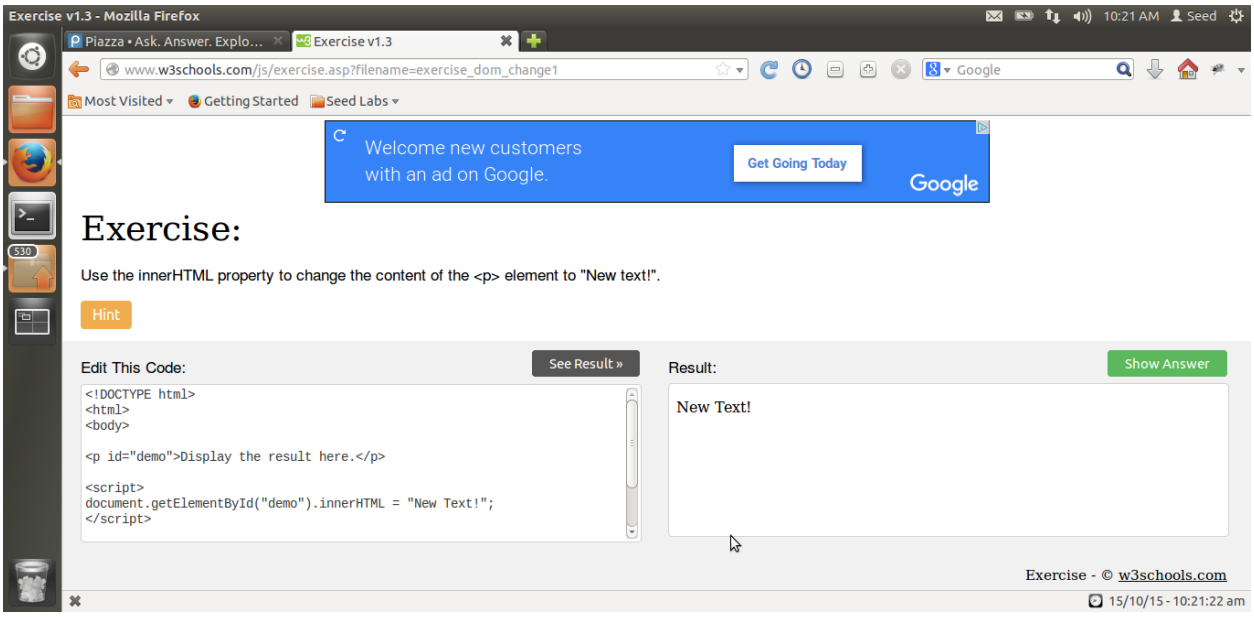
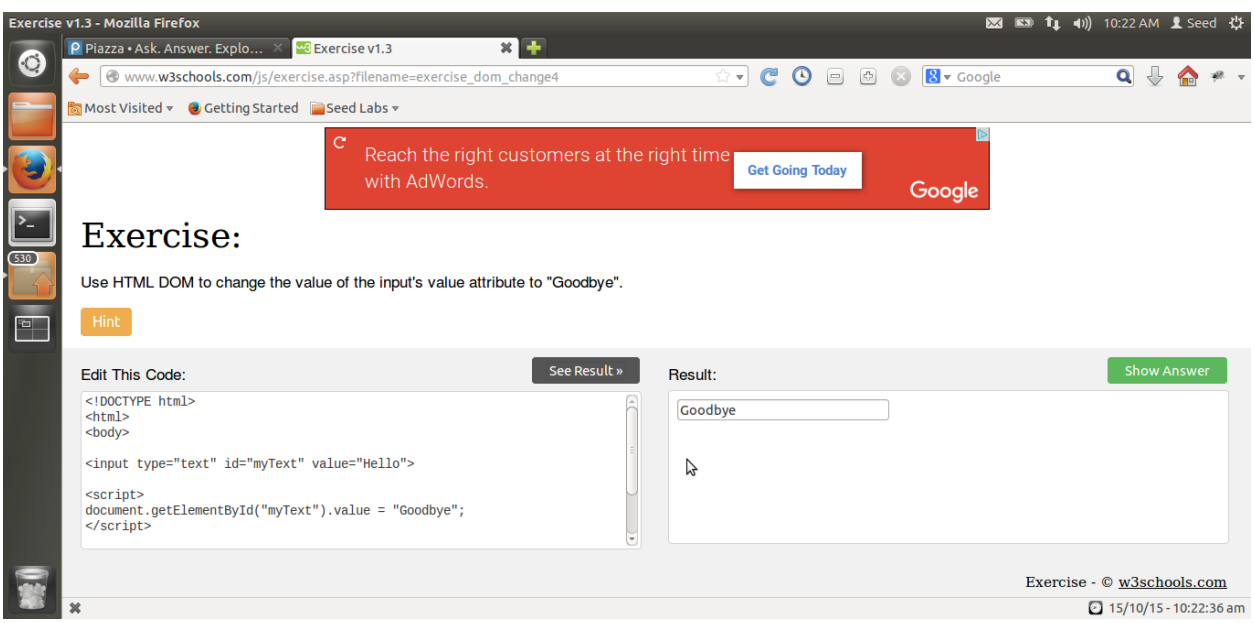
Yogesh Chaudhari
SUID: 244195971

1. DOM
   innerHTML:



By using getElementById() we can select any DOM element from HTML file using its specific ID. Here we can replace contents of html file or contents under any of its DOM elements by using .innerHTML() command on the respective element/node of HTML document. We are using .innerHTML on <p> element with ID = “demo” to replace contents under <p> tag in guven HTML file.

Changing attribute value:

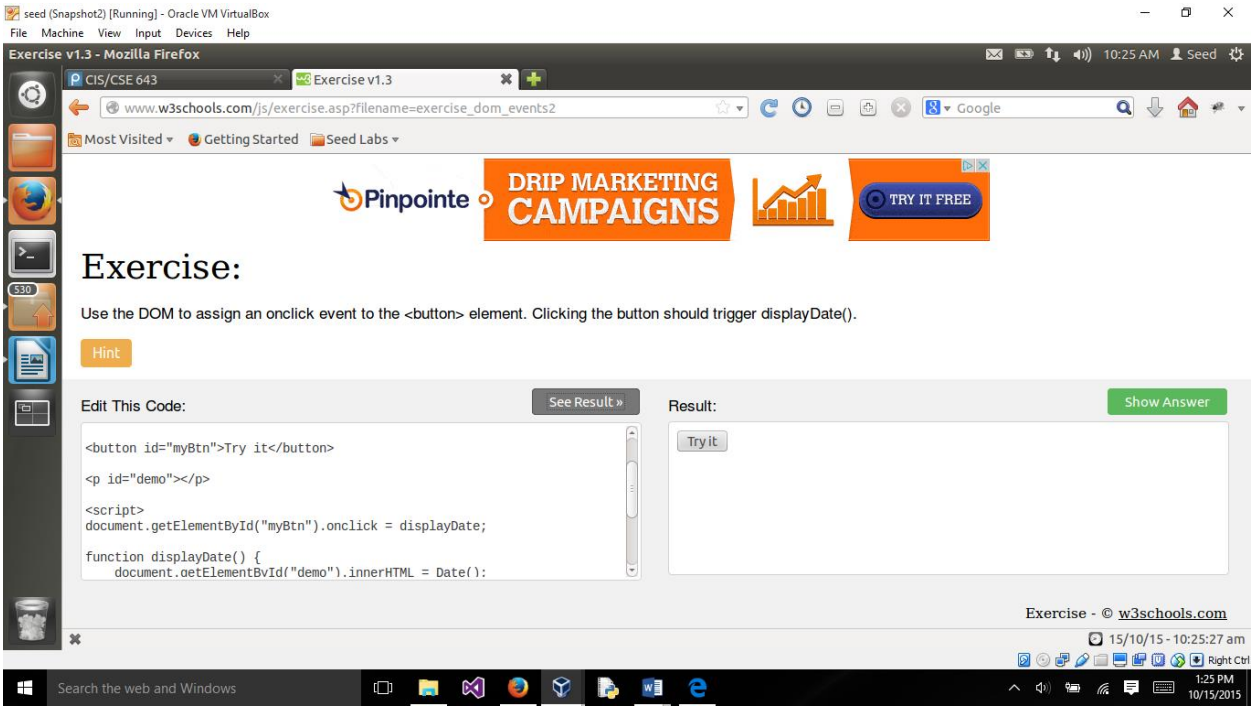


Here, by using .value = “Goodbye” (syntax: DOMelement/variable.attribute_name = value_of_attribute) command we are changing value attribute of <input> element specified with ID = “myText” to “Goodbye”.

Yogesh Chaudhari
SUID: 244195971

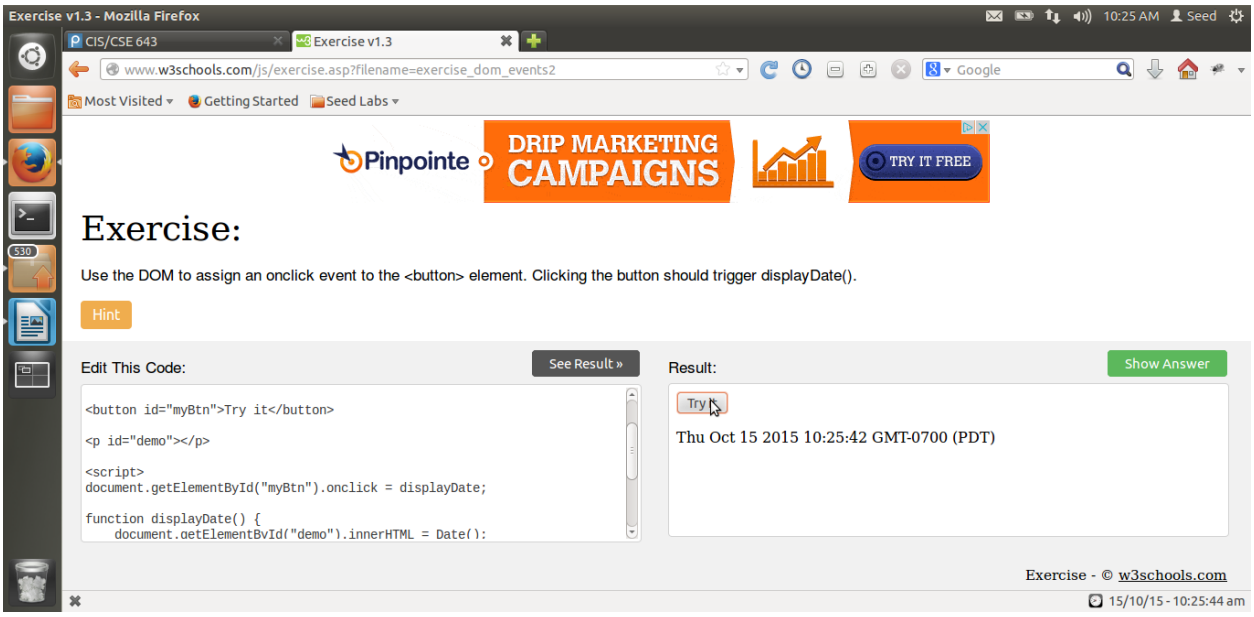
Onclick event:

Before clicking button:



Here by using JavaScript .onclick command on element <button> with ID “myBtn” we are setting an onclick event on <button> to call function displayDate() in javascript. DisplayDate() uses .innerHTML command to replace contents of <p> tag with current Date.

After clicking event:

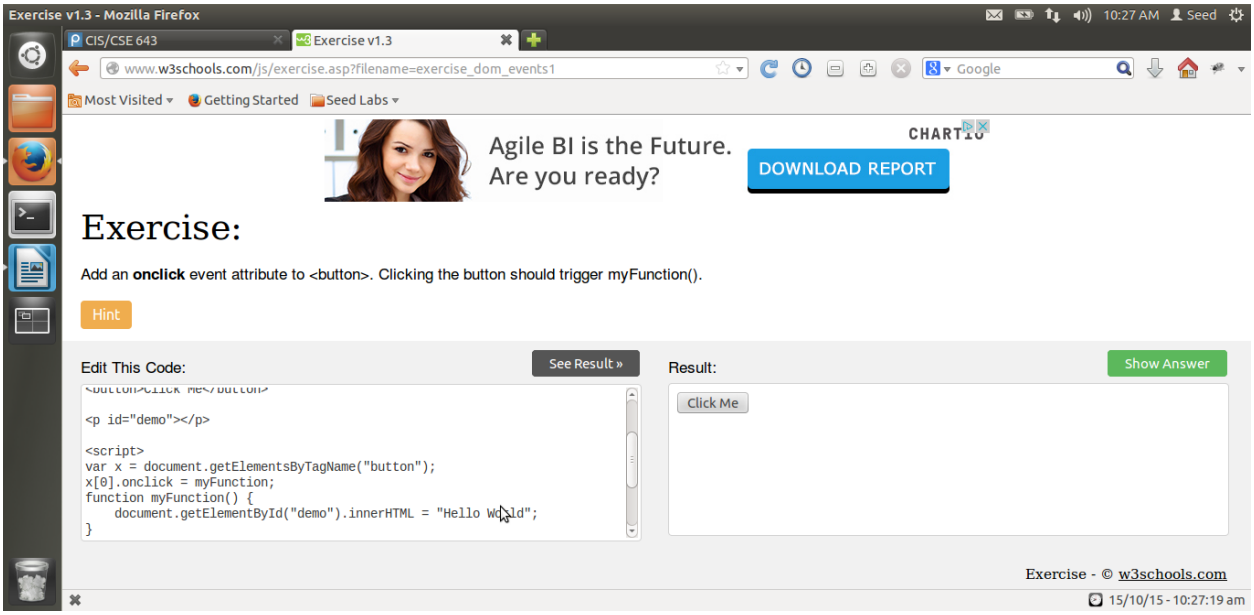


Yogesh Chaudhari
SUID: 244195971

Here, we can see that after clicking on try it button, an onclick event gets invoked and our code makes call to displayDate() function to display current-date in element with id “demo” of HTML.

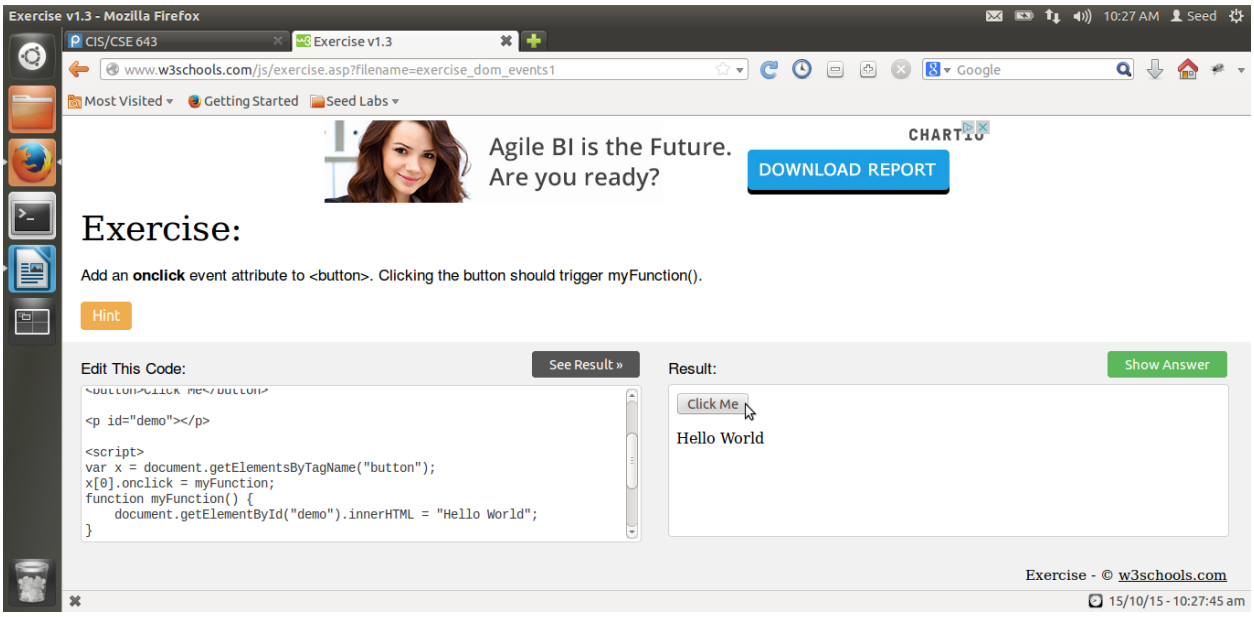
Event on tag button:

Before clicking button:



Here, we are using getElemntsByTagName(“button”) command to get list of all elements with tag <button> and storing it in variable x. Now, by using x[0] we select first button element of our html and apply onclick event on that using .onclick command.

After clicking button:



Yogesh Chaudhari
SUID: 244195971

Here, we can see that after clicking on the button onclick event got invoked and javascript made call to myFunction() which displays “Hello World” in element specified by ID “demo”.

Running cookie code:

Here, we are getting document cookies using document.cookie command and store them in variable x. Then using .innerHTML tag on element with ID “demo” we display cookies specified with this session.

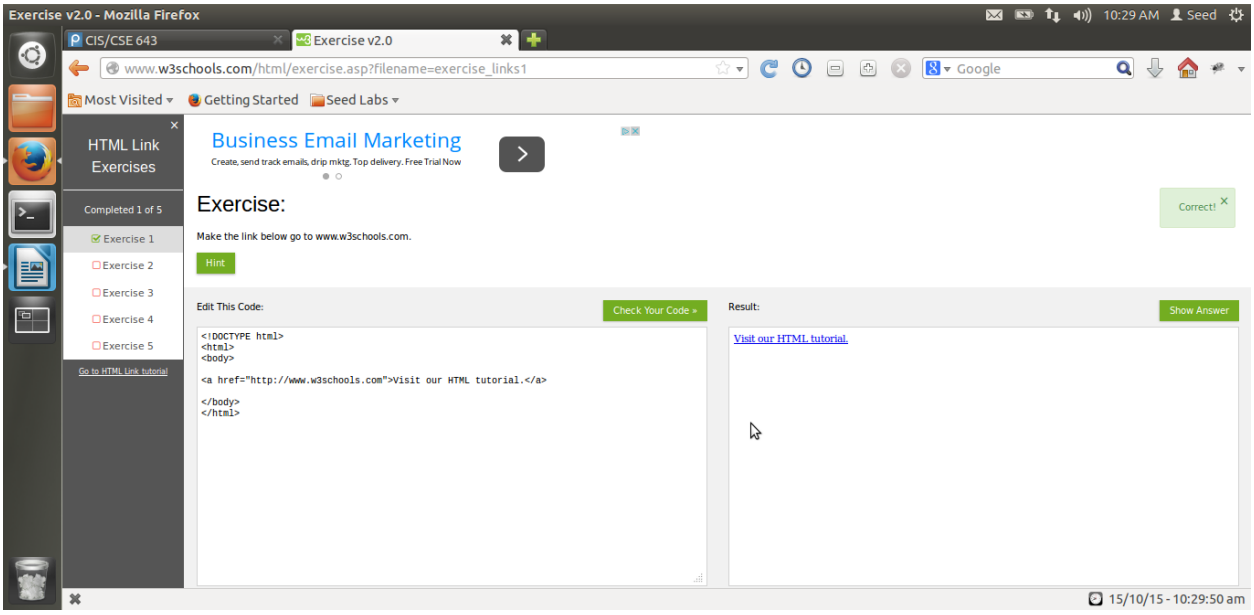
Here, we can see that in the cookies we have first cookie named ASPSESSIONID… = //something

This is a session ID value for this particular established session with server.

Yogesh Chaudhari
SUID: 244195971

2. HTML tags:

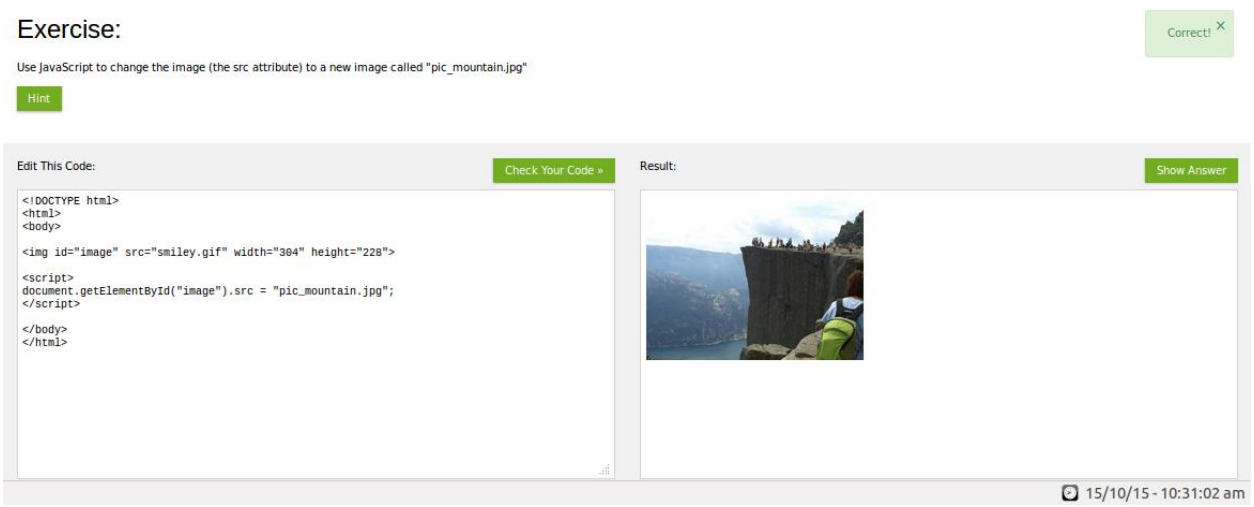
Links:



Here using href attribute in <a> tag element in HTML we can set URL which the link should follow to when clicked.

Image:

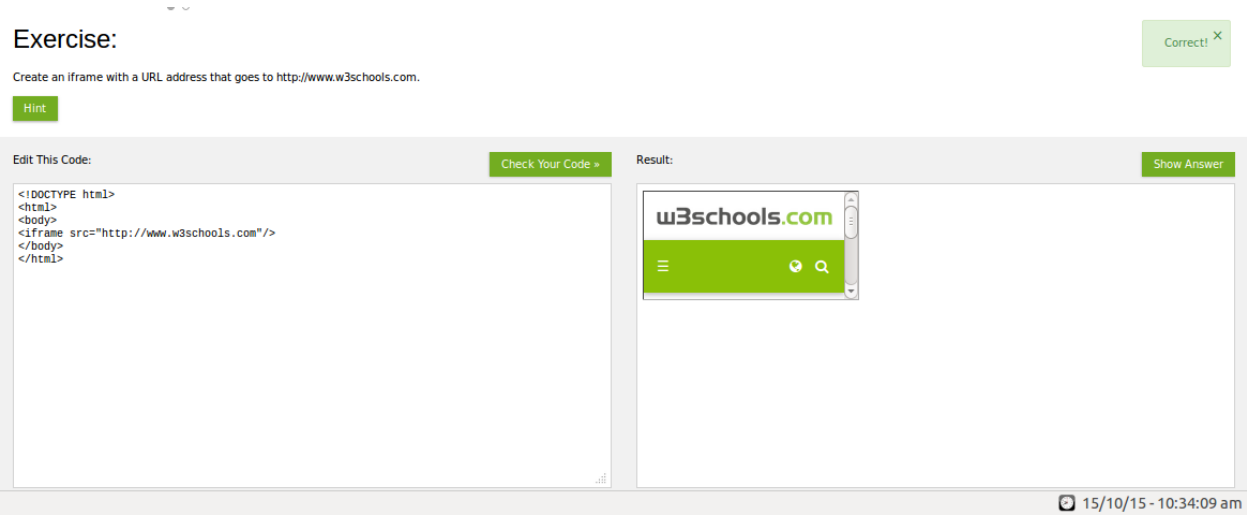
Changing image source:



Here using .src source-attribute setting method on img element with “image” ID in javascript we are changing src (source) value of <img> tag to “pic_mountains.jpg”.

Yogesh Chaudhari
SUID: 244195971

Iframe:

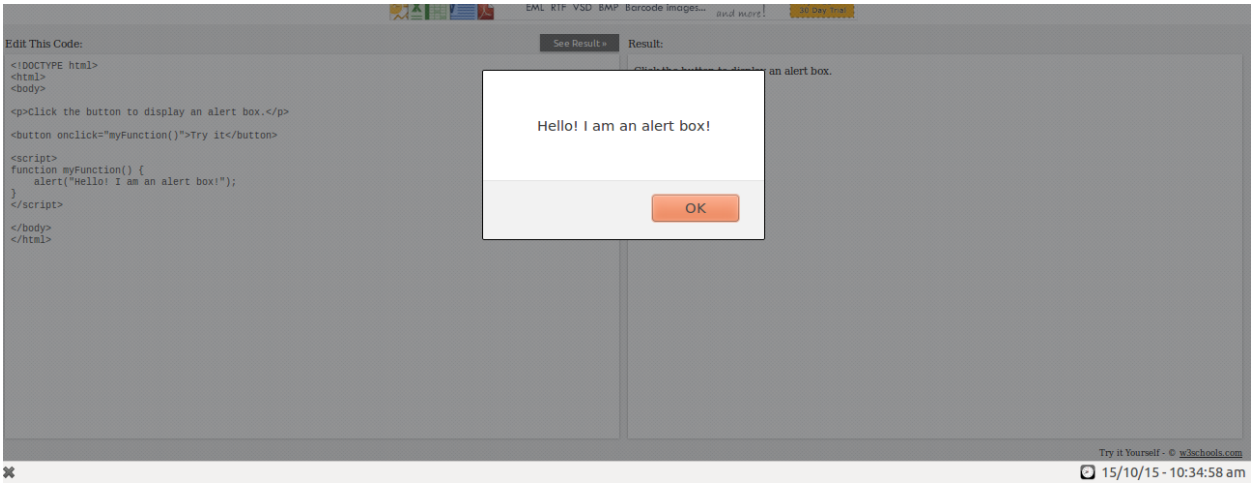


Here we are setting up the URL to be displayed in iframe element of HTML file using src attribute and URL as value for src.

Yogesh Chaudhari
SUID: 244195971

3. BOM:

Running alert code:

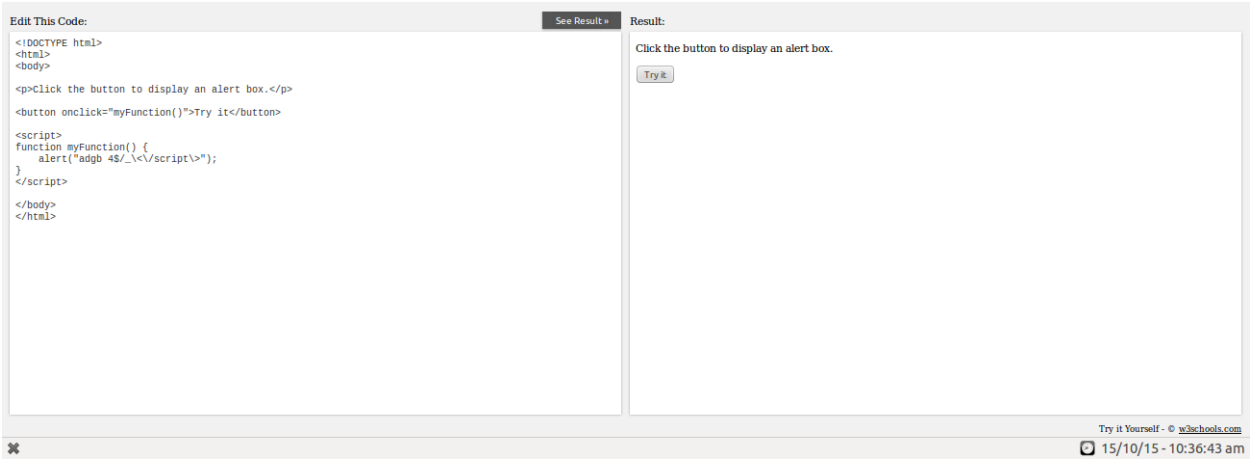


Here we are using alert() method of JavaScript in function myFunction() to show an alert dialog box on browser when button “Try it” is clicked. When try it button is clicked, onclick attribute of the button gets invoked which makes call to javascript function myFunction().

Yogesh Chaudhari
SUID: 244195971

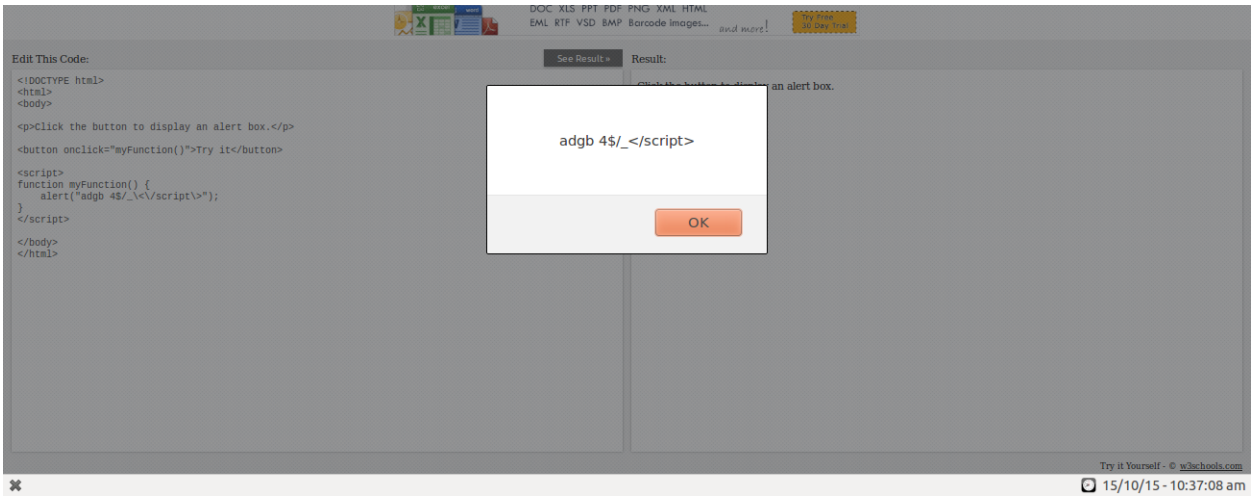
6. JS Strings:

Code for alert:



As browser interprets <, >, / characters and strings of any tags (such as <script>, </script>, <html>, <body>, etc.) as special characters, we cannot directly write them in the JavaScript when we want to use them in strings. For this purpose we precede them with backslash character "\", so that browser does not parse them as special characters and thinks of them as just normal string characters.

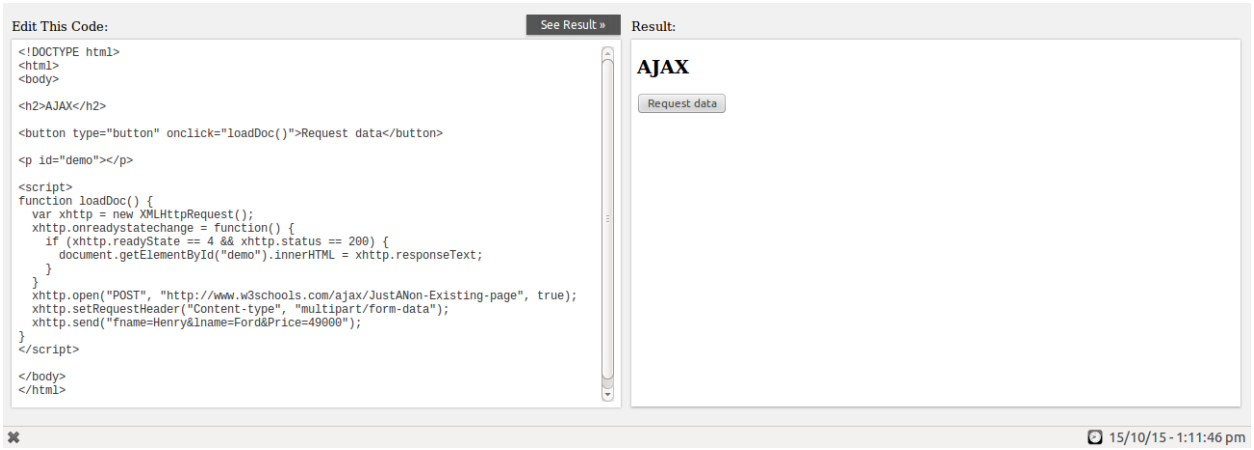
Onclick alert:



Here, we can see that after using "\" before all special characters such as "<,>,/", which are part of HTML syntax, browser parses them and turns them into string characters. By using alert() command we can display a dialog box displaying that string.

Yogesh Chaudhari
SUID: 244195971

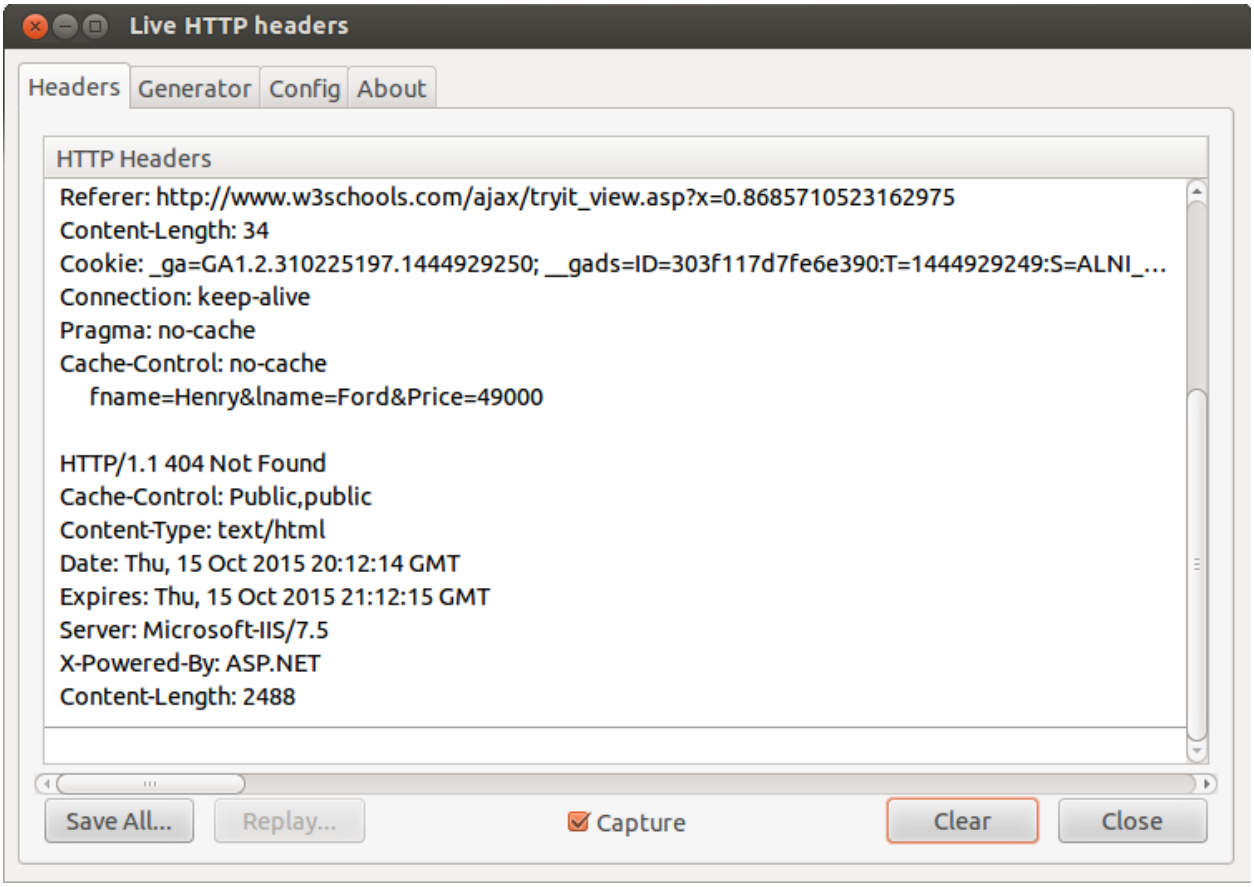
Ajax:



Here we are creating a variable xhttp which is an XMLHTTPRequest. By using open() method for this request we are providing 3 arguments which defines type of HTTPRequest i.e. GET/POST, URL where the file is located and true or false value for asynchronous flag. In this case we are sending an asynchronous POST request to URL http://www.w3schools.com/ajax/JustANon-Existing-page. We are setting up content type as multipart from-data in HTTP request header with command setRequestHeader() and passing content as fname=Henry, lname=Ford and Price=49000 to given URL with command send().

HTTP Live Header:

Yogesh Chaudhari
SUID: 244195971



From the content of Live HTTP Headers we can confirm that we have sent an asynchronous POST request to URL http://www.w3scholls.com/ajax/JustANon-Existing-page with content fname=Henry, lname=Ford and Price=49000. Our request sent back a HTTP 404 Error as the URL which we provided does not exist.

```
POST /ajax/JustANonExistingPage HTTP/1.1
Host: www.w3schools.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; charset=UTF-8
```

And from above screenshot we can confirm that the content-type for our POST request was “multipart/form-data”.