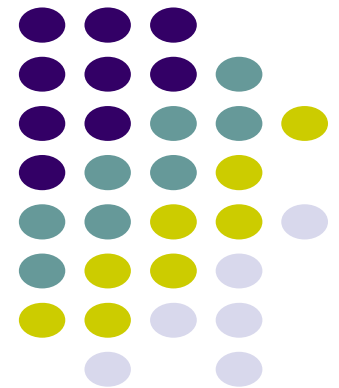
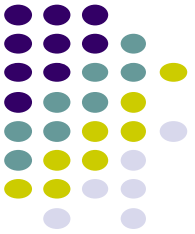


# JavaScript Input Validation, Changing CSS Styles and other DOM Methods

---

***Dr. Arul Xavier V M***

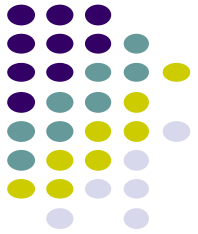




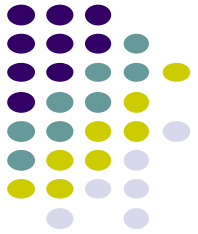
# Input Validation using JavaScript

- **Input validation** is an **essential aspect** of JavaScript to ensure that user-provided data meets certain criteria before being processed.
- JavaScript provides facility to validate the form on the **client-side** so data processing will be faster than server-side validation.
- Most of the web developers prefer **JavaScript form validation**.

# Input Validation Example



- In this example, we are going to validate the name and password.
  - The name and password can't be empty and password can't be less than 8 characters long.
  - Here, we are validating the form on form submit.
  - The user will not be forwarded to the next page until given values are correct.



```
<body>
  <form action="/server" name="myform" onsubmit="return validate()">
    <h3>User Registration</h3>
    <label>Name:</label>
    <input type="text" id="username" name="username">
    <span id="nameerror" style="color:red;"></span>
    <label>Password</label>
    <input type="text" id="password" name="password">
    <span id="passerror" style="color:red;"></span>
    <button type="submit">Save Details</button>
  </form>
</body>

<style>
  label,button {
    display: block;
    margin-top: 12px;
  }
</style>
```



## User Registration

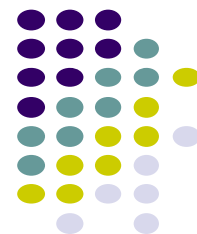
Name:

Password

Save Details

```
<script>
function validate(){
  var name = document.getElementById('username').value;
  var pass = document.getElementById('password').value;
  document.getElementById('nameerror').innerHTML="";
  document.getElementById('passerror').innerHTML="";
  if(name == ""){
    document.getElementById('nameerror').innerHTML = "Name is required!"
    return false;
  }
  if(pass == ""){
    document.getElementById('passerror').innerHTML = "Password is required!"
    return false;
  }
  if(pass.length < 8){
    document.getElementById('passerror').innerHTML = "Password must be 8 or more characters long"
    return false;
  }

  return true;
}
</script>
```



# Sample Outputs



127.0.0.1:5500/jsdemo2.html

### User Registration

Name:

Name is required!

Password

Save Details

When the Name field is empty

127.0.0.1:5500/jsdemo2.html

### User Registration

Name:

Password

Password is required!

Save Details

When the Password field is empty

127.0.0.1:5500/jsdemo2.html

### User Registration

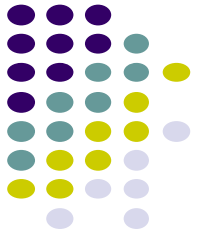
Name:

Password

Password can't be less than 8 characters long

Save Details

When the password length is less than 8



## Example 2: Input Validation

- Create JavaScript program to validate the following HTML form fields.
  - Person Name (should contain only alphabet letters)
  - E-mail ID (should not contain any invalid and must follow the pattern such as [name@domain.com](#))
  - Mobile Number (The phone number should contain 10 digits only)
- Also, validate the above fields to ensure it is not empty

In this example, we can use [Regular Expression](#) for reducing the number of lines in the code

## Example 2: Input Validation

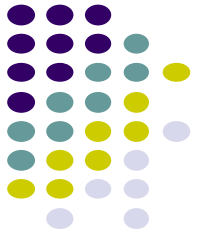


```
<body>
  <form action="/server" onsubmit="return validate()">
    <h3>User Registration</h3>
    <label>Name:</label>
    <input type="text" id="name" name="name">
    <span id="nameerror" style="color:red;"></span>
    <label>Email ID:</label>
    <input type="text" id="emailid" name="emailid">
    <span id="emailerror" style="color:red;"></span>
    <label>Mobile Number:</label>
    <input type="text" id="mobile" name="mobile">
    <span id="mobileerror" style="color:red;"></span>
    <button type="submit">Save Details</button>
  </form>
</body>
<style>
  label,button {
    display: block;
    margin-top: 12px;
  }
</style>
```

A screenshot of a web browser window displaying the "User Registration" form. The browser's address bar shows the URL "127.0.0.1:5500/jsdemo2.html". The form has a title "User Registration" and three input fields labeled "Name:", "Email ID:", and "Mobile Number:". Below the input fields is a "Save Details" button. The form is styled with a simple, clean layout.



## Java script Code



```
<script>
```

```
function validate(){
    var name = document.getElementById('name').value;
    var emailid = document.getElementById('emailid').value;
    var mobile = document.getElementById('mobile').value;
    const namerror = document.getElementById('nameerror')
    const emailerror = document.getElementById('emailerror')
    const mobileerror = document.getElementById('mobileerror')
    namerror.innerHTML="";
    emailerror.innerHTML="";
    mobileerror.innerHTML = "";
    if(name == ""){
        namerror.innerHTML = "Name is required!"
        return false;
    }
    const nameregex = /^[a-zA-Z]+$/
    if(nameregex.test(name) == false){
        namerror.innerHTML = "Name should contain alphabets only!"
        return false;
    }
}
```



```
if(emailid == ""){
    emailerror.innerHTML = "E-mail ID is required!"
    return false;
}
const emailregex = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,}$/;
if(emailregex.test(emailid) == false){
    emailerror.innerHTML = "Email ID format is wrong!!"
    return false;
}
if(mobile == ""){
    mobileerror.innerHTML = "Mobile Number is required!"
    return false;
}
const mobileregex = /^[987][0-9]{9}$/;
if(mobileregex.test(mobile) == false){
    mobileerror.innerHTML = "Mobile Number is Invalid"
    return false;
}
return true;
}
</script>
```

← → ↻ ⓘ 127.0.0.1:5500/jsdemo2.html

### User Registration

Name:  
 Name is required!

Email ID:

Mobile Number:

← → ↻ ⓘ 127.0.0.1:5500/jsdemo2.html

### User Registration

Name:

Email ID:  
 E-mail ID is required!

Mobile Number:

← → ↻ ⓘ 127.0.0.1:5500/jsdemo2.html

### User Registration

Name:

Email ID:  
 Email ID format is wrong!!

Mobile Number:

← → ↻ ⓘ 127.0.0.1:5500/jsdemo2.html

### User Registration

Name:

Email ID:

Mobile Number:  
 Mobile Number is required!

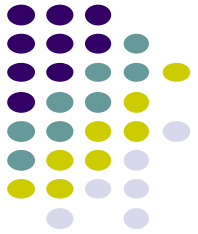
← → ↻ ⓘ 127.0.0.1:5500/jsdemo2.html

### User Registration

Name:

Email ID:

Mobile Number:  
 Mobile Number is Invalid



# Manipulating 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.

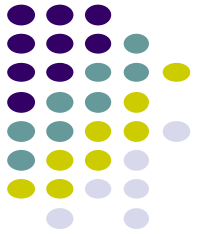
For example, the property may be **color, backgroundColor, borderBottom, marginLeft, etc...**

# 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

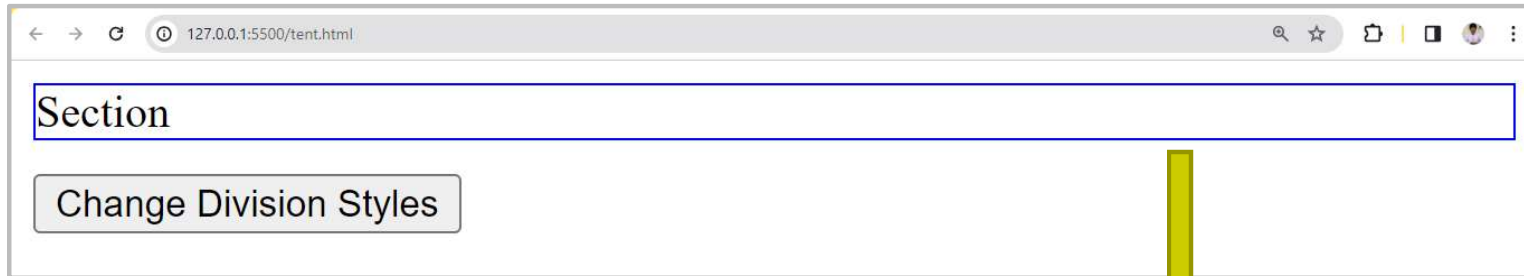
# Changing CSS properties in Javascript



- In this example, we can understand how to change the CSS styles using JavaScript code.

```
<script>
    function changeStyle(){
        document.getElementById('box').style.color = "blue";
        document.getElementById('box').style.fontSize = "30px";
        document.getElementById('box').style.backgroundColor = "pink";
        document.getElementById('box').style.padding = "16px";
        document.getElementById('box').style.width = "300px";
        document.getElementById('box').style.borderRadius = "8px";
    }
</script>
<body>
    <div id="box" style="border: 1px solid blue; margin-bottom: 12px;">
        Section
    </div>
    <button onclick="changeStyle()">Change Division Styles</button>
</body>
```

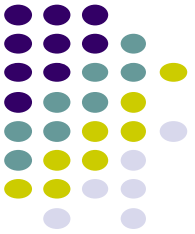
# Changing CSS properties in Javascript



When the button is clicked, the pervious CSS styles of DIV Element is changed



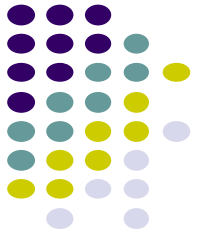
- The previous JavaScript code can be simplified as given below



```
<script>
    function changeStyle(){
        var element=document.getElementById('box');
        element.style.color = "blue";
        element.style.fontSize = "30px";
        element.style.backgroundColor = "pink";
        element.style.padding = "12px";
        element.style.width = "300px";
        element.style.borderRadius = "8px";
    }
</script>
```

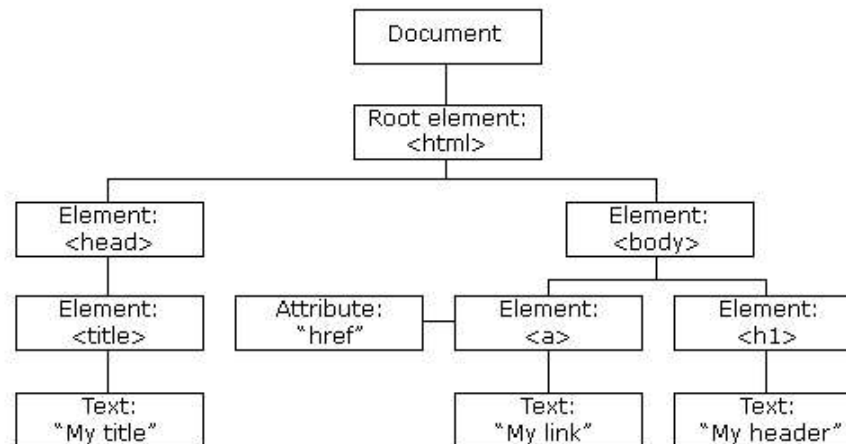


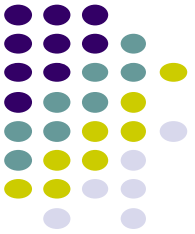
# The HTML DOM (Document Object Model)



- When a web page is loaded, the browser creates a **Document Object Model** of the page.
- The **HTML DOM** model is constructed as a tree of **Objects**:

The HTML DOM Tree of Objects

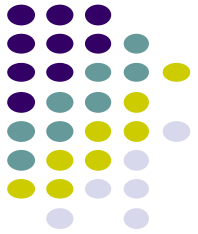




# What is the HTML DOM?

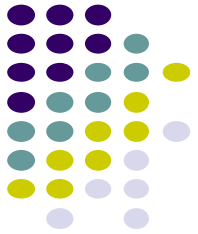
- The HTML DOM is a standard **object** model and **programming interface** for HTML.
- It defines:
  - The HTML elements as **objects**
  - The **properties** of all HTML elements
  - The **methods** to access all HTML elements
  - The **events** for all HTML elements

# JavaScript - HTML DOM Methods



- Finding HTML Elements
  - `document.getElementById(id)`
    - Find single element by element id
  - `document.getElementsByName(name)`
    - Find multiple elements by element name
  - `document.getElementsByClassName(name)`
    - Find multiple elements by class name
  - `document.getElementsByTagName(name)`
    - Find multiple elements by tag name

# Finding HTML Elements by CSS Selectors



- `document.querySelector(css_selector)`
  - The `querySelector()` method returns the first element that matches a specified CSS *selector(s)* in the document.
- `document.querySelectorAll(css_selector)`
  - To return all the matches, use the `querySelectorAll()` method instead.

## Finding HTML element by Id

```
<script>
```

```
    function changeColor(color){  
        var div = document.getElementById('box');  
        div.style.backgroundColor = color.value;  
    }
```

```
</script>
```

```
<body>
```

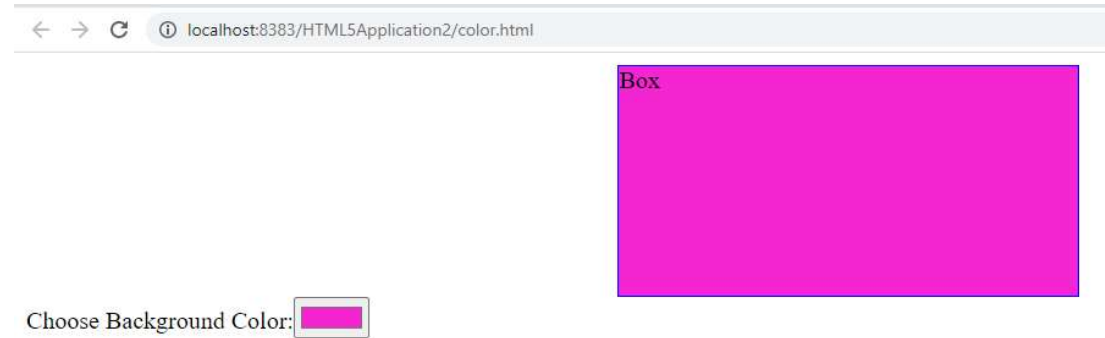
```
  <div id="box"  
    style="width:300px;height: 150px;border:1px solid blue;margin: auto;">
```

```
    Box
```

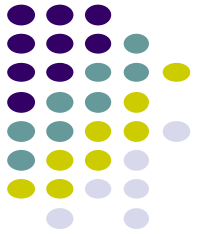
```
  </div>
```

```
  Choose Background Color:<input type="color" onchange="changeColor(this)">
```

```
</body>
```



## getElementsByName()



```
<script>

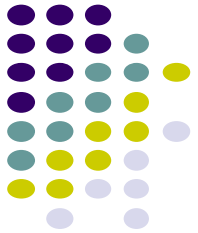
    function test(){
        var elements=document.getElementsByName('smart');
        var output = "<ol>";
        for(var i=0;i<elements.length;i++){
            if(elements[i].checked == true){
                output += "<li>"+elements[i].value+"</li>";
            }
        }
        output += "</ol>";
        document.getElementById('result').innerHTML = output;
    }
</script>

<body>

    Select your Favorite SmartPhone:
    <input name="smart" type="checkbox" value="Apple"> Apple
    <input name="smart" type="checkbox" value='Samsung'> Samsung
    <input name="smart" type="checkbox" value='Redmi'> Redmi
    <input name="smart" type="checkbox" value="OnePlus"> OnePlus
    <br> <br>
    <button onclick="test()">Save</button><br>
    <div id="result" style='color:red;'>
    </div>

</body>
```

# getElementsByTagName()



← → ↻ ⓘ localhost:8383/JSDemo/dom2.html

Select your Favorite SmartPhone: ☒ Apple ☒ Samsung ☒ Redmi ☐ OnePlus

Save

1. Apple
2. Samsung
3. Redmi

```
<script>
```

**document.getElementsByTagName(tagname)**

```
function register() {
```

```
var elements = document.getElementsByTagName('input');
```

```
var output="";
```

```
for(var i=0;i<elements.length;i++){
```

```
    if(elements[i].type === "text")
```

```
        output += elements[i].value+"<br>";
```

```
}
```

```
document.getElementById("result").innerHTML = output;
```

```
}
```

```
</script>
```

```
<body>
```

```
    <div>
```

```
        Register Yourself Here!:<br>
```

```
        <div>Name: <input type='text'></div>
```

```
        <div>Email: <input type='text'></div>
```

```
        <div>Phone: <input type='text'></div>
```

```
    </div>
```

```
    <button onclick="register()">Register Now!</button>
```

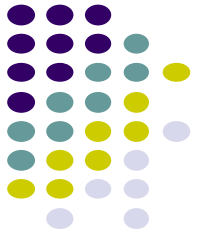
```
    <div id="result" style='color:red;'></div>
```

```
</body>
```





# document.getElementsByTagName(tagname)



Register Yourself Here!:

Name:

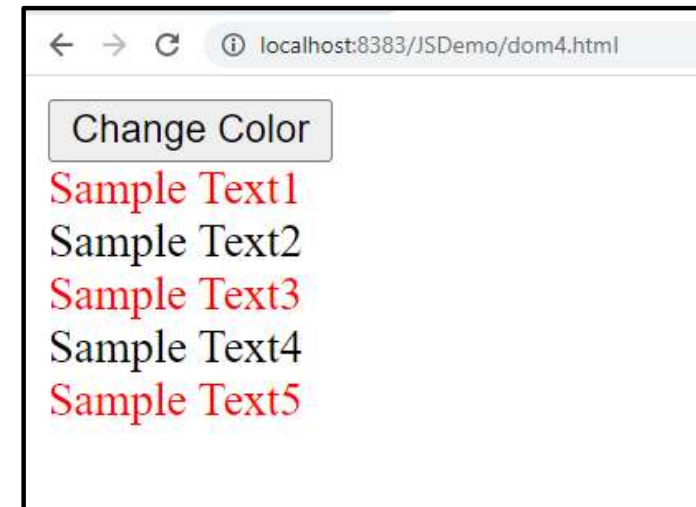
Email:

Phone:

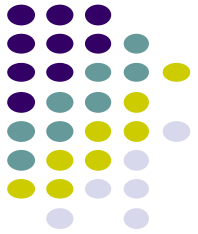
Arul  
arul@gmail.com  
9923232323

## document.getElementsByClassName(name)

```
<script>
  function changeColor() {
    var elements=document.getElementsByClassName('d1');
    for(var i=0;i<elements.length;i++){
      elements[i].style.color = "Red";
    }
  }
</script>
<body>
  <div>
    <button onclick="changeColor()">Change Color</button>
  </div>
  <br>
  <div class='d1'>Sample Text1</div>
  <div class='d2'>Sample Text2</div>
  <div class='d1'>Sample Text3</div>
  <div class='d2'>Sample Text4</div>
  <div class='d1'>Sample Text5</div>
</body>
```



# document.querySelector(css\_selector)



- The **querySelector()** method returns the first element that matches a specified *CSS selector(s)* in the document.
- It returns a **Single** element.

# document.querySelector(css\_selector)



```
<script>
    function test(){
        var element=document.querySelector(".sample");
        element.style.backgroundColor="yellow";
    }
</script>
<body>
    <h1 class="sample">Welcome to Java Scripts</h1>
    <p class="sample">This is a paragraph</p>
    <div class="sample">This is a division</div><br><br>
    <button onclick="test()">click</button>
</body>
```

← → ↻ localhost:8383/HTML5Application1/newhtml1.html

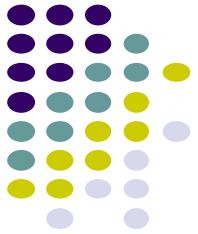
**Welcome to Java Scripts**

This is a paragraph

This is a division

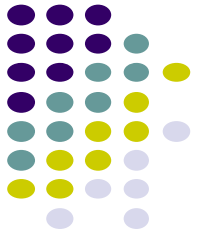
click

# document.querySelectorAll(css\_selector)



- The **querySelectorAll()** method returns all elements in the document that matches a specified CSS selector(s) as collection of elements. (**Similar to Array**)
- Every element can be accessed via its index number.

# document.querySelectorAll(css\_selector)



```
<script>
```

```
function test() {  
    var element=document.querySelectorAll(".sample");  
    for(var i=0;i<element.length;i++){  
        element[i].style.backgroundColor="yellow";  
    }  
}
```

```
</script>
```

```
<body>
```

```
<h1 class="sample">Welcome to Java Scripts</h1>  
<p class="sample">This is a paragraph</p>  
<div class="sample">This is a division</div><br><br>  
<button onclick="test()">click</button>
```

```
</body>
```

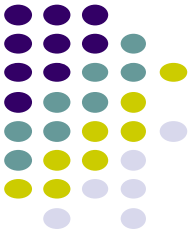
localhost:8383/HTML5Application1/newhtml1.html

Welcome to Java Scripts

This is a paragraph

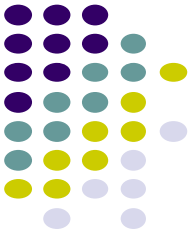
This is a division

click



# Working with Timer functions

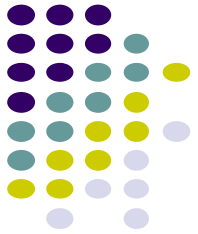
- **Animation** is a process of rapidly displaying sequence of the images or frames on web page.
- **Timer functions** can used to create such animations.
  - setTimeout()
  - setInterval()
  - clearTimeout()
  - clearInterval()



## Calling functions with Timer

- `setTimeout()` – executes the code after specified time.  
*`setTimeout(func_name, time);`*
- `setInterval()` – executes the code every time interval.  
*`setInterval(func_name, time);`*
- `clearTimeout()` and `clearInterval()` is used to cancel the timer.





# setTimeout() with function definition

To call a function after specified time(milliseconds) -  
`setTimeout(function_name , time_milliseconds)`

```
<script>  
    function sayHello(){  
        alert('Hello');  
    }  
    setTimeout(sayHello,2000);  
</script>
```

localhost:8383 says  
Hello

OK

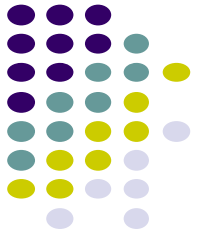
# setTimeout() in animation



```
<!DOCTYPE html>
<html>
  <head>
    <script>
      timer = '';
      function change(){
        var element=document.getElementById('box');
        if(element.style.backgroundColor === 'red')
          element.style.backgroundColor = 'yellow';
        else
          element.style.backgroundColor = 'red';

        timer = setTimeout(change,500);
      }
      function stop(){
        clearTimeout(timer);
      }
    </script>
  </head>
  <body>
    <div id='box' style="background-color: red;width:300px;height: 100px"></div><br>
    <button onclick="change()">Change Color</button>
    <button onclick="stop()">stop</button>
  </body>
</html>
```

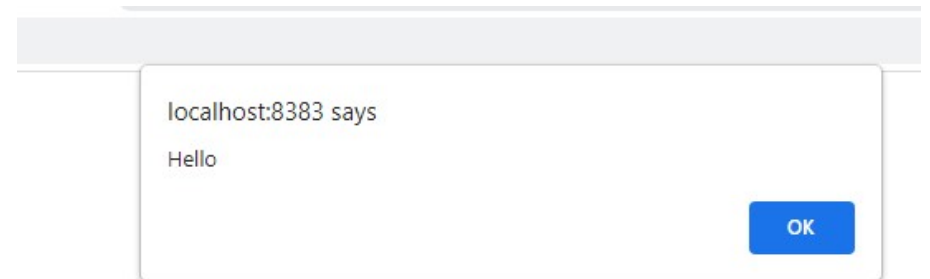
Activate Wind



# setInterval() with function definition

To call a function every specified time interval(milliseconds) -  
**setInterval(function\_name , time\_milliseconds)**

```
<script>  
    function sayHello(){  
        alert('Hello');  
    }  
    setInterval(sayHello,2000);  
</script>
```



# setInterval() in animation

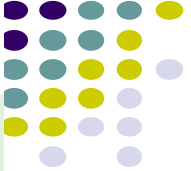


```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function init(){
        timer = setInterval(change,500);
      }
      function change(){
        var element=document.getElementById('box');
        if(element.style.backgroundColor === 'red')
          element.style.backgroundColor = 'yellow';
        else
          element.style.backgroundColor = 'red';
      }
      function stop(){
        clearInterval(timer);
      }
    </script>
  </head>
  <body>
    <div id='box' style="background-color: red;width:300px;height: 100px"></div><br>
    <button onclick="init()">Change Color</button>
    <button onclick="stop()">stop</button>
  </body>
</html>
```



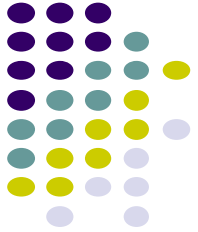
Activate Win

## Working with Animation – Moving HTML Element



```
<html>
  <head>
    <script>
      var timer=null;
      var element=null;
      function init(){
        element = document.getElementById('box');
        element.style.position = 'relative';
        element.style.left = '5px';
      }
      function move(){
        element.style.left = parseInt(element.style.left)+2+'px';
        timer = setTimeout(move,500);
      }
      function stop(){
        clearTimeout(timer);
      }
    </script>
  </head>
  <body onload="init()">
    <div id='box' style="background-color: red;width:100px;height:100px;"></div><br>
    <button onclick="move()">Move</button>
    <button onclick="stop()">Stop</button>
  </body>
</html>
```

# JavaScript Popup Dialog Boxes

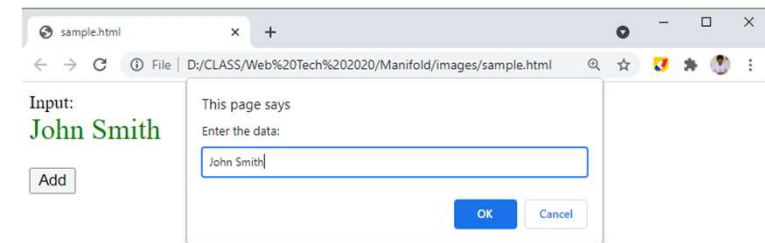


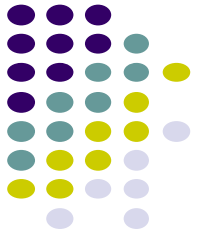
- Alert Dialog Box
- Prompt Box
- Confirm Box

# 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.

The screenshot shows a web browser window with a single tab titled 'sample.html'. The address bar displays the file path 'D:/CLASS/Web%20Tech%202020/Manifold/images/sample.html'. The main content area contains a form with the label 'Enter the amount:', a text input field containing '1000', and a button labeled 'Send Money'. Below the form, the text 'Amount Rs.1000 Paid Successful' is displayed in blue. A confirm dialog box is overlaid on the right side of the browser window. The dialog box contains the text 'This page says', 'Amount: 1000', and 'Do you want to confirm?'. At the bottom right of the dialog box are two buttons: 'OK' (blue) and 'Cancel' (white with a blue border).

**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>
```