Session 17

JavaScript
Part 2

1

W3C DOM Reading and Reference

- Background and introduction

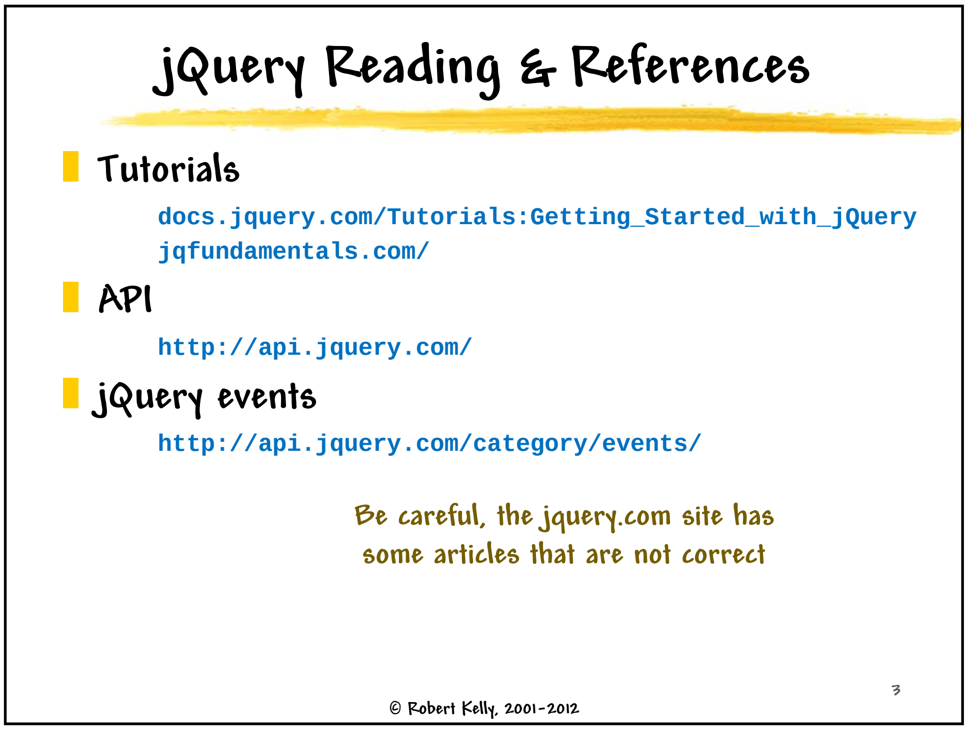
  www.w3schools.com/HTMLDOM/default.asp
- Reading – a good tutorial on the use of W3C DOM to modify html

  www.builderau.com.au/program/javascript/soa/Accessing-form-data-via-JavaScript-and-the-DOM/0,339028434,339277582,00.htm
- Reference:
  - JavaScript DOM properties – Flanagan book and

    www.javascriptkit.com/domref/elementproperties.shtml
  - W3C General spec

    www.w3.org/TR/DOM-Level-2/core.html

2



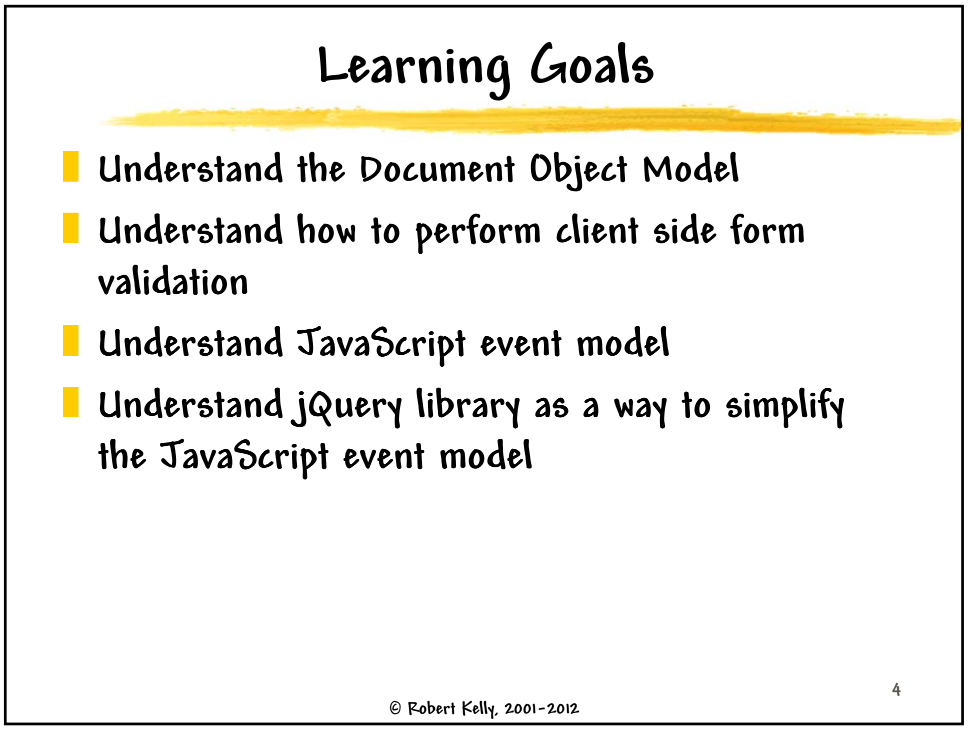
jQuery Reading & References

- Tutorials
  - docs.jquery.com/Tutorials:Getting_Started_with_jQuery
  - jqfundamentals.com/
- API
  - http://api.jquery.com/
- jQuery events
  - http://api.jquery.com/category/events/

Be careful, the jquery.com site has some articles that are not correct

© Robert Kelly, 2001-2012

3

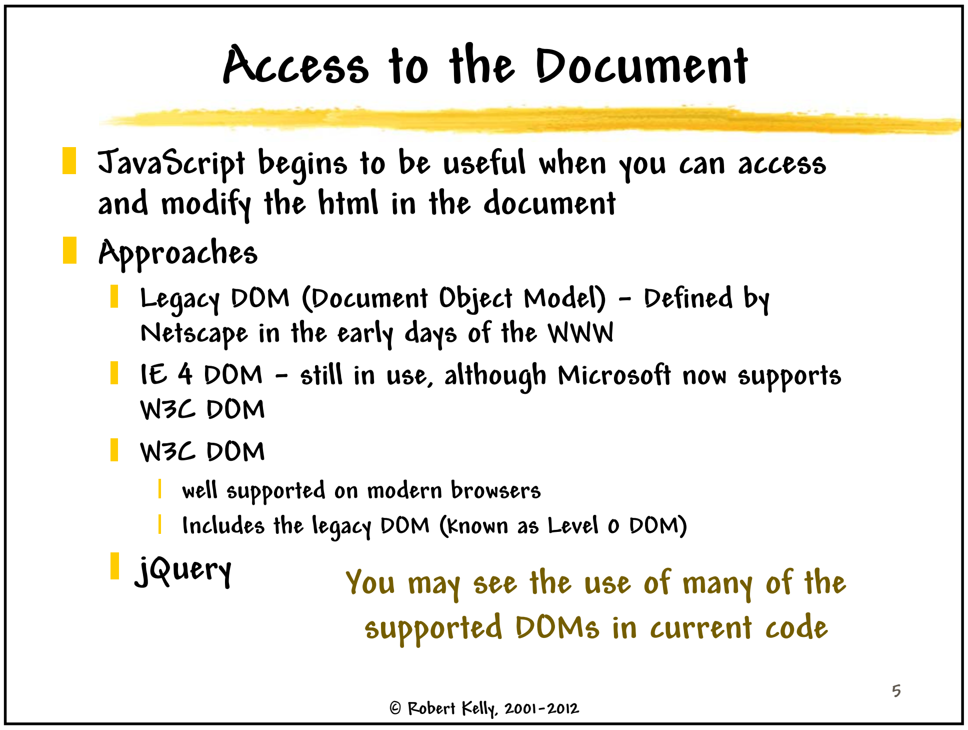


Learning Goals

- Understand the Document Object Model
- Understand how to perform client side form validation
- Understand JavaScript event model
- Understand jQuery library as a way to simplify the JavaScript event model

© Robert Kelly, 2001-2012

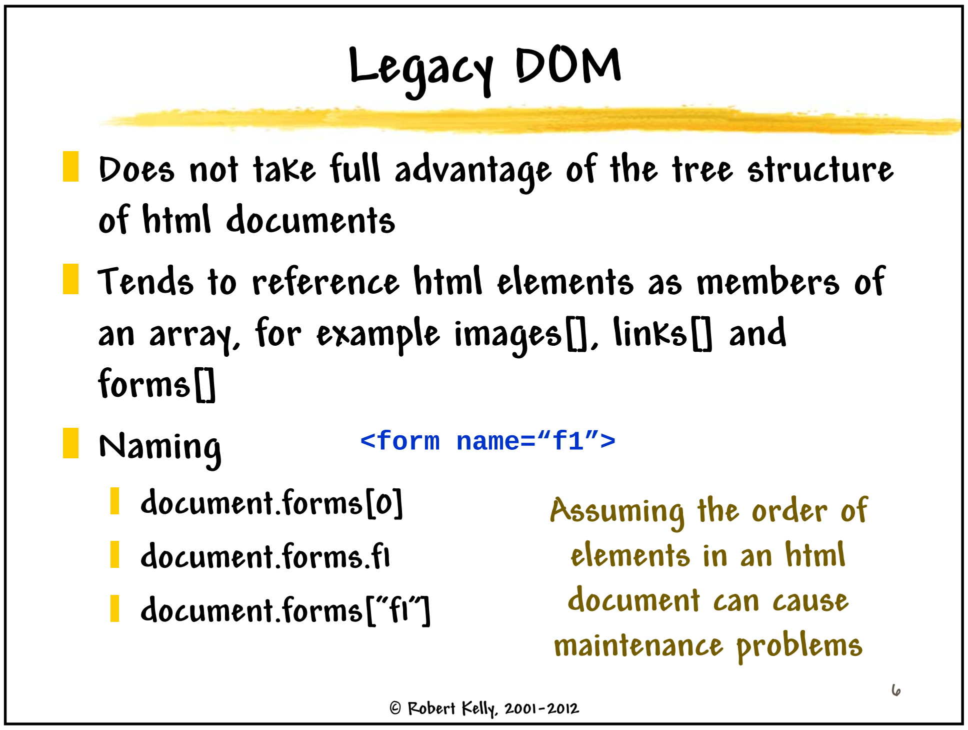
4



Access to the Document

- JavaScript begins to be useful when you can access and modify the html in the document
- Approaches
  - Legacy DOM (Document Object Model) – Defined by Netscape in the early days of the WWW
  - IE 4 DOM – still in use, although Microsoft now supports W3C DOM
  - W3C DOM
    - well supported on modern browsers
    - Includes the legacy DOM (known as Level 0 DOM)
  - jQuery

You may see the use of many of the supported DOMs in current code



Legacy DOM

- Does not take full advantage of the tree structure of html documents
- Tends to reference html elements as members of an array, for example images[], links[] and forms[]
- Naming
  - document.forms[0]
  - document.forms.f1
  - document.forms["f1"]

```
<form name="f1">
```

Assuming the order of elements in an html document can cause maintenance problems

W3C DOM

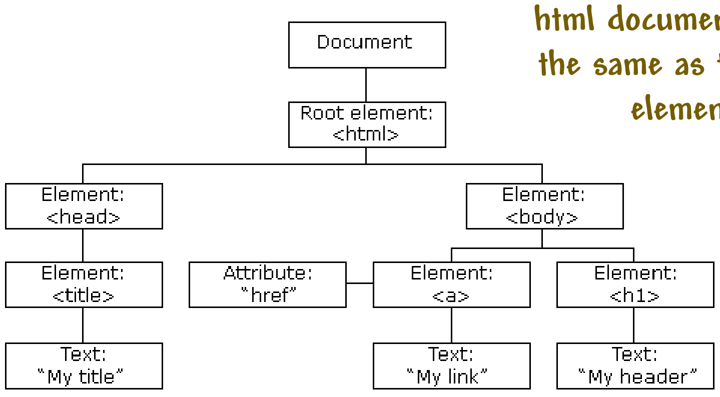
- Defines
  - a standard set of objects (object tree) for an html document
  - Set of methods (language independent) to access the html object
- Your Java and JavaScript (and other) programs can
  - Access a given node (element)
  - Walk the tree
  - Search for particular nodes or data (e.g., img tags)
  - Modify the nodes and insert sub-trees

Current version is DOM Level 3

7

DOM Access to html

Note that the root of the html document is not the same as the root element



8

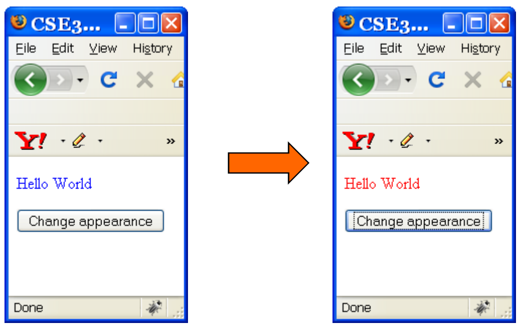
A Sampler of DOM Interfaces

- Document - getDoctype, getImplementation, getDocumentElement
- Node -getParentNode, getChildNodes, getFirstChild, getNextSibling, getAttributes, getLastChild
- NodeList - getLength, item
- Element - getTagName, getAttribute, setAttribute, removeAttribute, normalize
- Text - Inherited from Node
- Attr - Inherited from Node, but not considered part of the document tree (tree methods return null)
- CharacterData - getData, setData

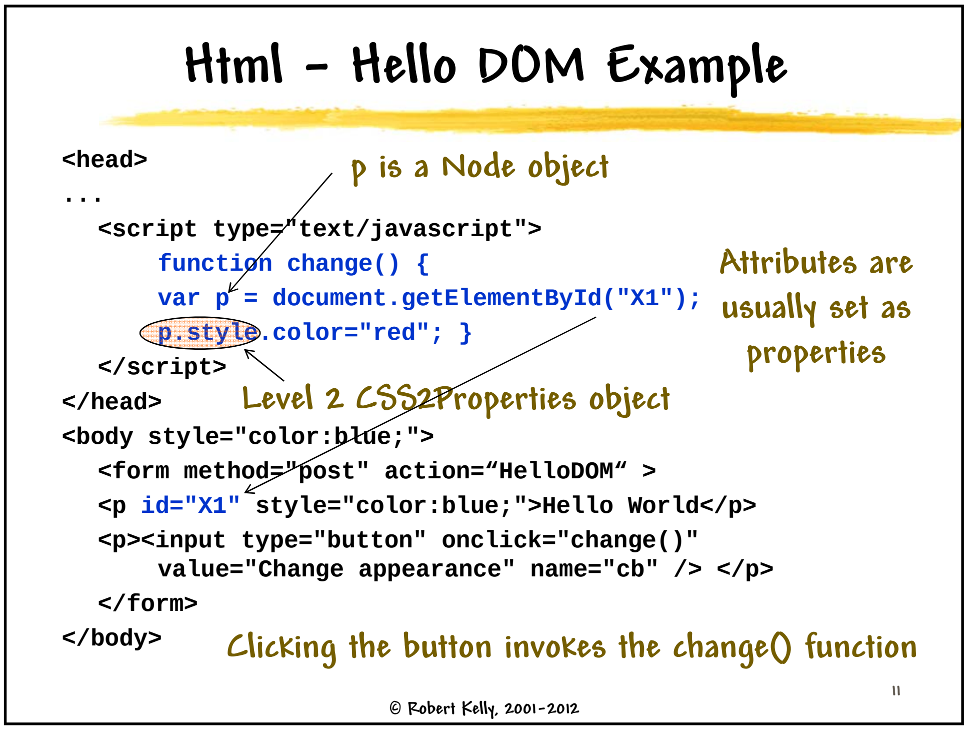
© Robert Kelly, 2001-2012 9

Example - Hello DOM

- Respond to an event
- Obtain a handle to an html element
- Modify the html element



© Robert Kelly, 2001-2012 10



Html - Hello DOM Example

p is a Node object

```
<head>
...
   <script type="text/javascript">
       function change() {
       var p = document.getElementById("X1");
       p.style.color="red"; }
   </script>
</head>
<body style="color:blue;">
   <form method="post" action="HelloDOM" >
   <p id="X1" style="color:blue;">Hello World</p>
   <p><input type="button" onclick="change()"
       value="Change appearance" name="cb" /> </p>
   </form>
</body>
```

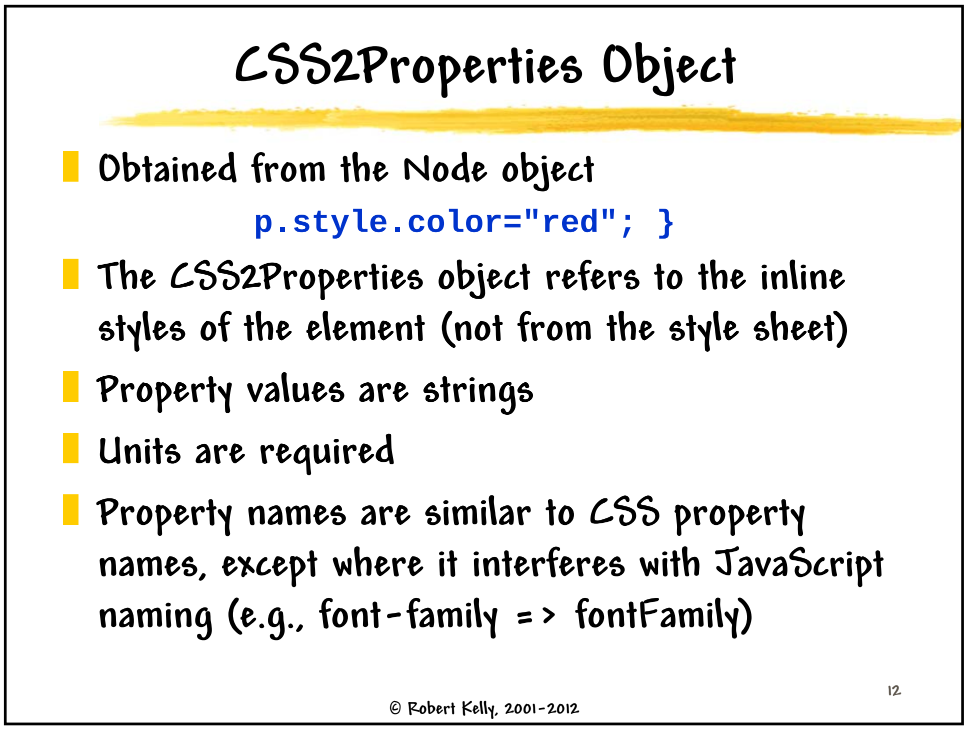
Attributes are usually set as properties

Level 2 CSS2Properties object

Clicking the button invokes the change() function

11

© Robert Kelly, 2001-2012



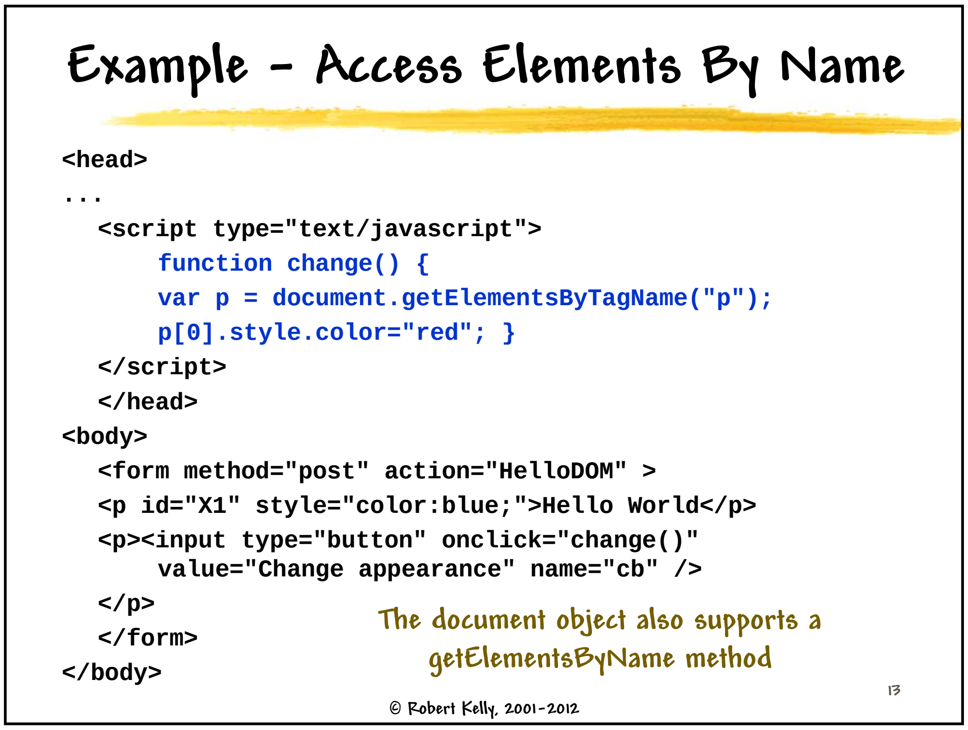
CSS2Properties Object

- Obtained from the Node object

  `p.style.color="red"; }`
- The CSS2Properties object refers to the inline styles of the element (not from the style sheet)
- Property values are strings
- Units are required
- Property names are similar to CSS property names, except where it interferes with JavaScript naming (e.g., font-family => fontFamily)

12

© Robert Kelly, 2001-2012



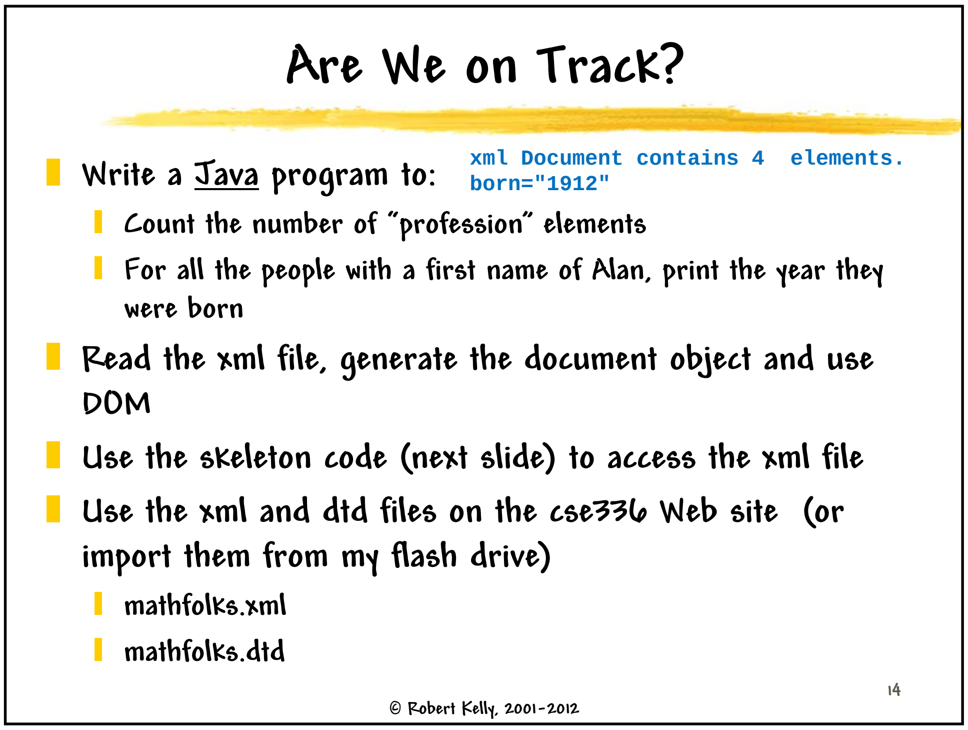
Example - Access Elements By Name

```
<head>
...
   <script type="text/javascript">
        function change() {
        var p = document.getElementsByTagName("p");
        p[0].style.color="red"; }
   </script>
   </head>
<body>
   <form method="post" action="HelloDOM" >
   <p id="X1" style="color:blue;">Hello World</p>
   <p><input type="button" onclick="change()"
        value="Change appearance" name="cb" />
   </p>
   </form>
</body>
```

The document object also supports a getElementsByName method

© Robert Kelly, 2001-2012

13



Are We on Track?

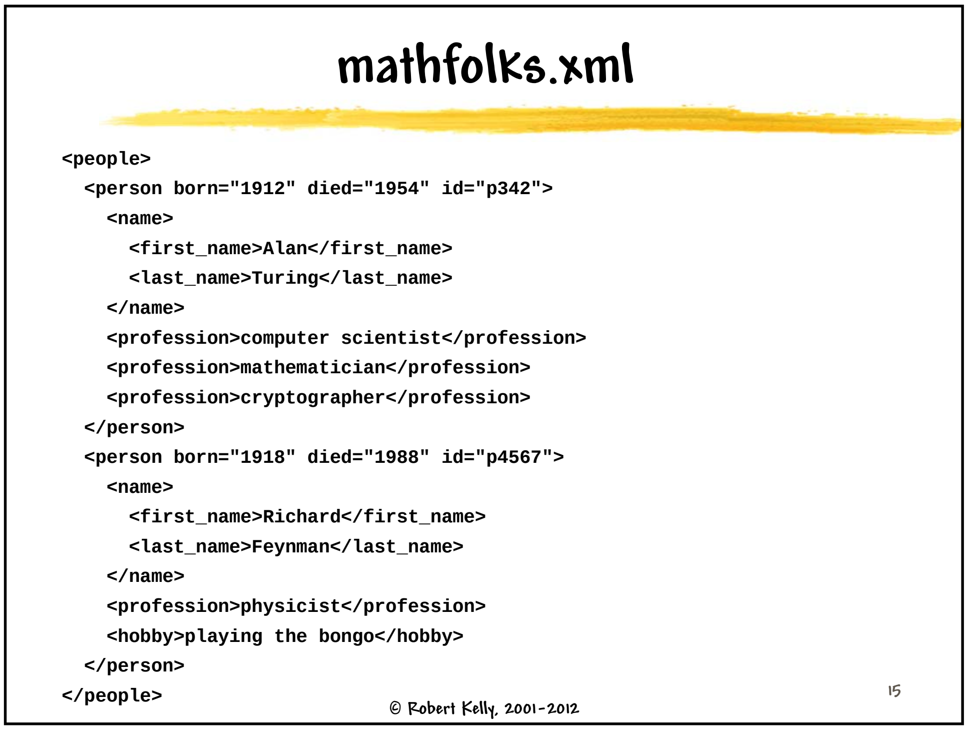
- Write a Java program to:
  - Count the number of "profession" elements
  - For all the people with a first name of Alan, print the year they were born

```
xml Document contains 4  elements.
born="1912"
```

- Read the xml file, generate the document object and use DOM
- Use the skeleton code (next slide) to access the xml file
- Use the xml and dtd files on the cse336 Web site (or import them from my flash drive)
  - mathfolks.xml
  - mathfolks.dtd

© Robert Kelly, 2001-2012

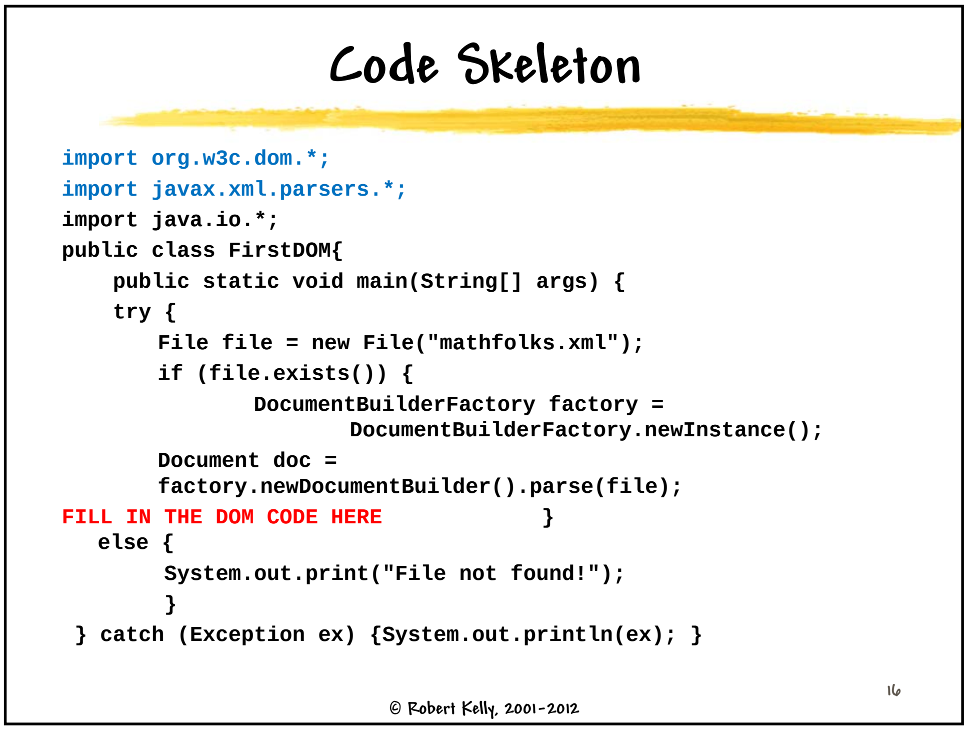
14



mathfolks.xml

```
<people>
  <person born="1912" died="1954" id="p342">
    <name>
      <first_name>Alan</first_name>
      <last_name>Turing</last_name>
    </name>
    <profession>computer scientist</profession>
    <profession>mathematician</profession>
    <profession>cryptographer</profession>
  </person>
  <person born="1918" died="1988" id="p4567">
    <name>
      <first_name>Richard</first_name>
      <last_name>Feynman</last_name>
    </name>
    <profession>physicist</profession>
    <hobby>playing the bongo</hobby>
  </person>
</people>
```

15



Code Skeleton

```
import org.w3c.dom.*;
import javax.xml.parsers.*;
import java.io.*;
public class FirstDOM{
    public static void main(String[] args) {
    try {
        File file = new File("mathfolks.xml");
        if (file.exists()) {
                DocumentBuilderFactory factory =
                        DocumentBuilderFactory.newInstance();
        Document doc =
        factory.newDocumentBuilder().parse(file);
FILL IN THE DOM CODE HERE             }
   else {
        System.out.print("File not found!");
        }
 } catch (Exception ex) {System.out.println(ex); }
```

16

DOM Code to Count Elements

Count elements

```
NodeList nodes = doc.getElementsByTagName("profession");
System.out.println("xml Document Contains " +
   nodes.getLength() + "  elements.");
```

Search tree

```
nodes = doc.getElementsByTagName("first_name");
for (int i = 0; i < nodes.getLength(); i++) {
  Node f = nodes.item(i);
  if (f.getTextContent().equals("Alan")) {
    Node grandParent = f.getParentNode().getParentNode();
    if (grandParent.hasAttributes())  {
      NamedNodeMap nnm = grandParent.getAttributes();
      System.out.println(nnm.getNamedItem("born"));
     }
   }
 }
```

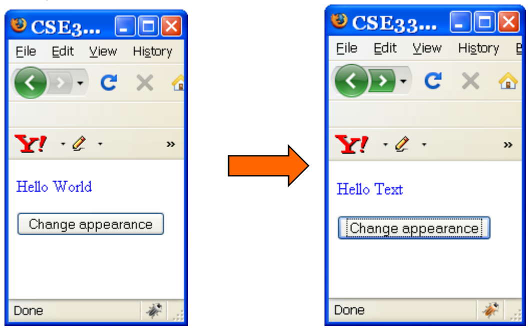
17

© Robert Kelly, 2001-2012

Example – Changing Element Contents

```
function change() {
   var p = document.getElementById("X3");
   p.innerHTML="Hello Text";
}
```

innerHTML is an element property that corresponds to all the markup and content within the element



Setting an innerHTML property parses html text into the html tree

innerHTML is not an official part of DOM, but well supported

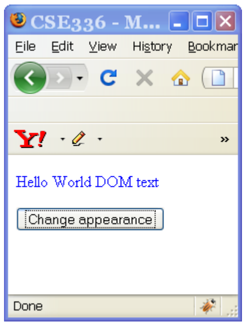
18

© Robert Kelly, 2001-2012

"Pure" DOM HTML Change

- DOM provides methods to delete, create, clone, and insert branches within the DOM tree

```
function change() {
       var p = document.getElementById("X6");
       var text = document.createTextNode(" DOM text");
       p.appendChild(text);
}
...
<p id="X6" class="blue">
       Hello World</p>
```



Hello World DOM text

© Robert Kelly, 2001-2012

19

Example – Changing Styles

- An easy way to change the appearance of an element is to change its class attribute

"class" is a reserved name in JavaScript, so the class property is "className"

```
<style type="text/css">
       .blue {color:blue;}
       .red {color:red;}
</style>
<script type="text/javascript">
       function change() {
       var p = document.getElementById("X4");
       p.className="red"; }
</script>
...
<p id="X4" class="blue">Hello World</p>
```

© Robert Kelly, 2001-2012

20

DOM Methods

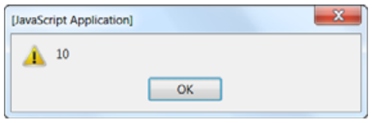
- DOM provides methods that allow you to access and modify the html tree
  - Collections of elements returned as a NodeList object

Example will display number of tags and id attributes of tags

```
function change() {
var elems = document.getElementsByTagName("*");
alert(elems.length);
for each (var e in elems) {
      if (e.hasAttribute("id")) {
            alert(e.id);
      }}}
...
<p id=“X5”>Hello World</p>
```

e is a Node object

Note the syntax of the JavaScript for each loop



21

© Robert Kelly, 2001-2012

Form Processing

- You can validate your form data in JavaScript with a function invoked by the onsubmit event
- If your form handler function returns false, the form data is not sent to the server

```
<form name=“z” onsubmit=“return isValid(...) >
<input name=“zipcode” ...  >
```

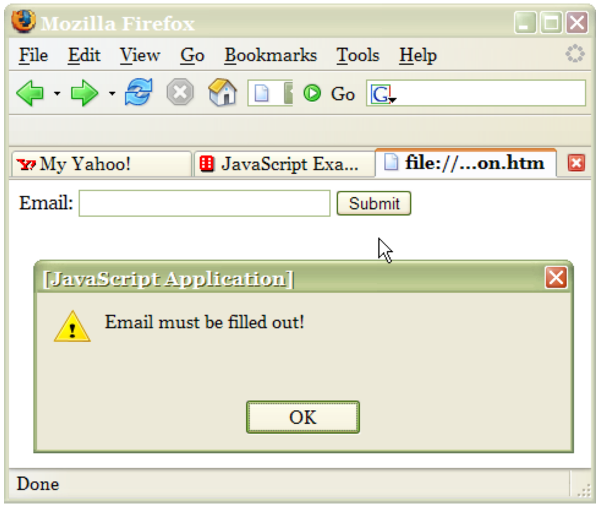
- You can easily reference form elements through the mandatory name attribute in the element

```
document.z.zipcode
```

22

© Robert Kelly, 2001-2012

Example – Validation (result)



23

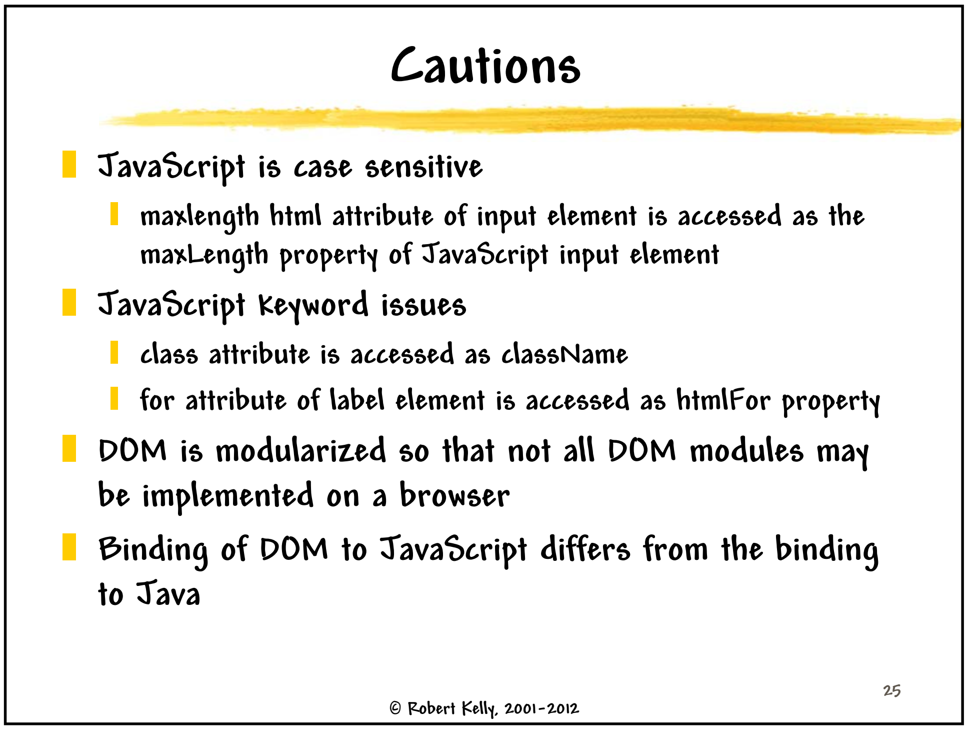
Example – Validation

The with keyword is used to specify an object to which the methods and properties in the block are referenced

```
<html> <head>
<script type="text/javascript">
function validate_form(thisform) {
   with (thisform) {
   if (validate_required(email,"Email must be filled
   out!")==false) {
   email.focus();return false} } }

function validate_required(field, alerttxt) {
   with (field) {
       if (value==null||value=="")
       {alert(alerttxt);return false}
       else {return true} } }
</script> </head>
<body>
<form action="submitpage.htm“
   onsubmit="return validate_form(this)" method="post">
   Email: <input type="text" name="email" size="30">
   <input type="submit" value="Submit">
</form> </body> </html>
```

24

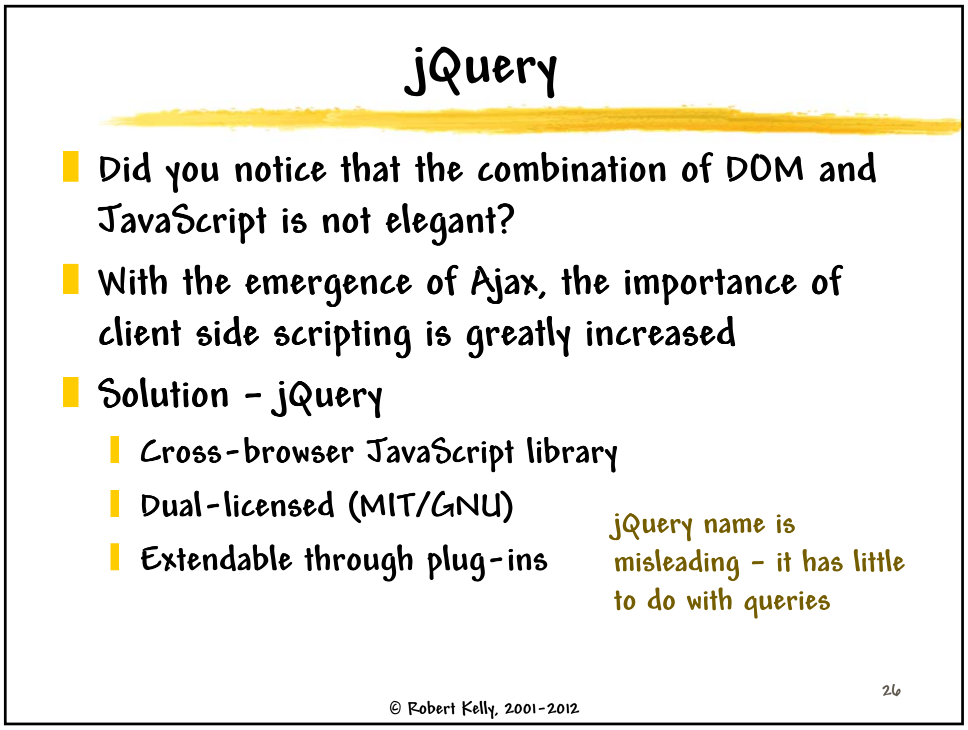


Cautions

- JavaScript is case sensitive
  - maxlength html attribute of input element is accessed as the maxLength property of JavaScript input element
- JavaScript keyword issues
  - class attribute is accessed as className
  - for attribute of label element is accessed as htmlFor property
- DOM is modularized so that not all DOM modules may be implemented on a browser
- Binding of DOM to JavaScript differs from the binding to Java

© Robert Kelly, 2001-2012

25



jQuery

- Did you notice that the combination of DOM and JavaScript is not elegant?
- With the emergence of Ajax, the importance of client side scripting is greatly increased
- Solution - jQuery
  - Cross-browser JavaScript library
  - Dual-licensed (MIT/GNU)
  - Extendable through plug-ins

jQuery name is misleading - it has little to do with queries

© Robert Kelly, 2001-2012

26

Getting Started With jQuery

- Download the jQuery library (uncompressed) from
  `http://code.jquery.com/jquery-1.7.2.js`
- Download the jQuery StarterKit
- Lecture material is based on
  `docs.jquery.com/Tutorials:Getting_Started_with_jQuery`

27

© Robert Kelly, 2001-2012

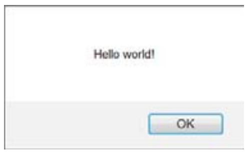
Hello jQuery World

```
<head>
<script type="text/javascript" src="jquery-1.7.2.js"></script>
<script type="text/javascript">
$(document).ready(function() {
  $("a").click(function() {
    alert("Hello world!");
  });
});
</script>
</head>
<body>
<a href="">Link</a>
</body>
</html>
```

The jQuery ready function provides a handler to execute when the page has loaded

Ready function parameter is typically an anonymous function

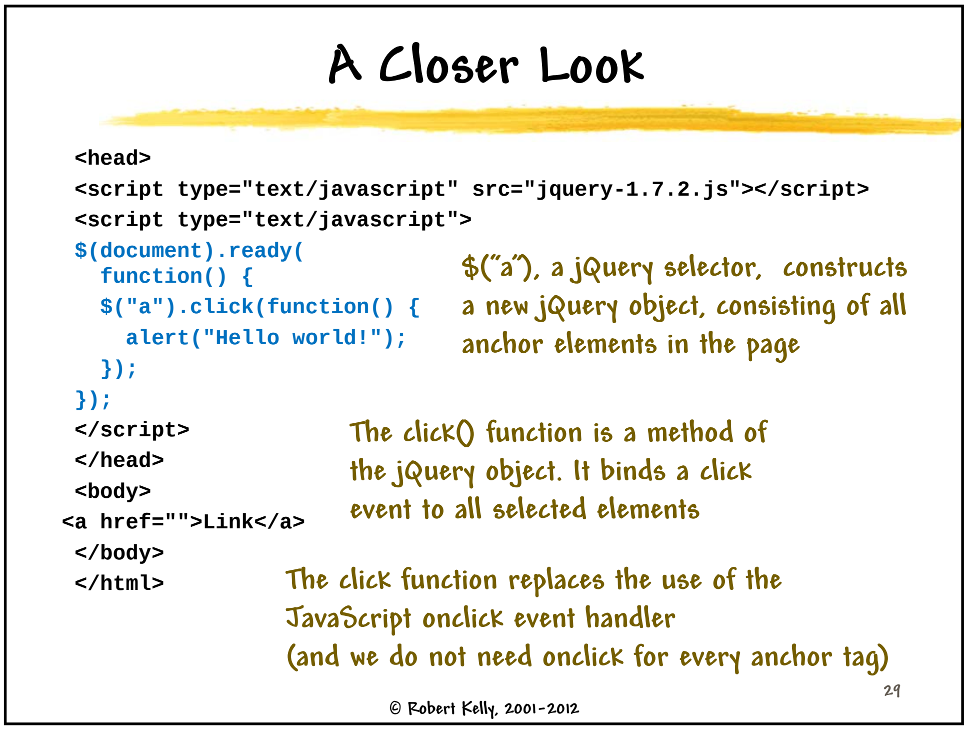
Link


Hello world!

$ (a valid JavaScript identifier) represents the jQuery function ($() constructs a new jQuery object)

28

© Robert Kelly, 2001-2012



A Closer Look

```
<head>
<script type="text/javascript" src="jquery-1.7.2.js"></script>
<script type="text/javascript">
$(document).ready(
  function() {
  $("a").click(function() {
    alert("Hello world!");
  });
});
</script>
</head>
<body>
<a href="">Link</a>
</body>
</html>
```

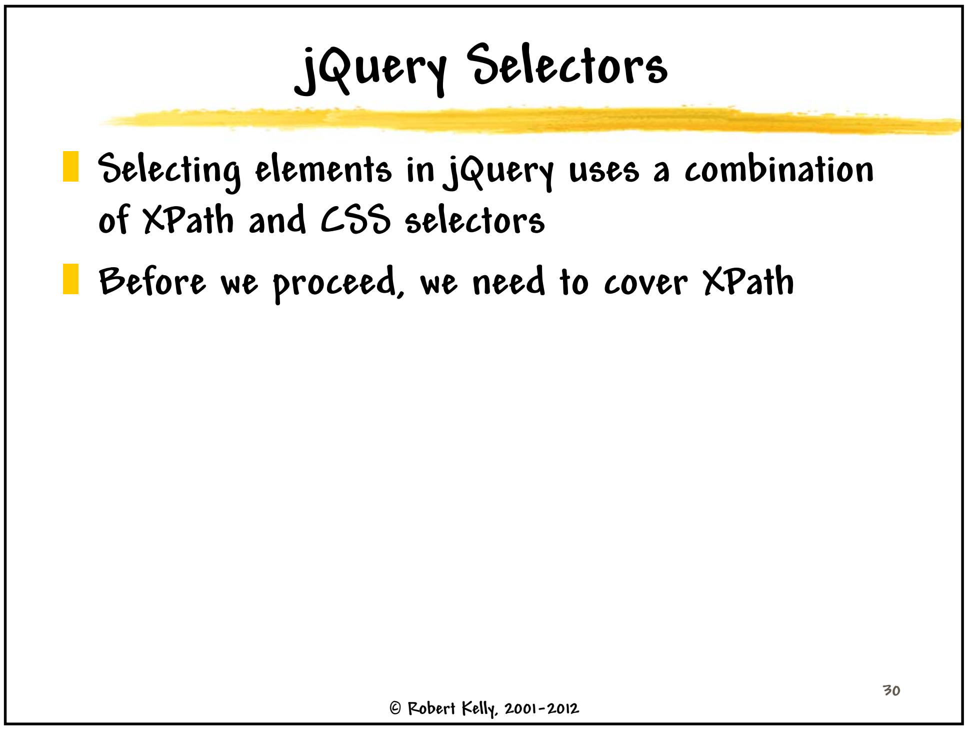
$("a"), a jQuery selector, constructs a new jQuery object, consisting of all anchor elements in the page

The click() function is a method of the jQuery object. It binds a click event to all selected elements

The click function replaces the use of the JavaScript onclick event handler (and we do not need onclick for every anchor tag)

29

© Robert Kelly, 2001-2012



jQuery Selectors

- Selecting elements in jQuery uses a combination of XPath and CSS selectors
- Before we proceed, we need to cover XPath

30

© Robert Kelly, 2001-2012

Have You Satisfied the Learning Goals?

- Understand the Document Object Model
- Understand how to perform client side form validation
- Understand JavaScript event model
- Understand jQuery library as a way to simplify the JavaScript event model

Assignment 7

- Implement your project form validation as a JavaScript
  - Do not replace the server side validation - use the JavaScript validation in addition