Marwadi University

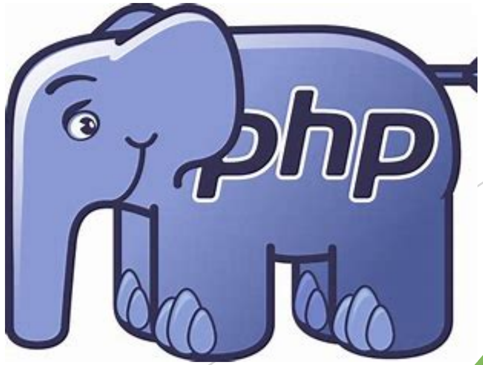
Web Application Development (PHP)

Unit – 5

Apply JavaScript functionalities, Validations on forms

Prof. Hardik Chavda

FOCA,MU



- Conditional statements
- Loops
- Dialog boxes
- DOM
- User Define Function
- Built in Functions
- Form Validations

JavaScript

- JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages.
- It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.
- It was introduced in the year **1995** for adding programs to the webpages in the **Netscape Navigator browser**.
- Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time.
- The traditional website uses js to provide several forms of interactivity and simplicity

Conditional statements and loops

- The **JavaScript if-else statement** is used *to execute the code whether condition is true or false*. There are three forms of if statement in JavaScript.
  1. If Statement
  2. If else statement
  3. if else if statement

```
<script>
var a=20;
if(a>10){
 document.write("value of a is greater than
 10");
}
</script>
```

```
<script>
var a=20;
if(a%2==0){
        document.write("a is even number");
}
else{
        document.write("a is odd number");
}
</script>
```

- The **JavaScript switch statement** is used *to execute one code from multiple expressions*.
- It is just like else if statement that we have learned in previous page.
- But it is convenient than *if..else..if* because it can be used with numbers, characters etc.

```
<script>
    var grade='B';
    var result;
    switch(grade){
        case 'A':
            result="A Grade";
            break;
        case 'B':
            result="B Grade";
            break;
        case 'C':
            result="C Grade";
            break;
        default:
            result="No Grade";
    }
    document.write(result);
</script>
```

Loops

- The **JavaScript loops** are used *to iterate the piece of code* using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.
- There are four types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop

```
<script>
for (i=1; i<=5; i++)
{
     document.write(i + "<br/>")
}
</script>
```

```
<script>
var i=1;
while (i<=5)
{
document.write(i + "<br/>");
i++;
}
</script>
```

```
<script>
var i=1;
do{
document.write(i + "<br/>");
i++;
}while (i<=5);
</script>
```

```
1
2
3
4
5
```

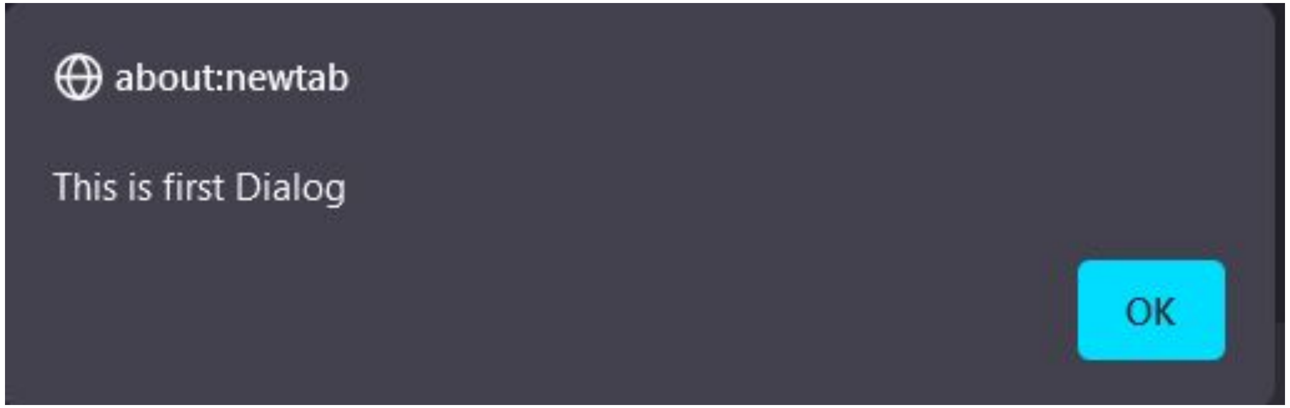
Dialog boxes

- JavaScript supports three important types of dialog boxes.
- These dialog boxes can be used to raise and alert, or to get confirmation on any input or to have a kind of input from the users.
- Here we will discuss each dialog box one by one.

Alert Dialog Box

- An alert dialog box is mostly used to give a warning message to the users.

```
<script>
    alert ("This is first Dialog");
</script>
```



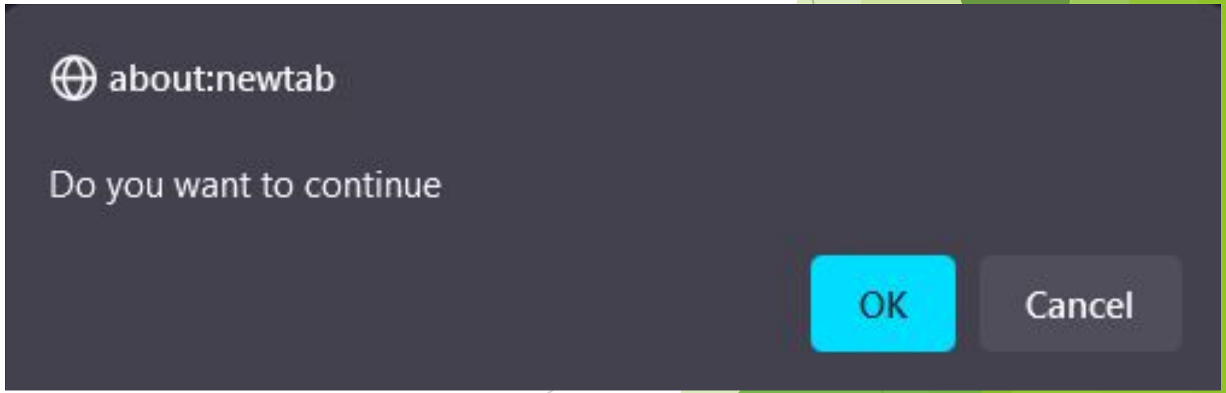
about:newtab

This is first Dialog

Confirmation Dialog

- A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: **OK** and **Cancel**.
- If the user clicks on the OK button, the window method **confirm()** will return true. If the user clicks on the Cancel button, then **confirm()** returns false. You can use a confirmation dialog box as follows.

```
<script>
confirm("Do you want to continue ?");
</script>
```



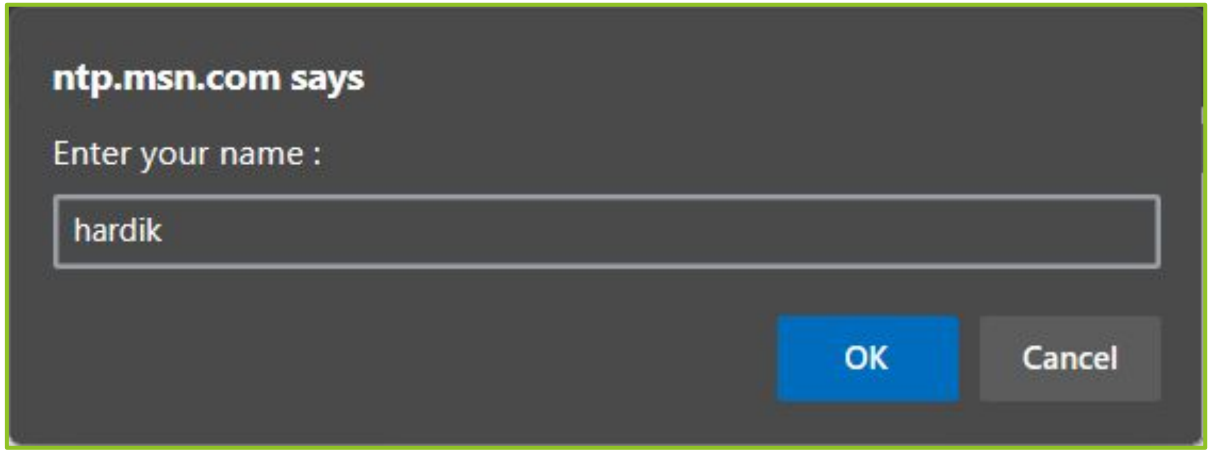
about:newtab

Do you want to continue

Prompt Dialog Box

- The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus, it enables you to interact with the user. The user needs to fill in the field and then click OK.
- This dialog box is displayed using a method called **prompt()** which takes two parameters:
  - (i) a label which you want to display in the text box and
  - (ii) a default string to display in the text box.
- This dialog box has two buttons: **OK** and **Cancel**. If the user clicks the OK button, the window method **prompt()** will return the entered value from the text box. If the user clicks the Cancel button, the window method **prompt()** returns **null**.

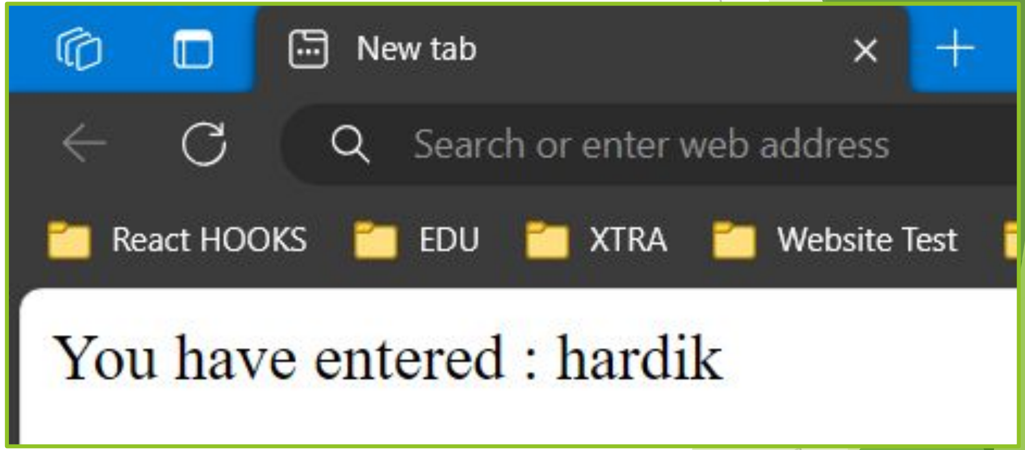
```
<script>
var data = prompt("Enter your name : ", “hardik");
document.write("You have entered : " + data);
</script>
```



ntp.msn.com says

Enter your name :

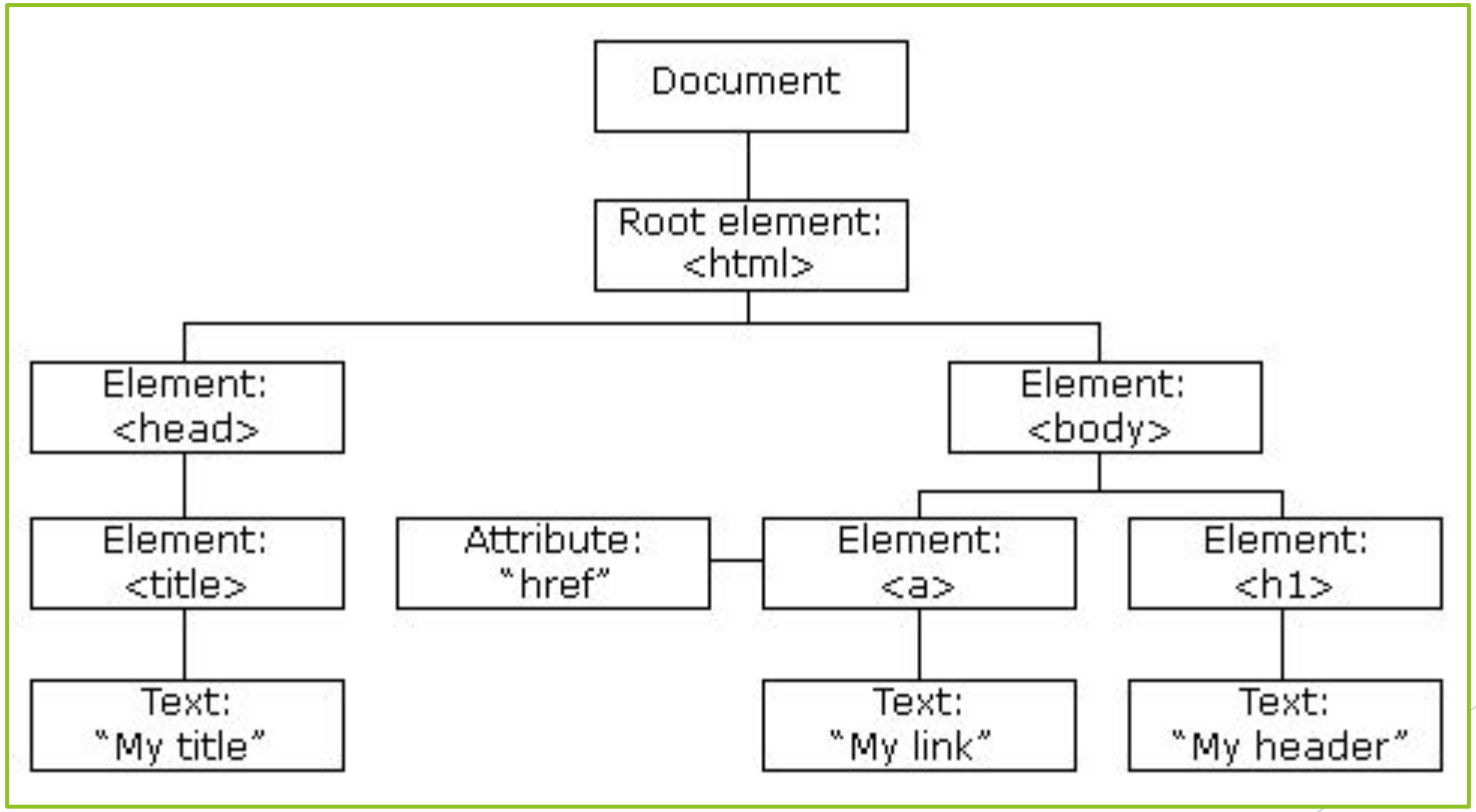
hardik



You have entered : hardik

DOM

- With the HTML DOM, JavaScript can access and change all the elements of an HTML document.
- When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.
- The **HTML DOM** model is constructed as a tree of **Objects**



What is the DOM?

- The DOM is a W3C (World Wide Web Consortium) standard.
- The DOM defines a standard for accessing documents:
- *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*
- 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

Finding HTML Elements

Method	Description
document.getElementById(id)	Find an element by element id
document.getElementsByTagName(name)	Find elements by tag name
document.getElementsByClassName(name)	Find elements by class name

Changing HTML Elements

Property	Description
element.innerHTML = new html content	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.style.property = new style	Change the style of an HTML element

User Define Function

- A **function** is a set of statements that take inputs, do some specific computation, and produce output.
- The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can call that function.
- Before, using a user-defined function in JavaScript we have to create one. We can use the above syntax to create a function in JavaScript.
- A function definition is sometimes also termed a function declaration or function statement.
- Below are the rules for creating a function in JavaScript:
  - Every function should begin with the keyword *function* followed by,
  - A user-defined function name that should be unique,
  - A list of parameters enclosed within parentheses and separated by commas,
  - A list of statements composing the body of the function enclosed within curly braces {}.

- Example

```
function calcAddition(number1, number2) {
    return number1 + number2;
}
console.log(calcAddition(6,9));
```

- OP:15
- In the above example, we have created a function named calcAddition,
- This function accepts two numbers as parameters and returns the addition of these two numbers.
- Accessing the function with just the function name without () will return the function object instead of the function result.

Built in Functions

- A JavaScript **method** is a property containing a **function definition**. In other words, when the data stored on an object is a function we call that a method.
- To differentiate between properties and methods, we can think of it this way: ***A property is what an object has, while a method is what an object does.***
- Since JavaScript methods are actions that can be performed on objects, we first need to have objects to start with. There are several objects built into JavaScript which we can use.
- Useful Built-in methods Date
- Math
- Date
- Strings
- Arrays
- Objects

Form Validations

- It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.
- 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.
- Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

```
<script>
function validateform(){

var name=document.myform.name.value;
var password=document.myform.password.value;

if (name==null || name=="")
{
  alert("Name can't be blank");
  return false;
}
else if(password.length<6)
{
  alert("Password must be at least 6 characters long.");
  return false;
  }
}
</script>
```

```
<body>
<form name="myform" method="post" action="abc.php" onsubmit="return validateform()" >

Name: <input type="text" name="name">
Password: <input type="password" name="password">

<input type="submit" value="register">

</form>
</body>
```