WEB ENGINEERING

INSTRUCTED BY

ENGR. M. ASIF SHAIKH

Lecture 4

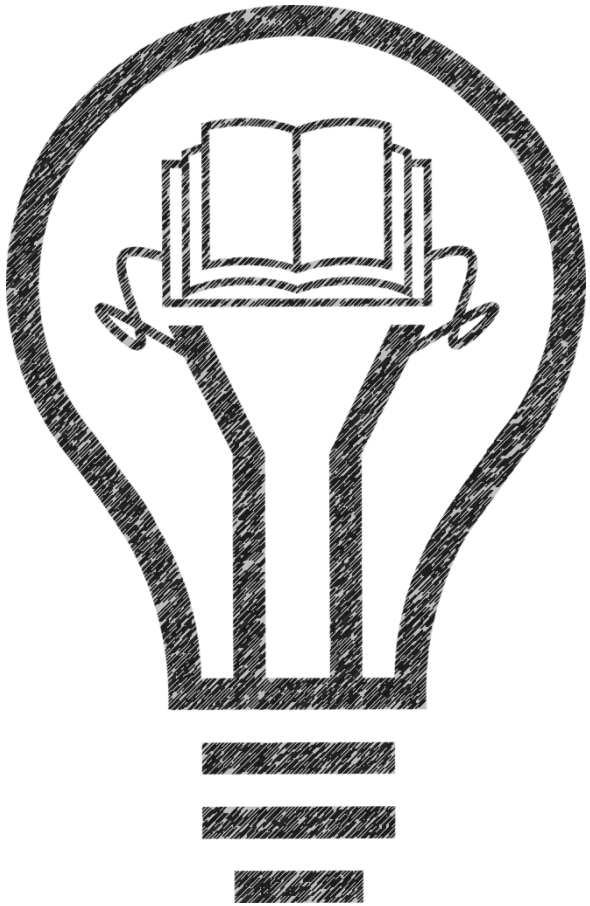
Javascript

Head Lines

- What is scripting languages
- What is JavaScript
- JavaScript main features
- Objects & Functions in JS
- Famous JavaScript Libraries
- Famous JavaScript Frameworks

Learning Objectives

- At the end of this session, students will be able to
  - Discribe JavaScript
  - Discuss JS Libraries
  - Discuss JS Frameworks
- This will cover CLO-1 & 2



Brainstorming

What is Scripting Language?

What's a Scripting Language?

- Language used to write programs that compute inputs to another language processor
  - One language embedded in another
    - Embedded JavaScript computes HTML input to the browser
    - Shell scripts compute commands executed by the shell

- Common characteristics of scripting languages
  - String processing – since commands often strings
  - Simple program structure, define things "on the fly"
  - Flexibility preferred over efficiency, safety
    - Is lack of safety a good thing? (Example: JavaScript used for Web applications…)

Why JavaScript?

- "Active" web pages
- Web 2.0
  - AJAX, huge number of Web-based applications
- Some interesting and unusual features
  - First-class functions - interesting
  - Objects without classes - slightly unusual
  - Powerful modification capabilities - very unusual
    - Add new method to object, redefine prototype, …
- Many security and correctness issues
- "The world's most misunderstood prog. language"

JavaScript History

- Developed by Brendan Eich at Netscape
  - Scripting language for Navigator 2
- Later standardized for browser compatibility
  - ECMAScript Edition 3 (aka JavaScript 1.5)
- Related to Java in name only
  - “JavaScript is to Java as carpet is to car”
  - Name was part of a marketing deal
- Various implementations available
  - SpiderMonkey C implementation (from Mozilla)
  - Rhino Java implementation (also from Mozilla)

Common Uses of JavaScript

- Form validation
- Page embellishments and special effects
- Navigation systems
- Basic math calculations
- Dynamic content manipulation
- Sample applications
  - Dashboard widgets in Mac OS X, Google Maps, Philips universal remotes, Writely word processor, hundreds of others…

JavaScript Objects

- JavaScript is an **object-based** language
  - It is **NOT object-oriented**
  - It has and uses objects, but does not support some features necessary for object-oriented languages
    - Class inheritance and polymorphism not supported
      - They can be "faked" but are not really there

Example 1: Add Two Numbers

```
<html>
      …
 <p>  … </p>
<script>
        var num1, num2, sum
        num1 = prompt("Enter first number")
        num2 = prompt("Enter second number")
        sum = parseInt(num1) + parseInt(num2)
        alert("Sum = " + sum)
</script>
        …
</html>
```

Example 2: Browser Events

```
<script type="text/JavaScript">
    function whichButton(event) {
        if (event.button==1) {
                alert("You clicked the left mouse button!") }
        else {
                alert("You clicked the right mouse button!")
        }}
</script>
…
<body onmousedown="whichButton(event)">
…
</body>
```

Mouse event causes page-defined function to be called

Other events: onLoad, onMouseMove, onKeyPress, onUnLoad

Language Basics

- JavaScript is case sensitive
  - onClick, ONCLICK, … are HTML, thus not case-sensitive
- Statements terminated by returns or semi-colons
  - x = x+1; same as x = x+1
- “Blocks” of statements enclosed in { …}
- Variables
  - Define using the var statement
  - Define implicitly by its first use, which must be an assignment
    - Implicit defn has global scope, even if occurs in nested scope!

JavaScript Blocks

- Use { } for grouping; not a separate scope

```
js> var x=3;
js> x
3
js> {var x=4; x}
4
js> x
4
```

- Not blocks in the sense of other languages

JavaScript Primitive Datatypes

- Boolean: true and false
- Number: 64-bit floating point
  - Similar to Java double and Double
  - No integer type
  - Special values NaN (not a number) and Infinity
- String: sequence of zero or more Unicode chars
  - No separate character type (just strings of length 1)
  - Literal strings using ' or " characters (must match)
- Special objects: null and undefined

Objects

- An object is a collection of named properties
- Think of it as an associative array or hash table
  - Set of name:value pairs
    - objBob = {name: “Bob", grade: 'A', level: 3};
  - Play a role similar to lists in Lisp / Scheme
- New members can be added at any time
    - objBob.fullname = 'Robert';
- Can have methods
- Can refer to this

Functions

- Functions are objects with method called “( )”
  - A property of an object may be a function (=method)
    - function max(x,y) { if (x>y) return x; else return y;};
    - max.description = “return the maximum of two arguments”;
  - Local declarations may appear in function body

- Call can supply any number of arguments
  - functionname.length : # of arguments in definition
  - functionname.arguments.length : # arguments in call
  - Basic types are passed by value, objects by reference

- “Anonymous” functions
  - (function (x,y) {return x+y}) (2,3);

Examples of Functions

- Curried functions
  - function CurriedAdd(x) { return function(y){ return x+y} };
  - g = CurriedAdd(2);

- Variable number of arguments
  - function sumAll() {
    var total=0;
    for (var i=0; i< sumAll.arguments.length; i++)
    total+=sumAll.arguments[i];
    return(total); }
  - sumAll(3,5,3,5,3,2,6)

JavaScript Libraries

- With many standard functionalities to help developers save time, programmers rely on JavaScript libraries.

- These fundamental assets help us create web pages using UI components, language utilities, math functions, and more.

1. **DOJO toolkit** 

- The Dojo is an open-source JavaScript library that helps develop cross-platform, JS, and Ajax-based websites in a faster manner.

2. **jQuery** 

- jQuery dramatically simplifies JS programming and is easy to learn and use. It is highly extensible and makes web pages load faster. jQuery wraps up a lot of standard functions making the job of the developer easy.

3. **Google Polymer** 

- Created by Google, Polymer is a JS library that allows developers to reuse HTML elements and create custom elements using HTML, CSS, and JS to create more interactive applications.

4. **D3.js** 

- D3 stands for Data-Driven Documents. With D3, you can apply data-driven transformations to DOM objects. The keyword with D3 is ‘data-driven,’ which means documents are manipulated depending on the data received.

5. **Pixi.js**

- Pixi js can create stunning digital content. This open-source, cross-platform 2D engine helps create games and interactive, animation-based websites.

6. **React**

- React is easy to understand and uses the JS library to build user interfaces for web applications. React is maintained by Facebook and a few other companies.

7. **PHP** 

- JsPHP is a Javascript library for PHP API to be available in the JS environment. It is open-source and provides a compelling interface for JS developers who work in PHP.

8. **MathJAX** 

- MathJAX, true to its name, is a cross-browser javascript library that can display math notations and uses markup.

9. **Parsley**

- We spend much time in front-end form validations on websites that need users to fill in information. Parsley library makes this form of the validation process simple.

JavaScript Frameworks

- A web development framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code.

- A JavaScript framework is an application framework written in JavaScript, where programmers can manipulate the functions and use them for their convenience.

1. **Angular**

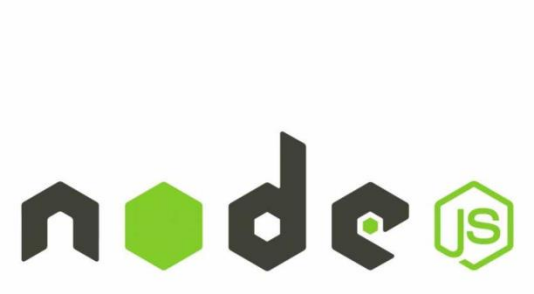
- One of the most powerful and efficient JavaScript frameworks, Angular is an open-source framework that is used for developing a Single Page Application (SPA).

- It extends the HTML into the application and interprets the attributes to perform data binding.

2. **React**

- Created by Facebook, the React framework has earned popularity very quickly.
- It is used to develop and operate the dynamic User Interface of web pages with high incoming traffic.
- It makes use of a virtual DOM, and hence, integration with any application is more straightforward.

3. Node.js



- Node.js is a server-side JavaScript run-time environment, which works on cross platforms and is open-source. The framework is capable of driving asynchronous I/O with its event-driven architecture. It works in the JavaScript Runtime environment and shows JAVA's similar properties like threading, packaging, and forming loops.

AJAX

- Ajax is a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications.

- With Ajax, web applications can send and retrieve data from a server asynchronously without interfering with the display and behavior of the existing page.

- AJAX is not a programming language.
- AJAX just uses a combination of:
  - A browser built-in XMLHttpRequest object (to request data from a web server)
  - JavaScript and HTML DOM (to display or use the data)

- AJAX is a developer's dream, because you can:
  - Read data from a web server - after the page has loaded
  - Update a web page without reloading the page
  - Send data to a web server - in the background

JSON

- JSON stands for JavaScript Object Notation
- JSON is a lightweight format for storing and transporting data
- JSON is often used when data is sent from a server to a web page
- JSON is "self-describing" and easy to understand

- **JSON Syntax Rules**
  - Data is in name/value pairs
  - Data is separated by commas
  - Curly braces hold objects
  - Square brackets hold arrays

```
"employees“  : [
    {"firstName":"John",  "lastName":"Doe"},
    {"firstName":"Anna",  "lastName":"Smith"},
    {"firstName":"Peter",  "lastName":"Jones"}
]
```

- In the example above, the object "employees" is an array. It contains three objects.

- Each object is a record of a person (with a first name and a last name).

Why Use JSON?

- The JSON format is syntactically similar to the code for creating JavaScript objects. Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.

- Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.

- JavaScript has a built in function for converting JSON strings into JavaScript objects:
  - JSON.parse()

- JavaScript also has a built in function for converting an object into a JSON string:
  - JSON.stringify()

Useful Links

- Scripting VS Programming Languages
  https://www.youtube.com/watch?v=g0Q-VWBX5Js
- JS Frameworks
  https://www.youtube.com/watch?v=Ka77djMkSwg
  https://hackr.io/blog/best-javascript-frameworks
- JS Libraries
  https://www.youtube.com/watch?v=ZrFfPIPA5gs
  https://hackr.io/blog/top-javascript-libraries
- JavaScript Form Validation
- https://www.youtube.com/watch?v=Gku9iMSMbWg&list=PLwGdqUZWnOp0j0wDMDapjCSS6YdPvmUMJ

- AJAX

  https://www.w3schools.com/js/js_ajax_intro.asp

- JSON

  https://www.w3schools.com/js/js_json_intro.asp