UNIT III CLIENT SIDE TECHNOLOGIES

- XML
- Document Type Definition
- XML Schema
- Document Object Model
- Presenting XML Using XML Parsers: DOM and SAX
- JavaScript Fundamentals
- Evolution of AJAX
- AJAX Framework
- Web applications with AJAX
- AJAX with PHP
- AJAX with Databases

Presented by,
B.Vijayalakshmi
Computer Centre
MIT Campus
Anna University

HTML
(Hyper Text Markup Language)

HTML
(Hyper Text Markup Language)

- It is **used to create web pages**.
- It is widely used language on the web.
- As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display
- We can **create only static website by HTML**
- HTML is being widely used to format web pages with the help of different tags available in HTML language
- HTML tags contain three main parts,
  - opening tag
  - content
  - closing tag

<tag> content </tag>

HTML Tags

Unclosed HTML Tags

Some HTML tags are not closed, for example br and hr.

**
 Tag**: br stands for break line, it breaks the line of the code.

<hr> Tag: hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

HTML Meta Tags

DOCTYPE, title, link, meta and style

HTML Text Tags

<p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , , <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <cite>, <q>, <code>, <ins>, <del>, <dfn>, <kbd>, <pre>, <samp>, <var> and

HTML Link Tags

<a> and <base>

HTML Image and Object Tags

<img>, <area>, <map>, <param> and <object>

HTML Tags Cont'd

HTML List Tags

, <ol>, <li>, <dl>, <dt> and <dd>

HTML Table Tags

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

HTML Scripting Tags

script and noscript

HTML Form Tags (An HTML form facilitates the user to enter data that is to be sent to the server for processing) form, input, textarea, select, option, optgroup, button, label, fieldset and legend

```
<form action="server url" method="get|post">
  //input controls e.g. textfield, textarea, radiobutton, button
</form>
```

- HTML basic tags Example
- HTML Extra Formatting tags Example
- HTML CSS Basic Example
- Form tags Example

Document Object Model (DOM)

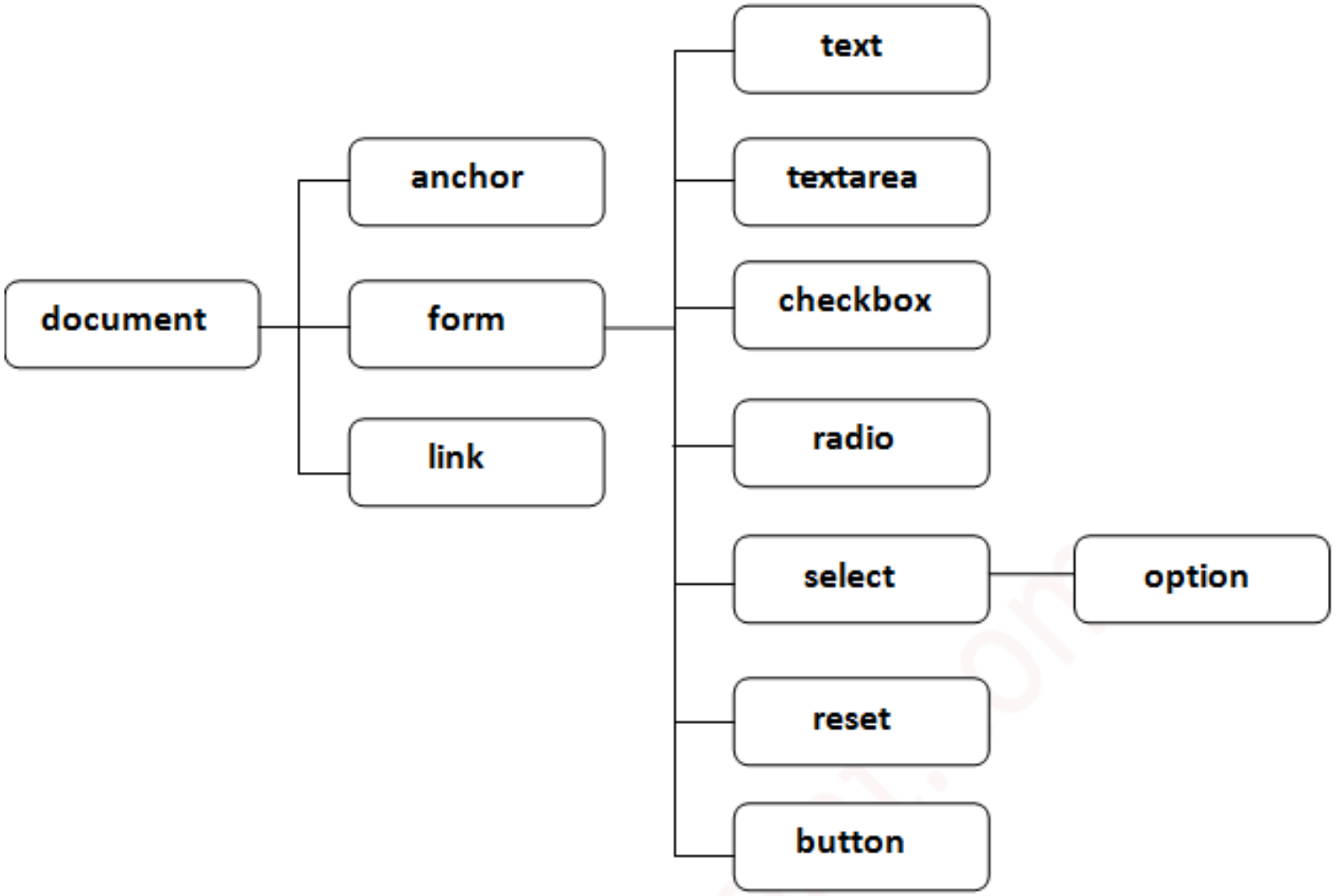
DOM

- The DOM is a W3C (World Wide Web Consortium) standard
- According to W3C
  - *It 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 W3C DOM standard is separated into 3 different parts:
  - Core DOM - standard model for all document types
  - HTML DOM - standard model for HTML documents
  - XML DOM - standard model for XML documents

Document Object Model

- When html document is loaded in the browser. The web browser builds a *model* of the web page (the *document*) that includes all the *objects* in the page (tags, text, etc) it becomes a document object
- It is the **root element** that represents the html document
- All of the properties, methods, and events available to the web developer for manipulating and creating web pages are organized into objects
- Those objects are accessible via scripting languages in modern web browsers

Properties of document object

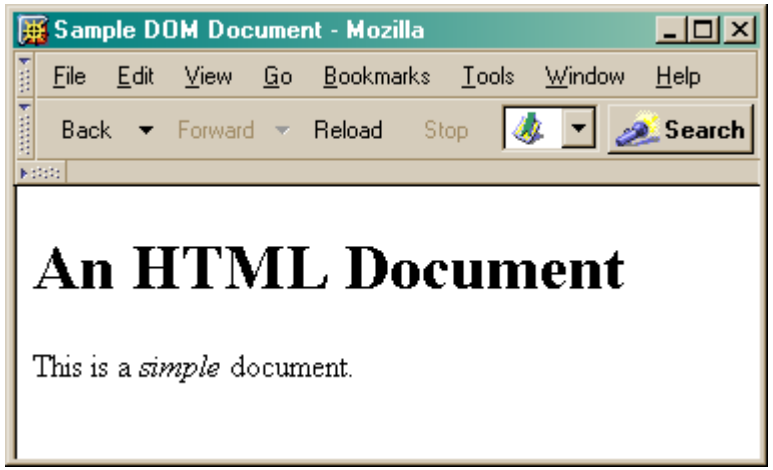


A Simple Html Document

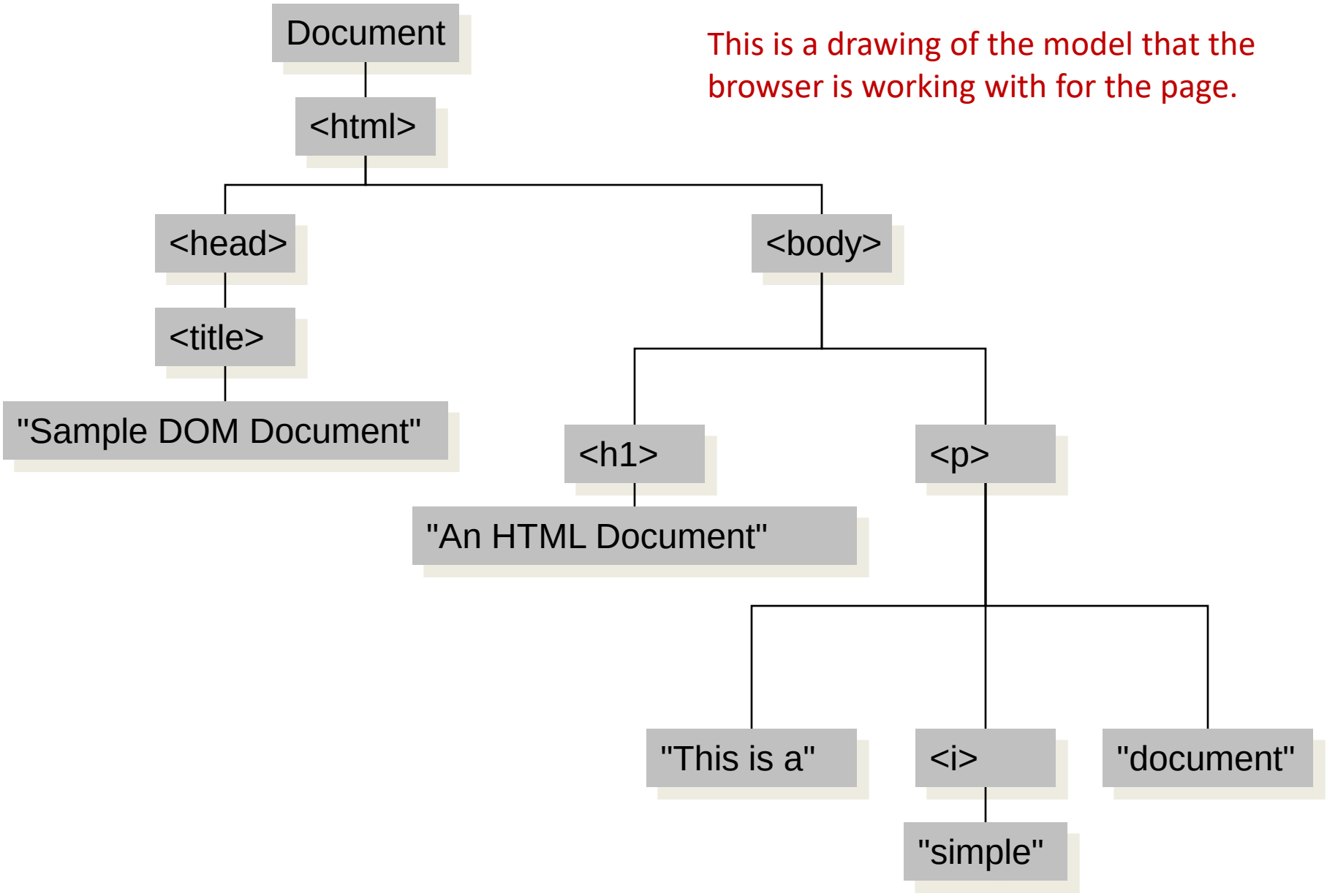
This is what the browser reads

```
<html>
  <head>
    <title>Sample DOM Document</title>
  </head>
  <body>
    <h1>An HTML Document</h1>
    <p>This is a <i>simple</i> document.
  </body>
</html>
```

This is what the browser displays on screen.

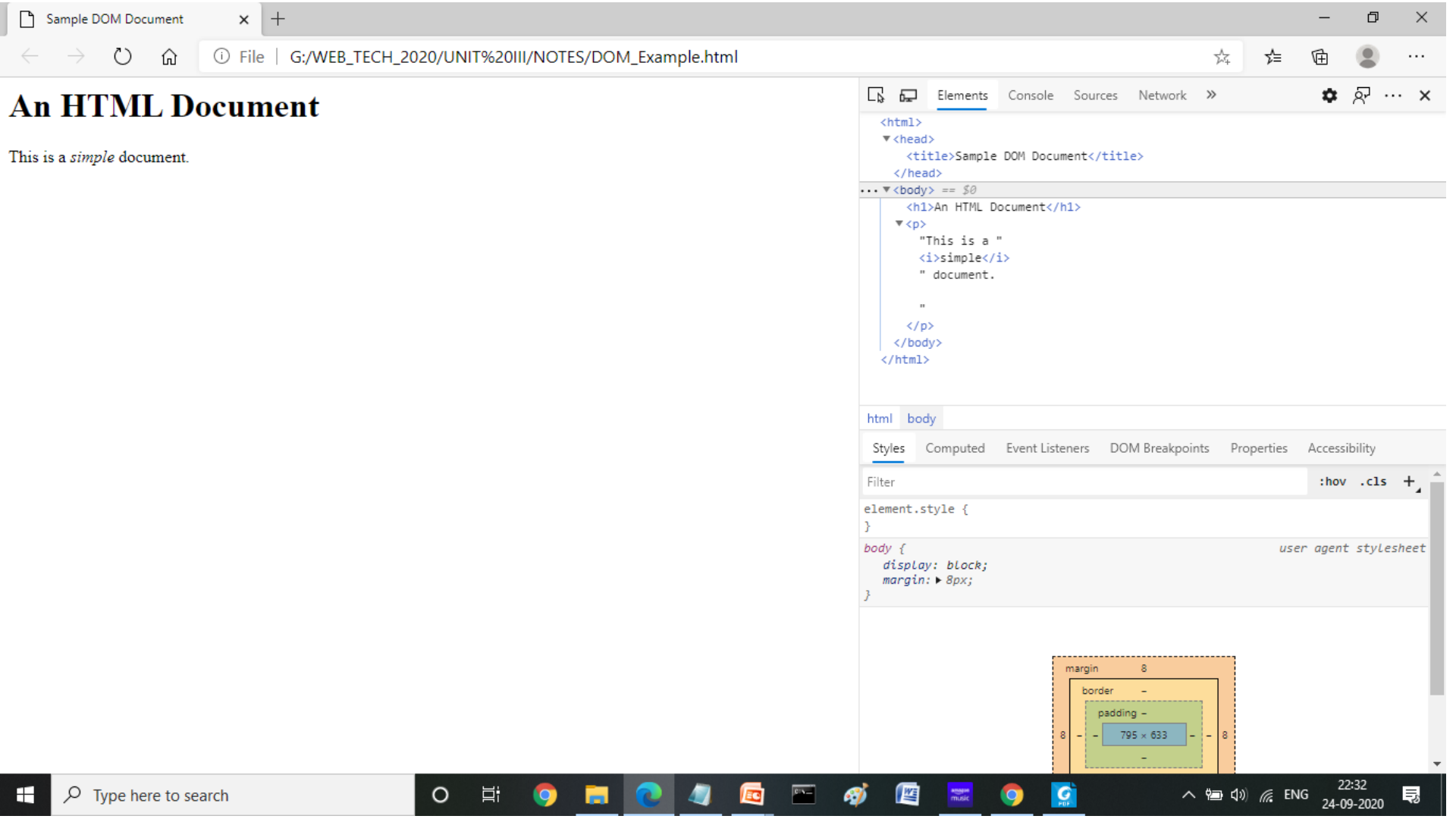


DOM Example



This is a drawing of the model that the browser is working with for the page.

Using, DOM Inspector



Why is this useful?

- The model is made available to scripts running in the browser, not just the browser itself
  - A script can find things out about the state of the page
  - A script can change things in response to events, including user requests

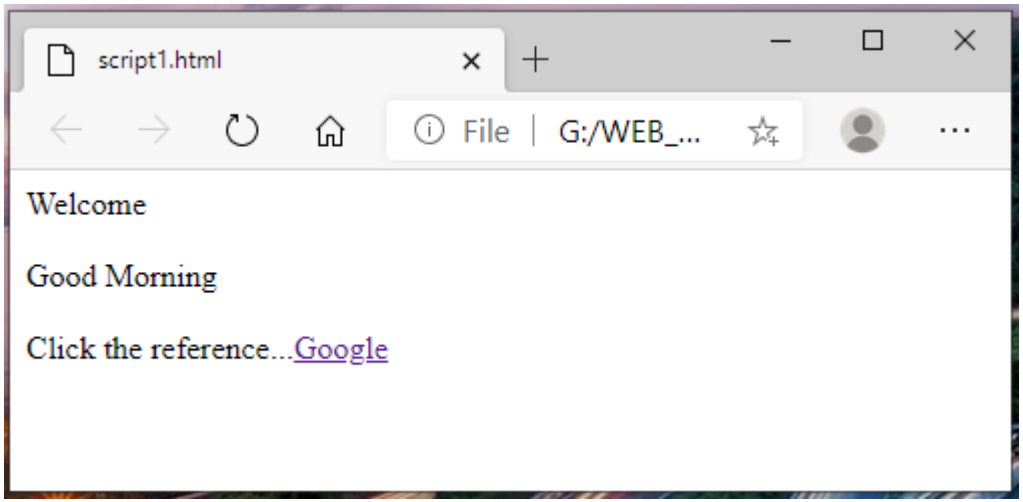
Methods of document object

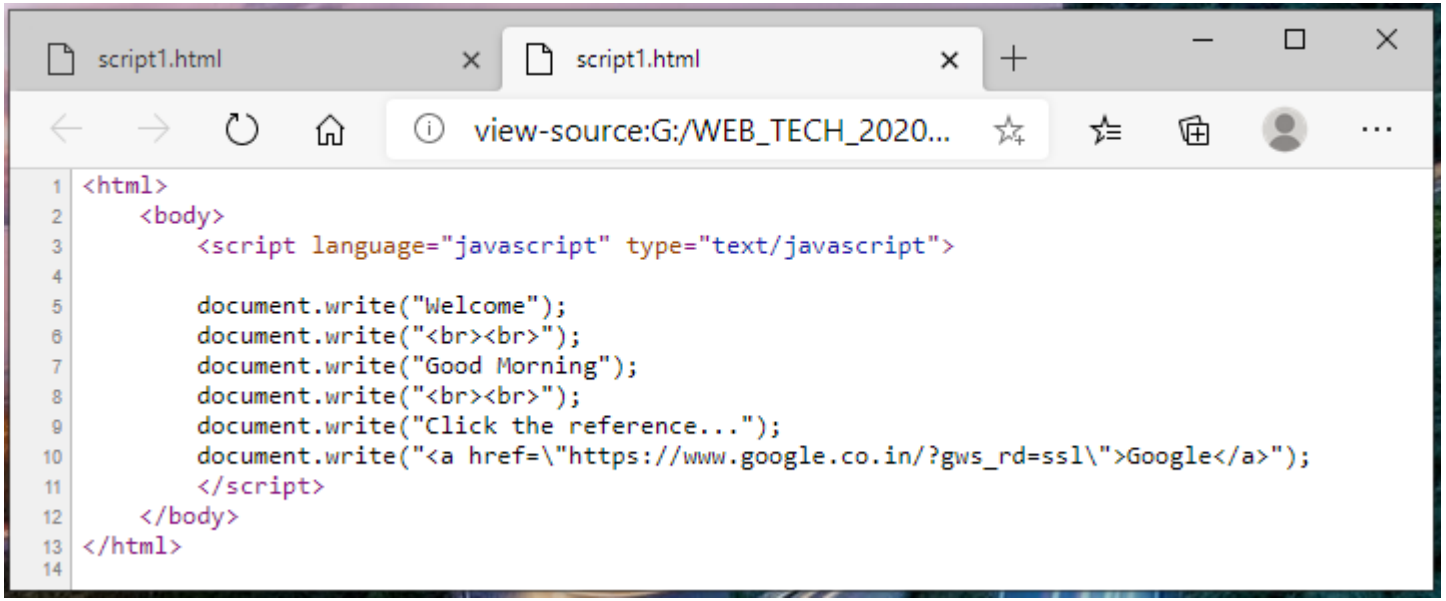
•Any document can be accessed and its contents can be changed using following

Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

write("string") method

Example

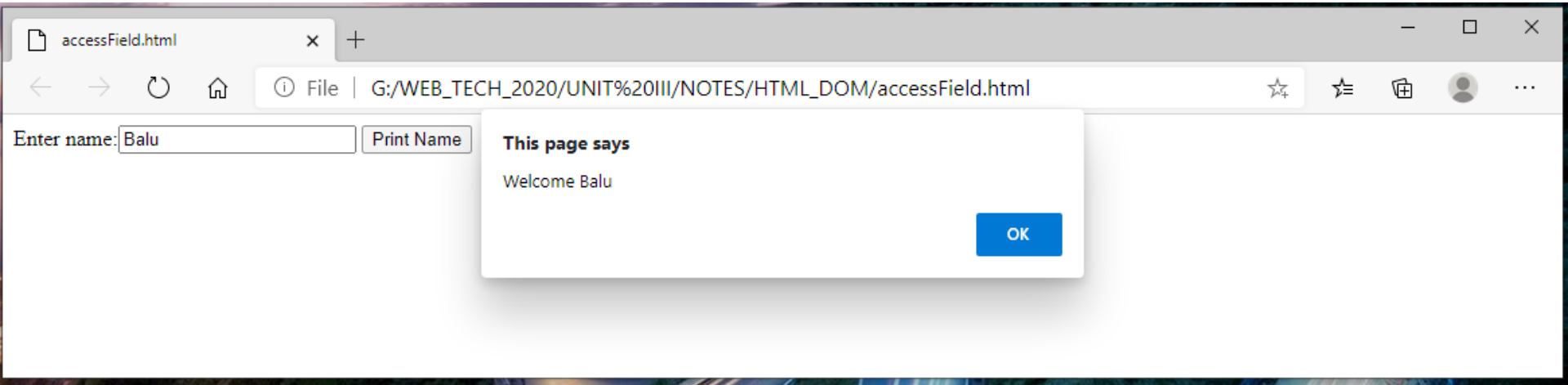




Accessing field value by document object

Example

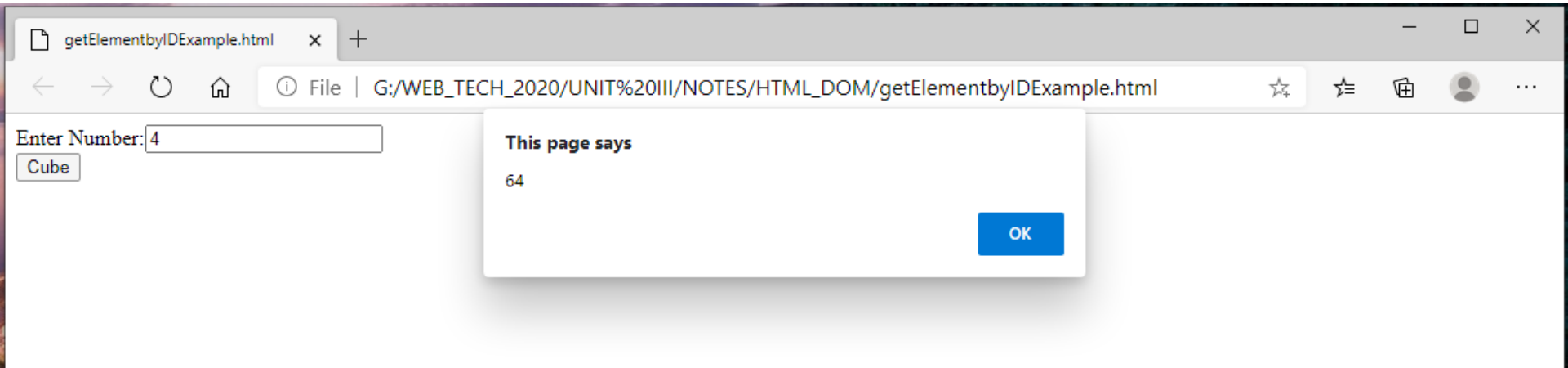
```
<html>
    <body>
        <script language="javascript" type="text/javascript">
        function printValue()
        {
            var name=document.form1.name.value;
            alert("Welcome "+name);
        }
        </script>
    <form name="form1">
    Enter name:<input type="text" name="name"/>
    <input type ="button" onclick="printValue()" value="Print Name"/>
    </form>
    </body>
</html>
```



document.getElementById() method

Example

```
<html>
    <body>
        <script language="javascript" type="text/javascript">
        function getCube()
        {
            var n=document.getElementById("num").value;
            alert(n*n*n);
        }
        </script>
    <form name="form1">
    Enter Number:<input type="text" id ="num" name="number"/><br/>
    <input type ="button" onclick="getCube()" value="Cube"/>
    </form>
    </body>
</html>
```

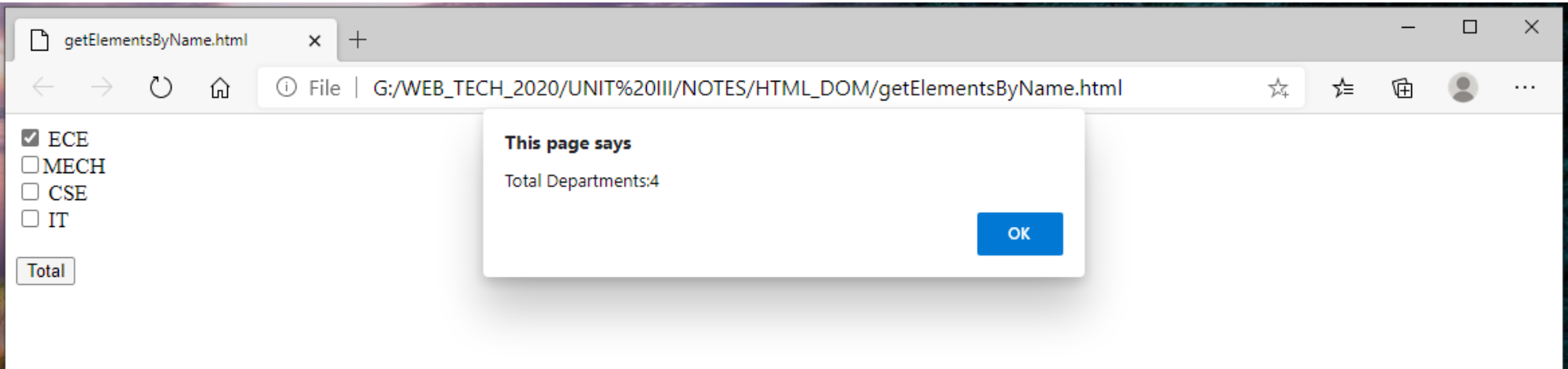


document.getElementsByName()

[Returns all the element of specified name]

Example

```
<html>
    <body>
        <script language="javascript" type="text/javascript">
        function totalElements()
        {
            var allDept=document.getElementsByName("dept");
            alert("Total Departments:"+allDept.length);
        }
        </script>
    <form name="form1">
    <input type="checkbox" name="dept" value="ece" checked> ECE<br>
    <input type="checkbox" name="dept" value="mech" >MECH<br>
    <input type="checkbox" name="dept" value="cse" > CSE<br>
    <input type="checkbox" name="dept" value="it" > IT<br>
    <br>
    <input type ="button" onclick="totalElements()" value="Total"/>
    </form>
    </body>
</html>
```



document.getElementsByTagName()

[returns all the element of specified tag name]

Example

```
<html>
    <body>
        <script language="javascript" type="text/javascript">
        function countPara()
        {
            var totalPara=document.getElementsByTagName("p");
            alert("Total p tags are:"+totalPara.length);
        }
        </script>
    <form name="form1">
    <p>This is paragraph</p>
    <p>Here we are going to count the total number of paragraphs by getElementsByTagName</p>
    <p> Let's see the simple example </p>
    <br>
    <input type ="button" onclick="countPara()" value="Count Paragraph"/>
    </form>
    </body>
</html>
```