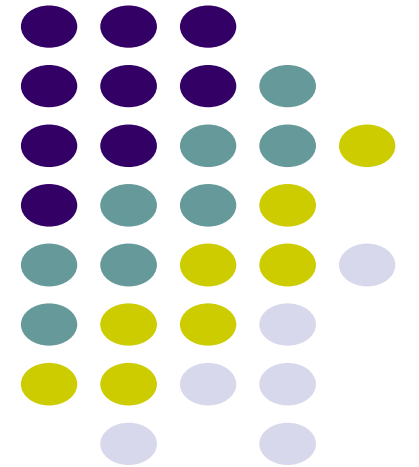


# JavaScript Events, DOM Method, Styles

---

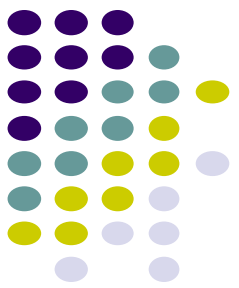




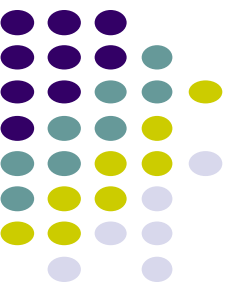
# Event Handling in JavaScript

- HTML events are “**state changes**” that happen to HTML elements by browser or user.
- When JavaScript is used in HTML pages, JavaScript can “**react**” on these events.
- Examples of HTML events:
  - An HTML web page has finished loading
  - An HTML input field was changed
  - An HTML button was clicked

# Common HTML Events



Event name	Description
onload	The browser has finished loading the page
onclick	The user clicks an HTML element
onchange	An HTML element has been changed
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pressing a keyboard key
onkeyup	The user releases a keyboard key
onfocus	The element gets the focus on it
onblur	The event occurs when an element loses focus
oninvalid	The event occurs when an element is invalid
onselect	The event occurs after the user selects some text (for <code>&lt;input&gt;</code> and <code>&lt;textarea&gt;</code> )
onsubmit	The event occurs when a form is submitted



# Writing Event in HTML

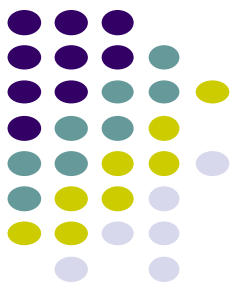
With single quotes:

```
<element event='some JavaScript'>
```

With double quotes:

```
<element event="some JavaScript">
```

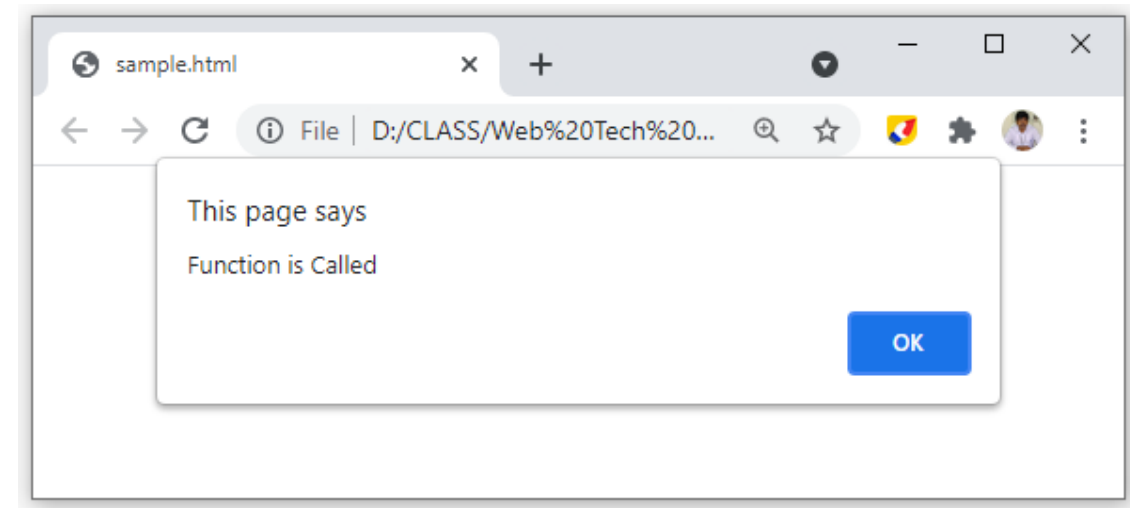
# Handling Event - **onload**



- **onload** event occurs whenever the browser has finished loading the page
- For example,
  - The alert function will be called when page loads.

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function test(){
        alert("Function is Called");
      }
    </script>
  </head>
  <body onload="test()">

  </body>
</html>
```



# Calling Functions when a button is clicked

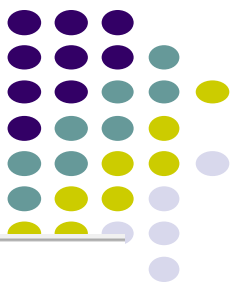


- **onclick** event occurs whenever a user clicks on a particular HTML element.

```
<html>
  <head>
    <script>
      function fun1(){
        alert("fun1 is called when 'Compute' button is clicked");
      }
      function fun2(){
        alert("fun1 is called when 'Convert' button is clicked");
      }
    </script>

  </head>
  <body>
    <input type="button" value="Compute" onclick="fun1()"> <br><br>
    <button onclick="fun2()">Convert</button>
  </body>
</html>
```

# Display inside HTML Element on event



```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script>
      function fun(){
        document.getElementById('box1').innerHTML = "Fun..";
      }
    </script>
  </head>
  <body>
    <div id="box1" style="color:blue;font-size: 20px;margin-left: 50px;">
    </div>
    <button onclick="fun()">Login</button>
  </body>
</html>
```



# Reading values from Input Elements

## Text Box Input Element

- To read the value from a text input fields, first we need to find the text box using its ID.
- Then, “**value**” property returns the value enter by the user.

HTML DOM object

Property to access the value text box entered by the user

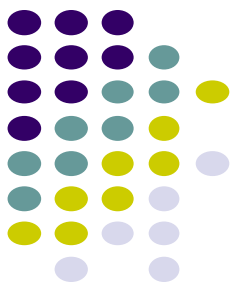
```
var data=document.getElementById('name').value;
```

variable

id of input Element



# Example1: Read the data and display it



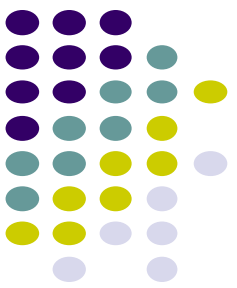
```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function read(){
        var name=document.getElementById('t1').value;
        document.write("Name: "+name);
      }
    </script>
  </head>
  <body>
    Enter Name:<input type="text" id="t1"><br><br>
    <button onclick="read()">click</button><br><br>
  </body>
</html>
```

localhost:8383/Exercise4/JSDemo3.html

Enter Name:

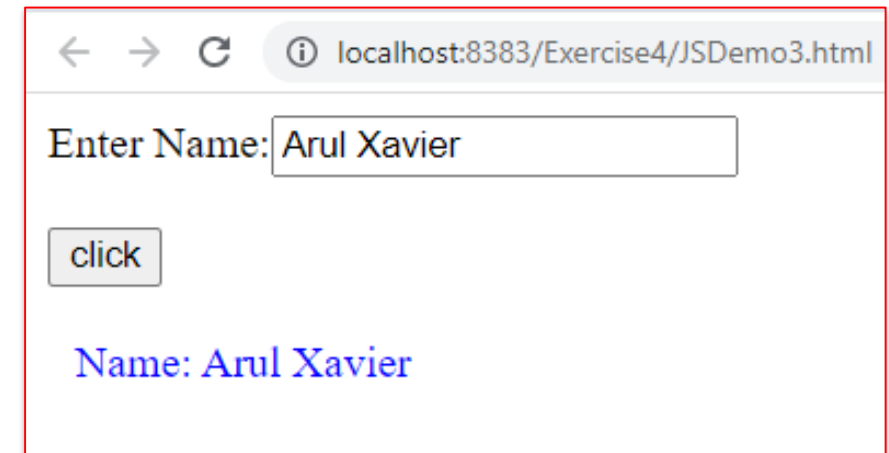
localhost:8383/Exercise4/JSDemo3.html

Name: Arul Xavier



## Example 2: Read data and display inside HTML Element

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function read(){
        var name=document.getElementById('t1').value;
        document.getElementById('box1').innerHTML = "Name: "+name;
      }
    </script>
  </head>
  <body>
    Enter Name:<input type="text" id="t1"><br><br>
    <button onclick="read()">click</button><br><br>
    <div id="box1" style="color:blue;margin-left: 10px;">
    </div>
  </body>
</html>
```



# Example: Add Two Numbers



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script>
    function add(){
      var a = parseInt(document.getElementById('tb1').value);
      var b = parseInt(document.getElementById('tb2').value);
      var sum = a+b;
      document.getElementById('result').value = sum;
    }
  </script>
</head>
<body>
  Number1: <input type="text" id="tb1"><br><br>
  Number2: <input type="text" id="tb2"><br><br>
  Result: <input type="text" id="result"><br><br>
  <button onclick="add()">ADD</button>
</body>
</html>
```



# Example: Add Two Numbers

Document x +

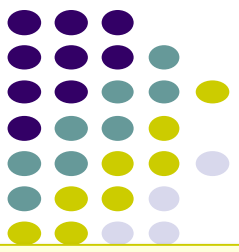
← → ↻ ⓘ File | D:/examples/vmax/ClassDemo/JSDemo2.html

Number1:

Number2:

Result:

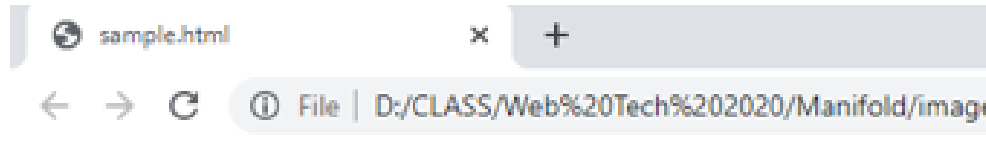
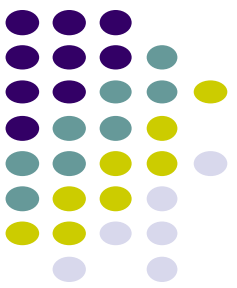
# Example 3: Dollar to Rupee Converter



## Dollar to Rupee Converter

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <script>
      function convert() {
        var a=document.getElementById("t1").value;
        var inr=a*67;
        document.getElementById("d1").innerHTML = inr;
      }
    </script>
  </head>
  <body>
    Enter the Dollars:<input type="text" id="t1">
    INR: <div id="d1" style="display:inline"></div><br>
    <button onclick="convert()">Convert</button>
  </body>
</html>
```

# Output



Dollar(\$): 10

Rupee(₹): 720

Submit

when submit button is clicked it  
calls convert function using onclick  
event

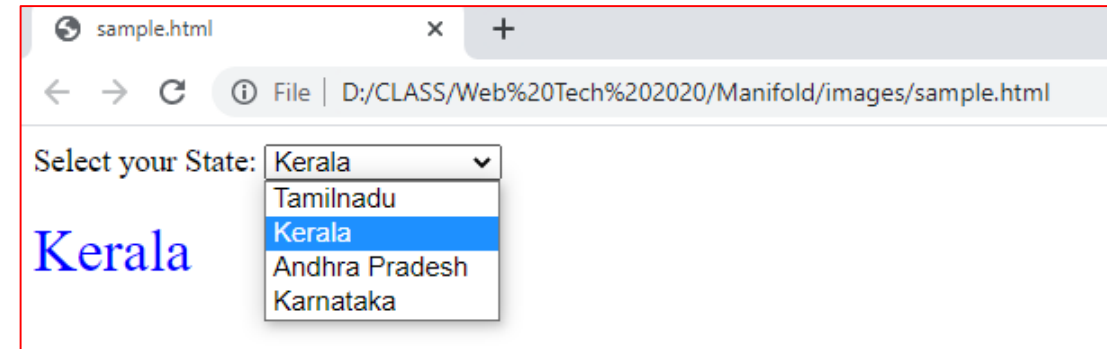
The value of Rupee is updated  
using value property when the  
button is clicked

# onchange event

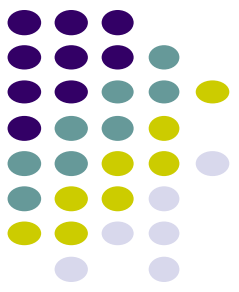


- The **onchange** event occurs when the value of an element has been changed.

```
<html>
<head>
  <script>
    function test(){
      var data=document.getElementById('state').value;
      document.getElementById('output').innerHTML = data;
    }
  </script>
</head>
<body>
  Select your State:
  <select id="state" onchange="test()">
    <option>Tamilnadu</option>
    <option>Kerala</option>
    <option>Andhra Pradesh</option>
    <option>Karnataka</option>
  </select>
  <br><br>
  <div id="output" style="color:blue;font-size:30px;"></div>
</body>
</html>
```



# The **onkeyup** event



- The **onkeyup** event occurs when user release the key while typing.

## HTML Code

```
<body>
  Enter Data: <input type="text" id="name" onkeyup="cube()"> <br><br>
  <div id="output" style="color:blue;font-size:30px;"></div>
</body>
```

## JS Code

```
<script>
  function cube(){
    var data=document.getElementById('name').value;
    var cube= data * data * data;
    document.getElementById('output').innerHTML = cube;
  }
</script>
```



# The **onkeyup** event

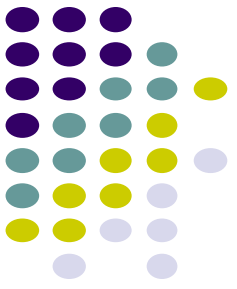


← → ↻ ⓘ File | C:/Users/Admin/Desktop/abc.html

Enter Data:

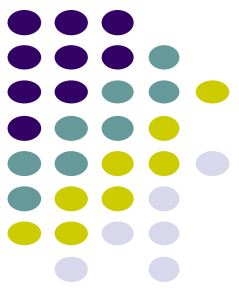
125

# JavaScript Popup Dialog Boxes



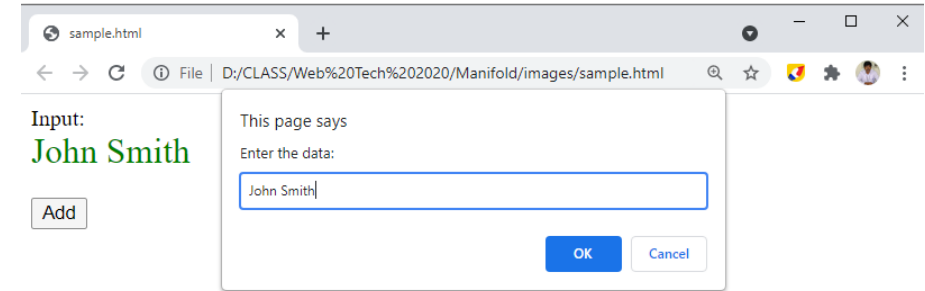
- Alert Dialog Box
- Prompt Box
- Confirm Box

# Creating Input Dialog Box



The `prompt()` method displays a dialog box that prompts the visitor for input.

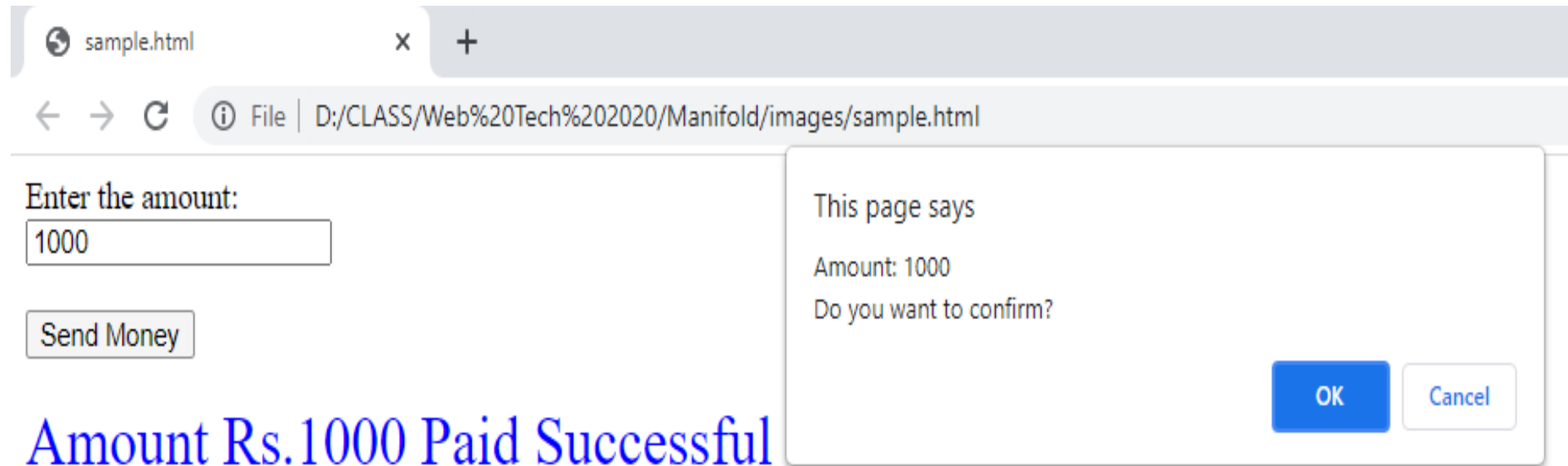
```
<html>
  <head>
    <script>
      function getInput(){
        var data = prompt("Enter the data: ");
        document.getElementById('output').innerHTML = data;
      }
    </script>
  </head>
<body>
  Input:<div id="output" style="color:green;font-size:25px;"></div><br>
  <button onclick="getInput()">Add</button>
</body>
</html>
```





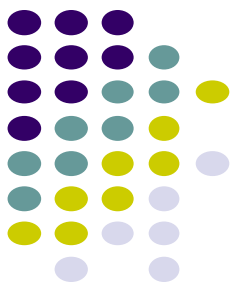
# Creating Confirm Dialog Box

- The **confirm()** method displays a dialog box with a specified message, along with an OK and a Cancel button.
- A confirm box is often used if you want the user to verify or accept something.



**See Next Slide for Program**

## Creating Confirm Dialog Box



```
<script>
  function pay(){
    var data = document.getElementById('amount').value;
    var status = confirm("Amount: "+data+"\nDo you want to confirm?");
    if(status==true){
      outstr = "Amount Rs."+data+" Paid Successful";
      document.getElementById('output').innerHTML = outstr;
    }
  }
</script>
<body>
  <div>Enter the amount:</div>
  <input type="text" id="amount"> <br><br>
  <button onclick="pay()">Send Money</button><br><br>
  <div id="output" style="color:blue;font-size:30px;"></div>
</body>
```



# Accessing CSS properties in Javascript

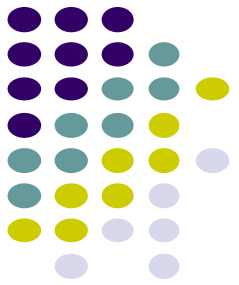
- The HTML DOM allows JavaScript to change the style of HTML elements.
- The HTML DOM also provides style object to every HTML element which is used to access the CSS properties of every HTML element.
- To change the style of an HTML element, use this syntax:

`document.getElementById(id).style.property = New_style`

Here, the **property** is a javascript variable of style object, which represents the CSS properties.

For example, the property may be **color, backgroundColor, borderBottom, marginLeft, etc...**

# Sample JavaScript Style Properties



CSS Properties	JavaScript Style Properties
color	color
background-color	backgroundColor
background-image	backgroundImage
border	border
border-color	borderColor
font-size	fontSize
fontWeight	fontWeight
margin	margin
margin-left	marginLeft
padding	padding
padding-top	paddingTop
position	position
text-align	textAlign
width	width

# Accessing CSS properties in Javascript



- Illustration of Changing **CSS Style** with **onmouseover**

```
<body>
  <div id='box' onmouseover="changeStyle()">
    Change Style
  </div>
</body>
<script>
  function changeStyle(){
    document.getElementById('box').style.color = "white";
    document.getElementById('box').style.fontSize = "30px";
    document.getElementById('box').style.backgroundColor = "blue";
    document.getElementById('box').style.padding = "12px";
    document.getElementById('box').style.width = "200px";
  }
</script>
```

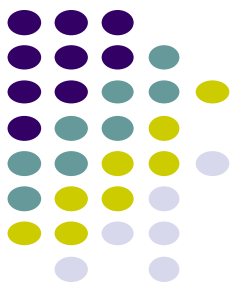




## Illustration of Changing **CSS Style** with **onmouseover**



- The previous JavaScript code can be simplified as given below



```
<script>
    function changeStyle(){
        var element=document.getElementById('box');
        element.style.color = "white";
        element.style.fontSize = "30px;";
        element.style.backgroundColor = "blue";
        element.style.padding = "12px";
        element.style.width = "200px";
    }
</script>
```