Web Technologies

Lecture 17

Introduction to JavaScript

Summary of Previous Lecture

- **Why User Interface should look Good?**
  - **Guidelines and Principles of User Interface Design**
  - **Principles of Screen Design**
  - **Interface Design Goals**
  - **Interaction Styles**
  - **Types of Interfaces**
  - **What are the Advantages of Style Guidelines?**
  - **What are Advantages of Good Interface?**
  - **What are Disadvantages of Bad Interface?**

Summary of Previous Lecture

- **What are the Elements of Visual Design?**
  - **Font**
    - **Six Typographic Terms**
    - **Font Size**
    - **Types of Fonts**
    - **Proportional Vs. Fixed width Fonts**
    - **Case of Text**
  - **Layout**
  - **Color**
    - **Guidelines for Color Use**
  - **Labels**

Today's Lecture Outline

- What is JavaScript?
- Embedding JavaScript with HTML
- JavaScript conventions
- Variables in JavaScript
- JavaScript operators
- Input output in JavaScript
- JavaScript functions
- Conditional Statements
- Looping Statements

1. JavaScript

- **JavaScript is a client-side scripting language**
- **JavaScript was designed to add interactivity to HTML pages**
- **JavaScript is used in millions of Web pages to improve the design, validate forms, detect browsers, create cookies, and much more**

1. JavaScript….

- **JavaScript is an interpreted language (means that scripts execute without preliminary compilation)**
- **JavaScript is usually embedded directly into HTML pages**
- **Everyone can use JavaScript without purchasing a license**

1. JavaScript….

- JavaScript is the programming language of HTML and the Web.
- JavaScript is the most popular scripting language on the internet, and works in all major browsers, such as Internet Explorer, Mozilla, Firefox, Netscape, Opera

1.1 JavaScript: Common Uses

- JavaScript gives HTML designers a programming tool
  - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax!
  - Almost anyone can put small "snippets" of code into their HTML pages
- JavaScript can react to events
  - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- JavaScript can read and write HTML elements
  - A JavaScript can read and change the content of an HTML element

1.1 JavaScript: Common Uses

- **JavaScript can be used to validate data**
  - A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing
- **JavaScript can be used to detect the visitor's browser**
- **JavaScript can be used to create cookies**
  - A JavaScript can be used to store and retrieve information on the visitor's computer

1.1 JavaScript: Common Uses

- **Cookie**
  - **A message given to a Web browser** by a Web server. The browser stores the message in a text file. **The message is then sent back** to the server each time the browser requests a page from the server.

2. Embedding JavaScript in HTML

- There are two methods to embed JavaScript in to HTML code
  - Internal Script: directly written in HTML code
  - External Script: written in separate file
- <script> tag is used to tell the browser that a script follows

2.1 Internal Scripts

- The <SCRIPT> tag is used to embed JavaScript code in HTML documents

```
<SCRIPT LANGUAGE="JavaScript">
        // JavaScript Statements...
</SCRIPT>
```

- JavaScript can be placed anywhere between <HTML> and </HTML> tags
- two possibilities are the <HEAD>…</HEAD> portion and the <BODY>…</BODY> portion

2.1 Internal Scripts…

Example:

```
<html>
<head>
    <title>Using Multiple scripts</title>
    <script language="javascript">
        // JavaScript statements...
    </script>

    <script language="javascript">
        // JavaScript statements...
    </script>
</head>

<body>
    <h1> This is another script...</h1>

    <script language="javascript">
        // JavaScript statements...
    </script>
</body>
</html>
```

2.2 External Script

- We place script in a separate file and include this in HTML code
- SRC attribute of the <SCRIPT> is used to include the external JavaScript file in HTML

```
<script src="myscripts.js">…</script>
```

- External Scripts are useful when you have lengthy scripts
- External Scripts improves the readability

2.2 External Script

- **External JavaScript Advantages**
- **Placing JavaScripts in external files has some advantages:**
  - **It separates HTML and code**
  - **It makes HTML and JavaScript easier to read and maintain**
  - **Cached JavaScript files can speed up page loads**

3. JavaScript Conventions

- **Using the Semicolon**
  - **With traditional programming languages, like C, C++ and Java, each code statement has to end with a semicolon (;).**
  - **Many programmers continue this habit when writing JavaScript, but in general, semicolons are optional! However, semicolons are required if you want to put more than one statement on a single line.**

```
<script language="javascript">
    document.write("Hello"); alert("Good bye") // Two Statements
    document.write("Hello")                    // No Semicolon
    alert("Good bye")                          // No Semicolon
    document.write("Hello");                   // Semicolon
    alert("Good bye");                         // Semicolon
</script>
```

3. JavaScript Conventions

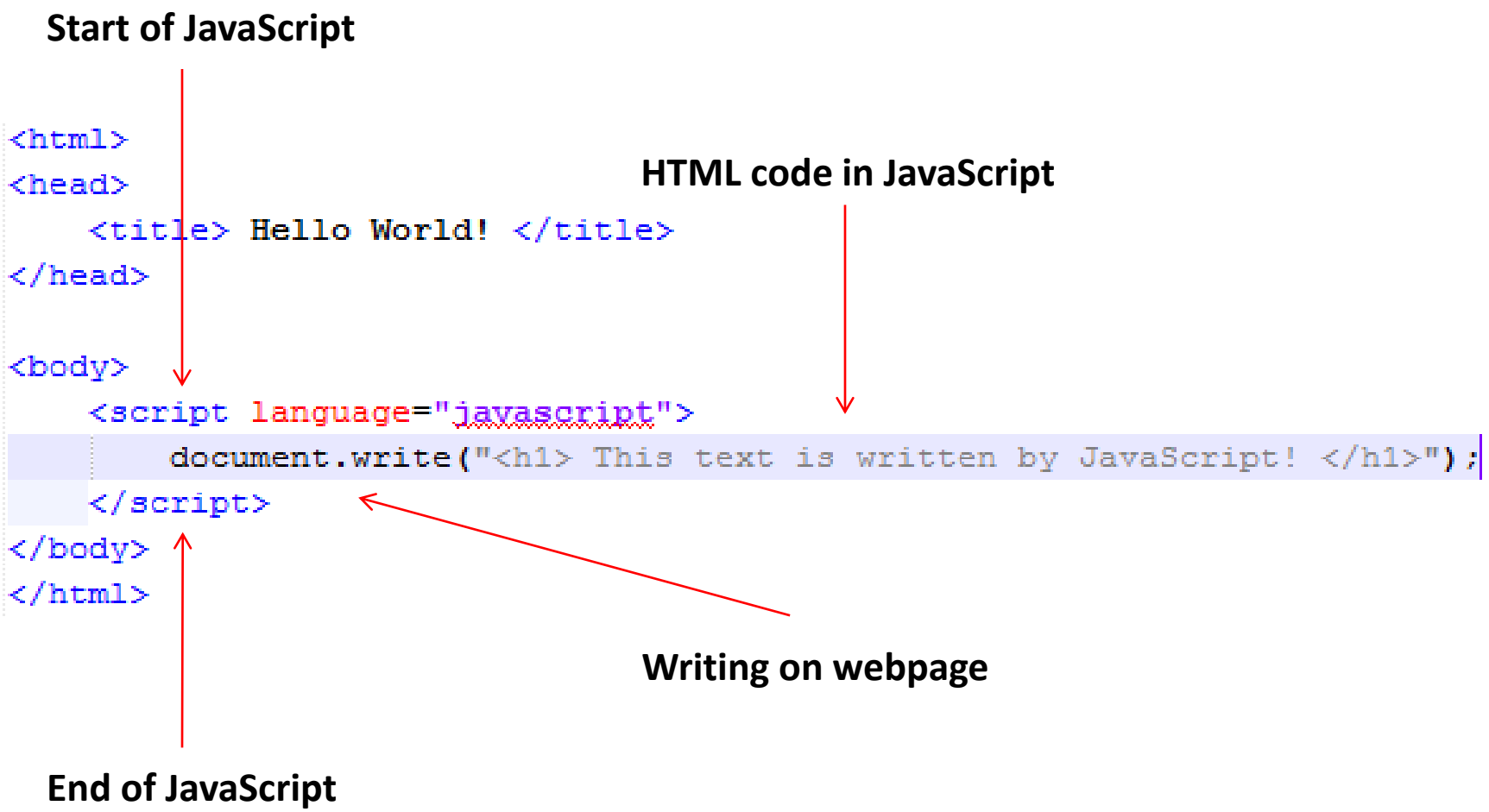
- **Case Sensitivity**
  - JavaScript is a case sensitive language
  - Variable names lastname and LastName are different.
- **Comments**
  - Single Line: //
  - Multiple lines: /* */

```
<script language="javascript">
    // This is single line comment
    and
    /*  This is
        multiple lines
        comment
    */
</script>
```

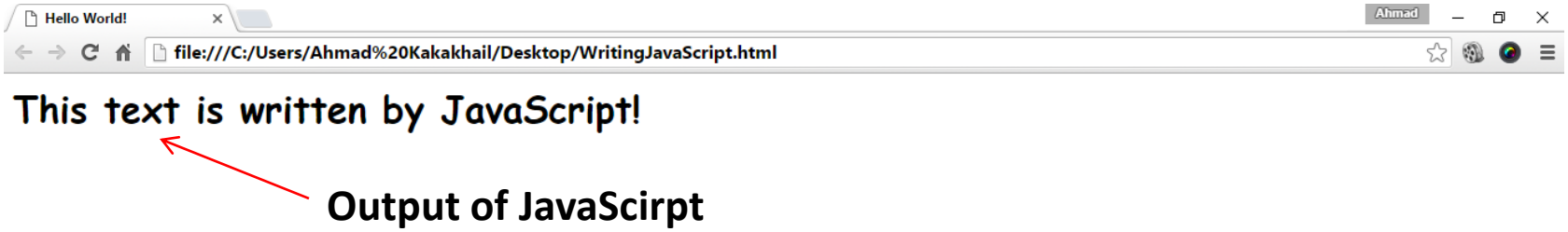
3. JavaScript Conventions

- **Using Quotes**
  - **You can use both type of quotes that is:**
    - **Single quotes: ‘something inside single quotes’**
    - **Double quotes: “something inside double quotes”**
  - **For Example:**
    - **document.write(“Hello World”)**
    - **document.write(“Hello World”)**

4. Writing JavaScript



4. Writing JavaScript…



4.1 Variables in JavaScript

- Variable is the name of a memory location which holds the data of a certain type (data types)
- There are four common data types in JavaScript
  - Numbers
  - Strings
  - Boolean
  - null values
- JavaScript is a loosely typed language which means you do not have to explicitly write the datatype, it can pick the datatype by itself.

4.1 Variables in JavaScript…

- The word "var" is used to declare a variable
  - var LastName = "Smith"
  - var AccountNumber = 1111
- Variable Naming
  - First character can not be a digit
  - Other characters may be digits, letters or underscore
  - Reserved words can not be used
  - Case Sensitive

4.1 Variables in JavaScript…

- **Variable Initialization**
  - **var *variableName* = *initialValue***
  - **var variableName1 = initialValue1, variableName2 = initialValue2, …**
- **Local & Global Variables:**
  - **A variable declared within a JavaScript function becomes Local and can only be accessed within that function.**
  - **Variables declared outside a function become Global, and all scripts and functions on the web page can access it.**

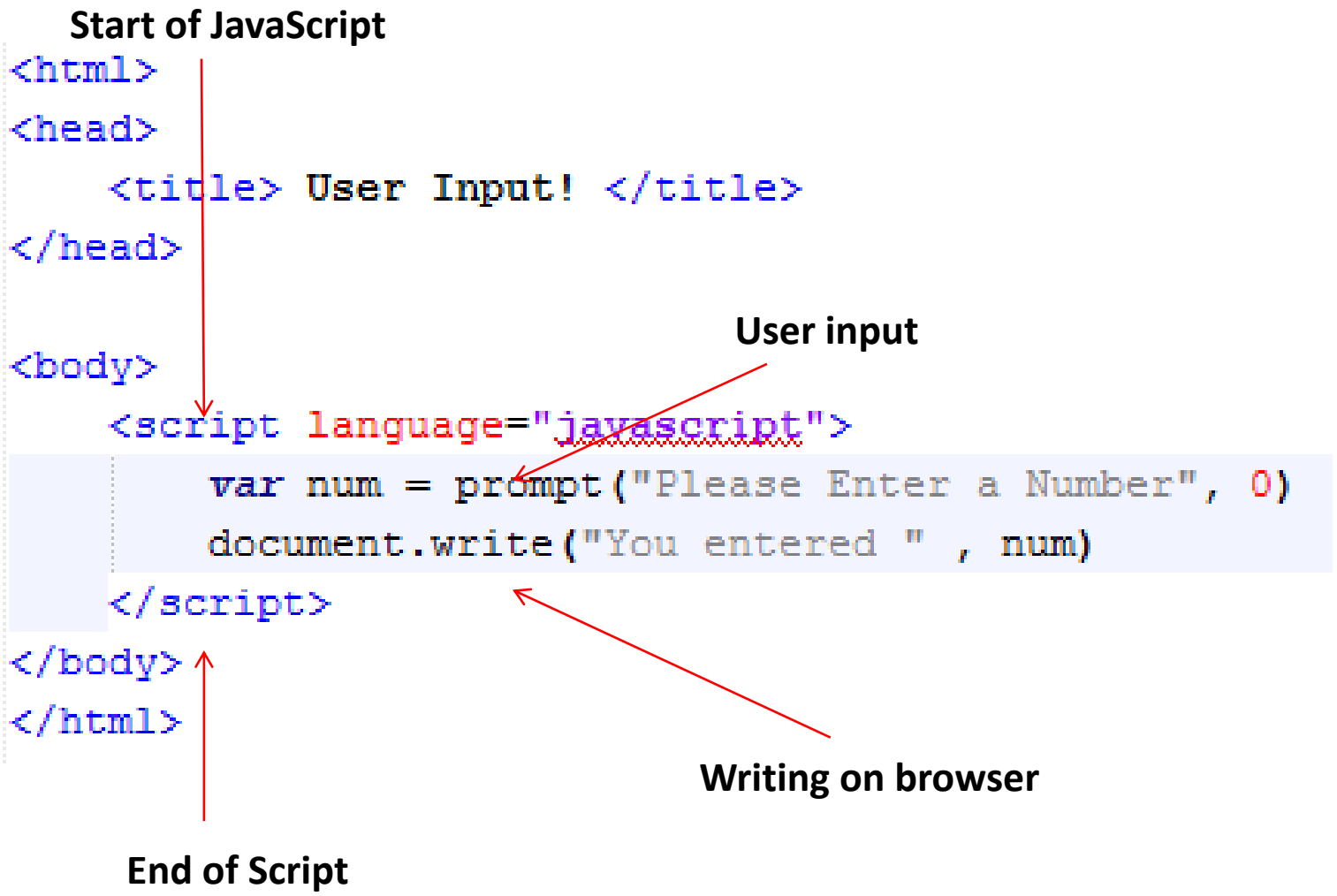
5. JavaScript Operators

- **An operator is simply a symbol that tells the compiler (or interpreter) to perform a certain action**
  - **Assignment Operator: =**
  - **Arithmetic Operators: +, - , *, /, %, ++, --**
  - **Logical Operators: &&, ||, !**
  - **Comparison Operators: ==, ===, !=, !==, <, >, <=, >=**

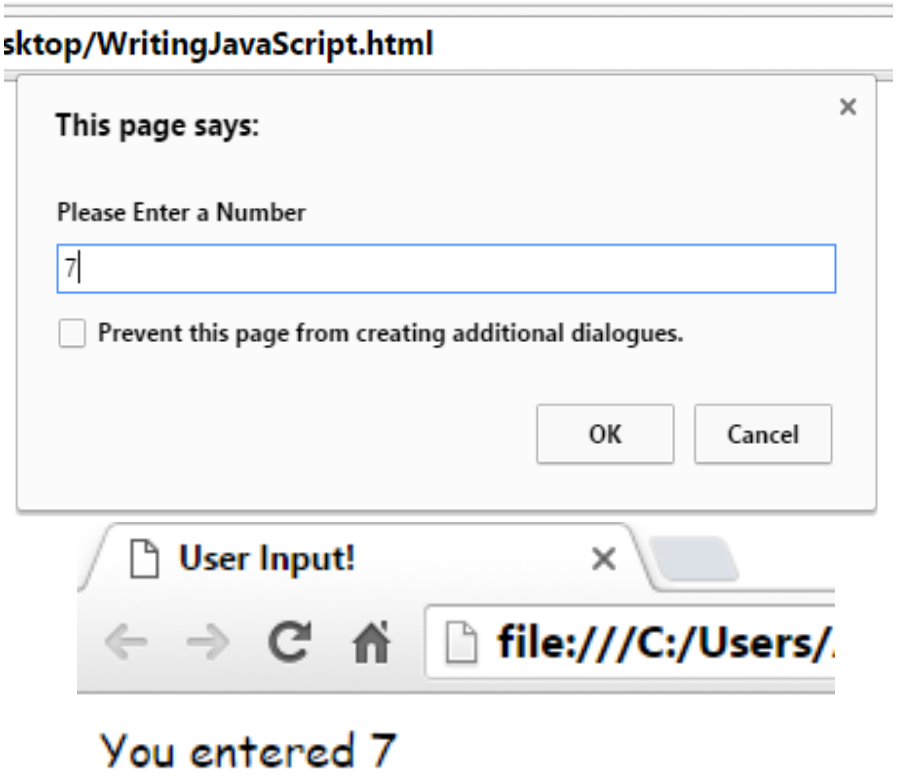
6. Input Output in JavaScript

- **write();**
  - **It is used to write on browser**
    - **document.write(“hello world”)**
    - **document.write(a)**
- **prompt();**
  - **It is used to take input from users**
    - **var num = prompt(“Please Enter a Number”, 0)**

6. Input Out put in JavaScript…



6. Input Out put in JavaScript…



7. JavaScript Function

- **A JavaScript function** is a block of JavaScript code, that can be executed when "asked" for.

- For example, a function can be executed when an **event** occurs, like when the user clicks a button.

7. JavaScript Function

- **Functions are used in JavaScript to perform various actions such as writing something, alerting users and taking input from users.**
- **There are two types of functions:**
  - **User defined functions: the functions made by the user to perform some actions:**
  - **Predefined functions: the functions that are defined previously and compiler uses it, user do not know about what is going on behind the code, but he should know how to call the function.**

7. JavaScript Function…

- **Functions are defined using the keyword function, followed by the name of the function and list of parameters**

```
function functionName([parameters])
{
        // some statements
}
```

- **Calling a function**
  - **The syntax of a function call is:**

```
functionName([arguments])
```

7. JavaScript Function…

```
<head>
    <title> Functions </title>
    <script language="javascript">
        function getName()
        {
            var name = prompt("Please Enter Your Name!", 'name')
            document.write("Welcome Mr. " , name, "!");
        }
    </script>
</head>
<body onload = "getName()">
</body>
</html>
```

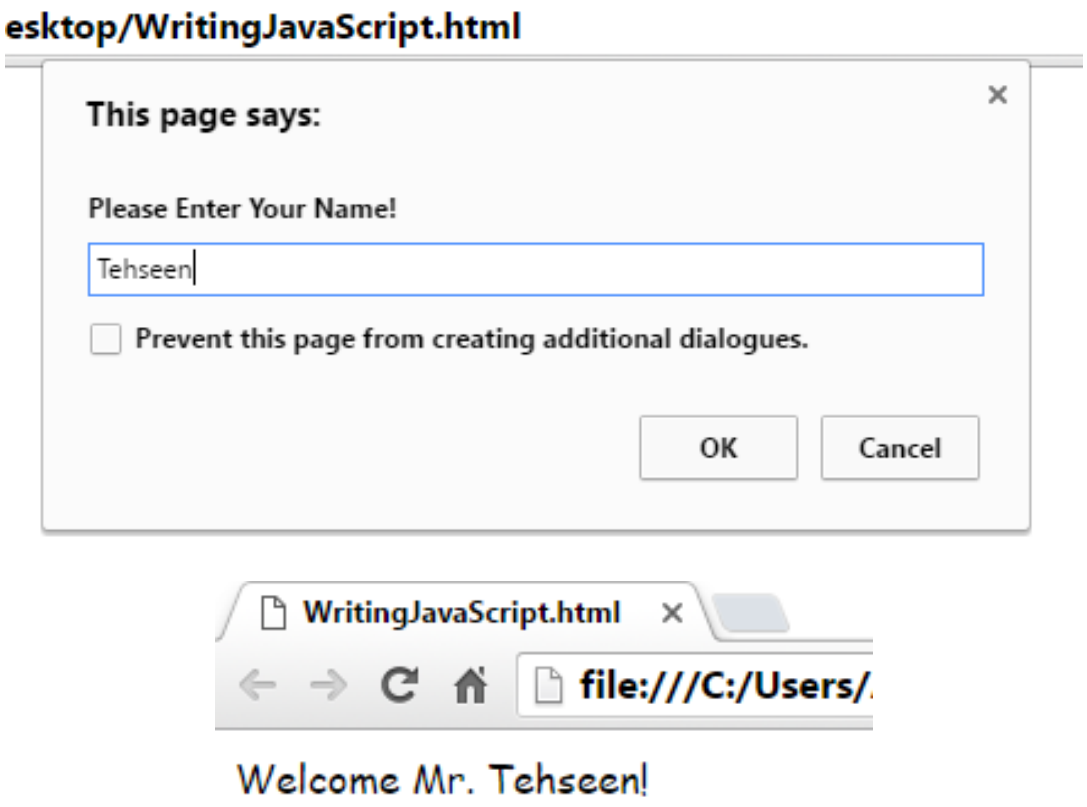
Start of the function

Asks user to enter name

Writes name on the webpage

Calling a function

7. JavaScript Function…



7. JavaScript Function…

- **Common events**
  - **onClick()**
  - **onDblClick()**
  - **onChange()**
  - **onFocus()**
  - **onMouseOver()**
  - **onMouseOut()**
  - **onSubmit()**
  - **onLoad()**

7. JavaScript Function…

- **Some common predefined math functions**
  - **Math.Sqrt (to calculate square root)**
  - **Math.Pow (to calculate power of number)**
  - **Math.Abs (to calculate absolute value)**
  - **Math.Max (to calculate maximum value)**
  - **Math.Min (to calculate minimum value)**
  - **Math.Floor (to calculate back rounded value)**
  - **Math.Ceil (to calculate forward rounded value)**
  - **Math.Round (to calculate rounded value)**
  - **Math.Random (to pick up some random value)**

8. Conditional Statements

- **If statement**
  - **if (condition)**
    // statement
  - **If (condition)**
    {
    // statements
    }
- **If-else statement**
  - **If (condition)**
    {
    // statement
    }
    else
    {
    // statement
    }

8. Conditional Statements…

```
<title> If Else </title>
<script language="javascript">
    function playGame()
    {
        var res = parseInt(Math.random(0, 10))
        var num = prompt("Please Enter A Number!", 0)

        if(num == res){
            document.write("You Won!");
        }
        else
        {
            document.write("You Lost! Correct Answer is ", res)
        }
    }
</script>
```

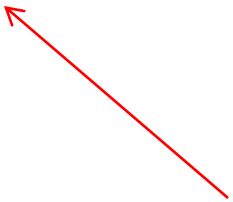
Random Number Generator

User's Input

If Condition

8. Conditional Statements…

```
<body>
    <p onclick="playGame()"> You Wanna Play? </p>
</body>
</html>
```

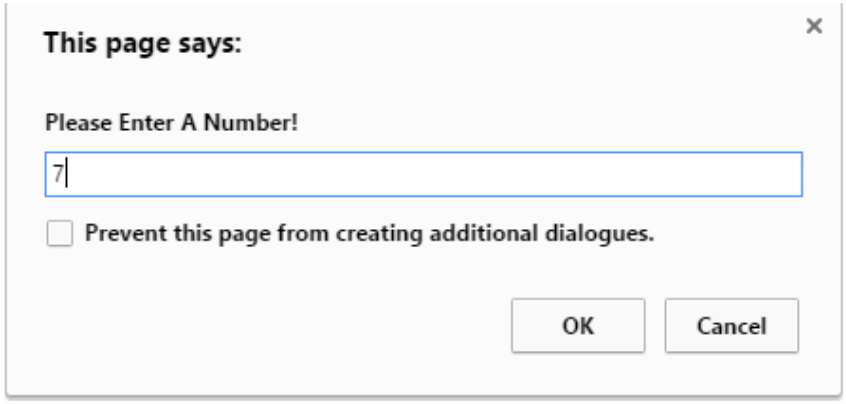


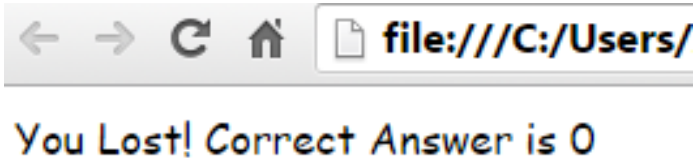
On Click Event

Text

8. Conditional Statements…

You Wanna Play?





9. Loops

- **For loop**
  - **for(var i=1; i==10; i==)**
    {
    // statements
    }
- **While loop**
  - **While(condition)**
    {
    // statements
    }

9. Loops

```
<body>
    <script language="javascript">
        for(var i=0; i<10; ++i)
        {
            document.write("This is For Loop Statement ", i, "<br>")
        }

        var j=0

        do{
            document.write("This is While Loop Statement ", j, "<br>")
            j++
        }
        while(j<10)
    </script>
</body>
```

For Loop

Do-While Loop

9. Loops

```
This is For Loop Statement 0
This is For Loop Statement 1
This is For Loop Statement 2
This is For Loop Statement 3
This is For Loop Statement 4
This is For Loop Statement 5
This is For Loop Statement 6
This is For Loop Statement 7
This is For Loop Statement 8
This is For Loop Statement 9
```

Output of For Loop

```
This is While Loop Statement 0
This is While Loop Statement 1
This is While Loop Statement 2
This is While Loop Statement 3
This is While Loop Statement 4
This is While Loop Statement 5
This is While Loop Statement 6
This is While Loop Statement 7
This is While Loop Statement 8
This is While Loop Statement 9
```

Output of Do-While Loop

10. JavaScript Output

- **JavaScript can "display" data in different ways:**
- **Writing into an alert box, using window.alert()**
  - <script>
    window.alert(5 + 6);
    </script>.
- **Writing into the HTML output using document.write().**
  - <script>
    document.write(5 + 6);
    </script>

10. JavaScript Output

- **Writing into an HTML element, using innerHTML.**
  - <script>
    document.getElementById("demo").innerHTML = 5 + 6;
    </script>
- **Writing into the browser console, using console.log().**
  - <script>
    console.log(5 + 6);
    </script>

Summary of Today's Lecture

- What is JavaScript?
- Embedding JavaScript with HTML
- JavaScript conventions
- Variables in JavaScript
- JavaScript operators
- Input output in JavaScript
- JavaScript functions
- Conditional Statements
- Looping Statements

THANK YOU