COMS W3101: SCRIPTING LANGUAGES: JAVASCRIPT (FALL 2017)

RAMANA ISUKAPALLI

RAMANA@CS.COLUMBIA.EDU

LECTURE-1

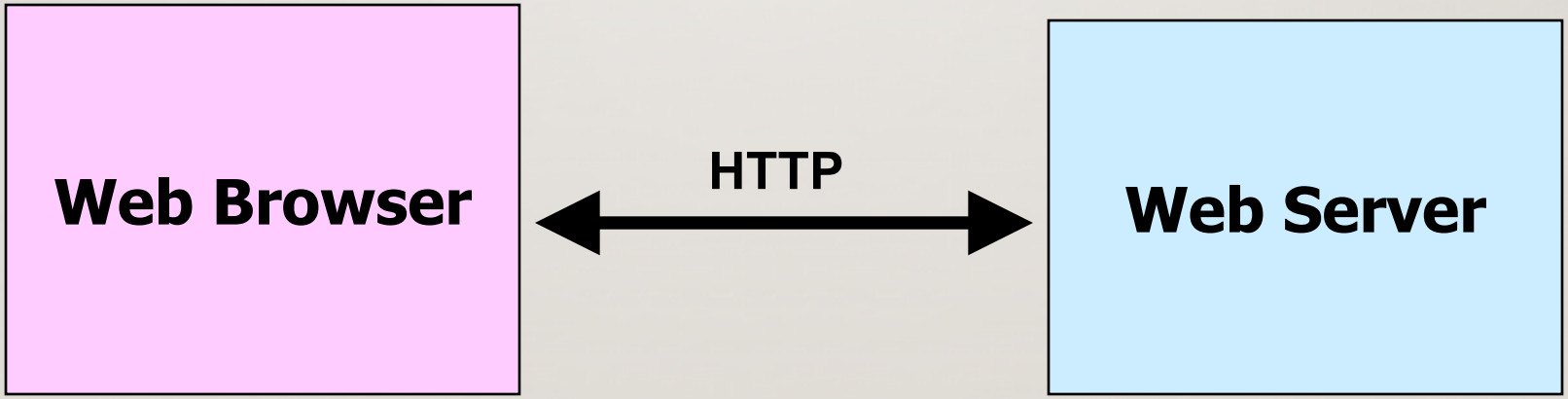
- Course overview
  - See http://www.cs.columbia.edu/~ramana
- Overview of HTML
  - Formatting, headings, images, colors, tables, forms, etc.
  - XHTML – difference with HTML
  - DHTML
    - What is it?
    - Why is it needed
- Javascript
  - Overview, what is it, why is it needed, etc.
  - How does 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
  - Browswer (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 formatsSimple 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
- 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>
          Sample 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>
- Form – <form>
- Image – <img>
- Lists –
- Ordered list –
- Unordered list --
- No support for frames – <noframes>

  - 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

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
  - values must be quoted
    - e.g., <table width="100%"
  - minimization is forbidden
    - <input checked="checked" /> instead of <input checked>

DHTML – DYNAMIC HTML

- Web requirements have became more 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 "Domain Object Model" (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.
- We will cover basic HTML first, before discussing Javascript.

JAVASCRIPT – OVERVIEW

- This course is concerned with client side JS
  - Executes on client (browser)
- Scripting – NOT compile/link.
- Helps provide dynamic nature of HTML pages.
- Included as part of HTML pages as
  - Regular code (viewable by user)
  - A file present in some location.
- NOTE: Javascript is NOT the same as JAVA

A SIMPLE JAVASCRIPT PROGRAM

```
<html>
  <head>
      <title> A simple Javascript program
      </title>
  </head>
  <body>
      <script>
         document.write ("A Simple Javascript program");
      </script>
  </body>
</html>
```

JAVASCRIPT CODE

- Javascript code in HTML
- Javascript code can be placed in
  - <head> part of HTML file
    - Code is NOT executed unless called in <body> part of the file.
  - <body> part of HTML file – executed along with the rest of body part.
  - Outside HTML file, location is specified.
    - Executed when called in <body>

WAYS OF DEFINING JAVASCRIPT CODE.

First:

```
<head>
    <script type="text/javascript">
   function foo(…) // defined here
    </script>
</head>

<body>
    <script type="text/javascript">
          foo(…)  // called here
    </script>
</body>
```

Second:

```
<head>
      …
</head>
<body>
      <script>
       function foo(…) // defined here
       {
              ..
       }
      foo( ) // Called here
      </script>
</body>
```

WAYS OF DEFINING JAVASCRIPT CODE, CONTD.

Third:

```
<head>
    // Any general location, that can be accessed.
    <script src="http://cnn.com/foo.js">
    </script>
</head>
<body>
<script>
    // Javascript code called here.
</script>
</body>
```

JAVASCRIPT – DATA TYPES

- Basic data types
  - number: E.g., 2, 3, 4, 10.6, 3.1415, 1.2e8, 3.2e-10
  - string: E.g., "abc", "Bob Doe"
  - boolean: true, false
- Complex data types
  - Objects
  - Functions

JAVASCRIPT – BASIC TYPES

- Not a strongly typed language
  - x = 5; x = "string" is perfectly acceptable.
- Case sensitive.
- A variable has a "var" prefix.
  - "var x = 5" is same as just, "x = 5".
- Re-declaration is possible
- Possible to mix and match while printing
  - document.write (6 + 10 + "xyz") → prints 16xyz
  - document.write ("xyz" + 6 + 10) → prints xyz610

JAVASCRIPT – OBJECTS

- var person = {
  firstName:"John",
  lastName:"Doe",
  email:“JohnDoe@gmail.com”,
  age: 38
  };
- To access values, use
  - object.propertyName or
  - object[“propertyName]
- Example
  - person[firstName]
  - person[“firstName”]

JAVASCRIPT – FINDING DATA TYPES

- Useful operator to find data types: typeof
  - typeof(10) → number
  - typeof(3.1415) → number
  - typeof("abcd") → string
  - name = "Bob"; typeof(name) → string
  - typeof(true) → boolean
  - value = false; typeof(value) → false
  - var x = 10; typeof(x) → number
  - typeof(x) → undefined.
  - typeof(person) → object

JAVASCRIPT – OPERATORS

- Arithmetic operators: Usual ones
  - +, -, *, /, %, ++, --
- Assignment operators
  - =, +=, -=, *=, /=, % =
- String operators:
  - + (concatenation operator)
- Comparison operators:
  - ==, !=, ===, >, <, >=, <=
- Logical operators
  - &&, ||, !
- Conditional operator
  - variable = (condition) ? value1: value2

JAVASCRIPT – CONTROL STATEMENTS

if statement:

```
if (cond1)
   <code-1>
else if (cond2)
   <code-2>
…
else
   <code-3>
```

Example:

```
if (x %2 == 0)
   document.write("x is a multiple of 2");
else if (x%3 == 0)
   document.write("x is a multiple of 3");
else
   document.write("x is not a multiple of 2 or 3);
```

JAVASCRIPT – CONTROL STATEMENTS

for statement:

```
for (i = 0; i < n; i++)
{
    // code
}
```

Example:

```
var x;
var mycars = new Array();
mycars[0] = "Saab";
mycars[1] = "Volvo";
mycars[2] = "BMW";
for (x in mycars)
 {
     document.write (mycars[x] + "<br />“);
 }
```

JAVASCRIPT – CONTROL STATEMENTS … CONTD.

do statement

```
do
{
    <code>
} while (cond);
```

while statement:

```
while (cond)
{
    <code>
}
```

switch statement:

```
switch (n)
{
   case n1:
        <code>
        break;
          …
   default:
        <code>
}
```

JAVASCRIPT – POPUP BOXES

- Alert box
  - alert (“alert text”);
- Confirm box
  - confirm (“confirm some text”);
- Prompt box
  - prompt (“prompt text”,“default value”)

JAVASCRIPT FUNCTIONS

Syntax

```
function <functionName> (params)
{
    // code
}
```

Note: Parameters do NOT have variable type.

1. Recall: Function definition can be in
   - <head> part of HTML file.
   - <body> portion of HTML file
   - An external file.
2. "return" value of the function is optional.

FUNCTIONS – EXAMPLE1

```
<html>
  <head>
    <title> Example of a simple function </title>
    <script type="text/javascript">
        function factorial (input)
         {
            product = 1;
            for (i=1; i <= input; i++)
                product *= i;
            document.write ("factorial of i " + product);
         }
    </script>
  </head>
  <body>
    <h1> Example of a simple function </h1>
      <script type="text/javascript">
        factorial (9);
      </script>
  </body>
</html>
```

FUNCTIONS – EXAMPLE2

```
<html>
  <head>
    <title>Browser Information example</title>
    <script type="text/javascript">
      function BrowserInfoFn( )
      {
        var browser = navigator.appName;
        var version = navigator.appVersion;
        var ver = parseFloat (version);
        document.write ("Broswer: " + browser + " version:“  + version + " ver: " + ver + "<br />");
      }
    </script>
  </head>
  <body>
    <h1>Browser Information example</h1>
    <script type="text/javascript">
      BrowserInfoFn( );
    </script>
    <hr>
</body>
</html>
```