Web Technologies and Programming

Lecture 18

Summary of Previous 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

Today's Lecture Outline

- **Dialog boxes in JavaScript**
- **HTML Document Object Model (DOM)**

1. Dialog boxes in JavaScript

- **JavaScript provides the ability to pickup user input or display small amounts of text to the user by using dialog boxes**
- **These dialog boxes appear as separate windows and their content depends on the information provided by the user**

1. Dialog boxes in JavaScript

- **JavaScript has three kind of popup boxes:**
- **Alert box**
- **Prompt box**
- **Confirm box**

1.1 Alert Box

- An alert box is simply a small message box that pops up and gives the user some information
- An alert dialog box is mostly used to give a warning message to the users
- When an alert box pops up, the user will have to click "OK" to proceed
- Syntax:
  - alert(“message”)

1.1 Alert Box…

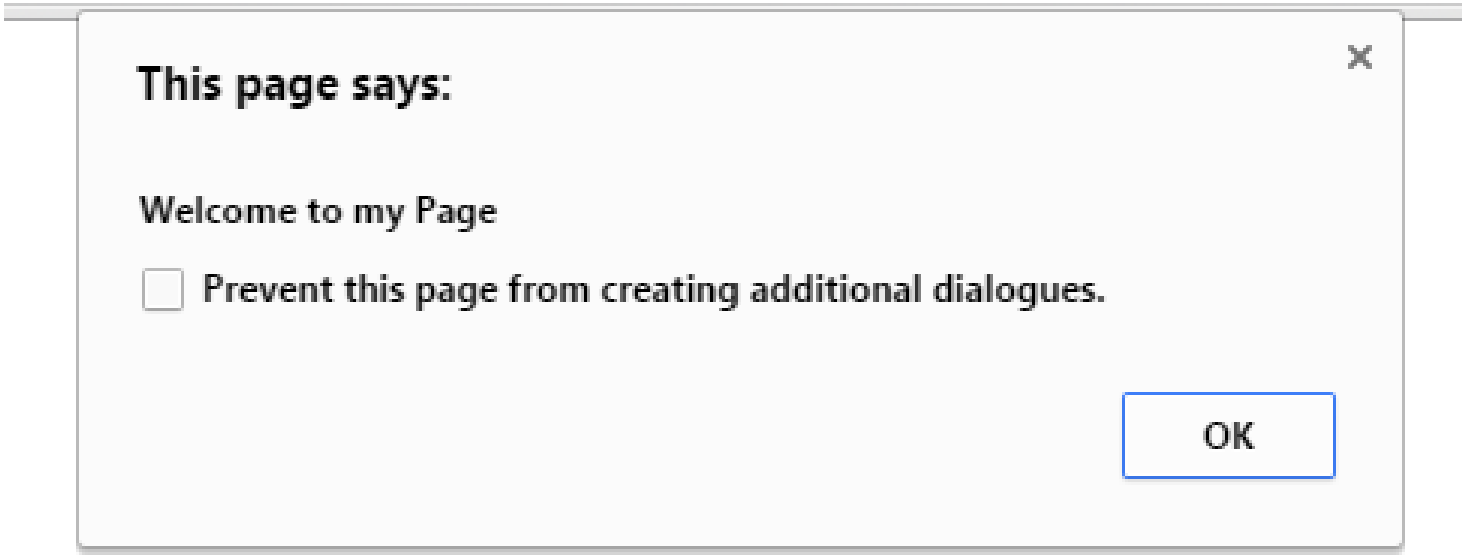
```
<script language="javascript">
    function alertme(message)
    {
        alert(message)
    }
    alertme("Welcome to my Page")
</script>
```

Start of Function

Displays an Alert Box

Calling the Function

1.1 Alert Box...



Alert Box

1.2 Prompt box

- A prompt box is often used if you want the user to input a value before entering a page
- When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value
- If the user clicks "OK" the box returns the input value
- If the user clicks "Cancel" the box returns null

1.2. Prompt box…

```
<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>
```

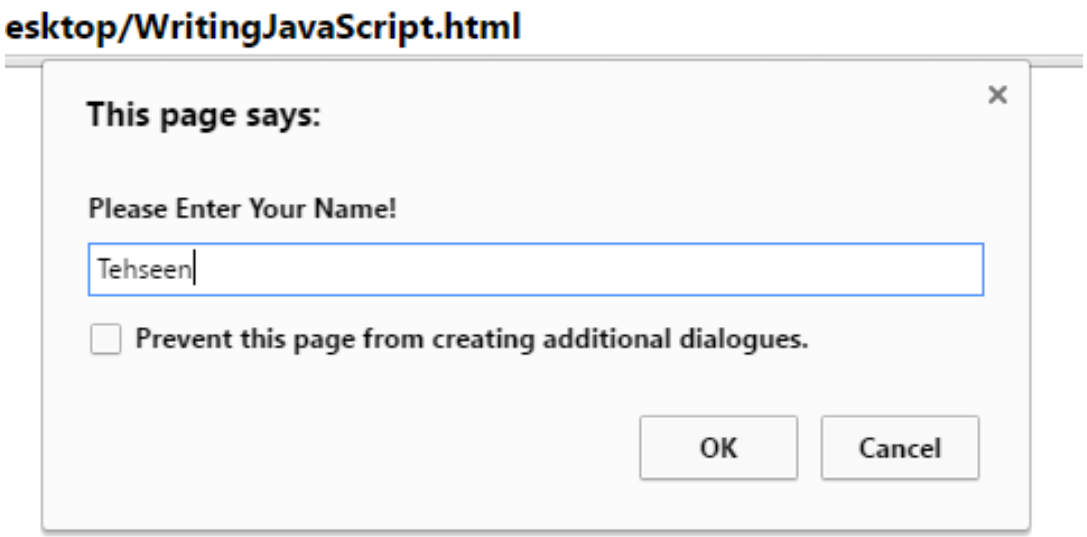
Start of the function

Asks user to enter name

Writes name on the webpage

Calling a function

1.2. Prompt box…



Prompt Box

Output

1.3 Confirm box

- A confirm box is often used if you want the user to verify or accept something
- When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed
- If the user clicks "OK", the box returns true
- If the user clicks "Cancel", the box returns false

1.3 Confirm box

User Input

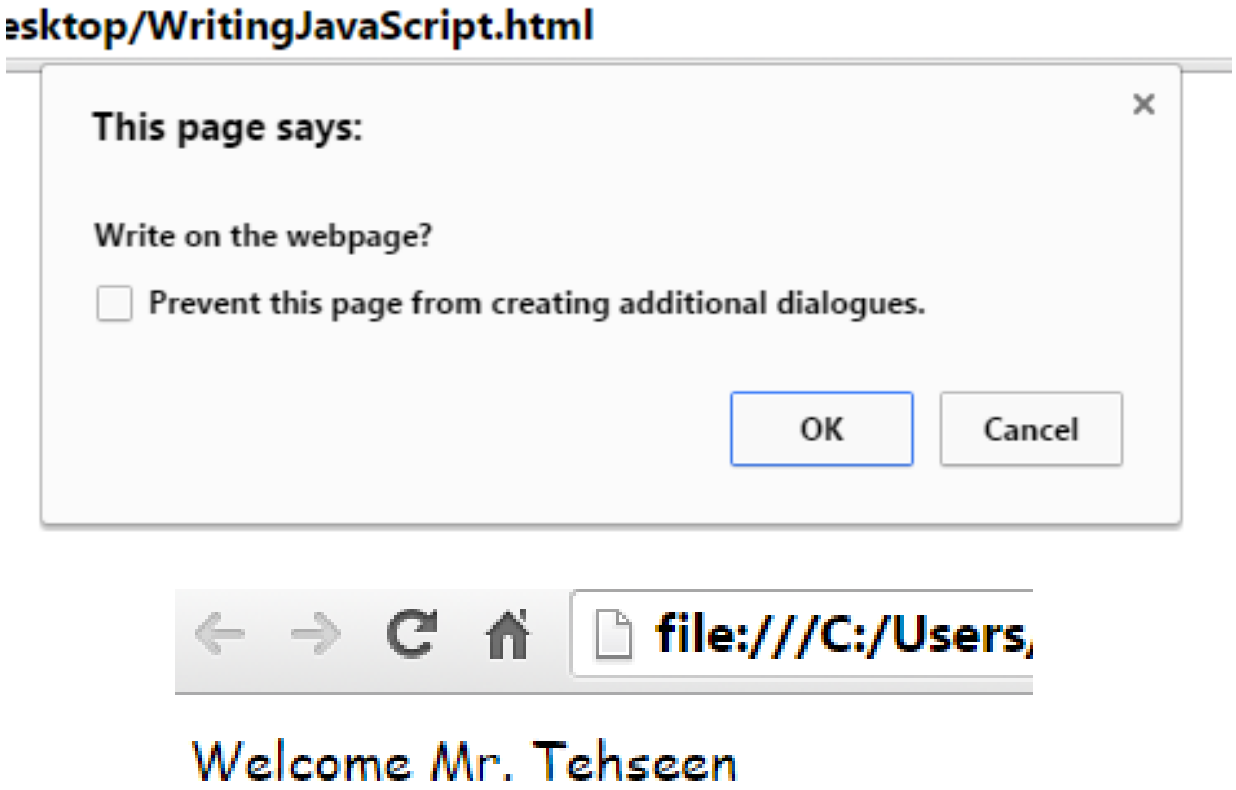
Confirmation

Not Confirmed

```
<script language="javascript">
    function conName()
    {
        var name = prompt("Your Name Please!")

        if(confirm("Write on the webpage?"))
            document.write("Welcome Mr. " + name)
        else
            document.write("Welcome Mr. Anonymous")
    }
    conName()
</script>
```

1.3 Confirm box



2. Document Object Model (DOM)

- **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."**

2. Document Object Model (DOM)

- **The W3C DOM standard is separated into 3 different parts:**
- **Core DOM - standard model for all document types**
- **XML DOM - standard model for XML documents**
- **HTML DOM - standard model for HTML documents**

2. HTML Document Object Model (DOM)

- Once html element are rendered (painted) in the browser window, browser can not recognize them
- To create interactive web pages it is vital that the browser continues to recognize HTML elements
- JavaScript enabled browsers do this because they recognize and uses DOM

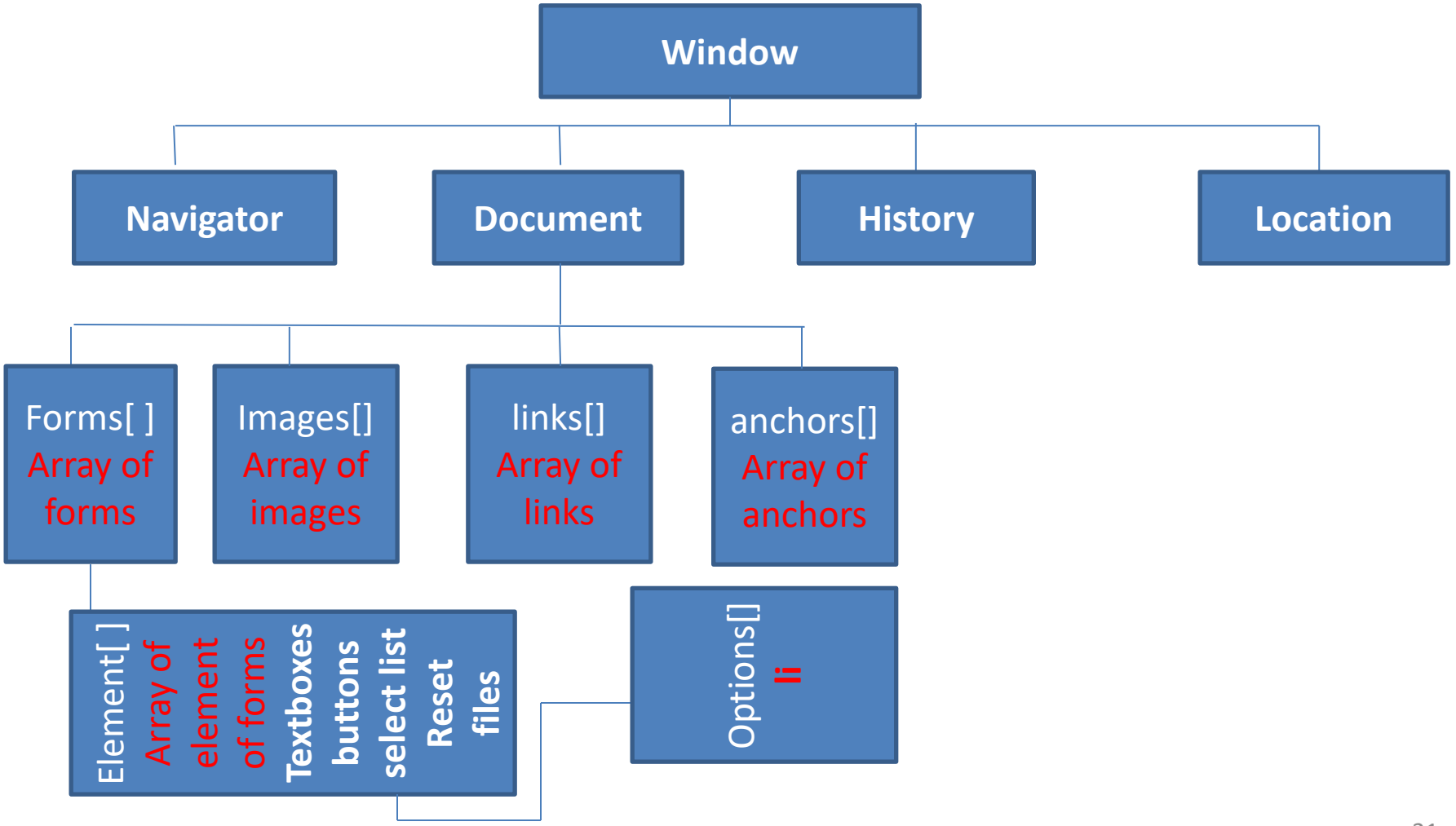
2. HTML Document Object Model (DOM)…

- The HTML DOM defines a standard set of objects for HTML, and a standard way to access and manipulate HTML documents
- All HTML elements, along with their containing text and attributes, can be accessed through the DOM
  - The contents can be modified or deleted, and new elements can be created

2. HTML Document Object Model (DOM)…

- **The HTML DOM is platform and language Independent**
  - **It can be used by any programming language like Java, JavaScript, and VBScript**
- **The HTML DOM can be thought of as a hierarchy moving from the most general object to the most specific**

2. Document Object Model (DOM)…



2. Document Object Model (DOM)…

- **When a web page is loaded, the browser creates a Document Object Model of the page.**
- **With the object model, JavaScript gets all the power it needs to create dynamic HTML:**
  - **JavaScript can change all the HTML elements in the page**
  - **JavaScript can change all the HTML attributes in the page**
  - **JavaScript can change all the CSS styles in the page**
  - **JavaScript can remove existing HTML elements and attributes**

2. Document Object Model (DOM)…

- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

2. Html Document Object Model (DOM)…

- HTML DOM methods are **actions** you can perform (on HTML Elements).
- HTML DOM properties are **values** (of HTML Elements) that you can set or change.
- In the DOM, all HTML elements are defined as **objects**.
- The programming interface is the properties and methods of each object.
- A **property** is a value that you can get or set (like changing the content of an HTML element).
- A **method** is an action you can do (like add or deleting an HTML element).

2. Document Object Model (DOM)…

- **document.forms[0].elements[0].value**
- **document.images[0].src**
- **document.links[0].href**

2.1 Retrieving HTML elements

- The getElementById() method is a workhorse method of the DOM
- It retrieves a specified element of the HTML document and returns a reference to it
- To retrieve an element, it must have an unique id
  - document.getElementById("element-id")

2.1 Retrieving HTML elements…

- **The returned reference can be used to retrieve element attributes**
  - **document.getElementById(“element-id”).attribute**
  - **document.getElementById(“pic”).src**
  - **document.getElementById(“pic”).height**

2.1 Retrieving HTML elements…

```
<html>
<head>
    <title> Retrieving Values </title>
    <script language="javascript">
        function getValue()
        {
            var a = document.getElementById("pic")
            alert(a.title)
            alert(a.height)
        }
    </script>
</head>
<body>
    <img src="Benz.jpg" id="pic" title="Mercedes Benz" onClick="getValue()">
</body>
</html>
```

Function Starts

Getting Reference to “pic”

Accessing Attributes

Setting id

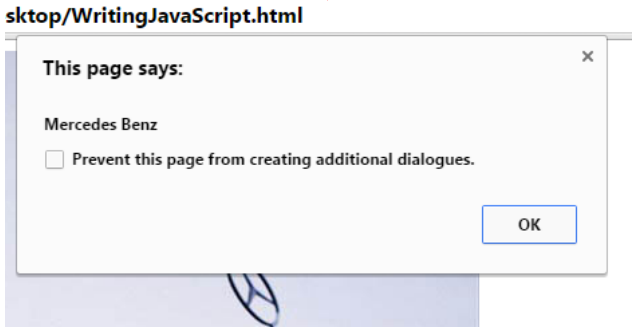
Image Tag

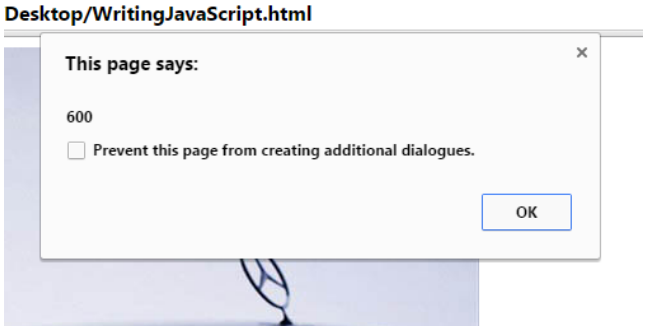
Calling Function

2.1 Retrieving HTML elements…

Image

Title Alert





Height Alert

2.2 Retrieving the text of an element…

- **innerHTML property defines both the HTML code and the text that occurs between that element's opening and closing**
  - **document.getElementById(“element-id”).innerHTML**

2.2 Retrieving the text of an element…

```
<html>
<head>
    <title> Retrieving Values </title>
    <script language="javascript">
        function getText()
        {
            var a = document.getElementById("myPara")
            alert(a.innerHTML)
        }
    </script>
</head>
<body>
    <p id="myPara" onClick="getText()"> Hello World! </p>
</body>
</html>
```

Getting Element Reference

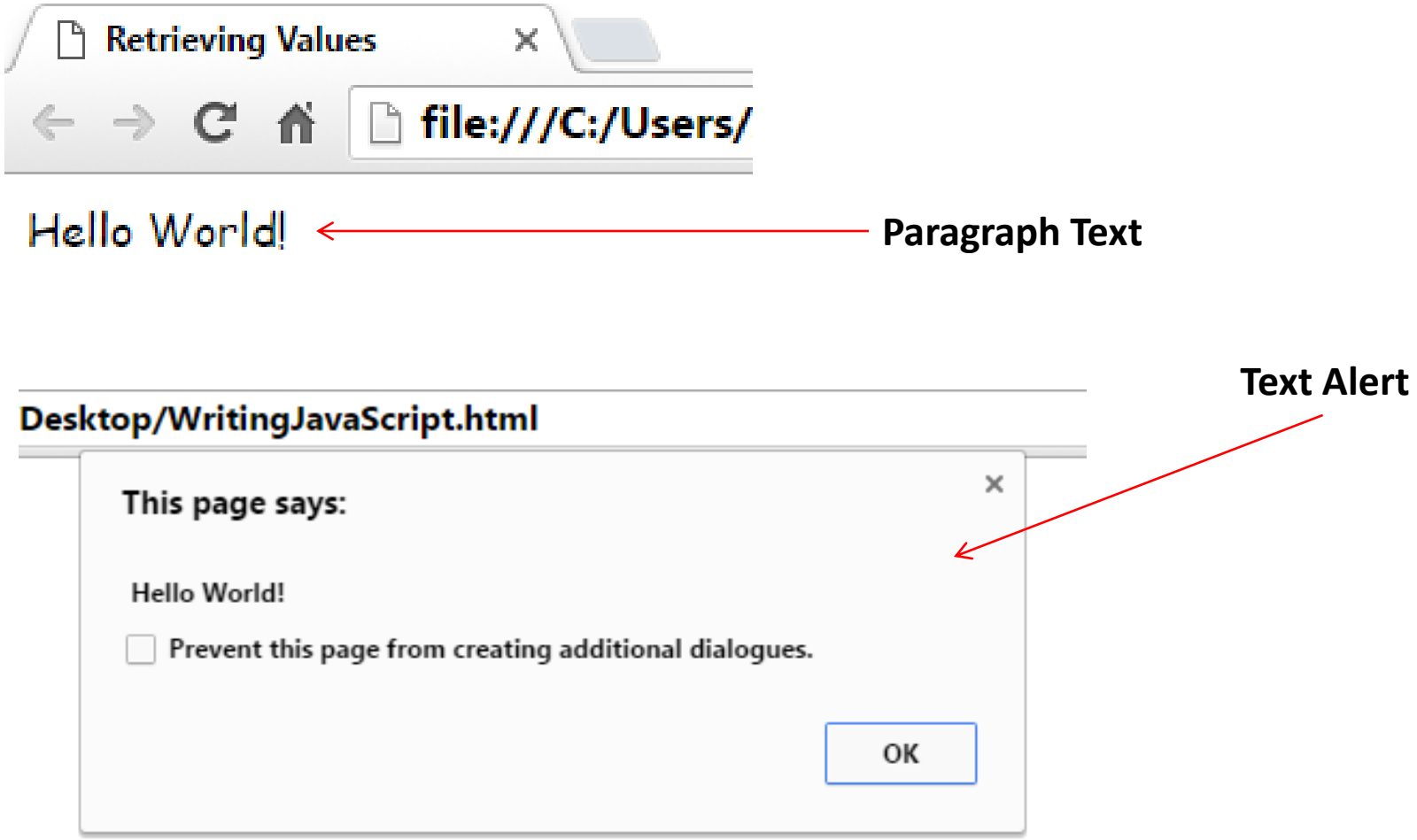
Getting Text

Paragraph id

Calling Function

Paragraph Text

2.2 Retrieving the text of an element



2.3 Getting value of attributes

- **getAttribute() method is used to retrieve values of attributes**
  - **document.getElementById("element-id").getAttribute("attribute-name")**
  - **document.getElementById("pic").getAttribute("src")**

2.3 Getting value of attributes…

```
<html>
<head>
    <title> Attribute Values </title>
    <script language="javascript">
        function getAtt()
        {
            var a = document.getElementById("pic")
            alert(a.getAttribute("title"))
        }
    </script>
</head>

<body>
    <img src="Benz.jpg" id="pic" title="Mercedes Benz" onClick="getAtt()">
</body>
</html>
```

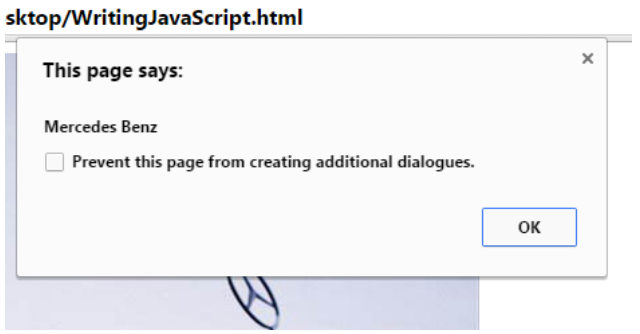
Getting Element Reference

Using getAttribute() Function

2.3 Getting value of attributes…

Image

Title Attribute Alert



2.4 Setting value of attributes

- **setAttribute() method is used to set values of attributes**
  - **document.getElementById(“element-id”).setAttribute(“attribute-name”, ”Value”)**
  - **document.getElementById(“pic”).setAttribute(“src”, ”abc.jpg”)**

2.4 Setting value of attributes…

```
<html>
<head>
    <title> Attribute Values </title>
    <script language="javascript">
        function setAtt()
        {
            var a = document.getElementById("pic")
            a.setAttribute("height", "75")
            a.setAttribute("width", "100")
        }
    </script>
</head>

<body>
    <img src="Benz.jpg" id="pic" title="Mercedes Benz" onClick="setAtt()">
</body>
</html>
```

Getting Reference Element

Changing Attribute Values

2.4 Setting value of attributes…

Before Setting Attribute Values

After Setting Attribute Values

3.0 HTML DOM Elements

- **Finding HTML Elements**
- To do so, you have to find the elements first. There are a couple of ways to do this:
  - Finding HTML elements by id
  - Finding HTML elements by tag name
  - Finding HTML elements by class name
  - Finding HTML elements by CSS selectors
  - Finding HTML elements by HTML object collections

3.1 HTML DOM Elements

- **Finding HTML Element by Id**
- The easiest way to find an HTML element in the DOM, is by using the element id.
- This example finds the element with id="intro":
- **Example:**
- var myElement = document.getElementById("intro");

3.2 HTML DOM Elements

- **Finding HTML Elements by Tag Name**
- This example finds all <p> elements:
- Example
- var x = document.getElementsByTagName("p");

3.3 HTML DOM Elements

- Finding HTML Elements by Class Name
- If you want to find all HTML elements with the same class name, use getElementsByClassName().
  This example returns a list of all elements with class="intro".
- **Example**
- var x = document.getElementsByClassName("intro");

3.4 HTML DOM Elements

- **Finding HTML Elements**
- **Methods**
  - **Descriptiondocument.getElementById(id)**
  - **Find an element by element**
  - **iddocument.getElementsByTagName(name)**
  - **Find elements by tag**
    namedocument.getElementsByClassName(name)
  - **Find elements by class name**

Summary of Today's Lecture

- Dialog boxes in JavaScript
  - Alert box
  - Prompt box
  - Confirm box
- What is DOM?
- HTML DOM
- Retrieving HTML elements
- getElementById()
- innerHTML
- getAttribute()
- setAttribute()

Summary of Today's Lecture

- **HTML DOM Elements**
- **Finding HTML Elements**
  - **Find an element by element**
  - **Find elements by tag**
  - **Find elements by class name**

THANK YOU