

# today: client side web technology: JavaScript

course: Web Technology

Maarten Lamers

[www.maartenlamers.com/WT2006/](http://www.maartenlamers.com/WT2006/)

Leiden University

## previous lecture

HyperText Markup Language (HTML)  
meant for *markup*, not for *programming*

HTML is used for

- structuring documents
- layout (not meant to, but OK), images, links, ...
- presenting information on the web

**previous lecture**

HTML forms provide interactivity between web-user and web application

using a GET or POST method HTTP request, data from an HTML form is sent to a web server

after processing the data, the server responds with a resulting document

HTML itself lacks possibility for interactivity, or instant feedback  
processing of the data happen on the server side

**adding dynamics: JavaScript**

programming language

commonly used within browsers to enhance HTML documents with interaction, dynamic content, and instant feedback

JavaScript code can be embedded within an HTML document and interpreted within the web-browser

## client side

"... *within HTML doc ... interpreted within the web-browser.*"  
therefore, *client side* technology!

reduces server and traffic overhead  
not for every small change in a webpage, or user action, a new page must be requested from the server

JavaScript language is not limited to web-browsers

- it may run on servers
- it may run from command line interpreters
- it may run everywhere in principle...
  
- just as Cantonese may be spoken on the South Pole

## JavaScript

(scripting) programming language  
1995 by Netscape and Sun corporations

popular, because:

- its code can be embedded into HTML
- it can change or add stuff to HTML documents
- it can control the web-browser
- it can interact with the user (react on what the user does)
- it is built into (*understood by, interpreted by*) common browsers

## JavaScript

most commonly used in HTML documents to

- open pop-up windows with specific size, location and other settings
- change images when the mouse rolls over them
- validate the content that a user typed into an HTML form; for example checking required fields, acceptable values, e-mail address format, ...

but much more possible

## intermezzo

### interpreted versus compiled

code written in an **interpreted** programming language (often called "script") may be executed from source form, by an **interpreter**. Any language may, in theory, be compiled or interpreted; therefore, it refers to languages' implementations rather than designs.

an interpreted program can not be as efficient as a **compiled** program, which has been processed by a language **compiler**. A language compiler converts source statements into something close to the strings of 0's and 1's that a processor ultimately is given to work on. Because this work is already done before a compiled program is run, it runs much more quickly.

(source: wikipedia.org, and whatis.com)

## what about Java?

*Java* is also a programming language  
 allows development of programs called *applets*  
 applets can be included in web-pages

applets are platform independent!  
 they are compiled to *bytecode*, which can be run on any  
 machine using a *Java Virtual Machine*  
 write once, run anywhere  
 efficient, since (semi-)compiled

features include graphics, networking, file I/O, ...  
 example: the Visible Human Viewer  
<http://www.dhpc.adelaide.edu.au/projects/vishuman2/>

## Java bytecode and Virtual Machine

### Java source code

```
public class MyClass
{
}
```

Java compiler

### Java bytecode

```
0xCAFEBAFE
0x0201
...
```

### executable machine code

```
001011101001001001011
001011010111010011110
...
```

Java VM

### Java bytecode

```
0xCAFEBAFE
0x0201
...
```

## JavaScript versus Java

JavaScript != Java;

their syntax is similar: both based on C programming language just as the Processing language

Java is compiled (to machine independent bytecode)

JavaScript is interpreted (hence the name *script*)

when part of the web:

- in Java you write *applets*, small portable applications that can be part of a web page
- JavaScript is tied into an HTML document and can control the document and browser

## Document Object Model (DOM)

an object-oriented description of an HTML document used by programs to access and change the document's content, structure and style

from the programming point-of-view, the HTML document is hierarchically made up of many objects that can be accessed and changed

for example, page object (`document`), browser window object (`window`), image object (`image`), button object (`button`), ...

JavaScript interacts with the DOM to add dynamics and interactivity to HTML documents

## object properties, methods and events

object can have properties, methods and events

properties are things that an object may have or be

```
document.title
document.lastModified
image.border
```

methods are actions that an object may perform

```
document.write("broodje kroket")
string.toUpperCase()
```

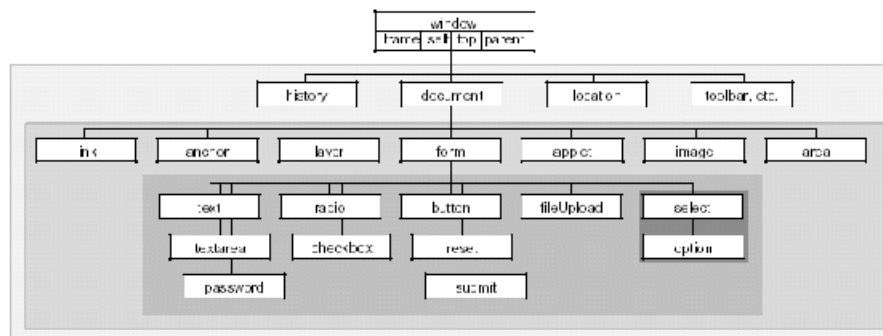
events are things that may happen to an object

```
<img onLoad="alert('April 30 is Koninginnedag');">
<a href="vla.html" onClick="alert('Vanille vla!');">
```

## DOM: object tree

DOM organizes all parts of the browser (window, document, images, buttons, text, ...) as a hierarchy of objects

JavaScript can use this model to access the objects



## DOM: object tree example

imagine the following HTML document is in the browser

```
<HTML>
  <BODY>
    <FORM NAME="ijssalon">
      <INPUT TYPE="text" NAME="bolletjes">
      <INPUT TYPE="text" NAME="smaak">
    </FORM>
  </BODY>
</HTML>
```

you can access the value of the text-input objects as:

```
window.document.ijssalon.bolletjes.value;
window.document.ijssalon.smaak.value;
```

## JavaScript language

the English version of Wikipedia.org has excellent entries for:

- "JavaScript syntax"  
all about variables, data types, operators, etcetera
- "Client-side JavaScript"  
shows you how to place JavaScript code inside HTML

many, many JavaScript tutorials exist on the web

**example: JavaScript and HTML**

```



```

```

<SCRIPT type="text/javascript">

function roundhalf(i) {
  // rounds a number to its nearest half-point. E.g. 6.33 -> 6.5
  return (Math.round(i*2)/2);
}

function finalgrade() {
  if (examgrade.value.length == 0) {
    alert ("You must fill in an exam grade.");
    examgrade.focus();
    examgrade.select();
    return false;
  }
  var fExam = parseFloat(examgrade.value);
  if (!(fExam > 0.0) && (fExam <= 10.0)) {
    alert (examgrade.value + " is not a valid exam grade.");
    examgrade.focus();
    examgrade.select();
    return false;
  }
  alert("final grade = " + ((passX.checked) || ((pass1.checked) &&
    (pass2.checked)) ? fExam : "FAIL"));
  return false;
}
</SCRIPT>

```

**assignment 1: JavaScript**

write a simple game in JavaScript  
either a new or existing game

turn-in date: Tuesday, March 7<sup>th</sup>, before 17:00h  
all details are in the assignment itself, which is on the course  
webpage

[www.maartenlamers.com/WT2006/](http://www.maartenlamers.com/WT2006/)

**reminder**

lab assistance is on Tuesdays 14:00 – 17:00h  
lab assistant is Amalia Kallergi (webtech2006@hotmail.com)