**3.2 Events:**

**One of the primary ways in which JavaScript is executed on a Web pages is through events**

**is a specific circumstance such as an action performed by a user, or an action performed by the browser that is monitored by JavaScript and that the script can respond to in some way.**

**JavaScript enabled Web pages are typically event driven Events are actions that occur on the webpage.**

**Events are signals generated when specific action occurs.**

**Event Handler:**

**When a function is assigned to an event handler that function is run when that function is run when that events occurs.**

**A handler that is assigned from a script used the syntax ' [element].[event] =[function];'**

**where [element] is a page element,[event] is the name of the selected event and [function] is the name of the function that occurs when the event takes place.**

**Event handling in html.**

<html>

<head>

<title> Event handling.html </title>

</head>

<body onload="window.alert( 'Body loaded');" onunload="window.alert\*( 'unloading...');">

<form action="http://www.google.org" onsubmit="window.alert('submitting..');" onreset="window.alert('Resetting...');">

<p>

<input type="text" name="Some Text" onkeypress="window.alert(' Text field got character.');" onselect="window.alert('Text selected.');" />

<br />

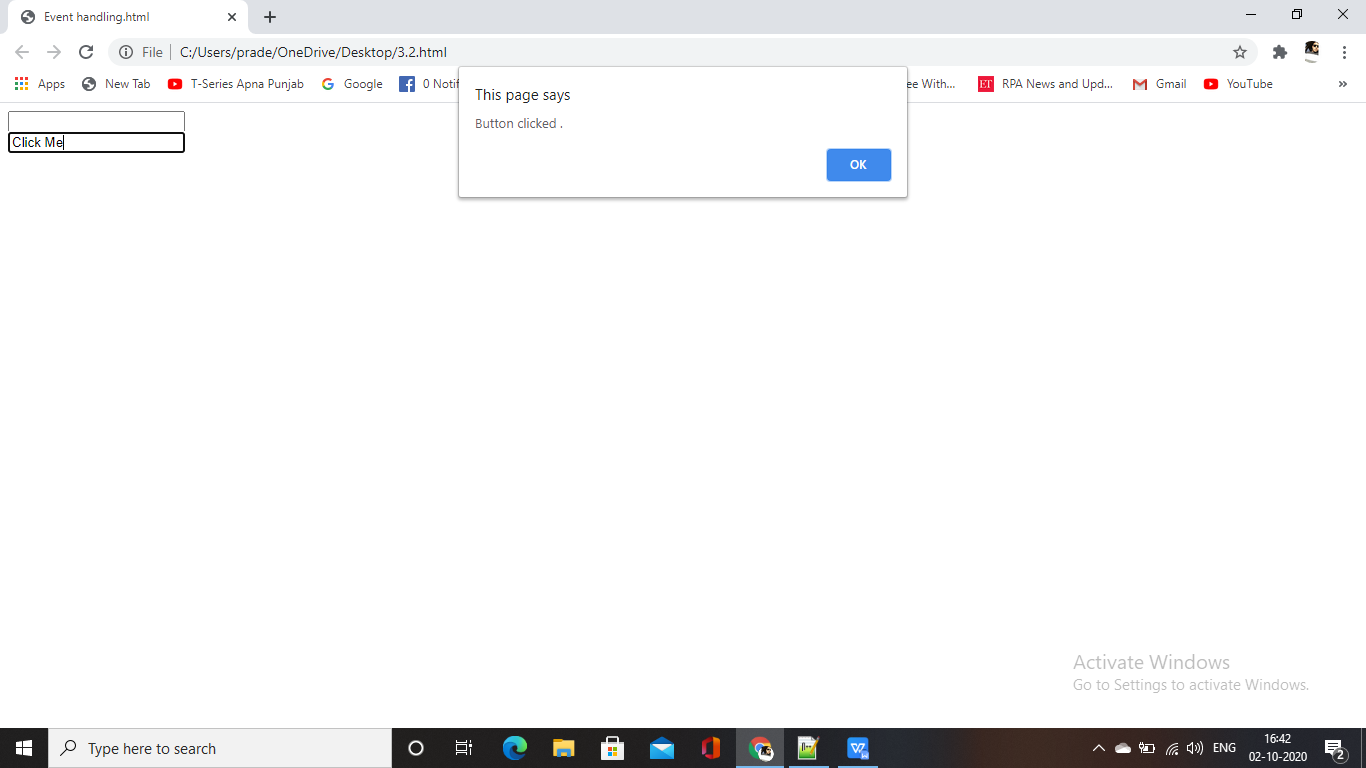
<input type=" button " name="aButton" value="Click Me" onclick="window.alert('Button clicked .');" />

</p>

</form>

</body>

</html>



**Mouse Event**

**JavaScript interaction with HTML is handled through events that occur when the user or browser manipulates a page.**

**When the page loads, that is an event. When the user clicks a button,that click,too,is an event.**

**Another examples of events are like click mouse button,pressing any key closing window,resizing window etc.**

<html>

<head>

<title> Events </title>

<script>

function react()

{

alert("please enter any value");

}

</script>

<body>

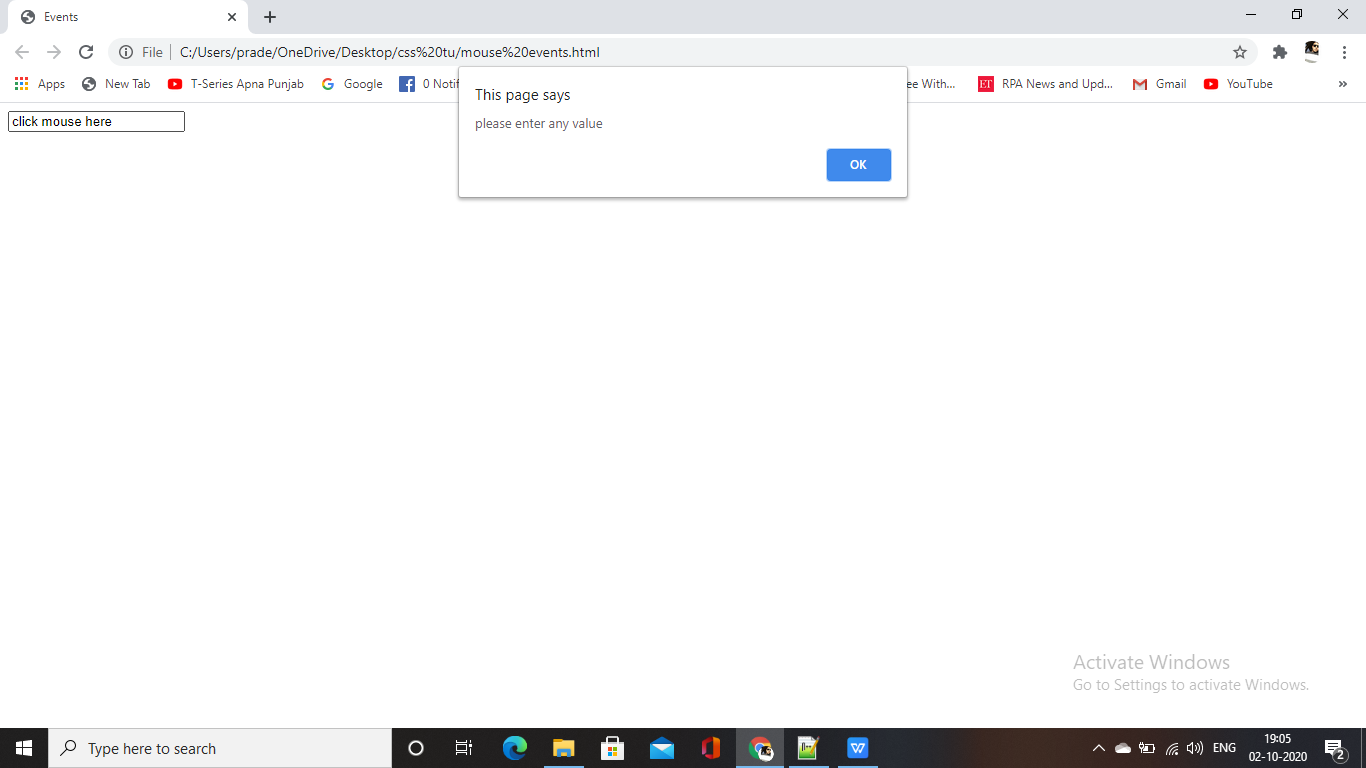
<form name="myform" action ="" method="post" onmousedown="react()">

<input value="click mouse here" >

</form>

</body>

</html>



**Using some of the mouse related events.**

<html>

<body>

<div id="panel"></div>

<script>

function clickResponse()

{

this.innerHTML+= "click detected<hr>";

}

function doubleclickResponse()

{

this.innerHTML+= "<br>Double-click detected<br>";

}

function mousedownResponse()

{

this.innerHTML+= "<br>Mouse button down<br>";

}

function mouseupResponse()

{

this.innerHTML+= "<br>Mouse button up <br>";

}

function init()

{

var panel = document.getElementById( "panel") ;

panel.innerHTML = "Click Here &gt;<br>";

panel.onclick = clickResponse ;

panel.onblclick = doubleclickResponse ;

panel.onmouseup = mouseupResponse ;

panel.onmousedown = mousedownResponse ;

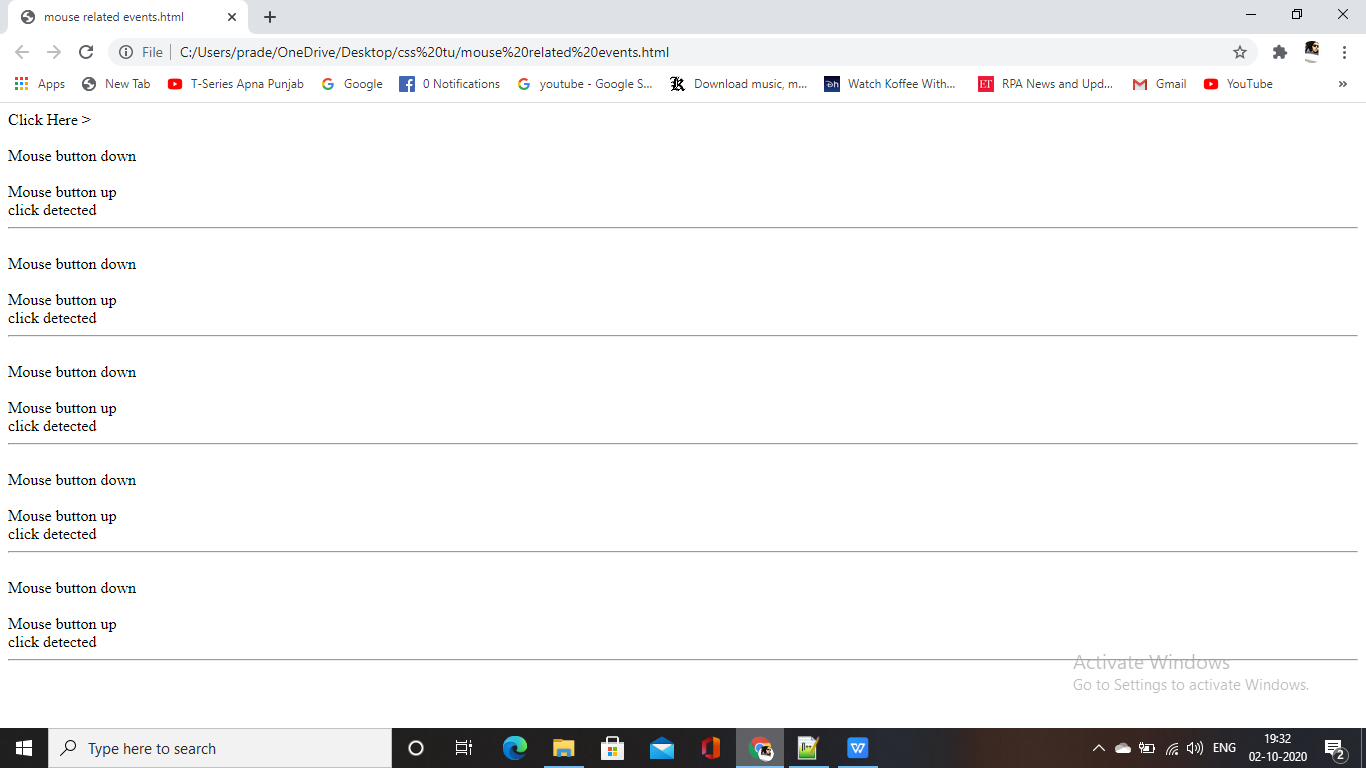
}

document.addEventListener("DOMContentLoaded", init, false);

</script>

</body>

</html>



**Mouseover and mouseout Events:**

* **The mouseover and mouseout events are triggered whenever the mouse moves on to or off of an object.The mousemove event the mouse is moved.**
* **The mouseover and mouseout will get fired each time you move the mouse over and around a child element.**
* **This means that you could be seeing many unnecessary event fires even though it seems like you are moving your mouse within a single region.**

<html>

<body>

<div id = "panel"></div>

<script>

var panel,flag;

panel = document.getElementById ("panel") ;

flag = true;

panel.innerHTML =" Move the mouse ...";

document.onmousemove = mousemove;

panel.onmouseover = mouseover;

panel.onmouseout = mouseout;

var panel, flag;

function mousemove (e)

{

If(flag)

{

panel.innerHTML= " Mouse is at X: " + e.x + ", y: " + e.y;

}

}

function mouseover ()

{

flag=false;

panel.innerHTML = "Mouse is Over";

}

function mouseout ()

{

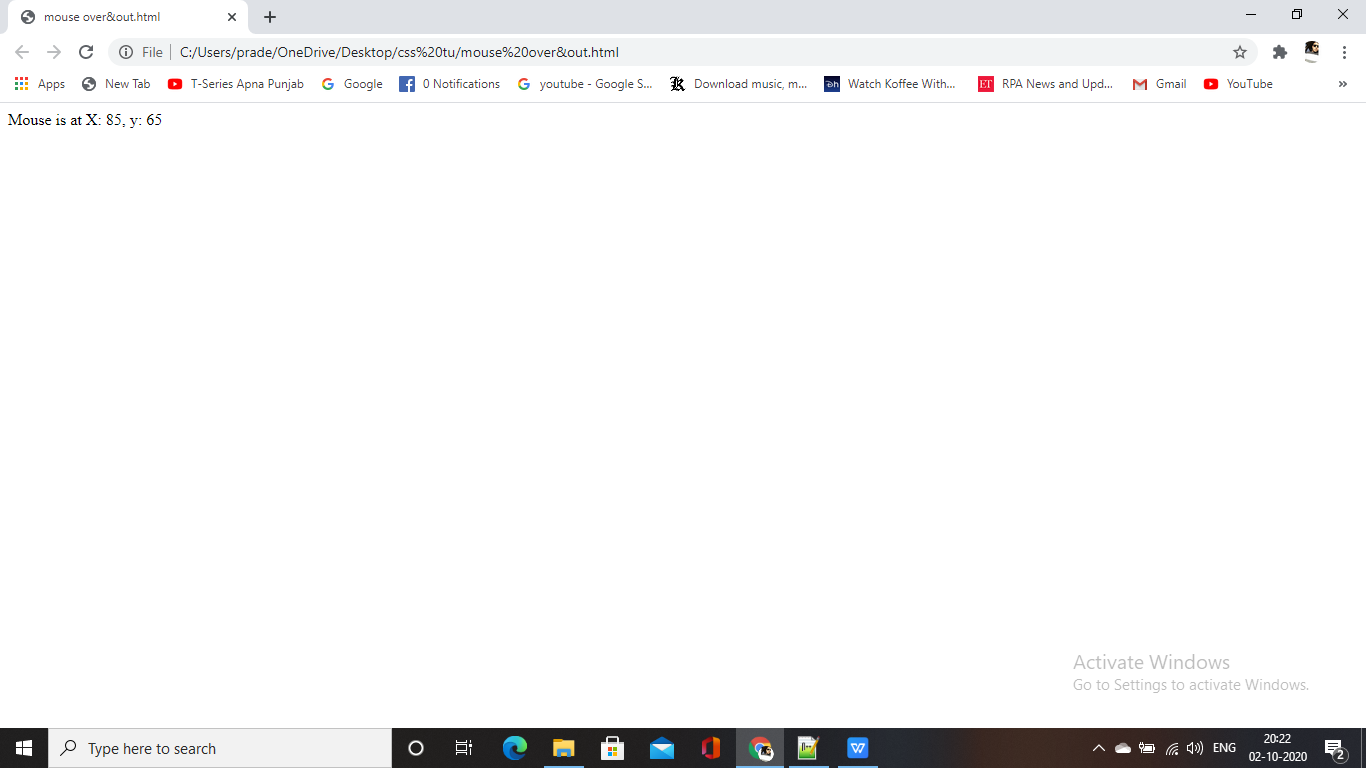
flag = true;

}

</script>

</body>

</html>



**Key Events:**

**Here, we will see how a document object react to key-related events. Key related events can be categorized as following ways:**

**1. Keydown: A key is pressed down.**

**2. Keypress: A character key is pressed.**

**3. Keyup: A key is released.**

**There is a fundamental difference between keypress and keydown.**

**1.Keydown triggers on any key press and gives scan-code.**

1. **Keypress triggers after keydown and gives char-code, but it is guaranteed for character keys only.**

**Use of keystrokes.**

<html xmlns= "http://www.W3.org/1999/xhtm1">

<head>

<title>Key Event</title>

</head>

<body>

<div id="panel"></div>

<script>

var panel;

panel = document.getElementById ("panel");

panel.innerHTML = "Press a key..";

document.addEventListener ("keydown", keydown, false);

document.addEventListener("keyup", keyup, false) ;

document.addEventListener ( "keypress", keypress, false);

function keydown()

{

panel.innerHTML += "<br/>Key Pressed: ";

}

function keyup()

{

panel.innerHTML += "<br/>Key Released";

}

function keypress (e)

{

var keynum = (Window.event) ? event.keyCode: e.which;

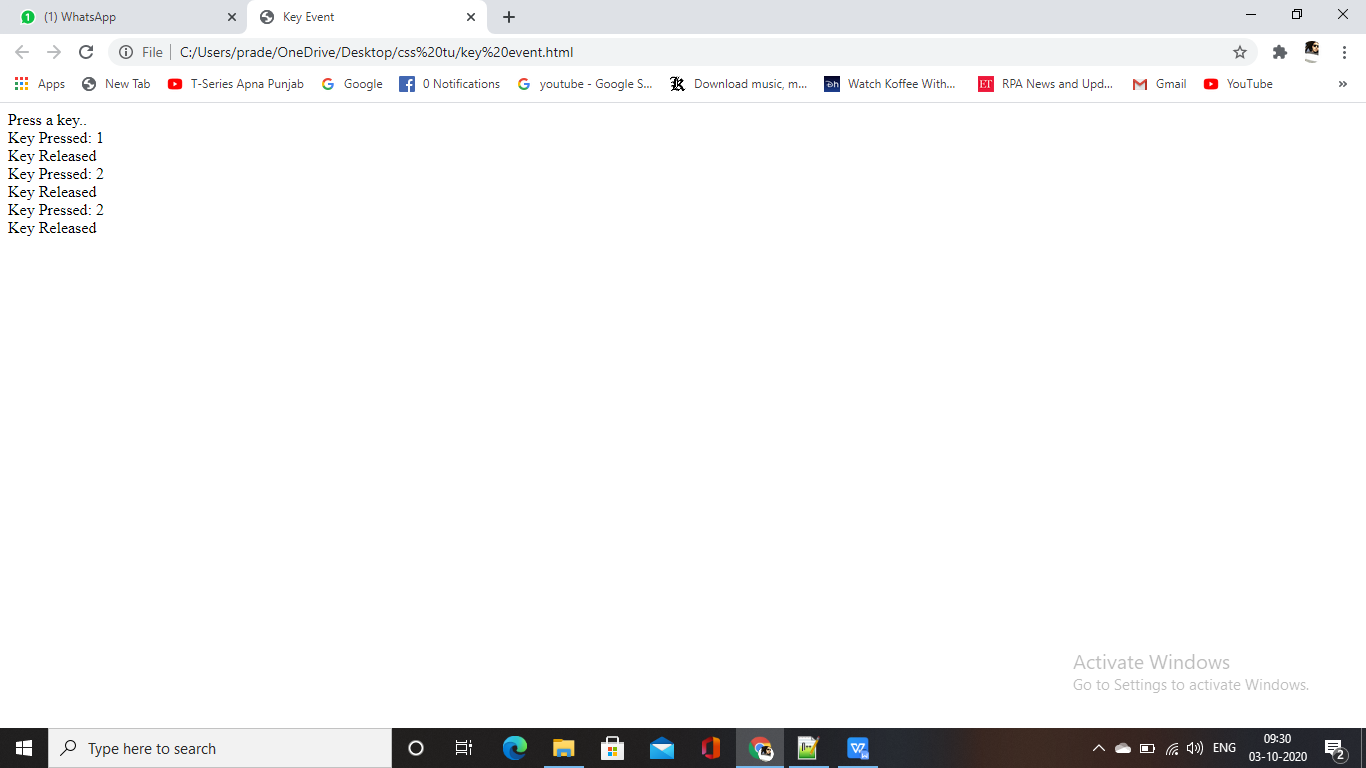
panel.innerHTML += String.fromCharCode(keynum) ;

}

</script>

</body

</html>



**Button Event handling**

<html>

<head>

<title>Button Event </title>

<Script>

function compute()

{

var a = document.getElementById("txt1").value

var b = document.getElementById("txt2").value

var c = parseInt(a) + parseInt(b)

document.getElementById("txt3").value = C

}

</script>

</head>

<body>

Enter value of A <input type="text" id="txt1" /><br/>

Enter value of B<input type="text" id="txt2" /><br/>

Sum= <input type="text " id="txt3" /><p>

<button id="myBtn" type= "button">Addition</button>

<script type="text/javascript">

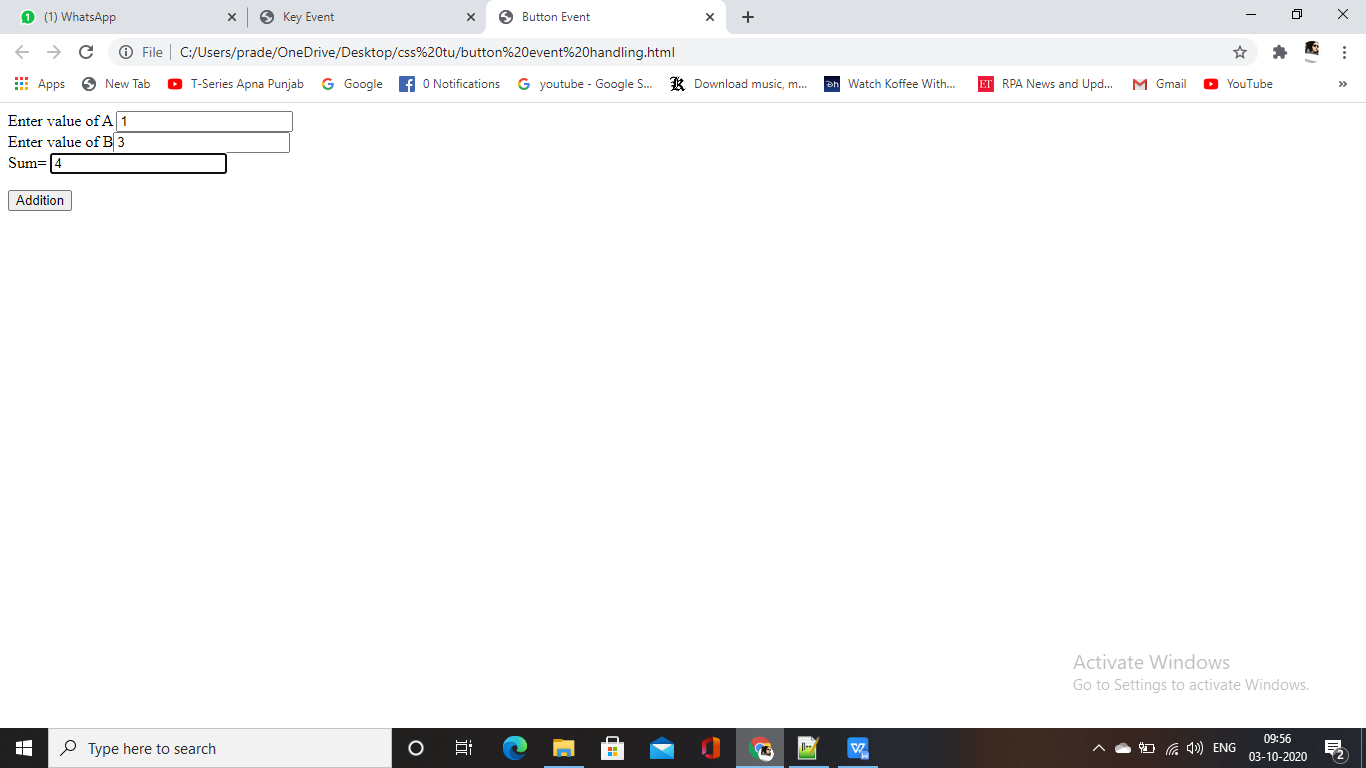
var button = window.document.getElementById("myBtn");

button.addEventListener("click", compute, false);

</script>

</body>

</html>



**Button event handling in JavaScript**

<html>

<head>

<title> EventHello.html</title>

<Script type="text/javascript" src="EventHello.js">

</script>

<meta http-equiv="Content-Script-Type" content="text/javascript" />

<script>

function sayHello(event)

{

window.alert("Hello World!\n\n" +

"Event type:" + event.type + "\n" +

"Event target element type: " + event.target.nodeName);

return;

}

</script>

</head>

<body>

<p>

<button id="myBtn" type="button">Click for a message! </button>

<script type= "text/javascript">

var button = window.document.getElementById("myBtn" ) ;

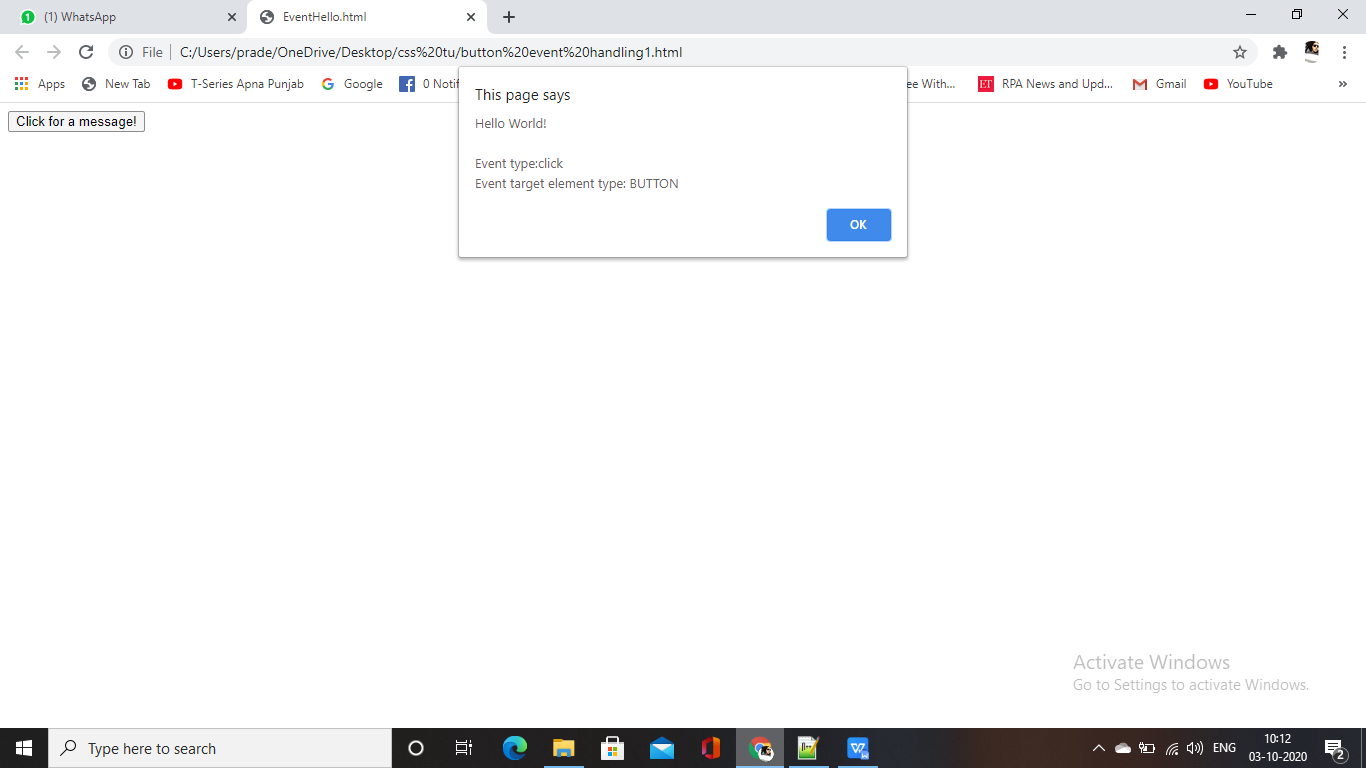
button.addEventListener("click" , sayHello, false)

</script>

</p>

</body>

</html>



**3.3 CHANGING ATTRIBUTE VALUES DYNAMICALLY**

**Earier we have seen we can change the attribute of an elements at scripting time. Java script provide**

**us facility to change an attribute of an element by assigning a new value to the attribute at runtime**

**We can Write the code of change in attribute value in a user defined function and can call that**

**function when an appropriate event occurs.**

**The change in any attribute value can be reflected to the user by highlighting the value or text by**

**some color Here we will show the updated value in blue and change the background color to pink.**

**The onchange event is associated with many elements (<input>, <select>) of a form object and**

**helpful to make call to a function where the change of attribute value code is written.**

<html>

<head>

<title> HTML Form </title>

<script type="text/javascript">

function Highlight(Element)

{

Element.style.color = 'blue'

Element.style.backgroundColor = 'pink'

}

</script>

</head>

<body>

<form name=" myform " action= "" method="post">

<p>

first Name : <input value= "om" type="text"

name="Fname " onchange="high1ight(this)"/><br>

Last Name: <input value="patel " type="text"

name="Lname" onchange= "highlight (this)"/><br>

Email: <input value= "ompatel@gmail.com" type="text"

name="email" onchange= "highlight(this) "/><br>

<input name="submit" value="submit" type="submit"/>

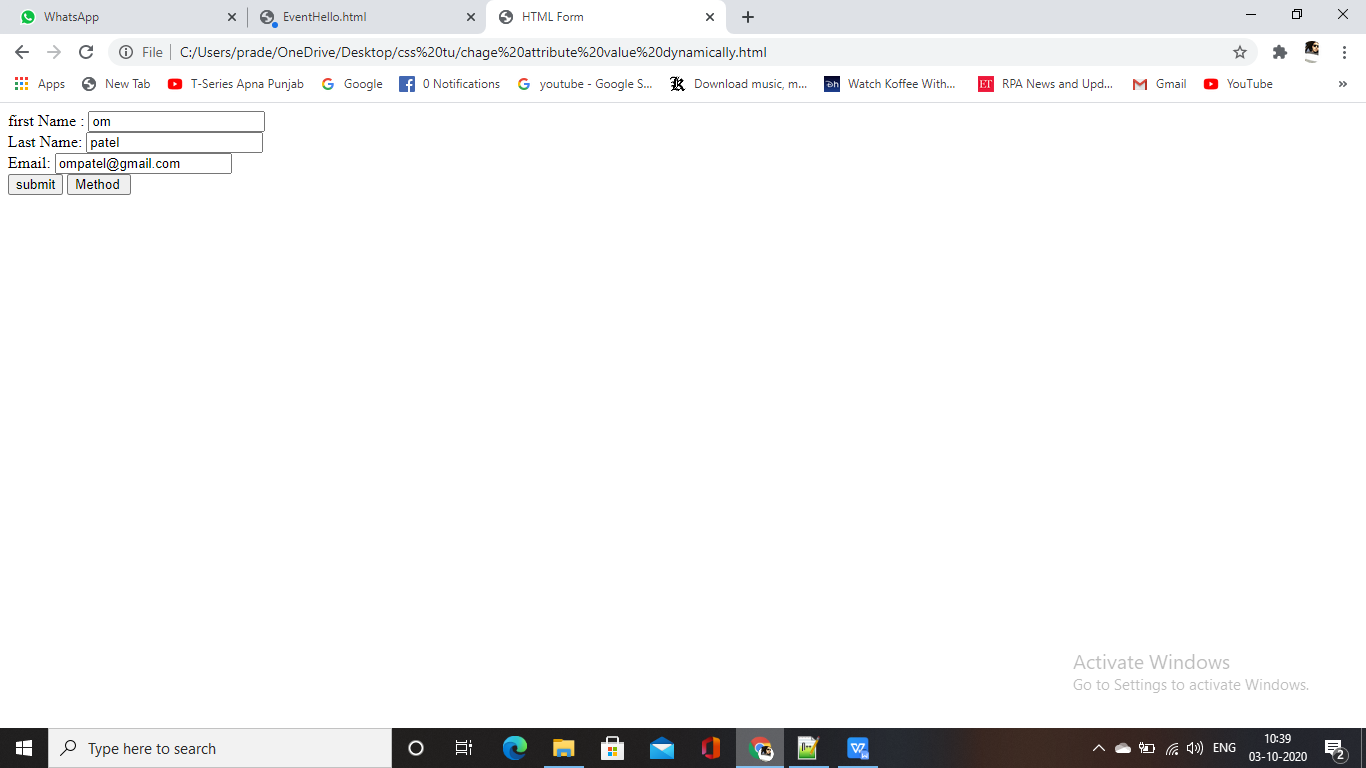
<input name= "reset" value= "Method " type="reset"/>

</p>

</form>

</body>

</html>



<html>

<head>

<title> HTML Form </title>

<script type="text/javascript ">

function OnSelectionChange(element)

{

var selected= element.options [element.selectedIndex].value;

alert("you have selected :"+selected);

}

</script

</head>

<body>

Select an item from the following list:<br />

<select onchange= "OnSelectionChange (this) ">

<option value="Apple" >Apple

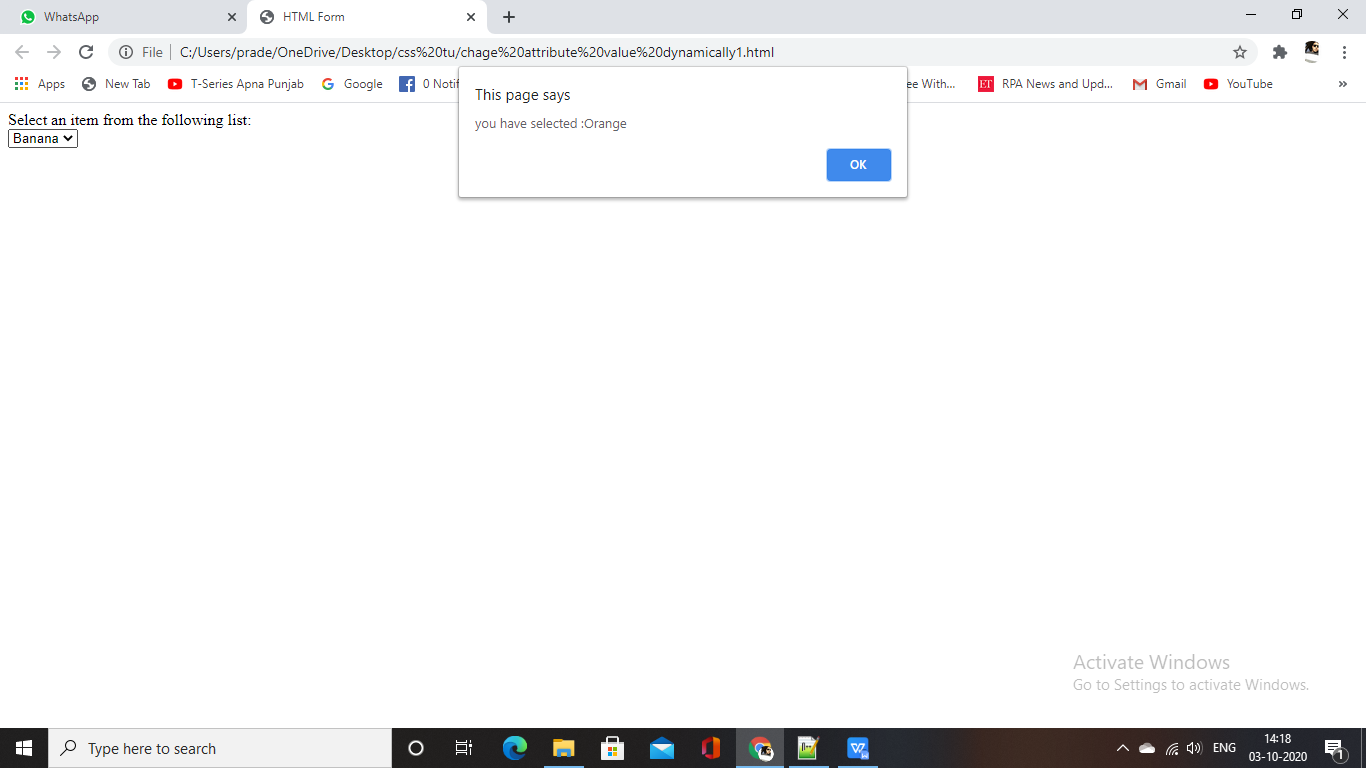
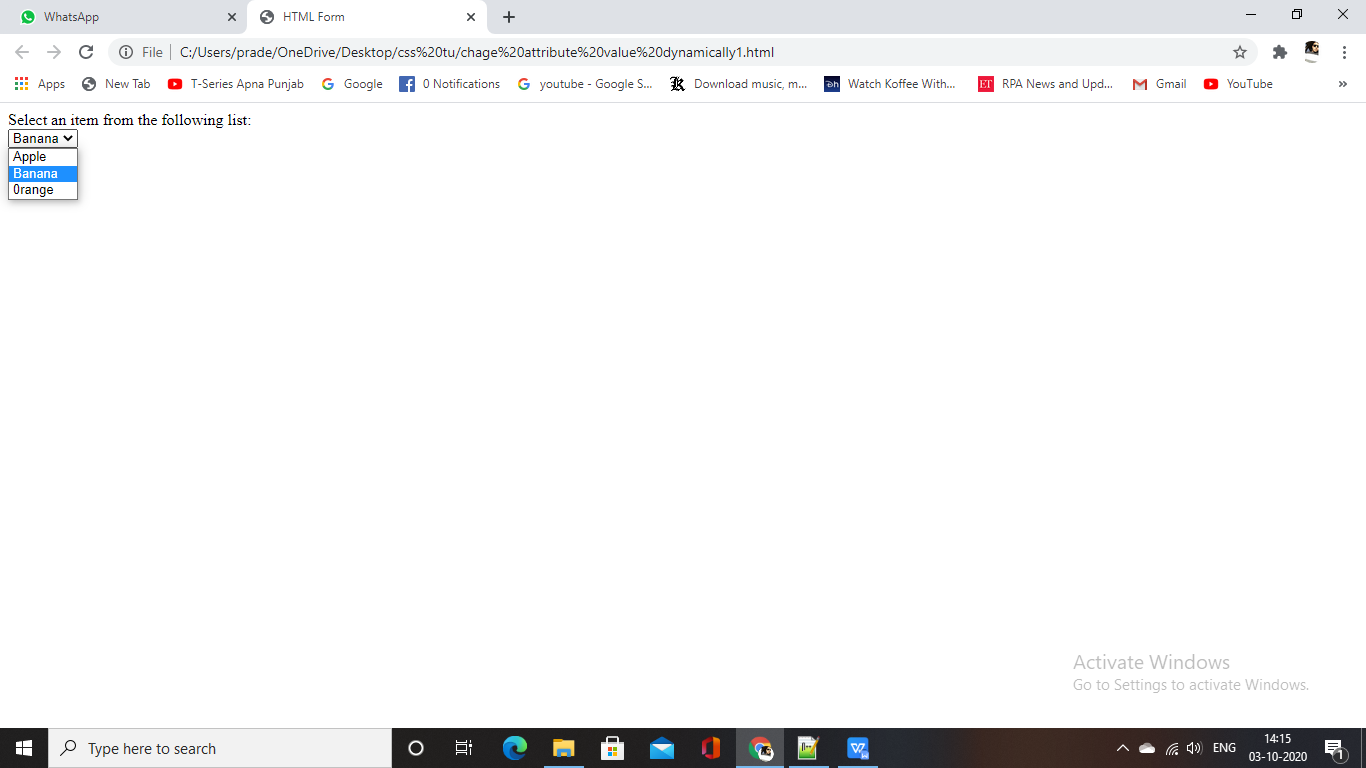
<option value="Banana" >Banana

<option value="Orange" >0range

</select>

</body>

</html>



**3.4 CHANGING OPTONLIST DYNAMICALLY**

**.As we have changed the attribute values dynamically in last section same way we can change options in the option list at runtime by writing the code in a function.**

**The purpose of an option list is to present a user two or more items for choice. Elements in option list are usualy set when the option list is created. However, you can change the content of an option list at runtime by using a JavaScript function.**

<html>

<head>

<title> HTML Form </title>

<script language="Javascript" type="text/javascript">

function updateList(ElementValue)

{

with (document.forms. myform)

{

if(ElementValue == 1)

{

OptionList [0].text = "Mango" ;

OptionList [0].value 1;

OptionList [1].text = "Banana";

OptionList [1].value = 2;

}

if (ElementValue== 2)

{

OptionList [0].text = "Potato";

OptionList [0].value = 1;

OptionList [1].text = "Cabbage";

OptionList [1].value 2

}

}

}

</script>

</head>

<body>

<form name="myform" action="" method="post ">

<p>

<select name="Option List " size="2">

<option Value=1>Mango

<option Value=2>Banana

</select>

<br>

<input type="radio" name="group1" value=1 checked="true"

onclick ="updateList(this.value)">Fruits

<input type="radio" name="group1" value=2

onclick ="updatelist(this.value)">Vegetables

<br>

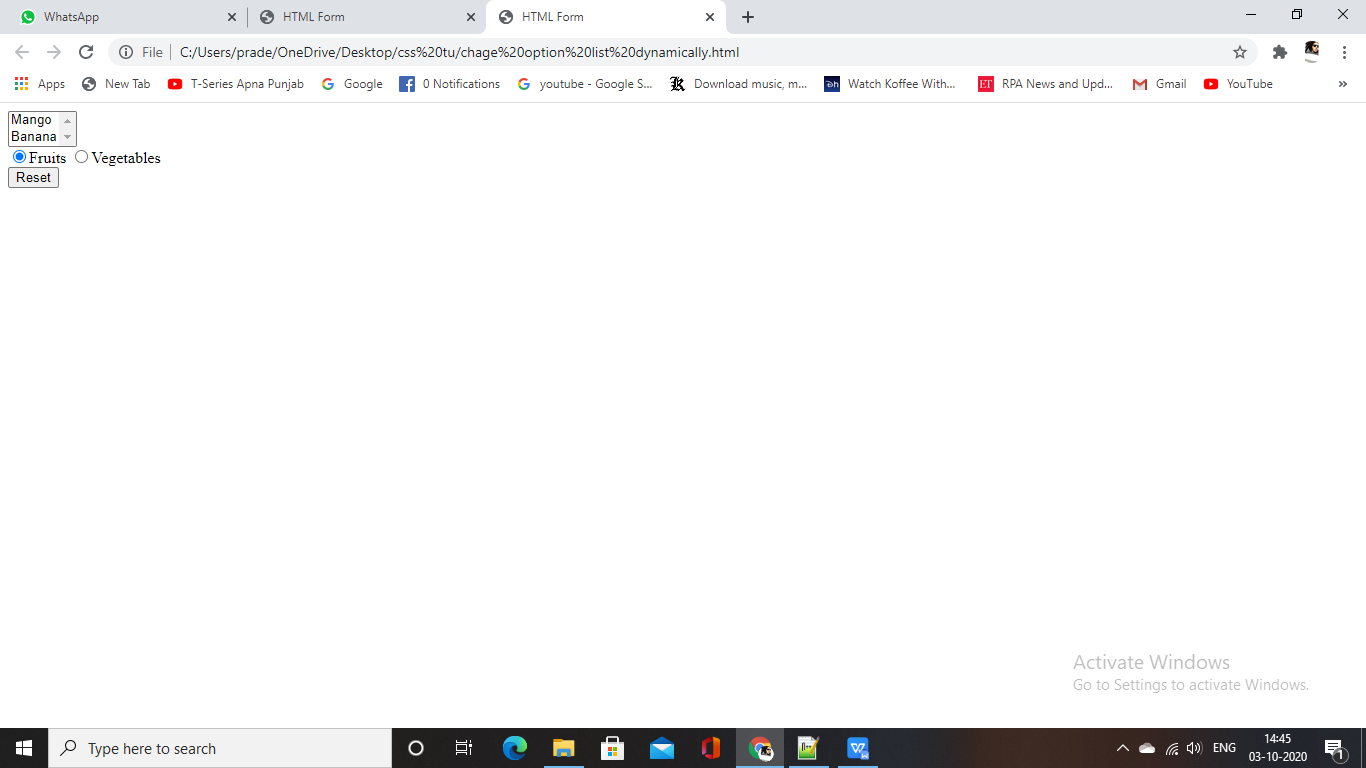
<input name="Reset" value="Reset" type="reset">

</p>

</form>

</body>

</html>



**3.5 EVALUATING CHECK BOX SELECTIONS**

**Checkboxes are an integral part of forms that allow a user to make a selection with a Simple Click of**

**the mouse. You can prompt users for specific input by using a list of checkboxes.**

**None, one or more checkboxes may be checked by the user according to the information they wish to send with the form.**

**A checkbox is created by using the input element with the type="checkbox" attribute-value pair**

**A check box in a form has only two states (checked or un-checked) and is independent of the state or**

**other check boxes in the form. Check boxes can be grouped together under a common name.**

**You can write a Java Script function that evaluates whether or not a check box was selected and then**

**processes the result according to the needs of your application.**

<html>

<head>

<title> HTML Form </title>

<Script language= "Javascript " type="text/javascript">

function selection()

{

var x = "You selected: "

with (document.forms.myform)

{

if (a.checked == true)

X+= a.value+" ";

if (b.checked == true)

X+=b.value+" ";

if(o.checked == true)

X+=o.value+" ";

if (p.checked == true)

X+=p.value+" ";

if (g.checked == true)

X+=g.value+" ";

}

document.write(x) ;

}

</script>

</head>

<body>

<form name="myform" action="" method="post ">

select your favourite Fruits :<br>

<input type="checkbox" name="a" value="apple">Apple

<input type="checkbox" name= "b" value="banana"> Banana

<input type="checkbox" name="o" value="orange">Orange

<input type="checkbox" name= "p" value="pear"> Pear

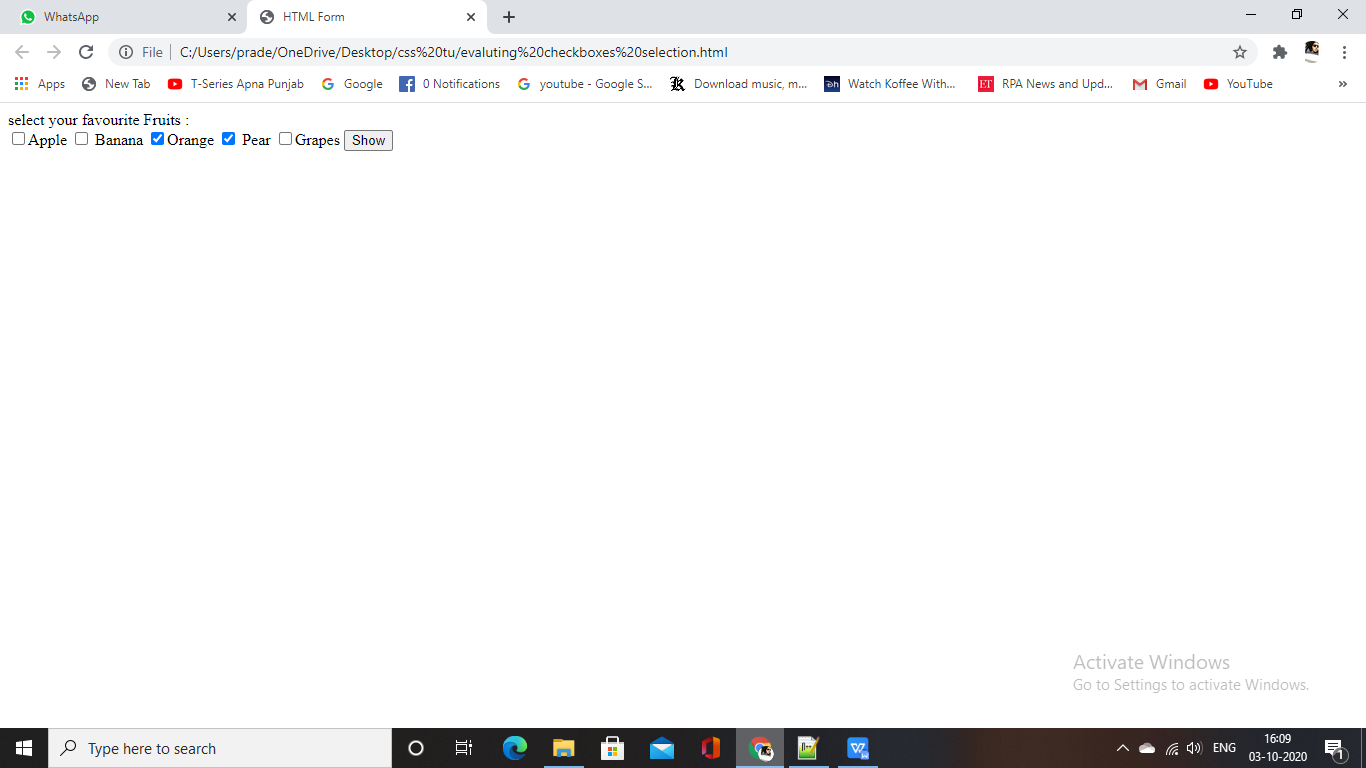
<input type="checkbox" name="g" value="grapes ">Grapes

<input type=reset value="Show" onclick ="selection()" >

</form>

</body>

<html>



**3.6 CHANGING LABELS DYNAMICALLY**

**Just likewe have changed an option list dynamically same way we can change a label of a button.**

**Following example make use of option list (consist of fruits name) and a button. Initially when the form will be loaded then option list shows the list of fruits and label of button is fruits.**

**Once the user click on the button its label will get changed to the "vegetables" and list will display all the fruits.**

**Here, button is acting as a toggle button and its label keep changing between fruits and vegetable on**

**each click.**

<html>

<head>

<title> HTML Form </title>

<Script language="Javascript" type="text/javascript">

function updateList(ElementValue)

{

with (document.forms.myform)

{

if (ElementValue == "Fruits")

{

toggleButton.value = "Vegetables";

OptionList [0].text = "Mango";

OptionList [0].value = 1;

OptionList [1].text = "Banana";

OptionList [1].value = 2;

}

if (ElementValue== "Vegetables")

{

toggleButton.value = "Fruits";

optionList [0].text ="Potato";

OptionList [0].value = 1;

OptionList [1].text = "Cabbage";

OptionList [1].value = 2;

}

}

}

</script>

</head>

<body>

<form name= "myform" action="" method="post">

<p>

<select name="OptionList" size="2">

<option value=1>Mango

<option value=2>Banana

</select>

<br>

<input name="togglebutton" value="fruits" type="reset"

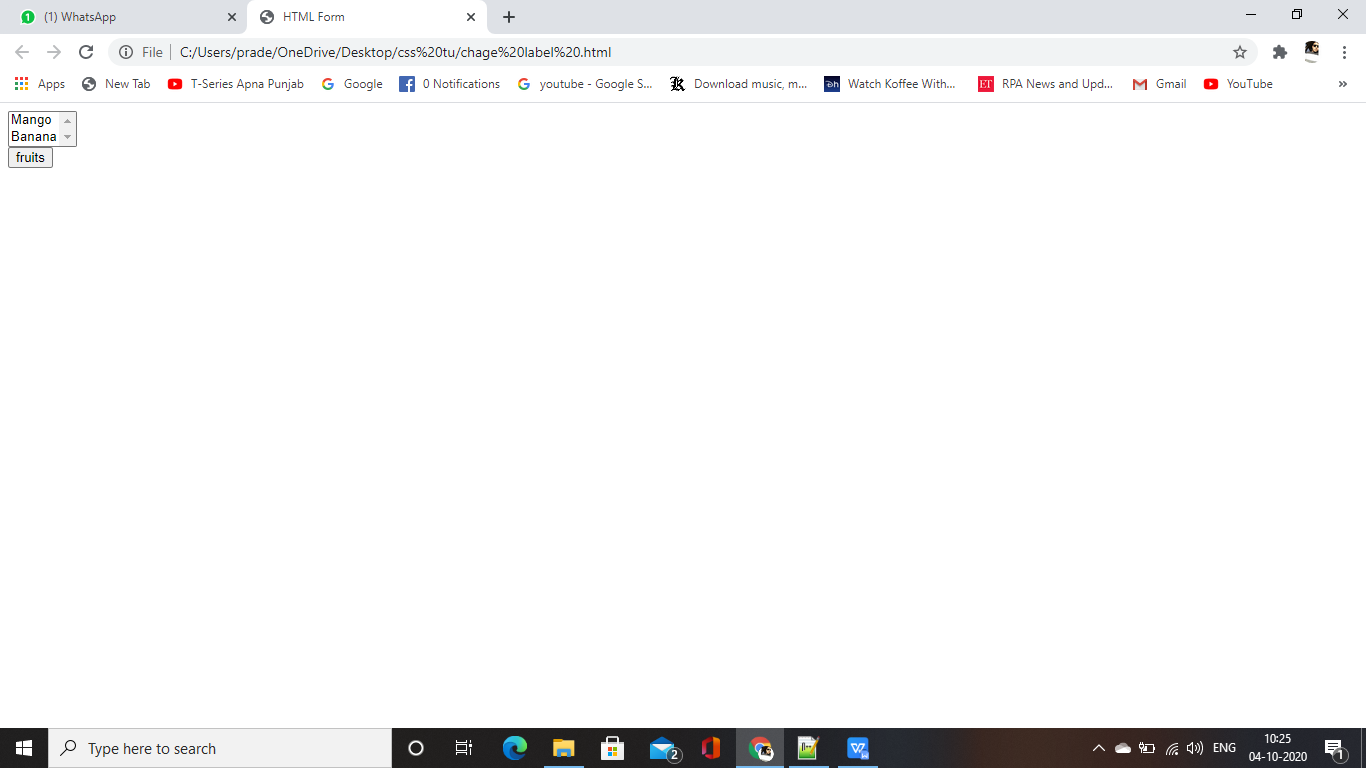
onclick="updatelist(this.value)" >

</p>

</form>

</body>

</html>



**3.7 MANTPULA TIN G FORM ELEMENTS**

**In last section, we have seen various example of manipulation of form elements before submitting any form to CGI application.**

**Javascript provide us facility to manipulate elements on a form after the user clicks the Submit button and before the form is actually submitted to the CGI application.**

**Javascript make it possible with help of hidden element which is similar to any html element except it does not appear on screen.**

**A hidden element has a name and value that is sent to the CGI program along with other elements of the form for processing.**

<html>

<head>

<title> HTML Form </title>

<script language="Javascript" type-"text/javascript">

function addEmail()

{

with (document.forms.myform)

{

if (Fname.value.length> 0 && Lname.value.length > 0)

{

Email.Value = Fname.value.charAt(0)+ Lname.value + '@xyz.com';

}

</script>

</head>

<body>

<form name= "myform" action="" method="post">

<p>

First Name: <input type="text" name="Fname"/> <br>

Last Name: <input type="text" name="Lname"/><br>

Email: <input type="hidden" name="Email"/><br>

<input name="Submit" value="Submit" type="button"

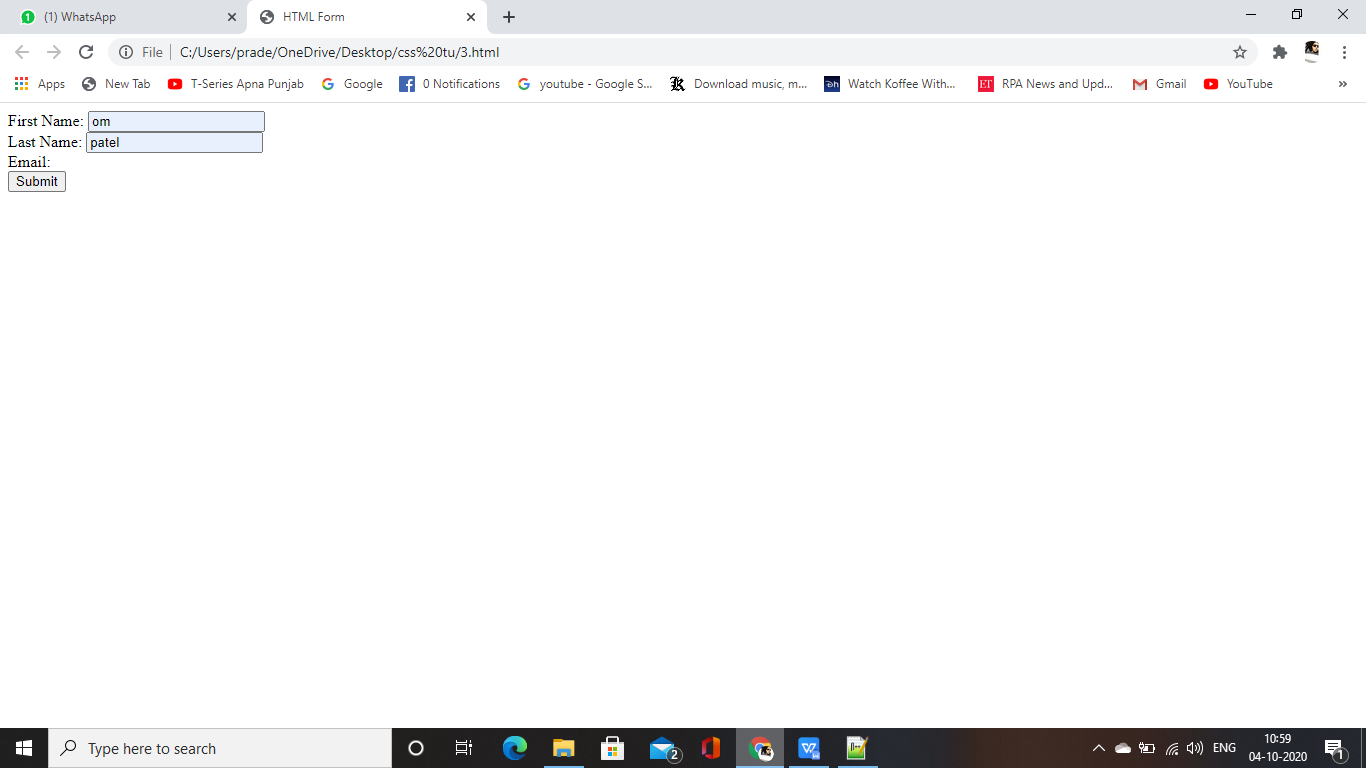
onsubmit="addEmail()"/>

</p>

</form>

</body>

</html>



**3.8 INTRINSIC JAVASCRIPT FUNCTIONS**

**.JavaScript has a special set of functions called intrinsic functions that mimic actions of the Submit button and Reset button of a form.**

**.An intrinsic function is often used to replace the Submit button and the Reşet button with your own graphical images, which are displayed on a form in place of these buttons.**

<html>

<head>

<title> HTML Form </title>

</head>

<body>

<form name="myform" action=" " method="post">

<p>

first Name: <input type= "text" name="Fname"/> <br>

Last Name: <input type="text" name="Lname"/><br

<img Src="11.jpg" onclick="javascript:document.forms.

myform.submit()" />

<img src="11.jpg" onclick="javascript:document.

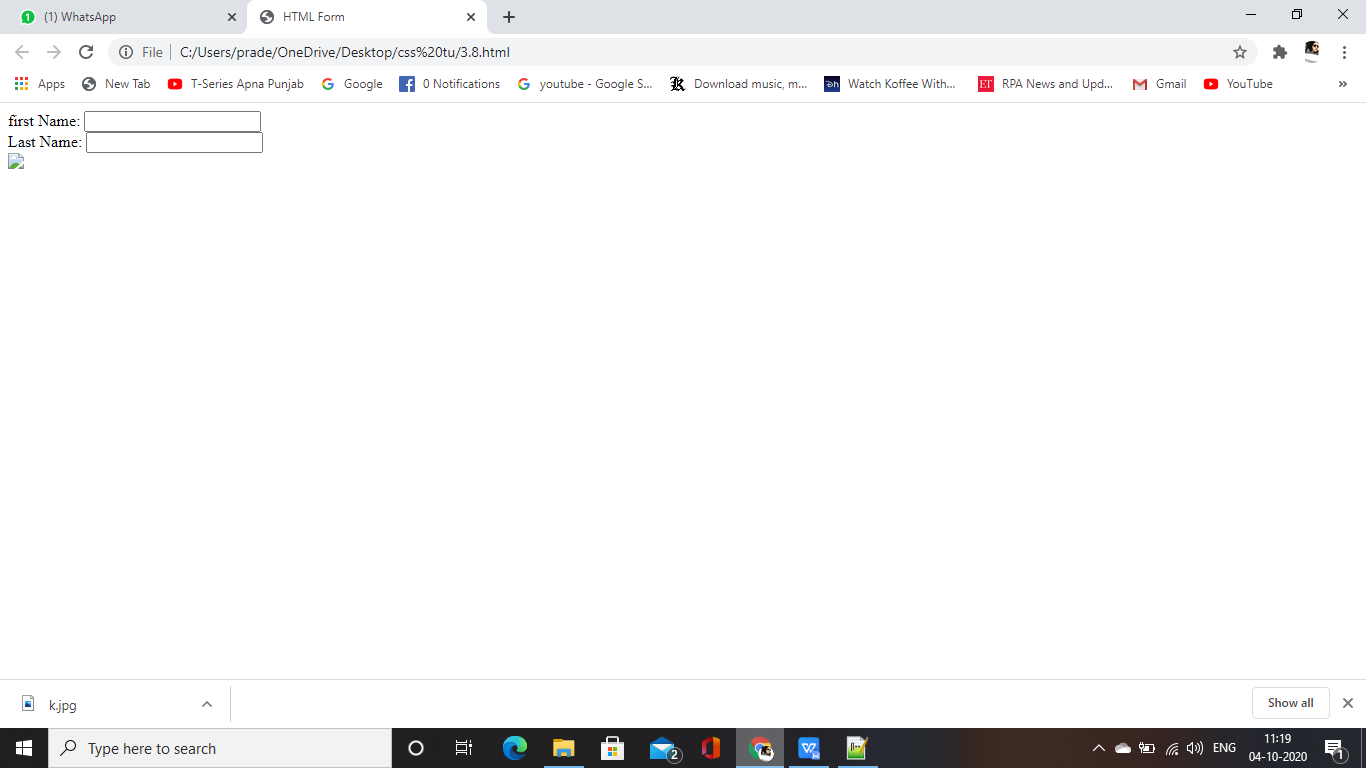
forms.myform.reset()"/>

</p>

</form>

</body>

</html>



**3.8.1 Disabling Elements**

**It is common to display a form with some elements disabled, which prevents the User from entering information into the element.**

**Following are the characteristics of disabled elements:**

**1. Cannot have focus.**

**2. Do not receive or fire mouse events.**

**3. Cannot receive user input.**

**4. Disabled elements are rendered in gray by default in browsers.**

<html>

<head>

<script type="text/javascript">

function ToggleDisabled ()

{

var text = document.getElementById ("myText");

if ('disabled' in text)

{

text.disabled = !text.disabled;

}

}

</script>

</head>

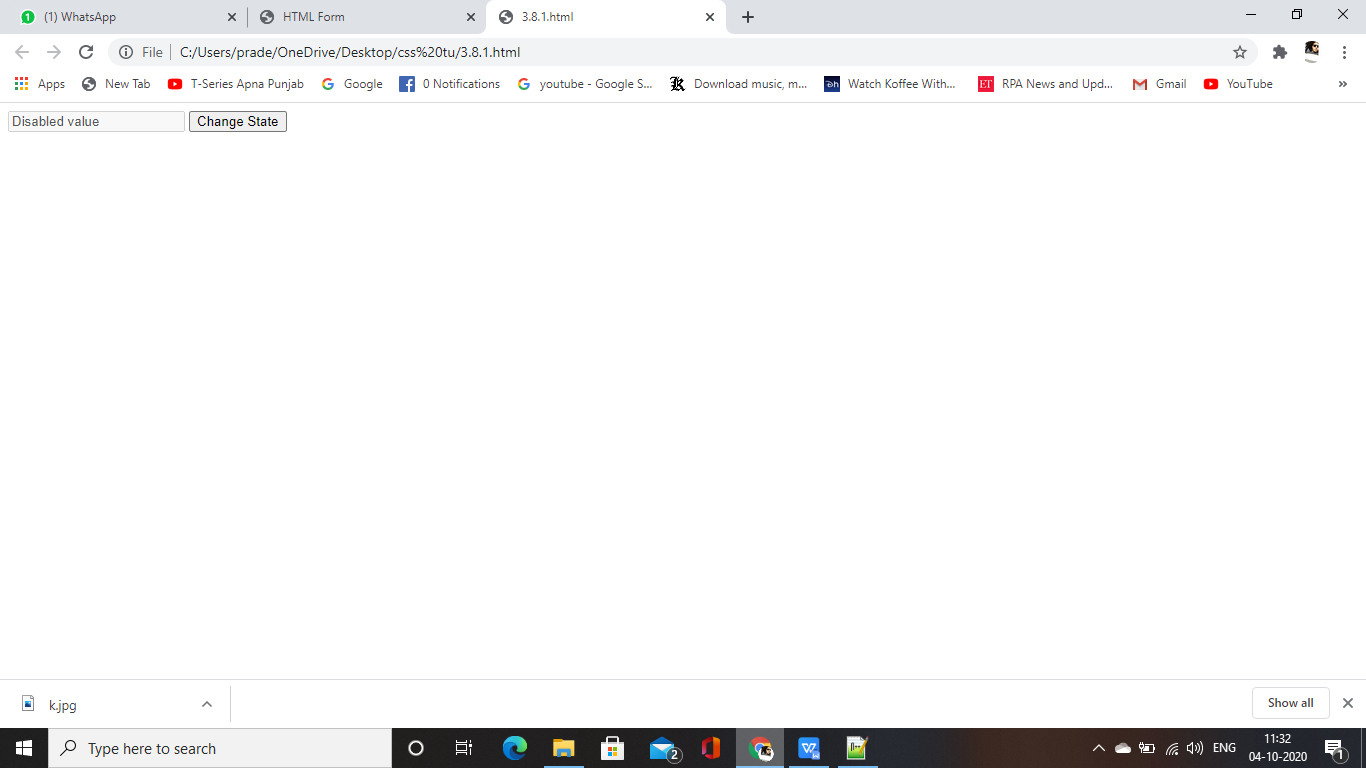
<body>

<input type="text" id="myText" disabled="disabled" value="Disabled value" />

<button onclick="ToggleDisabled ();">Change State</button>

</body>

</html>



**3.8.2 Read Only Elements**

**Most of the<input> element uses read only property to sets or retrieves whether the contents of the element are changeable.**

**If the state is read only, the user cannot modify the contents of the element, but the element continues to get focus and be selectable. If you want to prevent the user from interacting with the object, use the disabled property.**

**The possible values of read only property are true or false that indicates whether the contents of the element can be changed.**

<html>

<head>

<script type="text/javascript">

function changeState ()

{

var textarea = document.getElement ById ("myText" );

textarea.readOnly = !textarea.readOnly;

}

</script>

</head>

<body>

<textarea id="myText" rows="3" readonly="readonly" >Change this text!</textarea>

<button onclick="changeState();">Change me</button>

</body

</html>

