Introduction to XML
e<u>X</u>tensible <u>M</u>arkup <u>L</u>anguage

What is XML

- XML stands for eXtensible Markup Language.
- A markup language is used to provide information about a document.
- Tags are added to the document to provide the extra information.
- HTML tags tell a browser how to display the document.
- XML tags give a reader some idea what some of the data means.

Components: XML Documents

- Elements
- Attributes
- plus some other details

Example of an HTML Document

```
<html>
   <head><title>Example</title></head.
<body>
   <h1>This is an example of a page.</h1>
   <h2>Some information goes here.</h2>
</body>
</html>
```

Example of an XML Document

```
<?xml version=“1.0”/>
<address>
   <name>Alice Lee</name>
   <email>alee@aol.com</email>
   <phone>212-346-1234</phone>
   <birthday>1985-03-22</birthday>
</address>
```

Difference Between HTML and XML

- HTML tags have a fixed meaning and browsers know what it is.
- XML tags are different for different applications, and users know what they mean.
- HTML tags are used for display.
- XML tags are used to describe documents and data.

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number=“1” title=“Introduction”>
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number=“1” title=“Introduction”>
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Freely definable tags

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Start Tag

End Tag

Element

Content of the Element (Subelements and/or Text)

A Simple XML Document

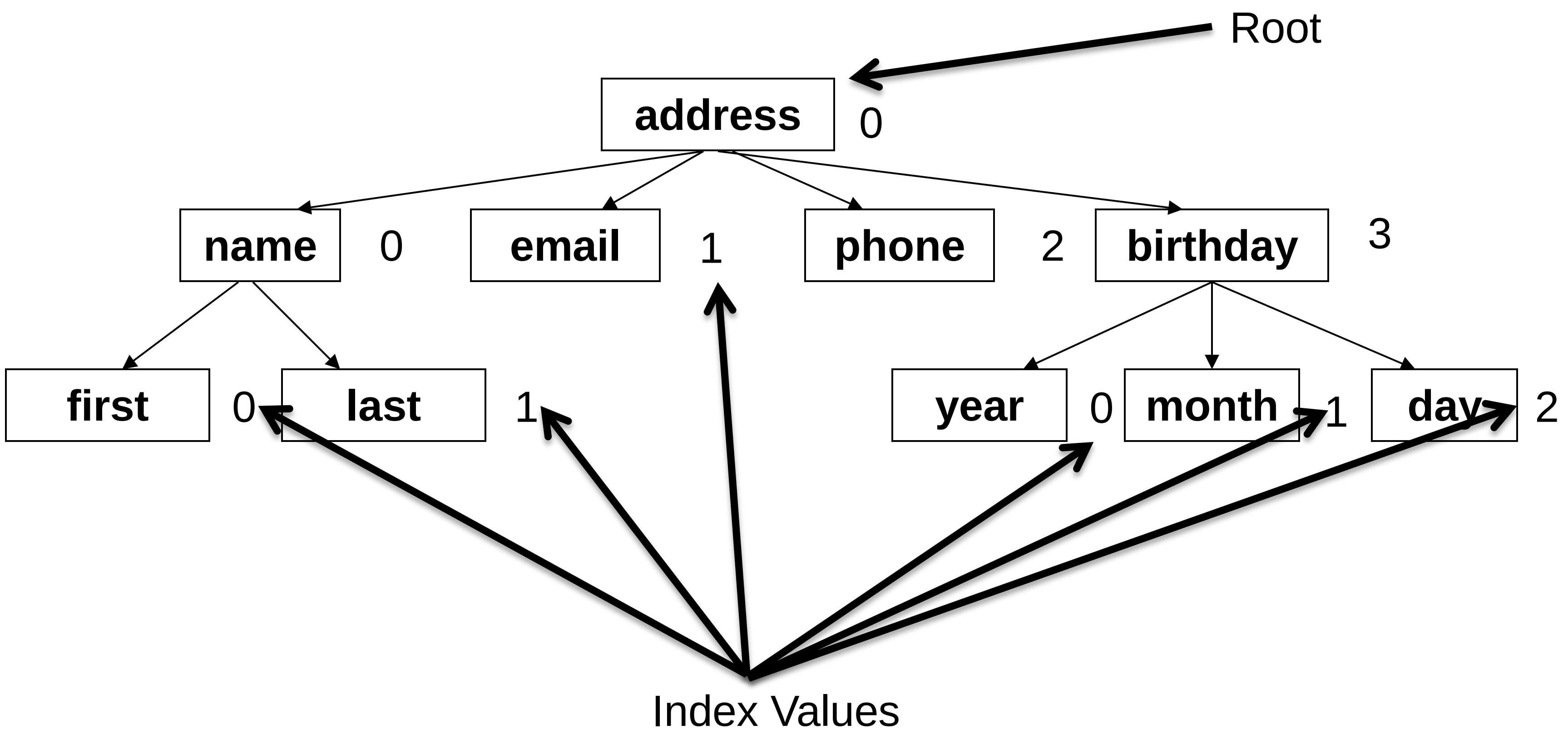
```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Attributes with **name** and **value**

Elements in XML Documents

- Freely definable **tags**: **`article`**, **`title`**, **`author`**
- **Elements**: **`<article> ... </article>`**
- Elements have a **name** (**`article`**) and a **content** (. . .)
- Elements may be nested.
- Elements may be empty: **`<this_is_empty/>`**
- Element content is typically parsed character data (PCDATA), i.e., strings with special characters, and/or nested elements (*mixed content* if both).
- Each XML document has exactly one root element and forms a tree.
- Elements with a common parent are ordered.

XML Trees


Root
address
0
name
0
email
1
phone
2
birthday
3
first
0
last
1
year
0
month
1
day
2
Index Values

Well-Formed XML Documents

- Every start tag has a matching end tag.
- Elements may nest, but must not overlap.
- There must be exactly one root element.
- Attribute values must be quoted.
- An element may not have two attributes with the same name.
- Comments and processing instructions may not appear inside tags.
- No unescaped `<` or `&` signs may occur inside character data.

Well-Formed XML Documents

- Every start tag has a matching end tag.
- Elements may nest, but must not overlap.
- There must be exactly one root element.
- Attribute values must be quoted.
- An element may not have to attributes with the same name.
- Comments and processing instructions may not appear inside tags.
- No unescaped `<` or `&` signs may occur inside character data.

Advantages of XML

- XML is text (Unicode) based.
  - Takes up less space.
  - Can be transmitted efficiently.
- One XML document can be displayed differently in different media.
  - Html, video, CD, DVD,
  - You only have to change the XML document in order to change all the rest.
- XML documents can be modularized. Parts can be reused.

XML Rules

- Tags are enclosed in angle brackets.
- Tags come in pairs with start-tags and end-tags.
- Tags must be properly nested.
  - **<name><email>…</name></email> is not allowed.**
  - **<name><email>…</email><name> is.**
- Tags that do not have end-tags must be terminated by a ‘/’.
  - <br /> is an html example.

More XML Rules

- Tags are case sensitive.
  - **<address> is not the same as <Address>**
- XML in any combination of cases is not allowed as part of a tag.
- Tags may not contain ‘<‘ or ‘&’.
- Tags follow Java naming conventions, except that a single colon and other characters are allowed. They must begin with a letter and may not contain white space.
- Documents must have a single *root* tag that begins the document.

XML Example Revisited

```
<?xml version=“1.0”/>
<address>
    <name>Alice Lee</name>
    <email>alee@aol.com</email>
    <phone>212-346-1234</phone>
    <birthday>1985-03-22</birthday>
</address>
```

- Markup for the data aids understanding of its purpose.
- A flat text file is not nearly so clear.

```
Alice Lee
alee@aol.com
212-346-1234
1985-03-22
```

Expanded Example

```
<?xml version = “1.0” ?>
<address>
    <name>
        <first>Alice</first>
        <last>Lee</last>
    </name>
    <email>alee@aol.com</email>
    <phone>123-45-6789</phone>
    <birthday>
        <year>1983</year>
        <month>07</month>
        <day>15</day>
    </birthday>
</address>
```

Some Javascript functions for XML

XMLHttpRequest: Create a request object for accesing XML data through server.

Var xml=new XMLHttpRequest();

responseText: Returns text from a file.

responseXML: returns the root of XML file

Open(): open a file

Send(): send a file on server.

getElementsByTagName(“TAG”): returns the element corresponding to tag “TAG”

Introduction to jQuery

- jQuery is a lightweight JavaScript Library
- It is an open-source that simplifies the interaction between HTML and JavaScript.
- jQuery contains all common functions used in HTML/JavaScript

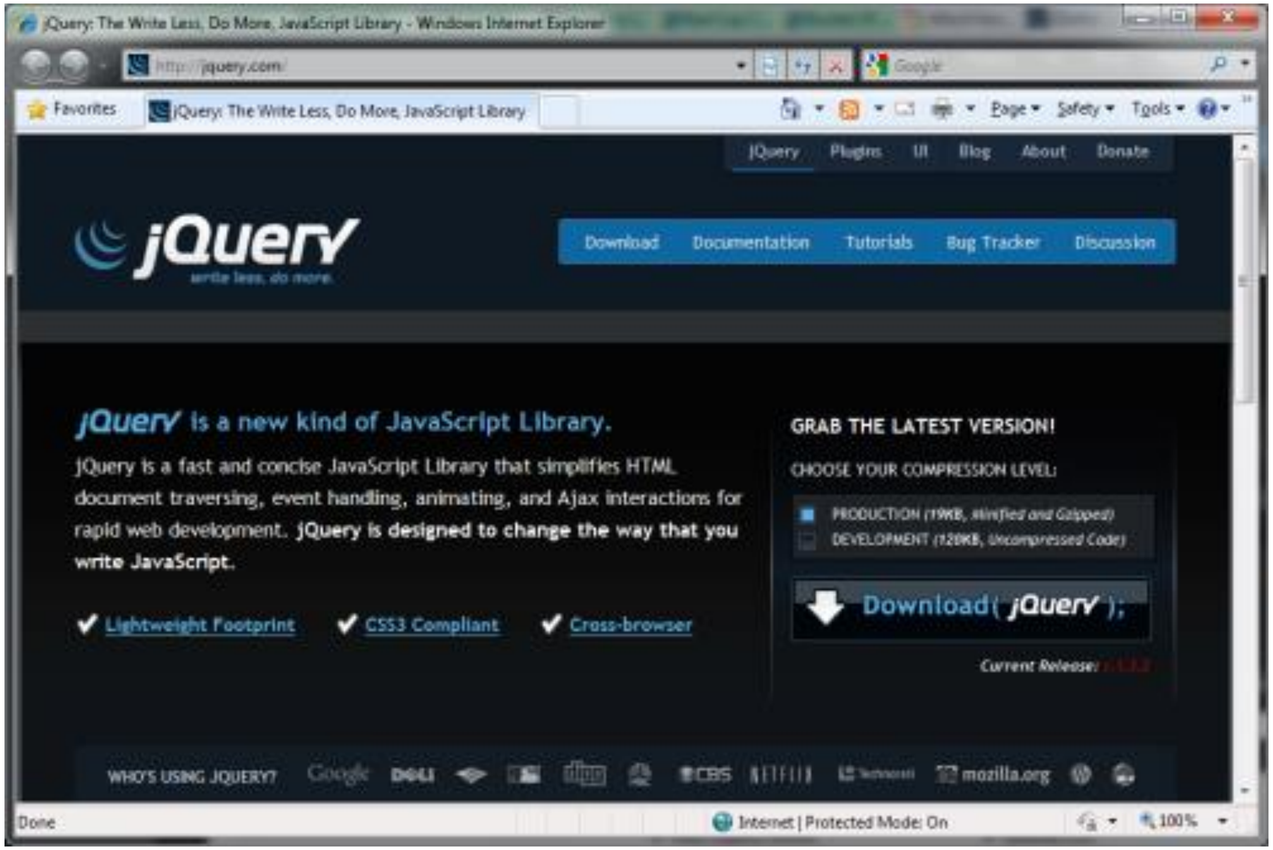
Features

- Easy to read and understand
- Programming support to OOP and EDP.
- It support the CSS.
- It has a great community
- It has great documentation

Getting Started

Download the latest version from

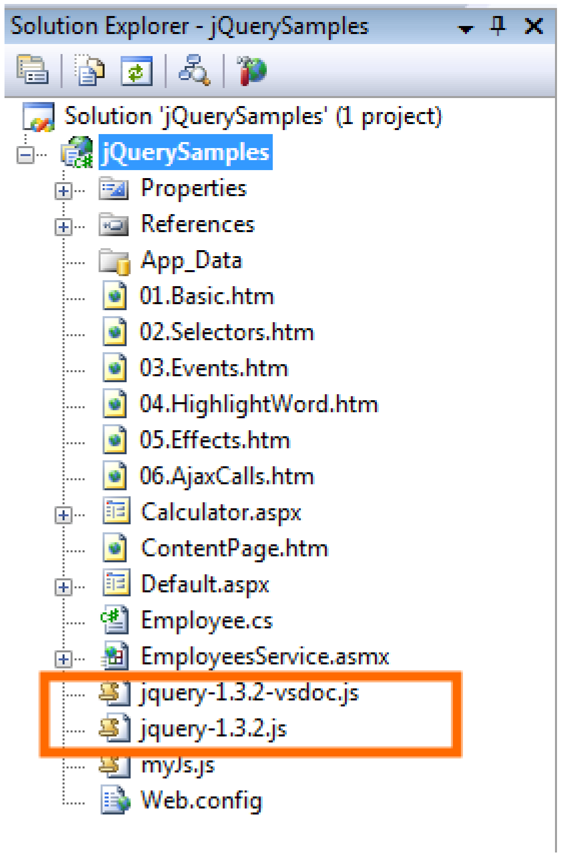
http://jquery.com



jQuery is a new kind of JavaScript Library.

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. **jQuery is designed to change the way that you write JavaScript.**

Save the file as **jquery.js** into your application folder


Solution Explorer - jQuerySamples

Solution 'jQuerySamples' (1 project)
- jQuerySamples
  - Properties
  - References
  - App_Data
  - 01.Basic.htm
  - 02.Selectors.htm
  - 03.Events.htm
  - 04.HighlightWord.htm
  - 05.Effects.htm
  - 06.AjaxCalls.htm
  - Calculator.aspx
  - ContentPage.htm
  - Default.aspx
  - Employee.cs
  - EmployeesService.asmx
  - jquery-1.3.2-vsdoc.js
  - jquery-1.3.2.js
  - myJs.js
  - Web.config

Reference it in your markup

```
<script src="jquery.js"/>
```

jQuery Core Concepts

The Magic $() function

```
var el = $(“<div/>”)
```

Create HTML elements on the fly

The Magic $() function

```
$(window).width()
```

Manipulate existing DOM elements

The Magic $() function

```
$("div").hide();

$("div", $("p")).hide();
```

Selects document elements

The Magic $() function

```
$(function(){…});
```

Fired when the document is ready for programming.

Better use the full syntax:

```
$(document).ready(function(){…});
```

The full name of $() function is

```
jQuery("div");
```

It may be used in case of conflict with other frameworks.

The library is designed to be isolated

```
(function(){
 var
    jQuery=window.jQuery=window.$=function(){
        // …
    };
 })();
```

jQuery uses closures for isolation

Almost every function returns jQuery, which provides a fluent programming interface and chainability:

```
$(“div”).show()
      .addClass(“main”)
      .html(“Hello jQuery”);
```

Three Major Concepts of jQuery

The $() function

Get > Act

Chainability

All Selector

```
$(“*”)            // find everything
```

Selectors return a pseudo-array of jQuery elements

Basic Selectors

By Tag:

```
$(“div”)

// <div>Hello jQuery</div>
```

By ID:

```
$(“#usr”)

// <span id=“usr”>John</span>
```

By Class:

```
$(“.menu”)

// <ul class=“menu”>Home</ul>
```

Yes, jQuery implements CSS Selectors!

Over to Programming

By Element

By ID:

By Class: