

# DCUKEU'QHLCXCUETRV

## 4.1 Overview of JavaScript: Origins

- Livescript
- Originally developed by Netscape
- Joint Development with Sun Microsystems in 1995
- Supported by Netscape, Mozilla, Internet Explorer

## 4.1 JavaScript Components

- Core
  - The heart of the language
- Client-side
  - Library of objects supporting browser control and user interaction EG: mouse clicks
- Server-side
  - Library of objects that support use in web servers
  - Eg: commun. With database management system

## 4.1 Java and JavaScript

- Differences
  - JavaScript has a different object model from Java
  - JavaScript is not strongly typed
- Java 1.6 has support for scripting
  - <http://java.sun.com/javase/6/docs/technotes/guides/scripting/index.html>
- Mozilla Rhino is an implementation of JavaScript in Java
  - <http://www.mozilla.org/rhino/>

## 4.1 Uses of JavaScript

- Provide alternative to server-side programming
  - Servers are often overloaded
  - Client processing has quicker reaction time
- JavaScript can work with forms
  - JavaScript can interact with the internal model of the web page (Document Object Model)
  - JavaScript is used to provide more complex user interface than plain forms with HTML/CSS can provide
    - <http://www.protopage.com/> is an interesting example
    - A number of toolkits are available. Dojo, found at <http://dojotoolkit.org/>, is one example

#### 4.1 Event-Driven Computation

- Users actions, such as mouse clicks and key presses, are referred to as *events*
- The main task of most JavaScript programs is to respond to events
- For example, a JavaScript program could validate data in a form before it is submitted to a server
  - ❑ *Caution:* It is important that crucial validation be done by the server. It is relatively easy to bypass client-side controls
  - ❑ For example, a user might create a copy of a web page but remove all the validation code.

#### 4.1 XHTML/JavaScript Documents

- When JavaScript is embedded in an XHTML document, the browser must interpret it
- Two locations for JavaScript server different purposes
  - ❑ JavaScript in the head element will react to user input and be called from other locations
  - ❑ JavaScript in the body element will be executed once as the page is loaded
- Various strategies must be used to ‘protect’ the JavaScript from the browser
  - ❑ For example, comparisons present a problem since < and > are used to mark tags in XHTML
  - ❑ JavaScript code can be enclosed in XHTML comments
  - ❑ JavaScript code can be enclosed in a CDATA section

#### 4.2 Object Orientation and JavaScript

- JavaScript is *object-based*
  - ❑ JavaScript defines objects that encapsulate both data and processing
  - ❑ However, JavaScript does not have true inheritance nor subtyping
- JavaScript provides *prototype-based inheritance*
  - ❑ See, for example this Wikipedia article for a discussion:

---

#### 4.2 JavaScript Objects

- Objects are collections of *properties*
- Properties are either *data properties* or *method properties*
- Data properties are either primitive values or references to other objects
- Primitive values are often implemented directly in hardware
- The Object object is the ancestor of all objects in a JavaScript program
  - ❑ Object has no data properties, but several method properties