

Web Technologies and Programming Lecture 21

jQuery, Data Description and Transformation, XML, DOM

Summary of the previous lecture

- The history object
- The navigator object
- The user browser
- The browser Engine
- The browser platform
- The browser language
- The screen object
- The form object
 - Accessing from element
 - Setting form element

Summary of the previous lecture

- Validating form data
- DATA Validation

Outline

- Introduction to jQuery
- DOM and jQuery
- jQuery Selectors and events
- DOM Objects
- XML
- Components of XML

1. Introduction to jQuery

- "jQuery is a lightweight JavaScript library that emphasizes interaction between JavaScript and HTML."
- Developed by John Resig at Rochester Institute of Technology
- "jQuery is free, open source software Dual-licensed under the MIT License and the GNU General Public License."
- "It's all about simplicity. Why should web developers be forced to write long, complex, book-length pieces of code when they want to create simple pieces of interaction?"

1.1 Why use jQuery?

- Rich Internet Applications (RIA)
- Dynamic HTML (DHTML)
- "Unobtrusive" JavaScript separation of behavior from structure
- Allows adding JavaScript to your web pages
- Much easier to use
- Eliminates cross-browser problems
- Mobile First Web Development

1.1 Why use jQuery?

- jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.
- The jQuery library contains the following features:
 - HTML/DOM manipulation
 - CSS manipulation
 - HTML event methods
 - Effects and animations
 - AJAX (Asynchronous JavaScript and XML.

2. jQuery and DOM

- DOM stands for Document Object Model
- "The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents.
- Aspects of the DOM (such as its "Elements") may be addressed and manipulated within the syntax of the programming language in use."

2. jQuery and DOM



2. jQuery and DOM

- jQuery is "DOM scripting"
- Hierarchal structure of a web page
- You can add and subtract DOM elements on the fly
- You can change the properties and contents of DOM elements on the fly
- jQuery uses DOM elements to manipulate a web page.

3. jQuery Syntax

- The jQuery syntax is tailor made for **selecting** HTML elements and performing some **action** on the element(s).
 - Basic syntax is: \$(selector).action()
 - A \$ sign to define/access jQuery
 - A (*selector*) to "query (or find)" HTML elements
 - A jQuery *action*() to be performed on the element(s)
- Examples:
 - \$(this).hide() hides the current element.
 - \$("p").hide() hides all elements.
 - \$(".test").hide() hides all elements with class="test".
 - \$("#test").hide() hides the element with id="test".

3.1 jQuery Selectors

- Query selectors allow you to select and manipulate HTML element(s).
- jQuery selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more.
- Majorly there are three types of selectors:
 - Element selector
 - Id selector
 - Class selector
- All selectors in jQuery start with the dollar sign and parentheses: \$().

3.1.1 jQuery element Selectors

• The jQuery element selector selects elements based on the element name.

• Example:

- \$("p")
- Using this all tags from a webpage will be selected.

3.1.2 jQuery id Selectors

- The jQuery **#id selector** uses the id attribute of an HTML tag to find the specific element.
- An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.
- Example:
 - \$("#test")
 - Using this tag with id of "text" from a webpage will be selected.

3.1.3 jQuery class Selectors

- The jQuery class selector finds elements with a specific class.
- Example:
 - \$(".test")
 - Using this, tags with class of "text" from a webpage
 will
 webpage

3.1.3 jQuery class Selectors

Syntax	Description	
\$("*")	Selects all elements	
\$(this)	Selects the current HTML element	
\$("p.intro")	Selects all elements with class="intro"	
\$("p:first")	Selects the first element	
\$("ul li:first")	Selects the first element of the first 	
\$("ul li:first-child")	Selects the first element of every 	
\$("[href]")	Selects all elements with an href attribute	
\$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"	
\$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"	
\$(":button")	Selects all <button> elements and <input/> elements of type="button"</button>	
\$("tr:even")	Selects all even elements	
\$("tr:odd")	Selects all odd elements	

3.2 jQuery events

- All the different visitor's actions that a web page can respond to are called events.
- An event represents the precise moment when something happens.

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

3.2.1 jQuery events syntax

- To assign a click event to all paragraphs on a page, you can do this:
 \$("p").click();
- The next step is to define what should happen when the event fires. You must pass a function to the event:
 - \$("p").click(function(){
 // action goes here!!
 });

3.2.2 Commonly used events

- \$(document).ready()
 - The method allows us to execute a function when the document is fully loaded.
- click()
 - The function is executed when the user clicks on the HTML element.
- dblclick()
 - he function is executed when the user double-clicks on the HTML element
- hover()
 - The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.
 - The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element
- blur()
 - The function is executed when the form field loses focus.

3.2.3 The on() method

• The on() method attaches one or more event handlers for the selected elements.

• Example:

```
- $("p").on({
    mouseenter: function(){
        $(this).css("background-color", "lightgray");
    },
    mouseleave: function(){
        $(this).css("background-color", "lightblue");
    },
    click: function(){
        $(this).css("background-color", "yellow");
    }
});
```

3.3 jQuery Effects

- We can apply some commonly used effects to our HTML elements by just calling a simple jQuery function.
- Some Common Effects are:
 - Hide() / show()
 - \$("p").hide();
 - \$("p").show();
 - fadeIn() / fadeOut()
 - slideDown() / slideUp()

4. DOM (Document Object Model)

- A Web page is made dynamic by applying JavaScript processing to the XHTML elements on that page
 - XHTML tags are also software objects , having properties and methods that can be programmed
 - These objects are programmed through JavaScript processing routines to make Web pages dynamic
- The DOM is the programming interface to the XHTML objects appearing on a Web page
- All XHTML 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.
- The XHTML DOM is platform and language independent.
 - It can be used by any programming language like Java, JavaScript, and VBScript.

4. DOM (Document Object Model)

- DOM stands for Document Object Model
- "The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents.
- Aspects of the DOM (such as its "Elements") may be addressed and manipulated within the syntax of the programming language in use."

4. DOM (Document Object Model) The Hierarchy



4.1 DOM (Document Object Model) Objects

- There are majorly 5 types of DOM Objects:
 - Window
 - Navigator
 - Location
 - History
 - Document

4.1.1 Window Object

- The window object is the "master" DOM object at the top of the DOM hierarchy
- Useful properties:
 - length: number of frames in window
 - frames: an array of window objects, one for each frame
 - parent: Since frames are window objects, sometimes parent window is needed
- Examples:
 - window.document : if frameless, accesses the top level document. If frames, accesses the top frame's document
 - window.frame[1].document : Access the document contained in the first frame
 - frame[1].parent.document : Access the document contained in the parent frame

4.1.2 Navigator Object

- Contains information about the browser
- Can be accessed as window.navigator or just navigator
- Useful properties:
 - appName: name of browser used
 - appVersion: version of browser used
 - platform: operating system in use
 - cookieEnabled: can the browser store cookies?

4.1.3 Location Object

- Contains information about the current URL
- Can be accessed as window.location or just location
- Useful properties:
 - href: retrieves entire URL
 - host: retrieves just the domain name (ex: yahoo.com)
 - pathname: retrieves just the path inside the domain (page name is at end)
 - hash: retrieves the anchor

4.1.4 History Object

- Contains information on the URLs that the browser has visited in this session within a window
- Can be accessed as window.history or just history
- Useful properties: next, previous (tells you the URL, but won't direct you there)
- Useful methods:
 - back: same as pressing the back arrow button
 - forward: same as pressing the forward arrow button
 - go: go back or forward a given number of pages; to go back
 3 pages:
 - history.go(-3);

4.1.5 Document Object

- This is the typically the most accessed object
- You can access all items in the document window through the document object
 - Forms, tables, paragraphs, lists, images, etc.
 - Consult a reference for properties and methods
- Frameless document: Access a
 - window.document or document
- Document contained in a frame:
 - window.frame[x].document, where x is the number or name of the frame

5. XML (Background)

- SGML (Standard Generalized Markup Language)
 - ISO Standard, 1986, for data storage & exchange
 - Metalanguage for defining languages (through DTDs)
 - A famous SGML language: HTML
 - Separation of content and display
 - Used in U.S. gvt. & contractors, large manufacturing companies, technical info. Publishers,...
 - SGML reference is 600 pages long
- XML
 - W3C recommendation in 1998
 - Simple subset (80/20 rule) of SGML: "ASCII of the Web", "Semantic Web"
 - XML specification is 26 pages long

5. XML

- XML stands for EXtensible Markup Language
- A meta-language for descriptive markup: you invent your own tags
- XML uses a **Document Type Definition (DTD) or an XML** Schema to describe the data
 - XML with a DTD or XML Schema is designed to be selfdescriptive
- Built-in internationalization via Unicode
- Built-in error-handling
 - A forgotten tag, or an attribute without quotes renders an XML document unusable
- Tons of support from the big IT companies

5.1 Why we use XML?

- Much of shareable data reside in computer systems and databases in incompatible formats
 - use conflicting hardware and/or software.
- One of the most time-consuming challenges for developers has been to exchange data between such systems over the Internet
- Converting the data to XML can greatly reduce the complexity and create data that can be read by many different applications
 - XML data is stored in plain text format hardware and software independent
- XML can be used to create new languages
 - Allows us to define our own markup languages

5.2 HTML and XML.

• XML is not a replacement for HTML

- In future Web development, XML is likely to be used to describe data while HTML will be used to format and display the same data (one interpretation of XML)
- XML and HTML were designed with different goals
 - XML was designed to describe data and to focus on what data is
 - XML describes only content, or "meaning"
 - HTML was designed to display data and to focus on how data looks.
 - HTML describes both structure (e.g. , <h2>,) and appearance (e.g.
, , <i>)
- XML is for computers while HTML is for humans
 - XML is used to mark up data so it can be processed by computers
 - HTML is used to mark up text so it can be displayed to users

5.3 Benefits of XML.

- Open W3C standard non-proprietary
- Representation of data across heterogeneous environments
 - Cross platform
 - Allows for high degree of interoperability
 - E.g., ability to exchange data between incompatible applications with incompatible data formats
- Strict rules that make it relatively easy to write XML parsers
 - Syntax
 - Structure
 - Case sensitive
- XML can make data more useful
 - s/w, h/w and application independence of XML makes data available to more users not only HTML browsers

5.3 Benefits of XML....

- XML tags are not predefined
 - You must "invent" your own tags
 - The tags used to mark up HTML documents and the structure of HTML documents are predefined
 - The author of HTML documents can only use tags that are defined in the HTML standard
- XML allows the author to define his own tags and his own document structure

5.4 Components of XML.

- XML declaration
- Processing instructions
 - Encoding specification (Unicode by default)
 - Namespace declaration
 - Schema declaration

Elements

- Each element has a beginning and ending tag
 - <TAG_NAME>...</TAG_NAME>
- Elements can be empty (<TAG_NAME />)

Attributes

- Describes an element; e.g. data type, data range, etc.
- Can only appear on beginning tag

5.4 Components of XML.



5.5 XML Declaration.

- The XML declaration looks like this: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
 - The XML declaration is not required by browsers, but is required by most XML processors (so include it!)
 - If present, the XML declaration must be first--not even whitespace should precede it
 - Note that the brackets are <? and ?>
 - The version attribute is required
 - encoding can be "UTF-8" (ASCII) or "UTF-16" (Unicode), or something else, or it can be omitted
 - An XML document is standalone if it makes use of no external markup (DTD) declarations
 - Default value for this attribute is no

5.6 XML Elements.

- An XML element is everything from the element's start tag to the element's end tag
- XML Elements are extensible and they have relationships
 - Related as parents and children
- XML Elements have simple naming rules
 - Names can contain letters, numbers, and other characters
 - Names must not start with a number or punctuation character
 - Names must not start with the letters xml (or XML or Xml ..)
 - Names cannot contain spaces

5.7 XML Attributes.

- XML elements can have attributes
- Data can be stored in child elements or in attributes
- Should you avoid using attributes?
 - Here are some of the problems using attributes:
 - attributes cannot contain multiple values (child elements can)
 - attributes are not easily expandable (for future changes)
 - attributes cannot describe structures (child elements can)
 - attributes are more difficult to manipulate by program code
 - attribute values are not easy to test against a Document Type Definition (DTD) - which is used to define the legal elements of an XML document
- Experience shows that attributes are handy in HTML but child elements should be used in their place in XML
 - Use attributes only to provide information that is not relevant to the data

5.8 XML Validation.

- There is a difference between a well-formed XML document and a valid XML document
- A well-formed XML document is one with correct XML syntax
- XML syntax is constrained by a grammar (DTD or Schema) that governs the permitted tag names, attachment of attributes to tags, and so on.
- A well-formed XML document that also conforms to a given DTD (Data type defination) or schema is said to be valid.
 - Every valid XML document is well-formed but the reverse is not necessarily the case

5.9 Sample XML Document.

- <?xml version='1.0'?>
- <bookstore>
- <book genre='autobiography' publicationdate='1981'
- ISBN='1-861003-11-0'>
- <title>The Autobiography of Benjamin Franklin</title>
- <author>
- <first-name>Benjamin</first-name>
- <last-name>Franklin</last-name>
- </author>
- <price>8.99</price>
- </book>
- <book genre='novel' publicationdate='1967' ISBN='0-201-63361-2'>
- <title>The Confidence Man</title>
- <author>
- <first-name>Herman</first-name>
- <last-name>Melville</last-name>
- </author>
- <price>11.99</price>
- </book>
- </bookstore>

5.9 XML Writing Rules.

- There must be one, and only one, root element
- All XML elements must have a closing tag
- Sub-elements must be properly nested
- Attributes are optional
 - Defined by an optional schema
- Attribute values must be enclosed in "" or "
- Processing instructions are optional
- XML is case-sensitive

5.10 XML Browser Support

- Netscape 6 supports XML
- Internet Explorer 5.0 supports the XML 1.0 standard
- Internet Explorer 5.0 and greater has the following XML support:
 - Viewing of XML documents
 - Displaying XML with CSS
 - Transforming and displaying XML with XSL
 - XML embedded in HTML as Data Islands
 - Binding XML data to HTML elements
 - Access to the XML DOM
 - Full support for W3C DTD standards

Summary

- Introduction to jQuery
 - Why use jQuery
- Jquery:
 - Syntax
 - Selectors
 - element Selectors
 - class Selectors
 - events
 - Effects

Summary

- DOM
- DOM Objects
 - Window
 - Navigator
 - Location
 - History
 - Document
- XML
- Components of XML

THANK YOU