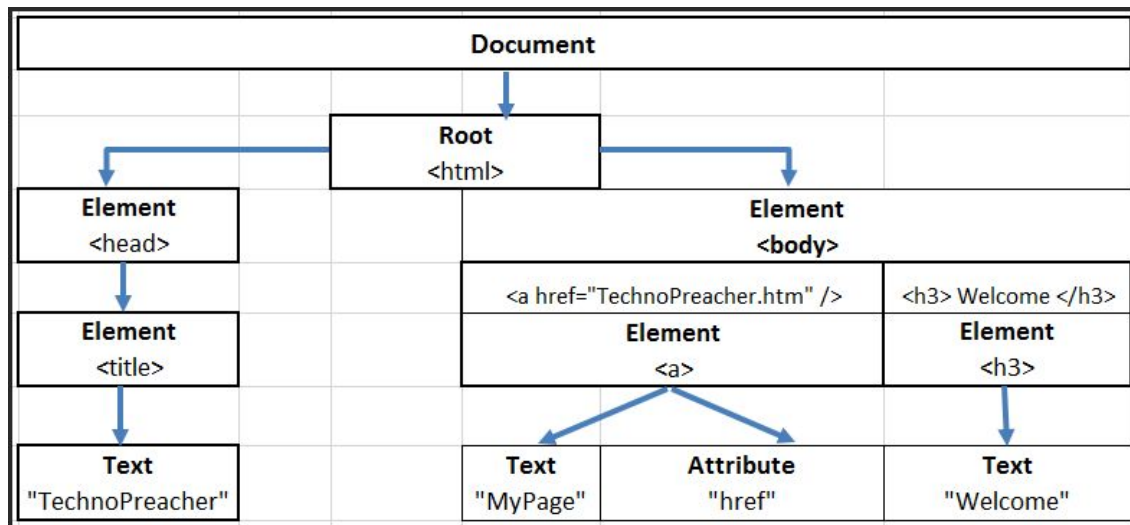
So, effectively, HTML DOM is a standard. This standard specify the way to manipulate, read, delete or add HTML elements and that can be used in programming to manage web page during run-time. The HTML DOM defines the objects method and properties of all HTML elements to access them. if the word HTML DOM strike in your mind then the key is

<b>What</b>	Standard
<b>Why</b>	Programming way for HTML Webpage
<b>WHO</b>	World Wide Web Consortium (W3C)

HTML tree structure can be consider as document. So, according to W3C HTML DOM standard everything in an html document is a Node. HTML element consider as an element node, text in html element is text node while HTML attribute is an attribute node.

<b>&lt;title&gt;</b>	<b>Aasish Katare</b>	<b>&lt;/title&gt;</b>
ElementNode	Text Node (value of textnode by innerHTML)	

So, it is basically all about node.



So now these node can be read, delete or modify by the programming language such as Javascript.If you notices the html document there is clear hierarchy of the nodes e.g. HTML node is the parent of all node, body and head are at same level and two form node are at same level. So it is Parent – Child – Sibling relationship among all the node.

## Relation among Nodes in HTML document

In HTML document the top node is root node. Every node can have zero or more children and exactly one parent node (except Mr Adam “root”). Node at same level called as sibling

<b>Root</b>			<html>
<b>Sibling</b>	<b>Child 1</b>		<head>
		<b>Child 1</b>	<title>TechnoPreacher</title>
			</head>
	<b>Child 2</b>		<body>
	<b>Sibling</b>	<b>Child 1</b>	<h1>QuickTestPro.CO.UK</h1>
		<b>Child 2</b>	<p>Aadhya! UFTSTAR</p>
			</body>
			</html>

In this example, “title” is element node and “TechnoPreacher” is text node. So, element node can contain the text node and the value of the text node can be found by innerHTML property.

Exercise - Accessing HTML element by JavaScript using HTML DOM



*Open Notepad (or Notepad++) | write following code | Save file “C:\Test\DomScript.htm” | Double click on “DomScript.htm” | Provide email “Nidhi@Prathvipur.com” | Click on button*

```

1 <html>
2 <body>
3 <a href="http://www.QuickTestPro.co.uk" id="myllink">TechnoPreacher</a>
4 <form >
5   Email: <input type="text" id="xpgv" value="contact@quicktestpro.co.uk">
6 </form>
7 <br>
8 <button type="button" onclick="javascriptFunction()">Access Email node</button>
9
10 <script type="text/javascript">
11 function javascriptFunction() {
12   var x=document.getElementById("xpgv");
13   var y=document.getElementById("xpgv").defaultValue;
14   document.write("<p>EMail is : " + x.value + "</p>");
15   alert('OLD EMAIL - ' + y);
16 }
17 </script>
18
19 </body>
20 </html>

```

#### Concept of Exercise - Accessing HTML element by JavaScript using HTML DOM

In STEP1, we have created a HTML page. The html elements can be access and modified by JavaScript with html DOM. Line-8 javascriptFunction() calls the function at line-11. We accessed InputText element with getElementById method of DOM document in Line-12. On Line-13, we access the default value of InputText element i.e. old value in text box. On Line-14, we are writing on webpage by using document.write method. “x.value” return the text in the InputText element. The value changed from [contact@quicktestpro.co.uk](mailto:contact@quicktestpro.co.uk) to “Nidhi@Prathvipur.com” after changing the text. So, by using JavaScript and DOM methods we can access the old or changing values (or other properties) of the html elements.

JavaScript, similar to VBScript, is a scripting language supported by all major browser. It is out of scope of the book but if you look at code you may figure out the logic from it. Method alert is similar to MsgBox while var is similar to Dim. Our aim is to understand HTML element manipulation. Later we will see the UFT way to access HTML DOM instead of JavaScript.

In this example, we encountered InputText element. So how many type of DOM element we can access? Following are the HTML element that can be access by HTML DOM.

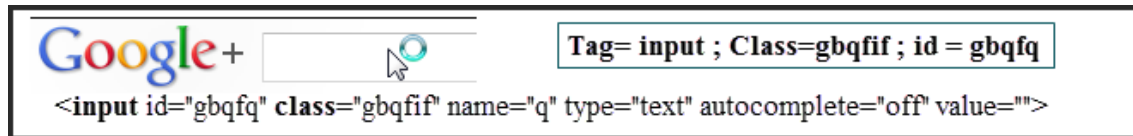
Anchor object	<b>Input Radio object</b>	<b>Image object</b>	<b>Select object</b>
Area object	Input Reset object	<b>Input Button object</b>	Style object
Base object	<b>Input Submit object</b>	<b>Input Checkbox object</b>	Table object
Body object	<b>Input Text object</b>	Input File object	td / th object
<b>Button object</b>	<b>Link object</b>	Input Hidden object	tr object
<b>Form object</b>	Meta object	Input Password object	<b>Textarea object</b>
Frame/IFrame	<b>Object object</b>	Option object	Frameset object

We have seen the “getElementById” method in document. So suppose we do not have the id attribute in the tag then how we can access the html element? To solve it, document provide various method to

access the html element. You can find all the method of document [REF]. Let's explore few popular methods.

1. getElementById() method – Always unique, return reference of single object
2. getElementsByTagName() - Return list of element
3. getElementsByClassName() - - Return list of element

This is the example of Google edit box. We can manipulate the same edit box by either of method for example document.getElementsByClassName("gbqfif");



So how we can change the html element property during run-time? Suppose we have label with id “biologist1” the following code will change the text Rani to Sippy.

```
<p id="biologist1">Rani</p>
document.getElementById("biologist1").innerHTML="Sippy"
```

## A Real Story Start from here – UFT and HTML DOM

UFT provide the simplest way to access HTML DOM. Using .Object on Page test object will provide access on all the object on webpage.

```
Systemutil.Run "iexplore.exe", "www.QuickTestPro.co.uk"
Set AllObjectOnPage = Browser("Browser").Page("Quick Test Pro").Object
Browser("Browser").CloseAllTabs
```

So, effectively, if you hear “HTML DOM” in UFT world then just think about “.Object”. This means that any operation that you can perform on a DOM object, you can also perform by running a **<WebObject>.Object** statement on the Web object. DOM implemented differently for IE and Firefox so, one code may not work on the other browser. The detail of IE and Firefox DOM is in reference [REF].

## Common DOM Methods & Properties

When we apply .Object in code it returns the reference to this DOM object. This document object represents the entire page. From this document object, we can access the rest of the object hierarchy by using properties and collections.

Methods	Description
getElementById Method	Returns a reference to the object with the specified ID

getElementsByName & getElementsByTagName	Retrieves a collection of objects based on the value of the NAME attribute.
<b>Properties</b>	<b>Description</b>
activeElement Property	Retrieves the object that has the focus when the parent document has focus.
cookie Property	Sets or retrieves the string value of a cookie.
documentElement Property	Retrieves a reference to the root node of the document.
readyState Property	Retrieves a value that indicates the current state of the object.
URL Property	Sets or retrieves the URL for the current document.
URLUnencoded	Retrieves the URL for the document, stripped of any character encoding.
<b>Collections</b>	<b>Description</b>
all, frames, images, and links Collections	Returns collection of elements contained by the object, img objects & HREF property

REF DOM document <http://msdn.microsoft.com/en-us/library/windows/apps/hh453143.aspx>

REF Internet Explorer DOM <http://msdn.microsoft.com/en-us/library/ms533022.aspx>.

REF Firefox DOM <https://developer.mozilla.org/En/DOM>.