Web-Based Information Systems

Fall 2006

CMPUT 410: JavaScript

Dr. Osmar R. Zaïane

University of Alberta

Publishing On the Web

- Writing HTML with a text editor allows to generate web pages. These pages are said static in the sense that they do not change.
- What if we want to personalize pages for particular visitors or events?
- What if we want to have actions on the page?
- What if the content of the page is from a database?
- Etc.

Course Content

- Introduction
- Internet and WWW
- Protocols
- HTML and beyond
- Animation & WWW
- CGI & HTML Forms
- **Javascript**
- Databases & WWW
- Dynamic Pages
- Perl & Cookies
- SGML / XML
- CORBA & SOAP
- Web Services
- Search Engines
- Recommender Syst.
- Web Mining
- Security Issues
- Selected Topics

Web-based Applications

Objectives

- Learn how JavaScript stores data, how a document is structured in JavaScript
- Learn event-based programming with JavaScript.
- Learn how JavaScript is event driven and how user actions are tracked
- See and analyze some concrete examples with JavaScript.

Content

I. JavaScript and the Details

- Variable identifiers and their types
- The notion of objets
- Arrays
- Control structures
  - Condition and selection
  - Iteration
- Procedures and functions

II. Event-Based Programming with JavaScript

- **What is an event?**
- What are the recognized events?
- Capturing events.

III. Practical Examples

- Data entry validation within a form;

Introduction to Variables

- A variable in Javascript has a type:
  - number (integer or non integer)
  - String
  - Boolean
  - Null
- JavaScript is not strongly typed.

Declaring Variables

The first time a variable is used it must be declared with the keyword **'var'.**

var identifier = value;

The identifier must start with a letter or underscore '_' and can have as many characters as necessary (letters, digits, underscore).

Javascript is sensitive to capital letters.
myvariable is different from *MyVariable* and x ≠ X

Type Conversion on the fly

- Because JavaScript is not strongly typed, it is possible to:
  - Change the type of a variable;
  - Do operations on variables of different types.
  - The major type, or default type, is string.

Variable Examples

```
<HTML>
<HEAD>
<TITLE>My First Java Script with variables</TITLE>
<script language="JavaScript">
 <!– hide script
    var myNumber=35;
    var myString="2004";
    var myOtherString="CMPUT410";
    var myAddition = myNumber+myNumber;
    var myConcatenation = myStyring + myOtherString;
    var myError = myNumber + myOtherString;
    var myCalculation = myNumber + myString;
    var myDream = myOtherString + myString;
 // end of hide -->
 </script>
</HEAD>
```


Variable Examples (con't)

```
<BODY>
<script language="JavaScript">
 <!-- hide script
        document.write("myAddition="+myAddition+"<BR>");
        document.write("myConcatenation="+myConcatenation+"<BR>");
        document.write("myError="+myError+"<BR>");
        document.write("myDream="+myDream+"<BR>");
        myError = myNumber * 3;
        document.write("myError="+myError+"<BR>");
        myNumber="Bye!";
        document.write("myNumber="+myNumber+"<BR>");
 // end of hide -->
 </script>
</BODY>
</HTML>
```

```
myAddition=70
myConcatenation=2004CMPUT410
myError=35CMPUT410
myDream= CMPUT4102004
myError=105
myNumber=Bye!
```


Content

JavaScript & Concept of Objects

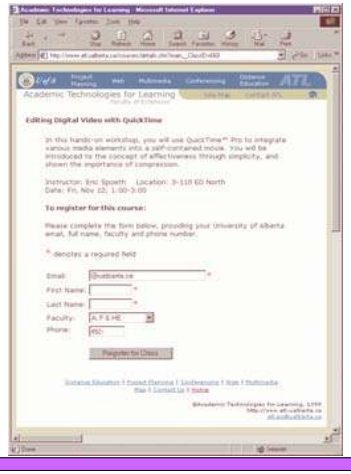
- JavaScript is not an object-oriented language.
- JavaScript is an object-based language.
- There are many pre-defined objects, but programmers can define their own objects.
- An object has attributes (specific properties) as well as methods (behaviour of objects).
- An attribute could be a value or recursively another object.

A Book is an Object

Title
Authors
Editors
Number of pages
Price
Set of Chapters
Set of figures and images
etc.

Each book has the same attributes with different values

What are the Objects, What are their Properties?



?

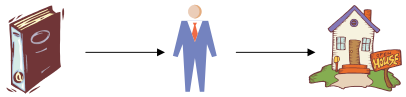
Access Object Properties

myObject.oneProperty

Object Name . Attributre Name

If the attribute is also an object, to access the property of the attribute's attribute:

myObject.oneProperty.aPropertyOfProperty
Ex: Book.Editor.Address



document.MyForm.Name.value

Access Object Methods

myObject.oneMethod(parameters)

Object Name . Method Name (parameters)

If there are no parameters:

myObject.oneMethod()

Ex: document.write("Hello!")

Predefined Object Classes

- There are many intrinsic pre-defined objects in JavaScript:

–Date	–Navigator
–String	–History
–Math	–Location
–Window	–Form
–Document	etc…

- These objects have their pre-defined attributes and methods.

Object Date

- The object Date needs to be instantiated with the keyword ***new***.

  var today= new date();

- The class Date doesn't have properties but the following methods:

•getDate()	•getYear()	
•getDay()	•setDate()	
•getHours()	•setHours()	etc…
•getMinutes()	•setMinutes()	
•getMonth()	•setMonth()	
•getSeconds()	•getSeconds()	
•getTime();	•setTime();	
•getTimezoneOffset()	•setYear()	

Example with Date

```
<HTML>
<HEAD>
<TITLE>My test with dates</TITLE><script language="JavaScript">
      var thisIsNow=new Date();
      var BirthDate = new Date(60,05,18);
</script></HEAD>
<BODY> <script language="JavaScript">
    document.write("Today we are the: "+thisIsNow+"<BR>");
    document.write("Alfred's birthdate is the "+ BirthDate +"<BR>");
    document.write("The date:"+ BirthDate.getDate()+ "/" +
                                (BirthDate.getMonth()+1) + "/" +
                                (BirthDate.getYear()+1900)+"<BR>");
     document.write("The time now is:" + thisIsNow.getHours() + ":" +
                                         thisIsNow.getMinutes() + ":" +
                                         thisIsNow.getSeconds()+"<BR>");
     thisIsNow.setYear(2010);
     document.write("The new date in the future is:<br>"+ thisIsNow);
  </script></BODY></HTML>
```


The Object String

- Where we define a string constant or a string variable, JavaScript creates an instance of an object String.
- The object String has one property, ***lenght***, and many methods:

•anchor()	astring.anchor(anchor)➔<A name="anchor">astring</A>
•big()	astring.big()➔<BIG>astring</BIG>
•blink()	astring.bink()➔<BLINK>astring</BLINK>
•bold()	astring.bold()➔<BOLD>astring</BOLD>
•fontcolor()	astring.fontcolor(#FF0000)➔<FONT color="#FF0000">astring</FONT>
•fontsize()	astring.fontsize(5)➔<FONT size=5>astring</FONT>
•italics();	astring.italics() ➔<I>astring</I>
•small()	astring.small()➔<SMALL>astring</SMALL>
•sub()	astring.sub() ➔_{astring}
•sup()	astring.sup() ➔^{astring}