Manual Prepared by **Prof.Sulalah Mirkar**	**Lab Manual**	**Academic Year- 2021-22**
Year:-	**Subject:- Web Programming**	**Semester:-**

	Experiment # 6	
		Date of Conduction:- / /2022
Class:-B.Tech IT/MBA.Tech IT/B.Tech (CS)		Division:-

Aim:

A. Java script functions
B. Regular expression
C. Event handling
D. Validations and Query

Prerequisites:-

- OOPS concept and HTMLS basic tags

Theory:-

```
//defining a function

function <function-name>()

{

   // code to be executed

};


//calling a function

<function-name>();
```

Function Declaration

- Function Function_Name(value1,value2)
- {
- return value1;
- Return value 2;
- }

1. Example 1

```
<html>
<script type="javascript">
Function add(a,b)
{
Return a+b;
}
Alert(add(1,2));
</script>
</body>
</html>
```

2. Example: Define and Call a Function

```
function ShowMessage() {
  alert("Hello World!");
    }
ShowMessage();
```

3. Example: Function Parameters

```
function ShowMessage(firstName, lastName) {
    alert("Hello " + firstName + " " + lastName);
}

ShowMessage("Steve", "Jobs");
ShowMessage("Bill", "Gates");
ShowMessage(100, 200);
```

4. Example :Iterate all Arguments

```
function ShowMessage() {

   for(var i = 0; i < arguments.length; i++){
      alert(arguments[i]);
   }
}

ShowMessage("Steve", "Jobs");
```

- Dynamic HTML (DHTML), functionality
  In 1997, Netscape and Microsoft released version 4.0 of Netscape Navigator and Internet Explorer respectively, adding support for Dynamic HTML (DHTML), functionality enabling changes to a loaded HTML document.

getElementById()

The getElementById() method accesses the first element with the specified id

- **Syntax:-** document.getElementById("*id*")
- **Id:-** The id of the element you want to access/manipulate.

5. Example 5

```
<html>
<head>
<script>
function getValue()
  {
  var x=document.getElementById("myHeader");

  }
</script>
</head>
<body>

<h1 id="myHeader“ >Click me!</h1>
```

```
</body>
</html>
```

Try and catch

```
try
{
   // code that may throw an error
}
catch(ex)
{
   // code to be executed if an error occurs
}
finally{
   // code to be executed regardless of an error occurs or not
}
```

Error Handling in JS

```
try
{
   var result  =  Sum(10, 20); // Sum is not defined yet
}
catch(ex)
{
   document.getElementById("errorMessage").innerHTML = ex;
}
```

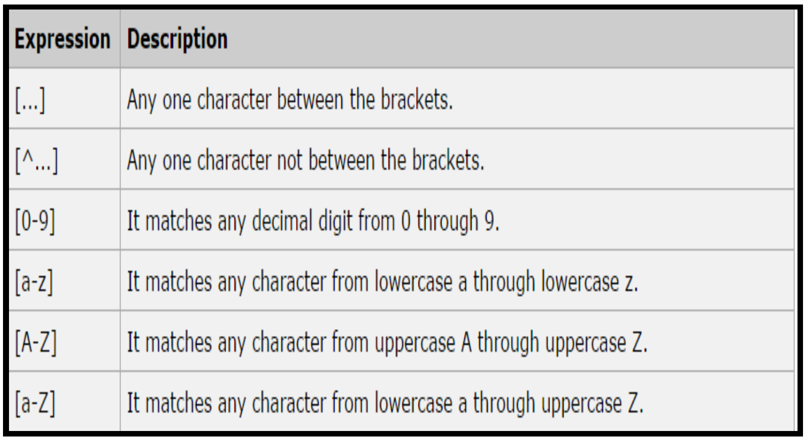
Regular Expressions

- A regular expression is an object that describes a pattern of characters.
- The JavaScript **RegExp** class represents regular expressions.
- **RegExp** define methods that use regular expressions to perform
- powerful pattern-matching
- search-and-replace functions on text.

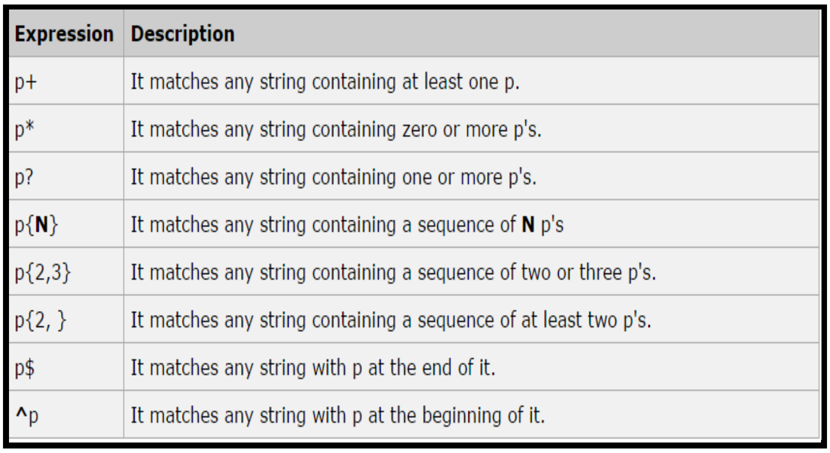
var pattern = new RegExp(pattern, attributes);
pattern: A string that specifies the pattern of the regular expression.

attributes: An optional string containing any of the attributes that specify

- "g“ =global
- "i“ =case-insensitive
- "m" = multiline matches.
- Brackets ([]) have a special meaning.
- They are used to find a range of characters.



Expression	Description
[...]	Any one character between the brackets.
[^...]	Any one character not between the brackets.
[0-9]	It matches any decimal digit from 0 through 9.
[a-z]	It matches any character from lowercase a through lowercase z.
[A-Z]	It matches any character from uppercase A through uppercase Z.
[a-Z]	It matches any character from lowercase a through uppercase Z.



Expression	Description
p+	It matches any string containing at least one p.
p*	It matches any string containing zero or more p's.
p?	It matches any string containing one or more p's.
p{**N**}	It matches any string containing a sequence of **N** p's
p{2,3}	It matches any string containing a sequence of two or three p's.
p{2, }	It matches any string containing a sequence of at least two p's.
p$	It matches any string with p at the end of it.
^p	It matches any string with p at the beginning of it.

Example 6

```
<script type="javascript">
function hello(name)
  {
 alert("Hello"  +name.value+ ".Nice to meet you");
return true;
```

```
  }
</script>

<form>


Enter your name: <input type="text" name="fname" id="1">
<input type="button" onclick ="hello(document.getElementById('1'))"
value="Submit">

</form>
```

<u>innerHTML</u>

- The innerHTML property sets or returns the HTML content (inner HTML) of an element.
- Syntax
- Return the innerHTML property:

➢ *HTMLElementObject*.innerHTML

- Set the innerHTML property:

➢ *HTMLElementObject*.innerHTML=*text*

Example 8

```
<html>
<body>
<h1>My Web Page</h1>
<p id="myP">This is a p element.</p>
<div id="myDIV">This is a div element.</div>
<script>
document.getElementById("myP").innerHTML = "Hello Dolly.";
document.getElementById("myDIV").innerHTML = "How are you?";
</script>
</body>
</html>
```

<u>Use String replace() With a Regular Expression</u>

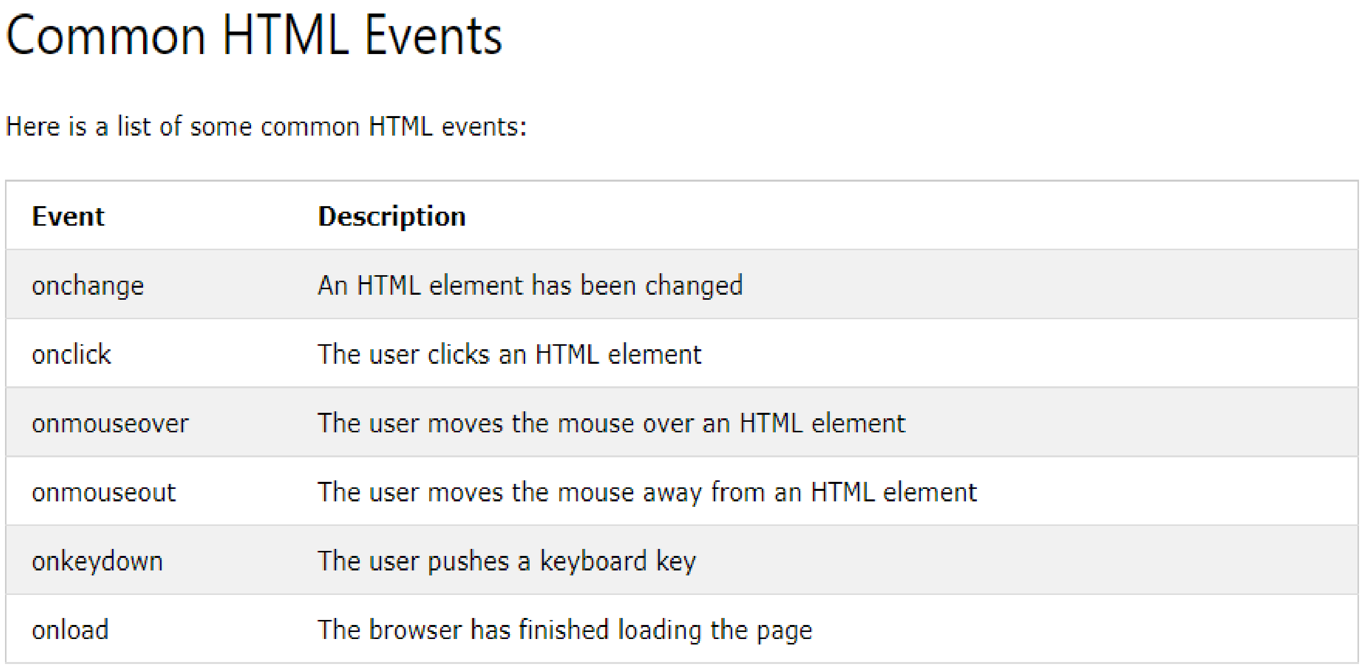
- var str = "Visit Microsoft!";
  var res = str.replace(/microsoft/i, "MPSTME");

6. Example 9

```
<html>
<body>
<h2>JavaScript Functions</h2>
<p>This example calls a function to convert from Fahrenheit to Celsius:</p>
<p id="demo"></p>
<script>
function toCelsius(f) {
  return (5/9) * (f-32);
}

document.getElementById("demo").innerHTML = toCelsius(77);
</script>
</body>
    </html>
```

7. Example 10

```
<html>
<body>
<p>Click the button to display the date.</p>
<button onclick="displayDate()">The time is?</button>
<script>
function displayDate() {
  document.getElementById("demo").innerHTML = Date();
}
</script>
<p id="demo"></p>
</body>
</html>
```



Common HTML Events

Here is a list of some common HTML events:

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

8. Example 11

```
<html>
<body>
<h2>JavaScript Arrays</h2>
<p>JavaScript array elements are accessed using numeric indexes (starting from 0).</p>
<p id="demo"></p>
<script>
var cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars[0];
</script>
</body>
</html>
```

Change the Array Value

- var cars = ["Saab", "Volvo", "BMW"];
  cars[0] = "Opel";
  document.getElementById("demo").innerHTML = cars[0];

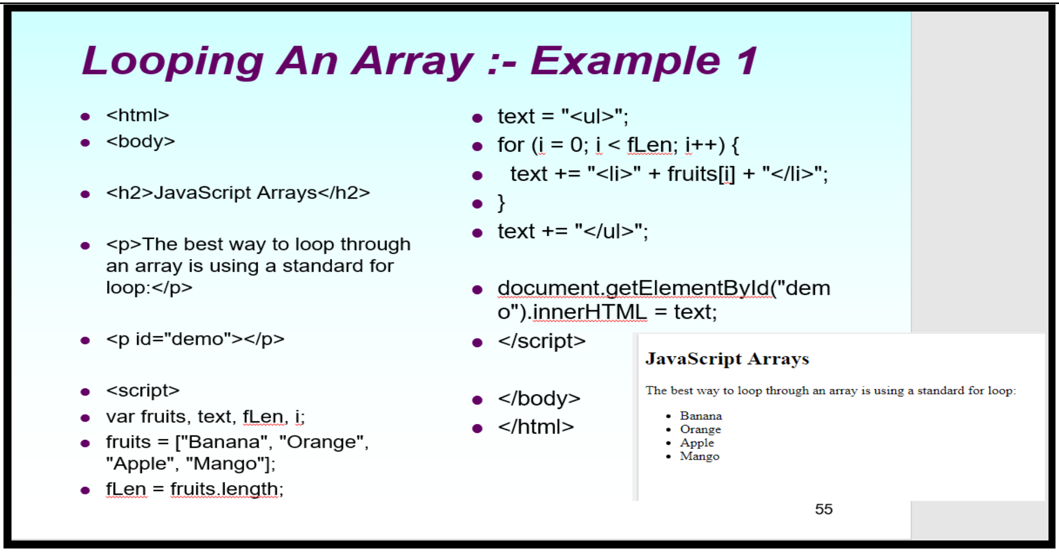
Length() function

var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.length;

Access the first and last element of an Array

- fruits = ["Banana", "Orange", "Apple", "Mango"];
  var first = fruits[0];
- fruits = ["Banana", "Orange", "Apple", "Mango"];
  var last = fruits[fruits.length - 1];

9. Example 12



Looping An Array :- Example 1

- <html>
- <body>
- <h2>JavaScript Arrays</h2>
- <p>The best way to loop through an array is using a standard for loop:</p>
- <p id="demo"></p>
- <script>
- var fruits, text, fLen, i;
- fruits = ["Banana", "Orange", "Apple", "Mango"];
- fLen = fruits.length;
- text = "<ul>";
- for (i = 0; i < fLen; i++) {
-   text += "<li>" + fruits[i] + "</li>";
- }
- text += "</ul>";
- document.getElementById("demo").innerHTML = text;
- </script>
- </body>
- </html>

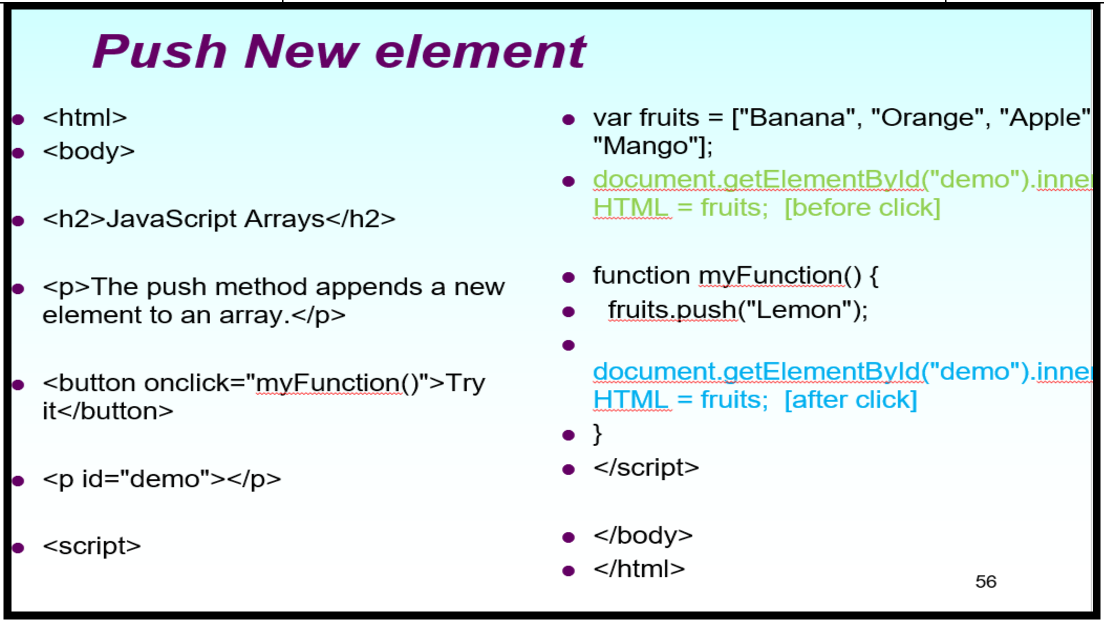
JavaScript Arrays

The best way to loop through an array is using a standard for loop:

- Banana
- Orange
- Apple
- Mango

55

10. Example 13



Push New element

- <html>
- <body>
- <h2>JavaScript Arrays</h2>
- <p>The push method appends a new element to an array.</p>
- <button onclick="myFunction()">Try it</button>
- <p id="demo"></p>
- <script>
- var fruits = ["Banana", "Orange", "Apple" "Mango"];
- document.getElementById("demo").inne HTML = fruits; [before click]
- function myFunction() {
- fruits.push("Lemon");
-
  document.getElementById("demo").inne HTML = fruits; [after click]
- }
- </script>
- </body>
- </html>

56

ARRAY METHODS

- var fruits = ["Banana", "Orange", "Apple", "Mango"];
  document.getElementById("demo").innerHTML = fruits.toString();

- var fruits = ["Banana", "Orange", "Apple", "Mango"];
  document.getElementById("demo").innerHTML = fruits.join(" * ");

- OUTPUT :---Banana * Orange * Apple * Mango

- var fruits = ["Banana", "Orange", "Apple", "Mango"];
  fruits.pop();   // Removes the last element ("Mango") from fruits

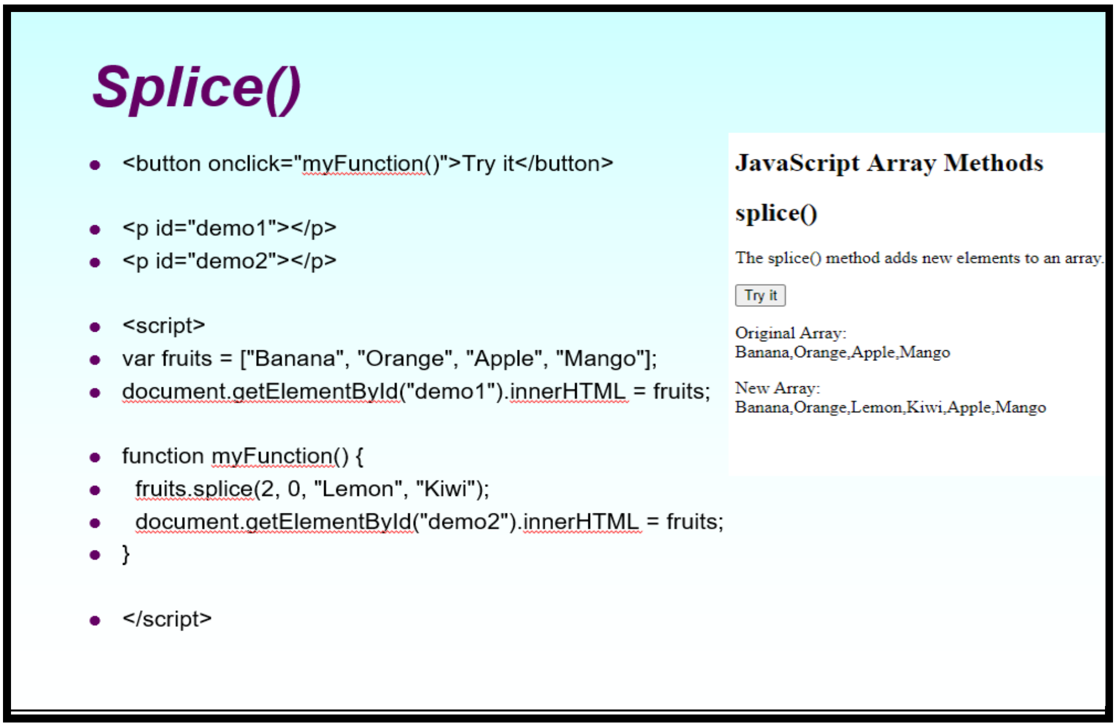
- var fruits = ["Banana", "Orange", "Apple", "Mango"];
  fruits.push("Kiwi");   // Adds a new element ("Kiwi") to fruits

- var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.shift();   // Removes the first element "Banana" from fruits

- var fruits = ["Banana", "Orange", "Apple", "Mango"];
  fruits[0] = "Kiwi";   // Changes the first element of fruits to "Kiwi"

11. Example 14



Splice()

```
• <button onclick="myFunction()">Try it</button>

• <p id="demo1"></p>
• <p id="demo2"></p>

• <script>
• var fruits = ["Banana", "Orange", "Apple", "Mango"];
• document.getElementById("demo1").innerHTML = fruits;

• function myFunction() {
•   fruits.splice(2, 0, "Lemon", "Kiwi");
•   document.getElementById("demo2").innerHTML = fruits;
• }

• </script>
```

JavaScript Array Methods

splice()

The splice() method adds new elements to an array.

Original Array:
Banana,Orange,Apple,Mango

New Array:
Banana,Orange,Lemon,Kiwi,Apple,Mango

Concatenates

```
var myGirls = ["Cecilie", "Lone"];
var myBoys = ["Emil", "Tobias", "Linus"];
var myChildren = myGirls.concat(myBoys);   // Concatenates (joins) myGirls and myBoys
document.getElementById("demo").innerHTML = myChildren
```

Slice

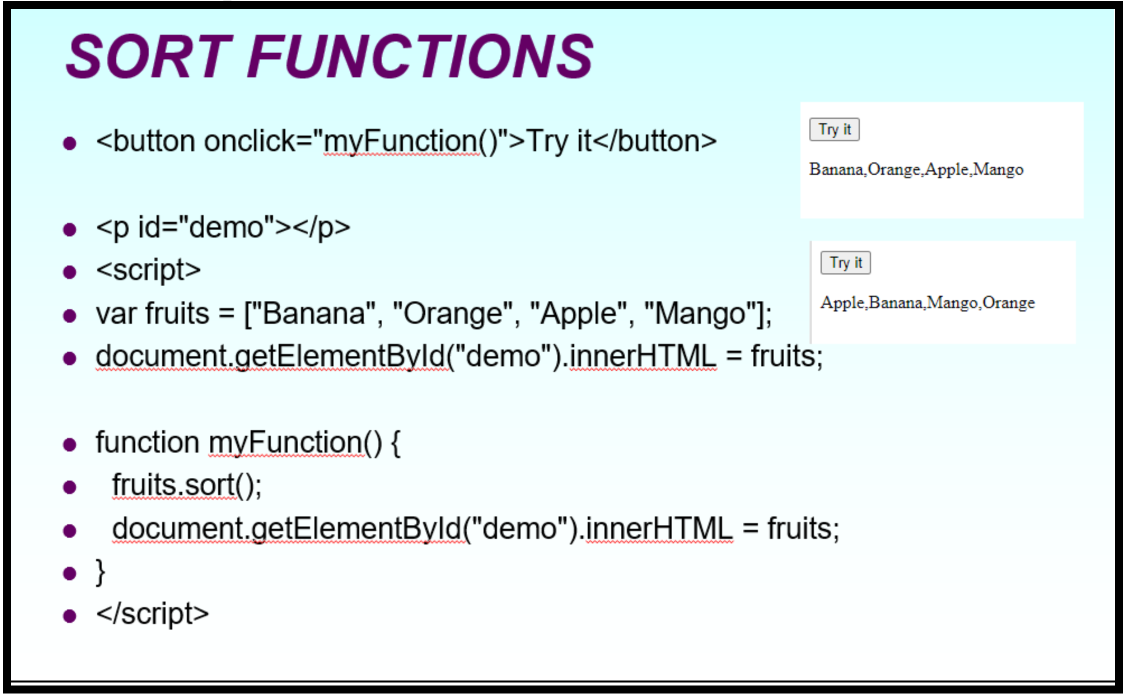
```
var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
var citrus = fruits.slice(1);
document.getElementById("demo").innerHTML = fruits + "<br><br>" + citrus;
```

/****output*****/

- Banana,Orange,Lemon,Apple,Mango

  Orange,Lemon,Apple,Mango

12. Example 15



SORT FUNCTIONS

```
<button onclick="myFunction()">Try it</button>

<p id="demo"></p>
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;

function myFunction() {
  fruits.sort();
  document.getElementById("demo").innerHTML = fruits;
}
</script>
```

Try it

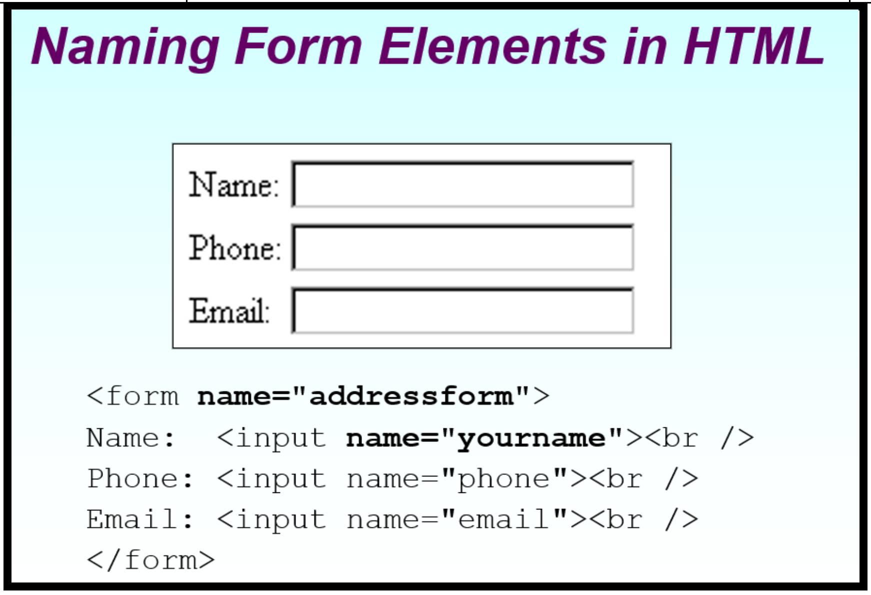
Banana,Orange,Apple,Mango

Try it

Apple,Banana,Mango,Orange

JavaScript Form Validation

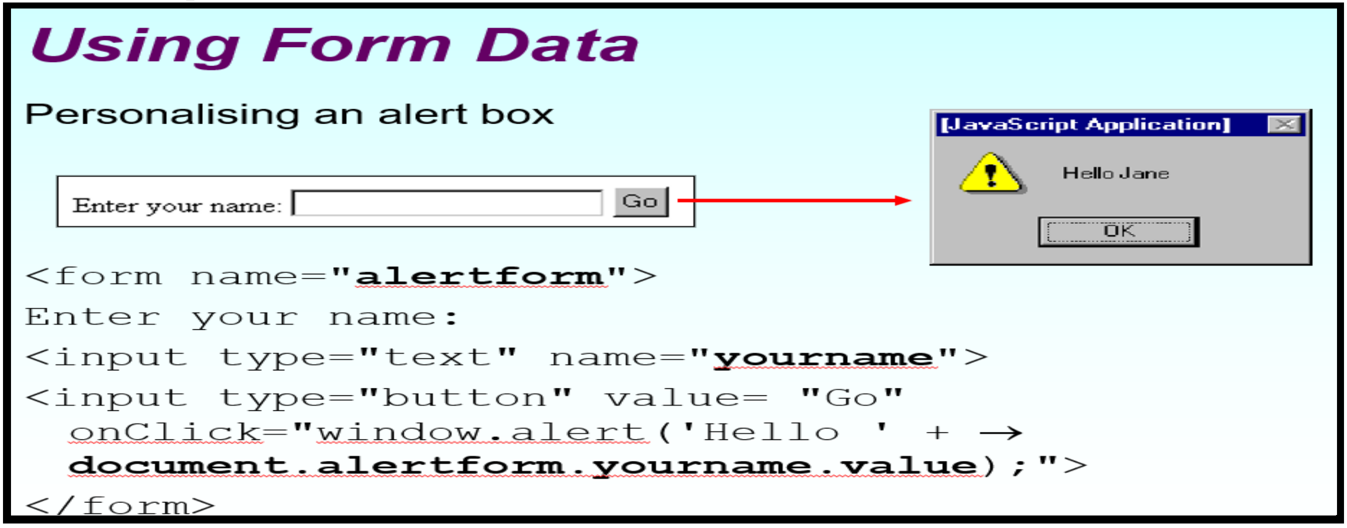
- Form data that typically are checked by a JavaScript could be:
- has the user left required fields empty?
- has the user entered a valid e-mail address?
- has the user entered a valid date?
- has the user entered text in a numeric field?



Naming Form Elements in HTML

```
<form name="addressform">
Name:  <input name="yourname"><br />
Phone: <input name="phone"><br />
Email: <input name="email"><br />
</form>
```

13. Example 16

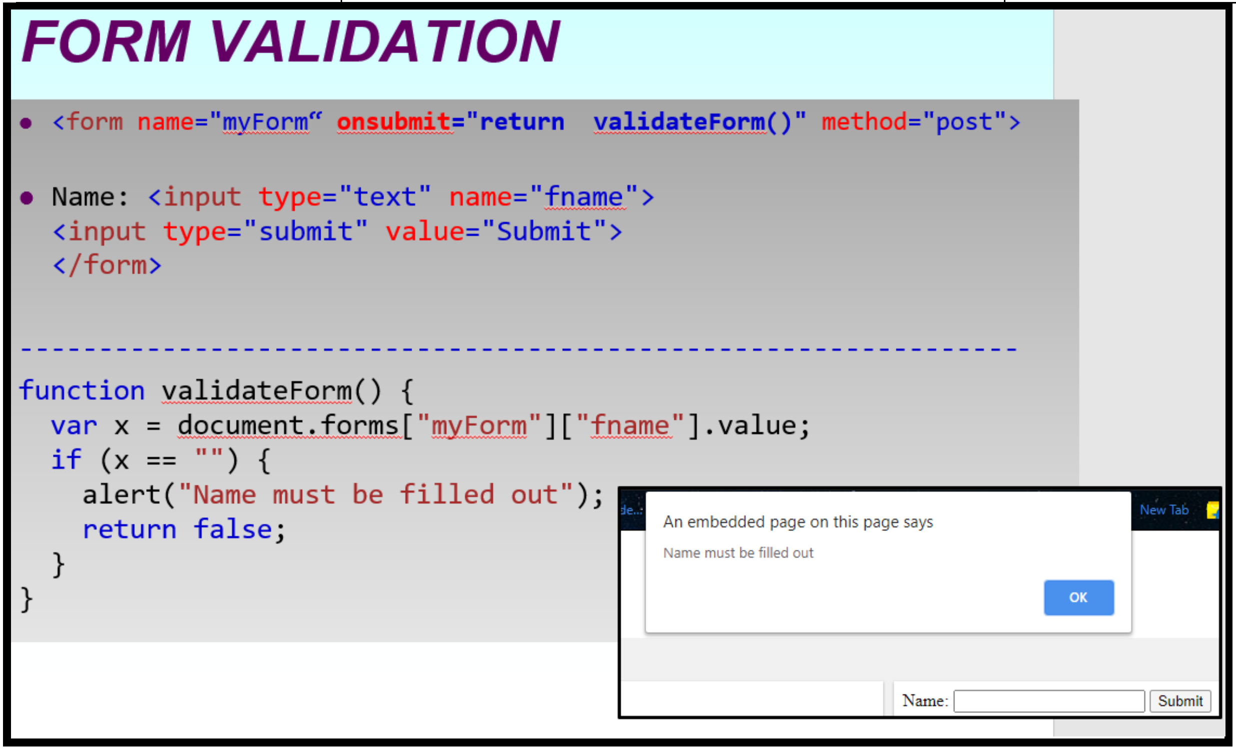


Using Form Data

Personalising an alert box

```
<form name="alertform">
Enter your name:
<input type="text" name="yourname">
<input type="button" value= "Go"
  onClick="window.alert('Hello ' + →
  document.alertform.yourname.value);">
</form>
```

14.Example 17



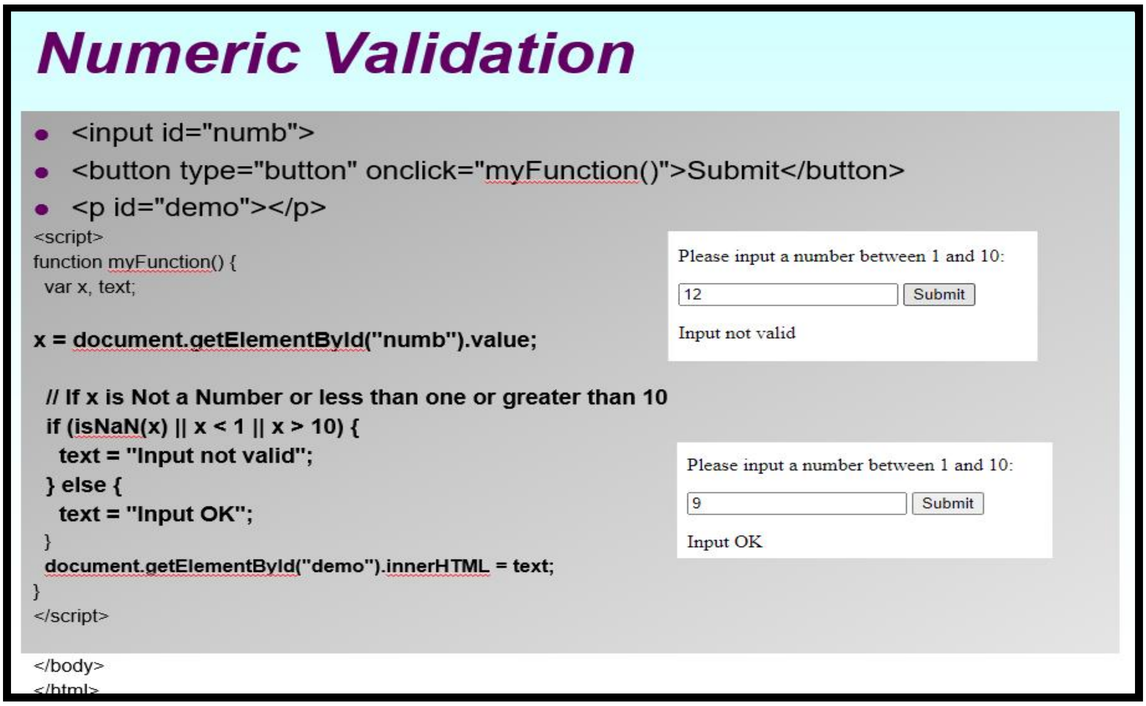
FORM VALIDATION

```
• <form name="myForm“ onsubmit="return  validateForm()" method="post">

• Name: <input type="text" name="fname">
  <input type="submit" value="Submit">
  </form>


-----------------------------------------------------------------

function validateForm() {
  var x = document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
    return false;
  }
}
```

An embedded page on this page says

Name must be filled out

15.Example 18



Numeric Validation

- <input id="numb">
- <button type="button" onclick="myFunction()">Submit</button>
- <p id="demo"></p>

```
<script>
function myFunction() {
  var x, text;

x = document.getElementById("numb").value;

  // If x is Not a Number or less than one or greater than 10
  if (isNaN(x) || x < 1 || x > 10) {
    text = "Input not valid";
  } else {
    text = "Input OK";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>

</body>
</html>
```

Please input a number between 1 and 10:

12 Submit

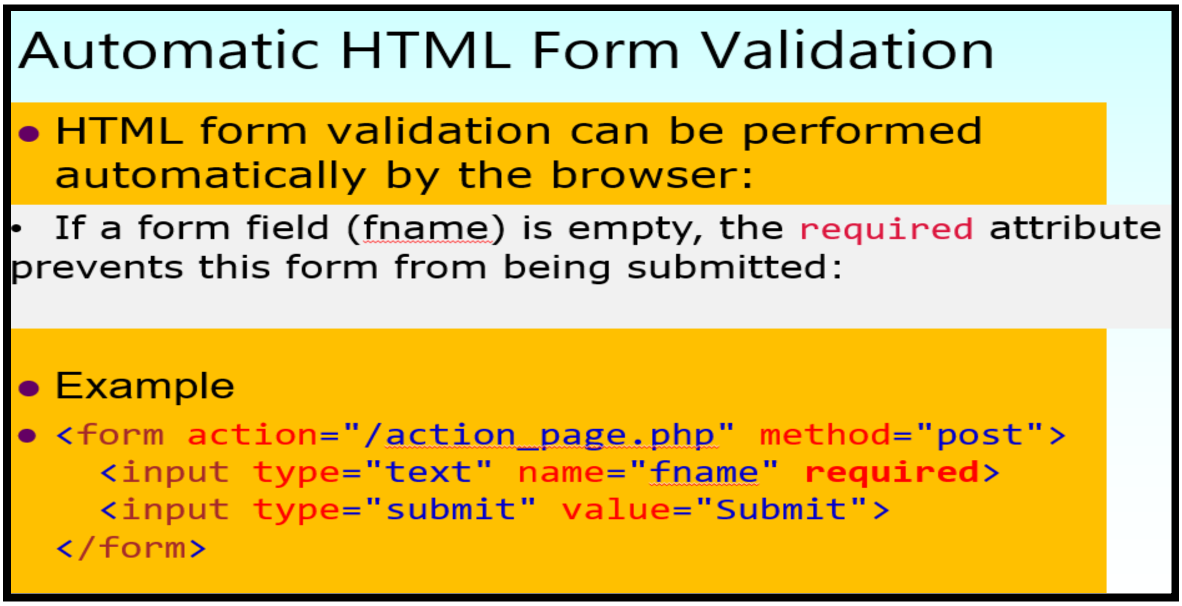
Input not valid

Please input a number between 1 and 10:

9 Submit

Input OK

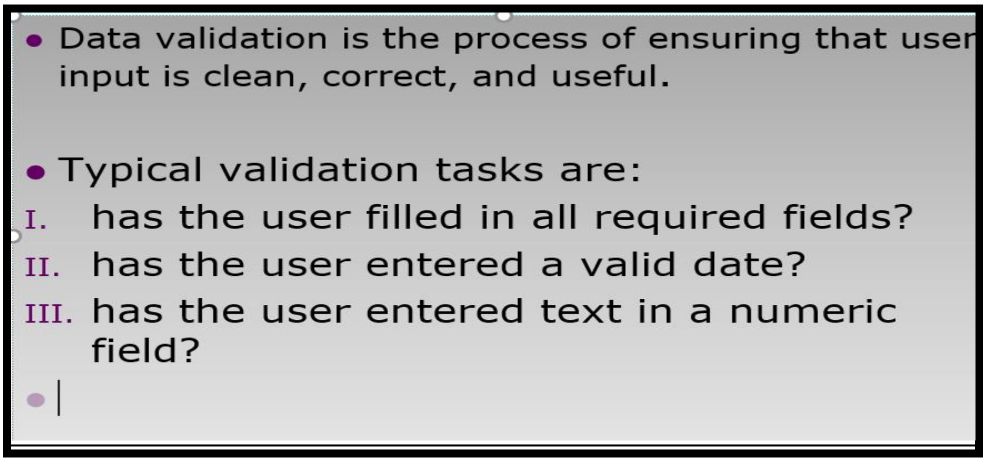
HTML form validation



Automatic HTML Form Validation

- HTML form validation can be performed automatically by the browser:
- If a form field (fname) is empty, the required attribute prevents this form from being submitted:
- Example
- <form action="/action_page.php" method="post">
    <input type="text" name="fname" required>
    <input type="submit" value="Submit">
  </form>

Data Validation



- Data validation is the process of ensuring that user input is clean, correct, and useful.
- Typical validation tasks are:

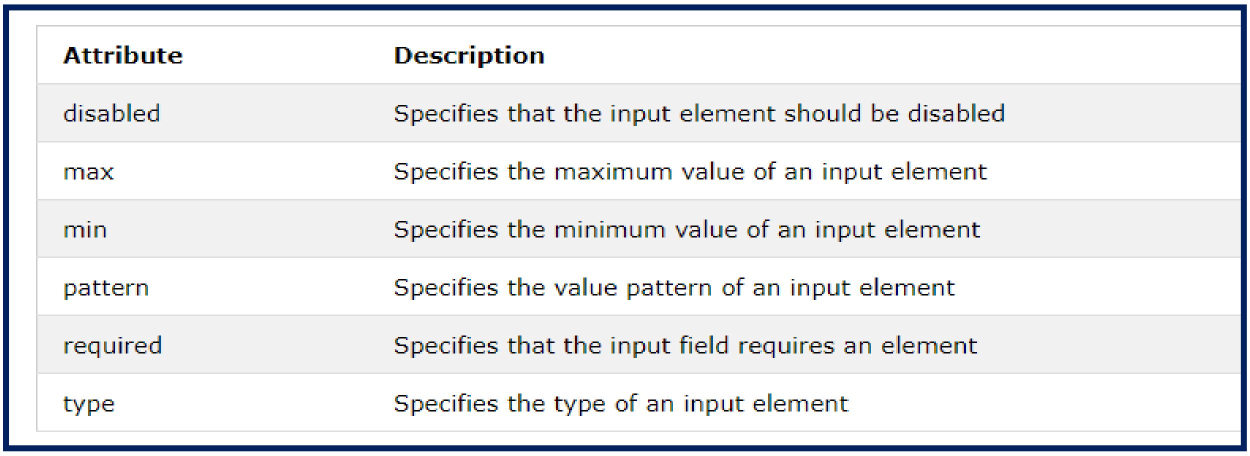
I. has the user filled in all required fields?
II. has the user entered a valid date?
III. has the user entered text in a numeric field?

HTML Validations

- HTML constraint validation is based on:

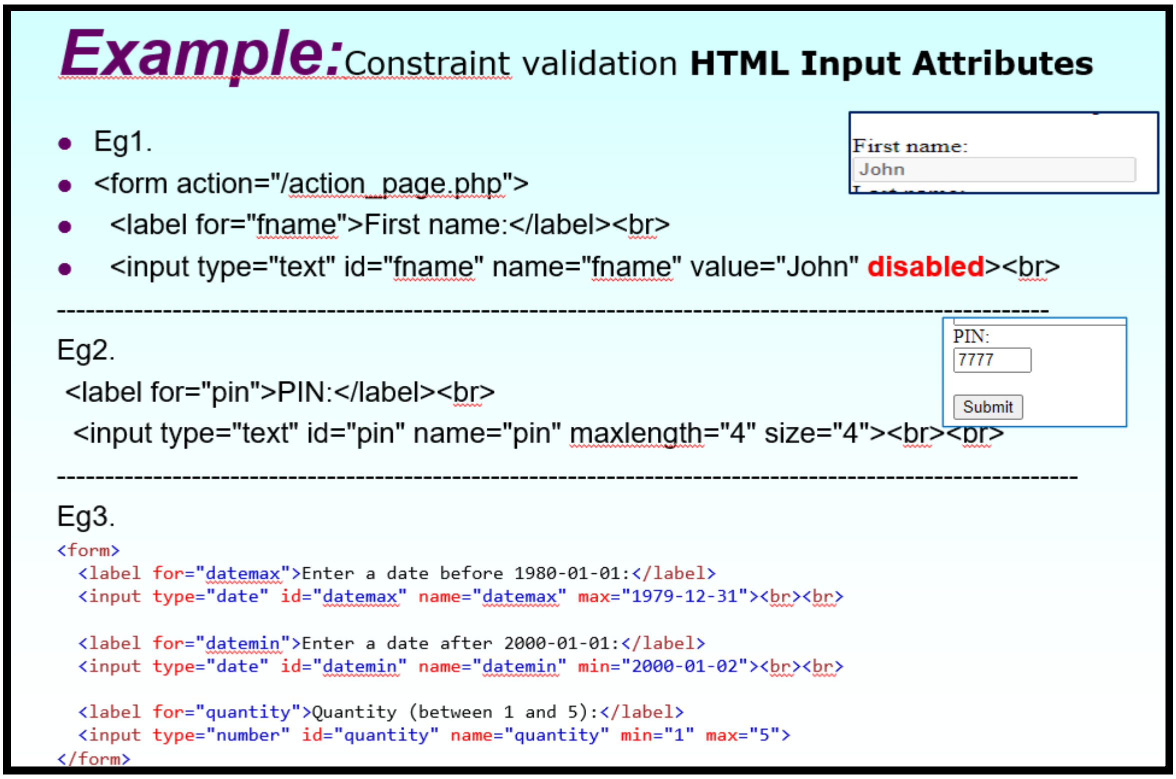
I. Constraint validation **HTML Input Attributes**
II. Constraint validation **CSS Pseudo Selectors**
III. Constraint validation **DOM Properties and Methods**

Constraint validation **HTML Input Attributes**



Attribute	Description
disabled	Specifies that the input element should be disabled
max	Specifies the maximum value of an input element
min	Specifies the minimum value of an input element
pattern	Specifies the value pattern of an input element
required	Specifies that the input field requires an element
type	Specifies the type of an input element

16. Example 19



Example: Constraint validation **HTML Input Attributes**

- Eg1.
- <form action="/action_page.php">
- <label for="fname">First name:</label>
- <input type="text" id="fname" name="fname" value="John" **disabled**>

First name:
John

Eg2.

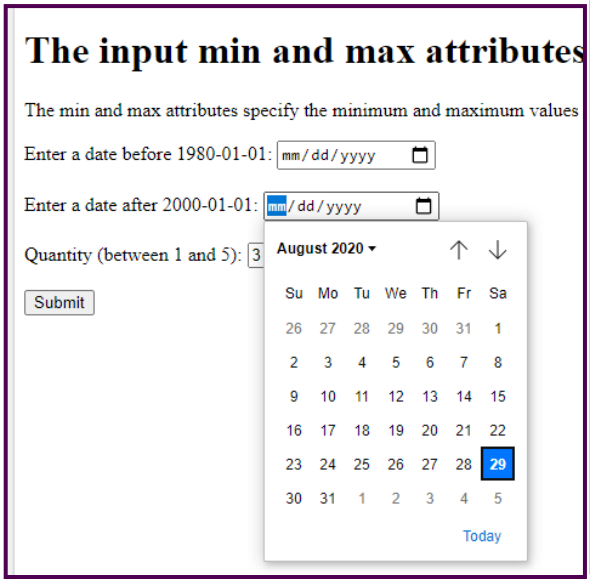
```
<label for="pin">PIN:</label><br>
 <input type="text" id="pin" name="pin" maxlength="4" size="4"><br><br>
```

PIN:
7777
Submit

Eg3.

```
<form>
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>

  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>

  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```

Output



The input min and max attributes

The min and max attributes specify the minimum and maximum values

Enter a date before 1980-01-01: mm/dd/yyyy

Enter a date after 2000-01-01: mm/dd/yyyy

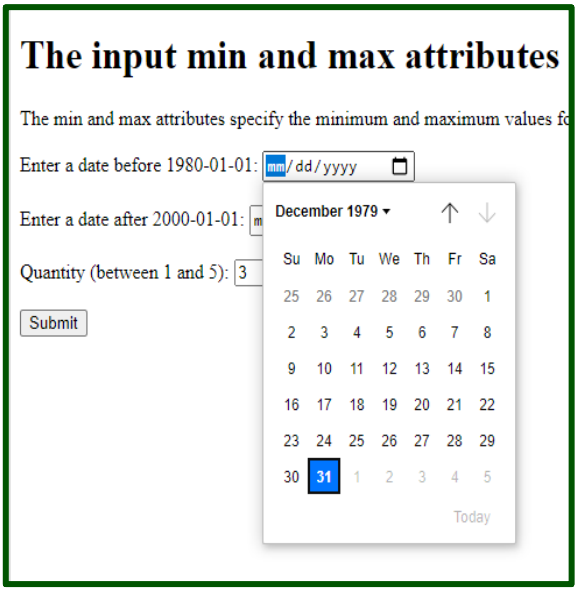
Quantity (between 1 and 5): 3

Submit

August 2020

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Today



The input min and max attributes

The min and max attributes specify the minimum and maximum values fo

Enter a date before 1980-01-01: mm/dd/yyyy

Enter a date after 2000-01-01: m

Quantity (between 1 and 5): 3

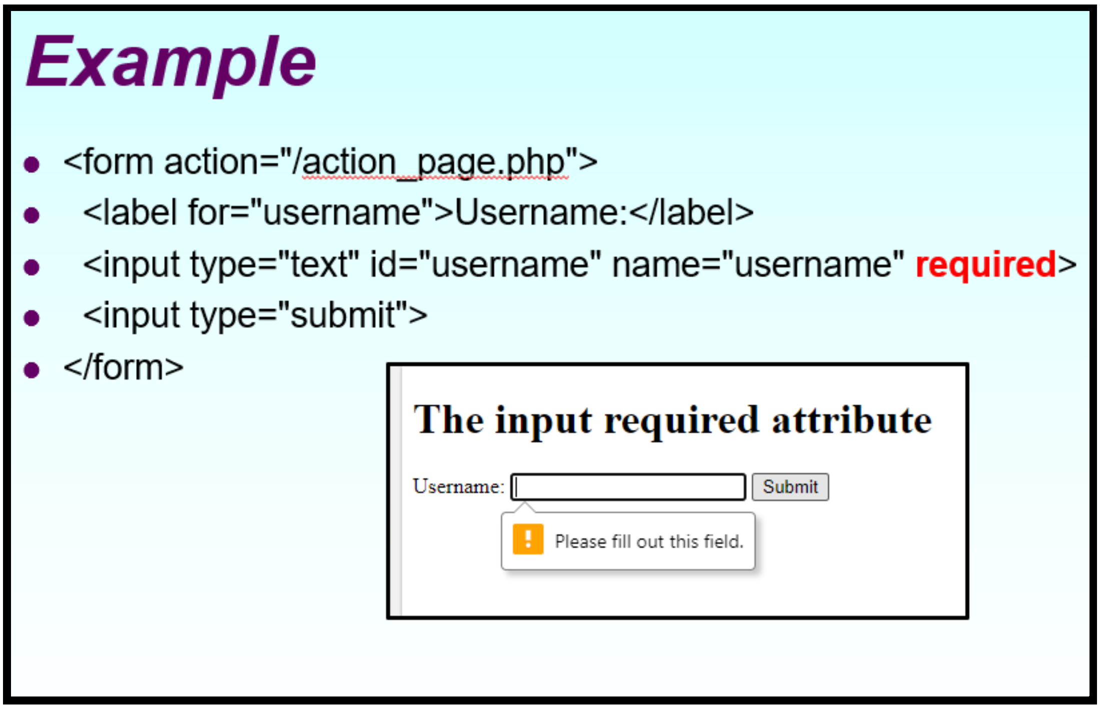
Submit

December 1979

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Today

17.Example 20



Example

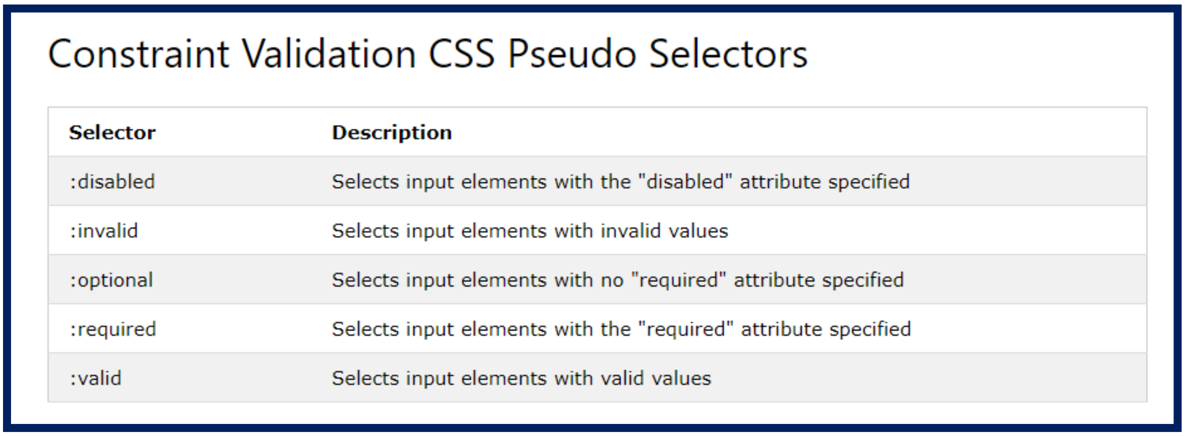
- <form action="/action_page.php">
-   <label for="username">Username:</label>
-   <input type="text" id="username" name="username" **required**>
-   <input type="submit">
- </form>

The input required attribute

Username: Submit

Please fill out this field.

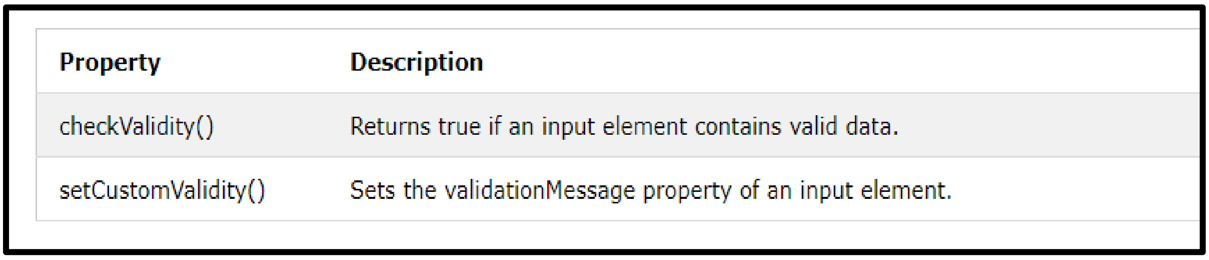
II.Constraint validation **CSS Pseudo Selectors**



Constraint Validation CSS Pseudo Selectors

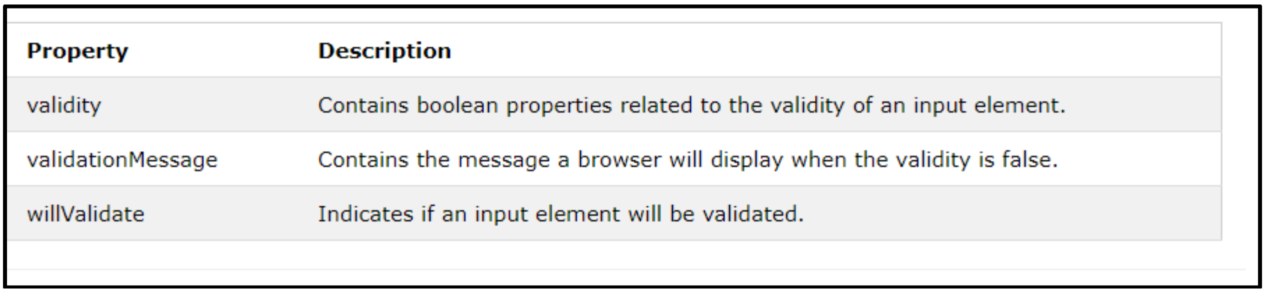
Selector	Description
:disabled	Selects input elements with the "disabled" attribute specified
:invalid	Selects input elements with invalid values
:optional	Selects input elements with no "required" attribute specified
:required	Selects input elements with the "required" attribute specified
:valid	Selects input elements with valid values

III.Constraint Validation DOM Methods



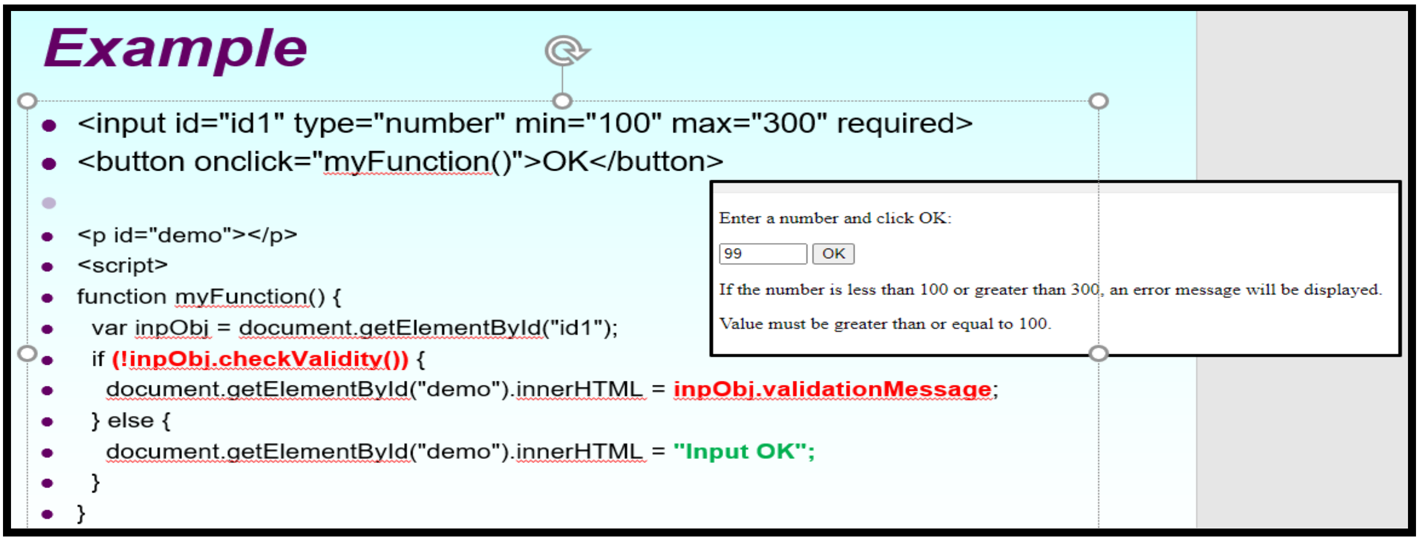
Property	Description
checkValidity()	Returns true if an input element contains valid data.
setCustomValidity()	Sets the validationMessage property of an input element.

Constraint Validation DOM Properties



Property	Description
validity	Contains boolean properties related to the validity of an input element.
validationMessage	Contains the message a browser will display when the validity is false.
willValidate	Indicates if an input element will be validated.

18. Example 21



Example

```
<input id="id1" type="number" min="100" max="300" required>
<button onclick="myFunction()">OK</button>

<p id="demo"></p>
<script>
function myFunction() {
  var inpObj = document.getElementById("id1");
  if (!inpObj.checkValidity()) {
    document.getElementById("demo").innerHTML = inpObj.validationMessage;
  } else {
    document.getElementById("demo").innerHTML = "Input OK";
  }
}
```

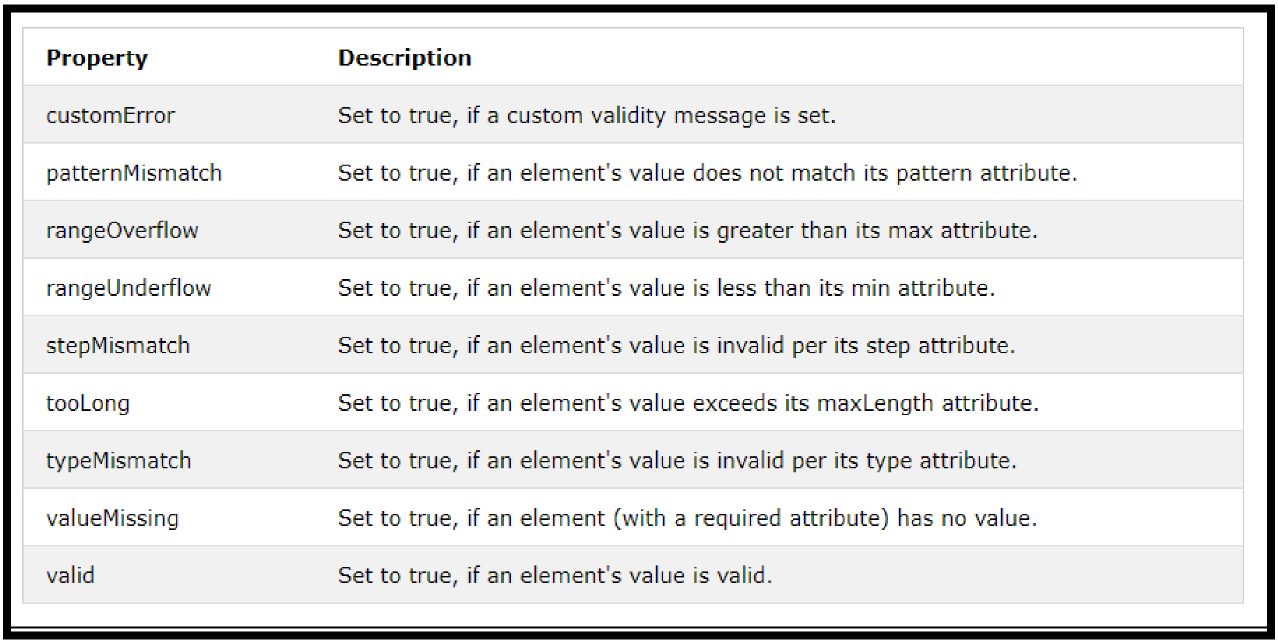
Enter a number and click OK:

99 OK

If the number is less than 100 or greater than 300, an error message will be displayed.

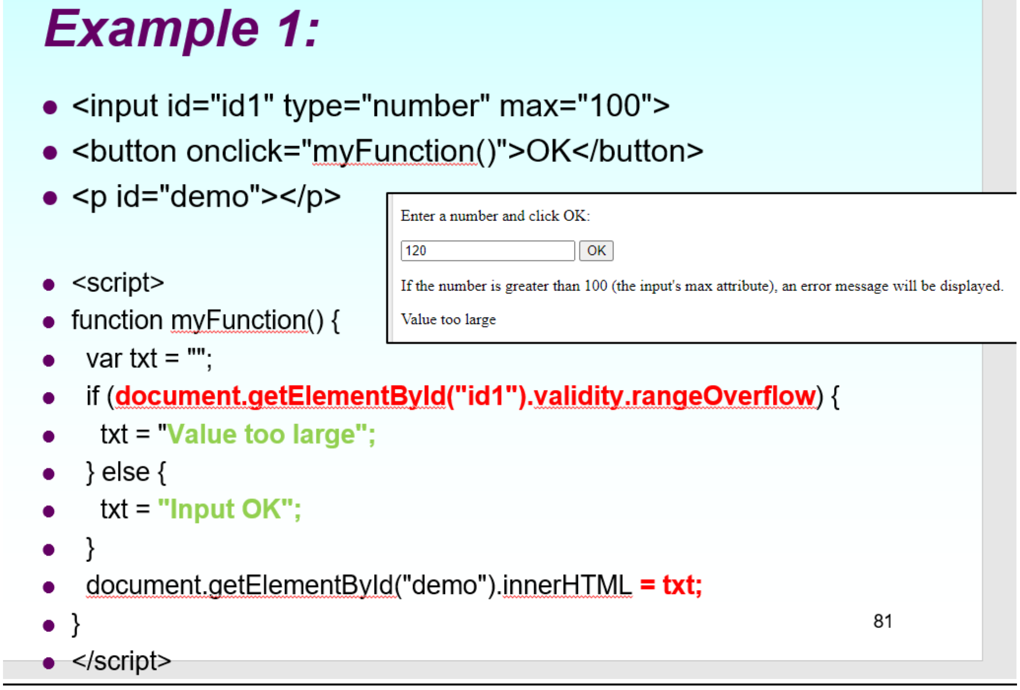
Value must be greater than or equal to 100.

Validity Properties



Property	Description
customError	Set to true, if a custom validity message is set.
patternMismatch	Set to true, if an element's value does not match its pattern attribute.
rangeOverflow	Set to true, if an element's value is greater than its max attribute.
rangeUnderflow	Set to true, if an element's value is less than its min attribute.
stepMismatch	Set to true, if an element's value is invalid per its step attribute.
tooLong	Set to true, if an element's value exceeds its maxLength attribute.
typeMismatch	Set to true, if an element's value is invalid per its type attribute.
valueMissing	Set to true, if an element (with a required attribute) has no value.
valid	Set to true, if an element's value is valid.

19. Example 22



Example 1:

```
• <input id="id1" type="number" max="100">
• <button onclick="myFunction()">OK</button>
• <p id="demo"></p>

• <script>
• function myFunction() {
•   var txt = "";
•   if (document.getElementById("id1").validity.rangeOverflow) {
•     txt = "Value too large";
•   } else {
•     txt = "Input OK";
•   }
•   document.getElementById("demo").innerHTML = txt;
• }
• </script>
```

Enter a number and click OK:

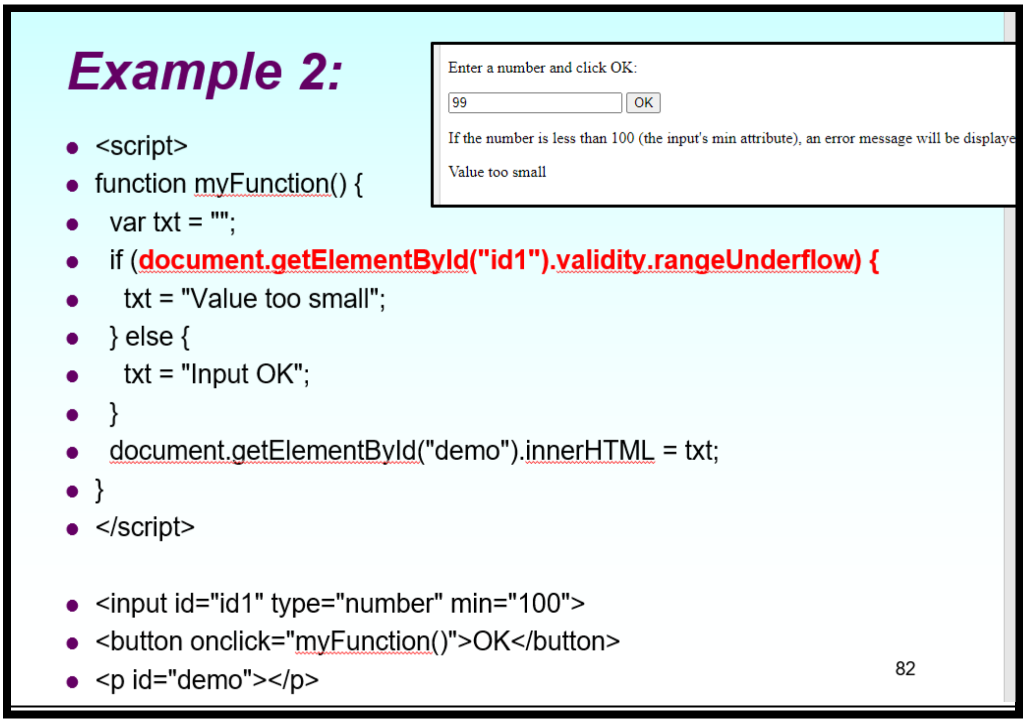
120 OK

If the number is greater than 100 (the input's max attribute), an error message will be displayed.

Value too large

81

20. Example 23



Example 2:

Enter a number and click OK:

99 OK

If the number is less than 100 (the input's min attribute), an error message will be displaye

Value too small

```
<script>
function myFunction() {
  var txt = "";
  if (document.getElementById("id1").validity.rangeUnderflow) {
    txt = "Value too small";
  } else {
    txt = "Input OK";
  }
  document.getElementById("demo").innerHTML = txt;
}
</script>

<input id="id1" type="number" min="100">
<button onclick="myFunction()">OK</button>
<p id="demo"></p>
```

82

PART B

Functions , Events and Arrays

1.Write a Program to display Exponent of a two numbers using a JavaScript functions.

2.Write a Program to display a Moon weight and input will be actual earth weight .

3.Write a java script program to create an Array for the students whose length is 5 . and display with ordered list.

4.push the new students into an array and display the new array list on MouseOver Event on paragraph tags.

5.Splice ,new 3 students into an array on image Onclick Event.

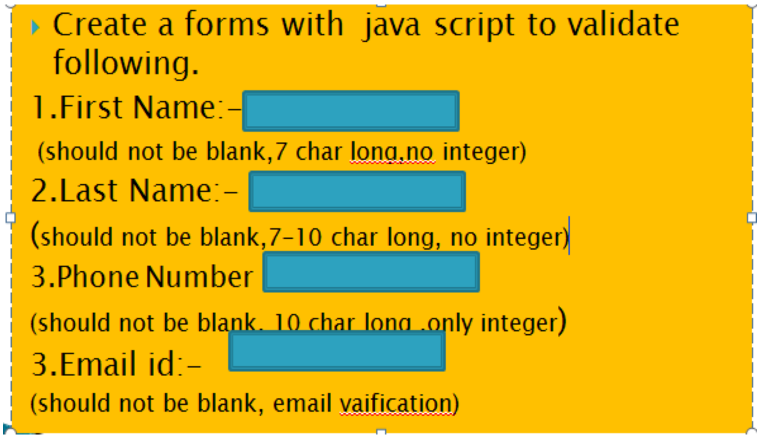
6.concate the two array called girls and boys students on division Onclick Event.

7.slice the students array into 3 halves using slice function of java Script.

8.<u>Validations 1</u>

a. Write a Program to Accept <u>User Name </u>, <u>User Age </u>,<u>User Phone Number.</u>
b. Check No user input is left or blank.
c. User Name length should be 10 char long.
d. User Age should be more than 20 and less than 90.
e. User Phone length should be 10 digit long.
f. Note:- User inner html for displaying an error message.

9.Form Validations 2



- Create a forms with java script to validate following.

1.First Name:-

(should not be blank,7 char long,no integer)

2.Last Name:-

(should not be blank,7-10 char long, no integer)

3.Phone Number

(should not be blank, 10 char long ,only integer)

3.Email id:-

(should not be blank, email vaification)

Code:-

[Note: - Execute program from 1-2 and paste executed html code]

Input and Output:-

[Note: - Paste Output of Code from 1-2 [use print scrn button]]

observation and Learning:-

[Note: - Write your observation and learning]

Conclusion:-

[Note: - Write conclusion in your own word]

Questions:-

1. What is meaning of inner html?
2. What is the meaning of DOM?
3. Why Javascript is need to use on HTML web page?