

COMSW3101: SCRIPTING LANGUAGES: JAVASCRIPT (SPRING 2020)

RAMANA ISUKAPALLI

RAMANA@CS.COLUMBIA.EDU

LECTURE- I

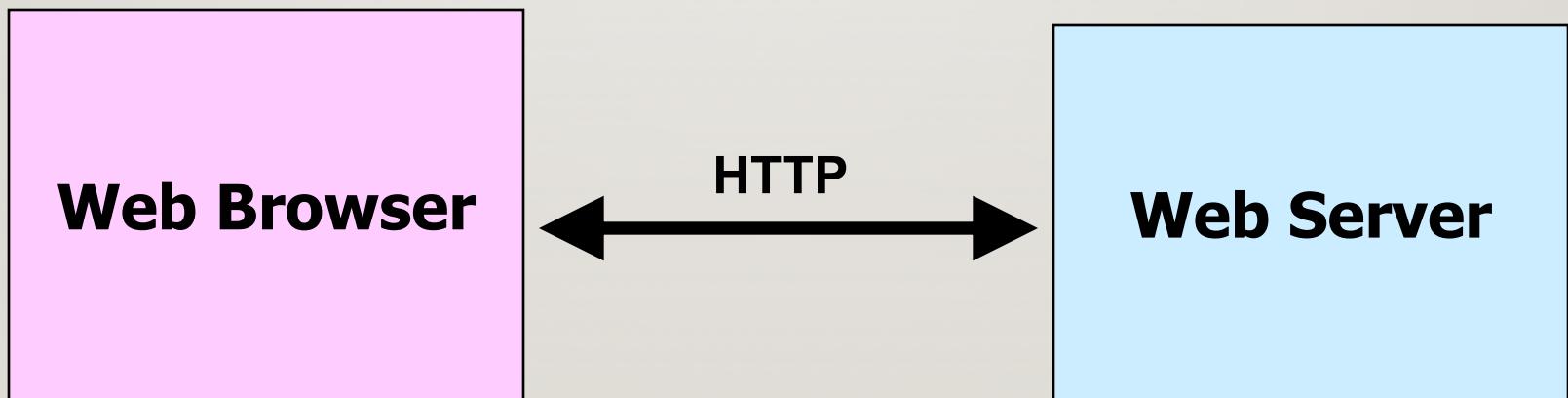
- Course overview
 - See <http://www.cs.columbia.edu/~ramana>
- Overview of HTML
 - Formatting, headings, images, colors, tables, forms, etc.
 - XHTML – difference with HTML
 - DHTML
 - What is it?
 - Why is it needed
- Javascript
 - Overview, what is it, why is it needed, etc.
 - How does it fit with HTML

PREREQUISITES

- A good background in at least one programming language is recommended.
- Ability to learn quickly.

OVERVIEW OF HTML

- HTTP: Communication protocol between
 - Any web server (e.g., www.cnn.com) and
 - Browser (e.g., firefox, IE, Opera, etc.)
- HTML – Hyper-Text Markup Language
 - Format in which web data is stored.



HTML ... CONTD.

- Format in which a web server stores the content.
- Transferred over to the client (using HTTP).
- Hypertext – stores data of many formats
- Simple text – with different fonts, sizes, colors, paragraphs, etc.
 - Audio, video, image files, etc.
 - Uses **markup tags**, e.g., `<h1>` Heading `</h1>`
 - ⇒ Can arrange data in tables, bullets, web links, forms, etc.
- HTML details
 - <http://www.w3schools.com/html/default.asp>
 - <http://www.w3.org/TR/html4/>

A TYPICAL HTML PAGE

```
<html>      <!-- Beginning of the HTML page -->  
  <head>    <!--Typically has page title, useful for search engines -->  
    <title>  
      My Web page  
    </title> <-- Page title -->  
  </head>  
  <body>    <-- Body of the web page, has main content-->  
    Content  
  </body>  
</html>      <-- End of the HTML page -->
```

HTML TAGS

- Headings – <h1>, <h2>, <h3>
- Anchor – <a>
- Table – <table>
- Table row – <tr>
- Table cell – <td>
- No support for scripts –
<noscript>
 - These tags are used to format a web page content
 - A complete list of tags can be found at
<http://w3schools.com/tags/default.asp>
- Form – <form>
- Image –
- Lists –
- Ordered list –
- Unordered list --
- No support for frames –
<noframes>

XHTML

- XHTML
 - EXtensible HyperText Markup Language
 - Combines HTML with strict syntax of XML
- Almost identical to HTML
- XHTML is a stricter and cleaner version of HTML.
- XHTML is HTML defined as an XML application.
- XHTML consists of
 - DOCTYPE declaration
 - head
 - body

XHTML RULES

- XHTML elements must be
 - Properly nested – e.g., <head> <title>.... </title> </head>
 - Always closed – e.g., <body> .. </body>
 - In lowercase
- XHTML documents must have one root element
- XHTML
 - Attribute names must be in lower case
 - E.g., <table WIDTH="100%"> is wrong.
 - Attribute values must be quoted
 - e.g., <table width="100%">
 - Attribute minimization is forbidden
 - <input checked="checked" /> instead of <input checked>

ANOTHER HTML EXAMPLE

```
<html>
  <head>
    <title>DOM Tutorial
  </title>
  </head>
  <body>
    <h1>DOM Lesson one </h1>
    <p> Hello world! </p>
  </body>
</html>
```

- <html> node is the root node
 - Has no parent node
- Parent node of the <head> and <body> nodes is the <html> node.
- Parent node of the "Hello world!" text node is the <p> node
- <html> node has two child nodes
 - <head> and <body>
- <head> node has one child node
 - <title> node
- <title> node has one child node
 - text node "DOM Tutorial"
- <h1> and <p> nodes are siblings
 - Both child nodes of <body>

HTML TREE STRUCTURE

- Follow the standard “tree” nomenclature
- Top node is called the root
- Every node, except the root, has exactly one parent node.
 - Root has none.
- A node can have any number of children
- Leaf is a node with no children
- Siblings are nodes with the same parent

ACCESSING HTML NODES

- `getElementById (<id>)`
- `getElementsByTagName(<tag>)`
- A combination of the above
 - Using the tree and parent/child relationship.

HTML PROPERTIES

- For any HTML element (node) x ,
 - $x.innerHTML$ - the inner “HTML” value of x
 - $x.innerText$ – the inner “text” value of x
 - $x.nodeName$ - the name of x
 - $x.nodeValue$ - the value of x
 - $x.parentNode$ - the parent node of x
 - $x.childNodes$ - the child nodes of x
 - $x.attributes$ - the attributes nodes of x

BACK TO THE EXAMPLE ...

- `document` - the current HTML document
- `getElementById("intro")` - the element with the id "intro"
- `childNodes[0]` - the first child of the element
- `nodeValue` - the value of the node (e.g., text)

HTML METHODS

- For any HTML element (node) x
 - `x.getElementById(id)`
 - get the element with a specified id
 - `x.getElementsByTagName(name)`
 - get all elements with a specified tag name. Tag = “**body**”, for example.
 - `x.appendChild(node)`
 - insert a child node to x
 - `x.removeChild(node)`
- Details can be found at
https://www.w3schools.com/js/js_htmldom_document.asp

HTML DOM – OBJECT MODEL

- Each node is an **object**.
- Objects have methods
- Can use methods to retrieve or change HTML content dynamically.
- We will cover HTML DOM again later.
⇒ Basis for Dynamic HTML (**DHTML**)

DHTML – DYNAMIC HTML

- Web requirements are very demanding.
 - Not just “**static**” requirements.
 - Check validity of input given on a web page.
 - Ability to manipulate data **dynamically** based on
 - User input
 - Already available data.
 - Provide animation
 - Highlight a text area with a different color.
 - Change behavior of images on mouse clicks, focus, etc.
- Solution: **DHTML**
 - Ability to change HTML content dynamically.

DHTML

- Components of HTML to support dynamic nature of content:
 - CSS – cascading style sheets
 - To present the data
 - HTML DOM
 - Ability to access and change different portions (e.g., head, body, input, etc.) of a web page.
 - Javascript
 - Run scripts for various purposes
 - Running scripts, creating cookies, animation, etc.
- This course is about Javascript.

HTML FORMS

- We covered some HTML tags earlier.
- HTML form
 - Another HTML tag
 - Useful to send information from browser to server
 - Can use other HTML tags
 - <input>
 - <button>
 - <submit>
 - <select> and <option>
 - <textarea>
- Javascript functions can be used to verify HTML forms' input

HTML FORM 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() == false) {
        document.getElementById("demo").innerHTML =
            inpObj.validationMessage;
    }
}
</script>
```