Programming for Digital Media
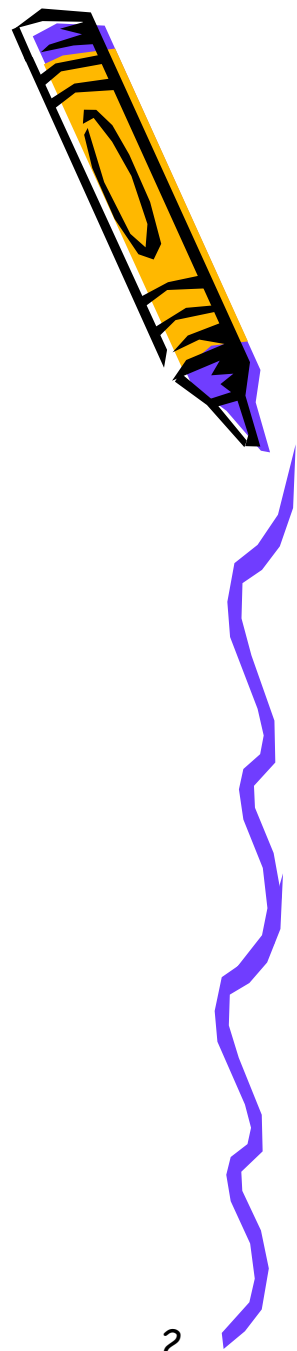
Lecture 7
JavaScript
By: A. Mousavi and P. Broomhead
SERG, School of Engineering Design, Brunel University, UK

Topics

- **Ajax (Asynchronous JavaScript and XML)**

    – What it is and used for
    – Ajax in operation
    – Functionalities
    – Features that enrich *web experience*
    – Example
    – Further Reading
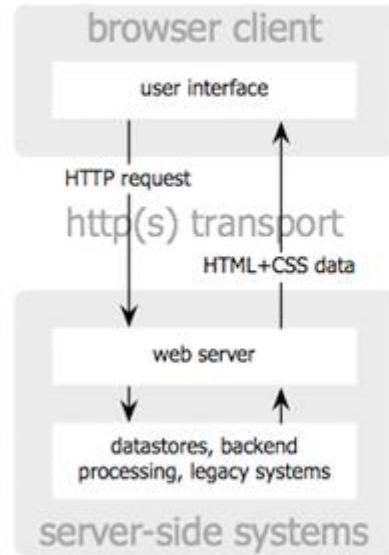
Post-back vs. Real-Time

Updating web document content:

1. Post-Back: Refreshing content by requesting update from the server. Each time the whole document is reproduced – *not good enough for close to real-time application such as: flight arrivals and departures, stock exchange, live matches commentary etc…*

2. Ajax proposed to address this issue – continuously updating a section of the document that needs frequent update – *not the whole document*

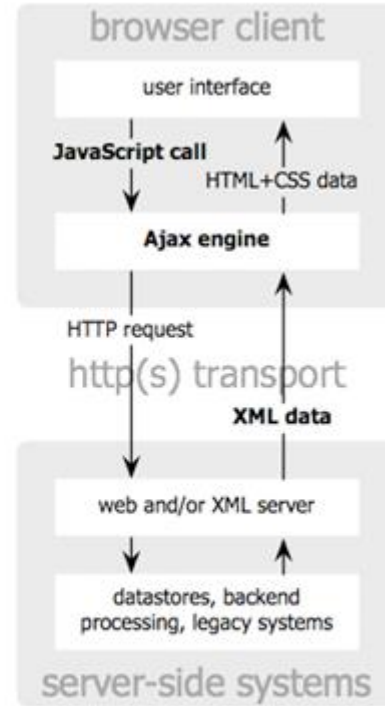# Classical web application vs. Ajax powered web application
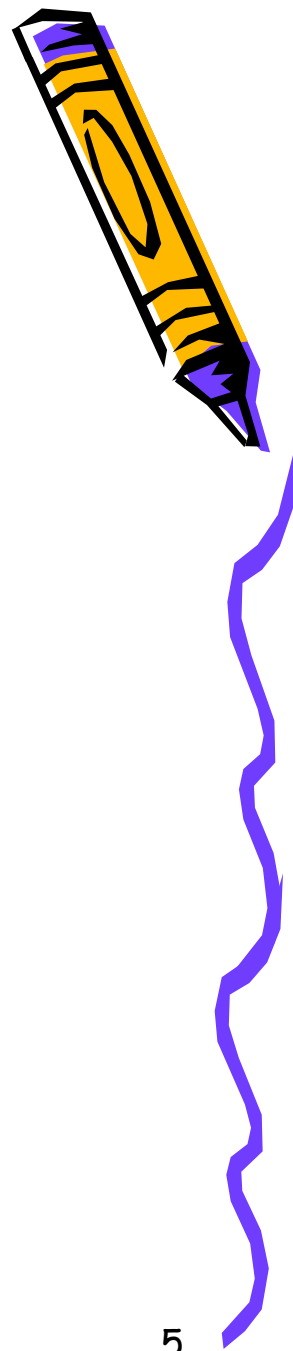
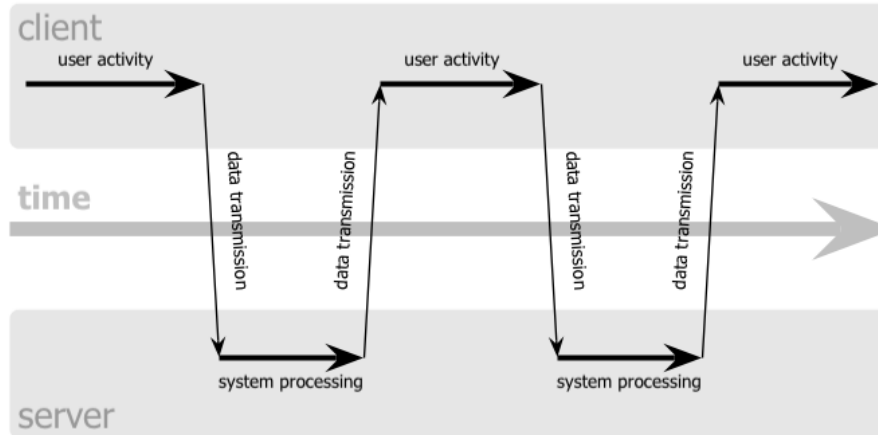

synchronous   *asynchronous*

Classical web application vs. Ajax powered web application

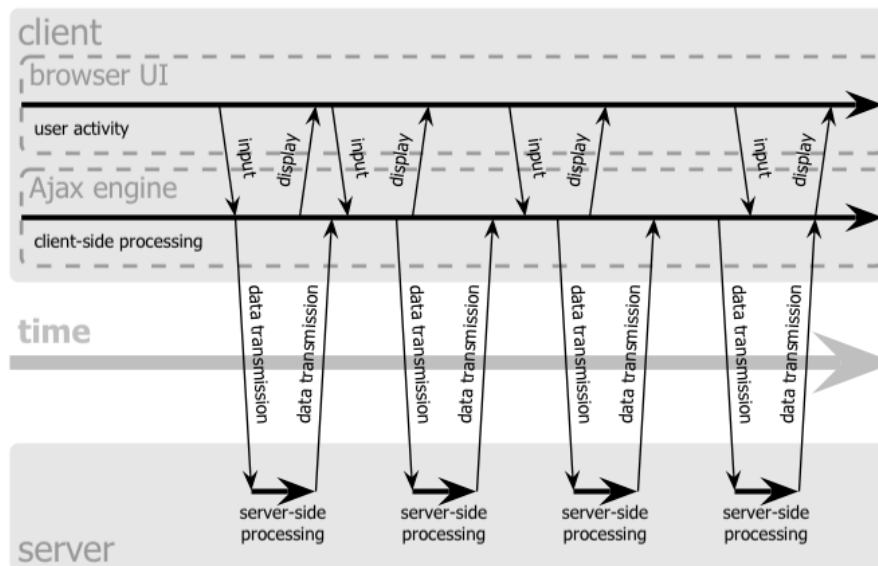Source: Ajax: A New Approach to Web Applications, by J. J. Garrett, 2005

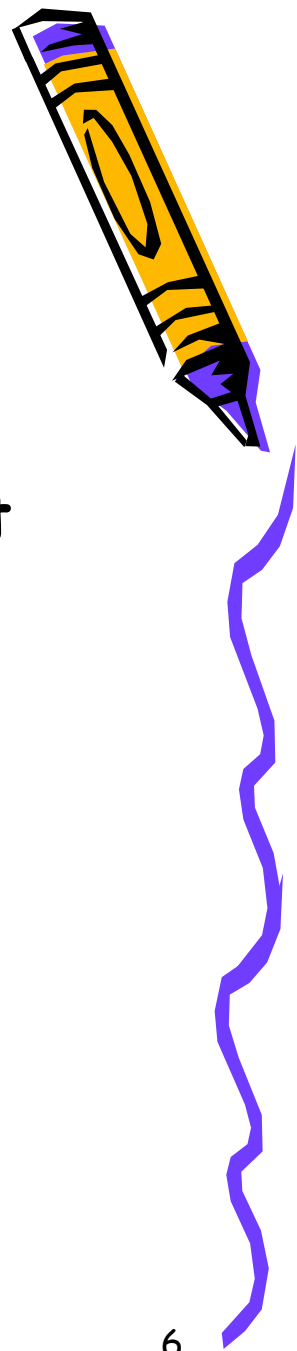# Synchronous vs. Asynchronous document content management

classic web application model (synchronous)

client

user activity → → user activity → → user activity →

data transmission / data transmission / data transmission / data transmission

time →

system processing   system processing

server

Ajax web application model (asynchronous)

client

browser UI

user activity

input / display / input / display / input / display / input / display

Ajax engine

client-side processing

time →

data transmission (repeated) / data transmission

server-side processing   server-side processing   server-side processing   server-side processing

server

Source: Ajax: A New Approach to Web Applications, by J. J. Garrett, 2005

5

Asynchronous JavaScript and XML

- **Real-time page update (no more waiting for refreshing)**

- **Make changes to the instant of the document on-fly**

- **Makes web application look more like desktop application**

- **Utilises the facilities in the web browser to maximise *web experience***

Basics

- **Ajax uses**

  – XHTML and CSS to control UI

  – JavaScript to manage the document

  – XmlHttpRequest object to retrieve asynchronous data
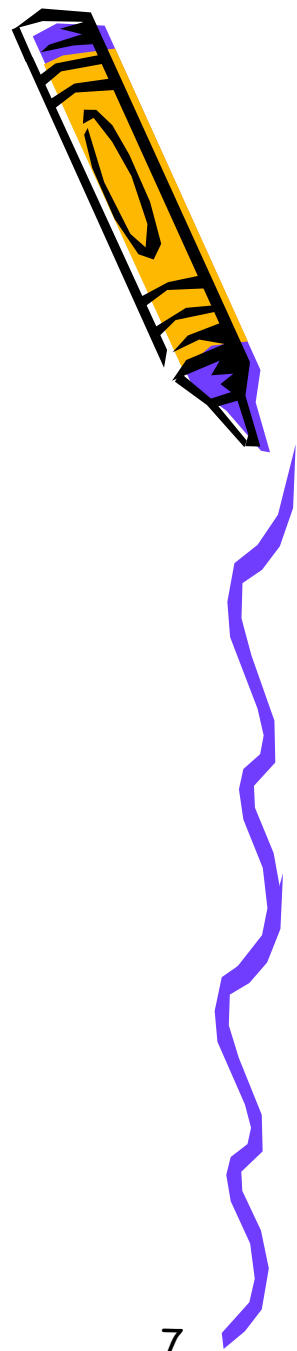
Ajax key features

1. **Seamless and smooth progression (Continuity) – vital for e-commerce applications**

2. **Allows for real-time update without redrawing the whole page at refresh intervals**

3. **Powerful graphical interface changing looks and feel on-fly**

4. **No plug-ins required**

5. **Works with all programming languages**

6. **Combines standards and protocols from HTML, CSS and JavaScript**

Also see: http://ajaxpatterns.org/

How does Ajax work?

1. **Ajax engine:** **Acts as an intermediary between the user and the server.**

2. **At the start of the session the browser loads an Ajax engine rather than the webpage – written in JavaScript hidden in a frame**

3. **The engine handles both rendering of the interface and communicating with the server the engine sees and communicates with the server on the user's behalf.**

4. **The Ajax engine allows the user's interaction with the application to happen asynchronously — independent of communication with the server.**

How does Ajax work continued?
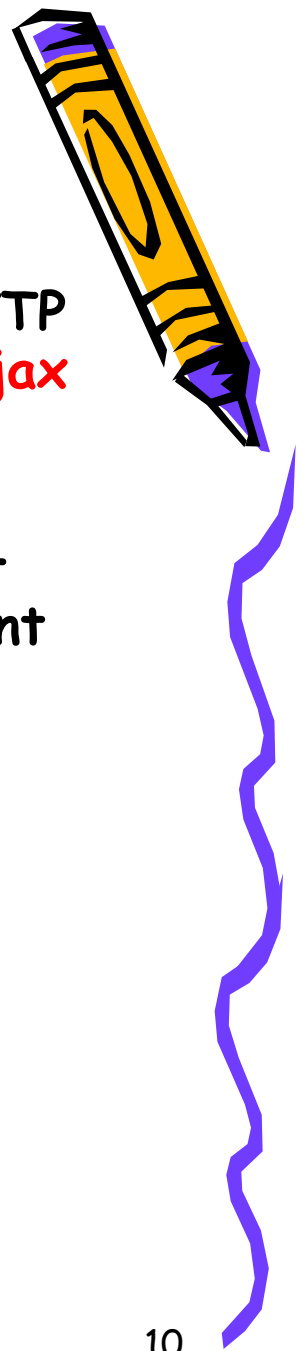
5. Every user action that normally would generate an HTTP request takes the form of a JavaScript call to the **Ajax engine** instead.

6. **The engine** handles any response to a user action that does not require interaction with the server (e.g. client side validations, …) —

7. If communication with the server is necessary (e.g. validating a user) **the engine** makes those requests asynchronously, usually using XML, without stalling a user's interaction with the application.

Ajax shortcomings

- **Continuous communication with server (browser-side caching can reduce this)**

- **Knowledge of JavaScript and other server-side languages are necessary**

- **Many prefer not to directly code using Ajax, but use for example JQuery Ajax support – could get close to the real thing with less complexity and almost all functionalities**
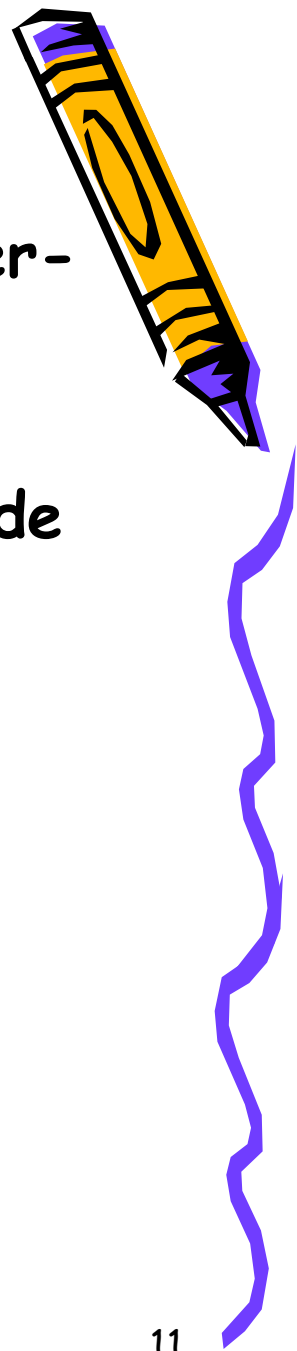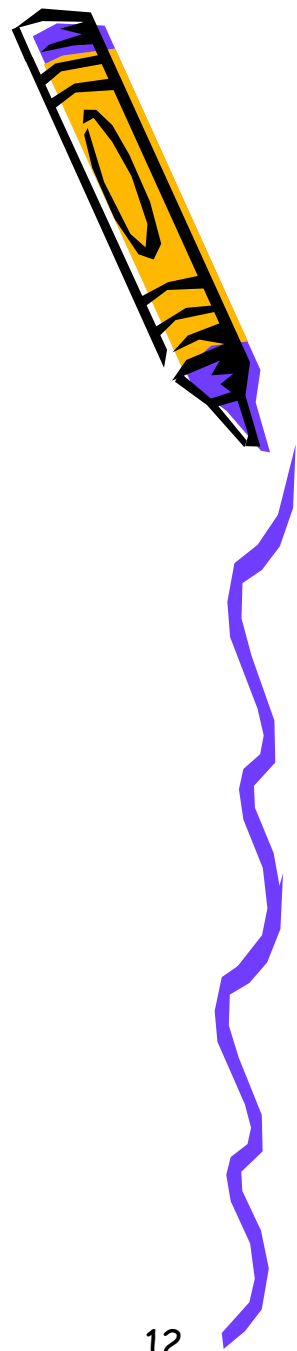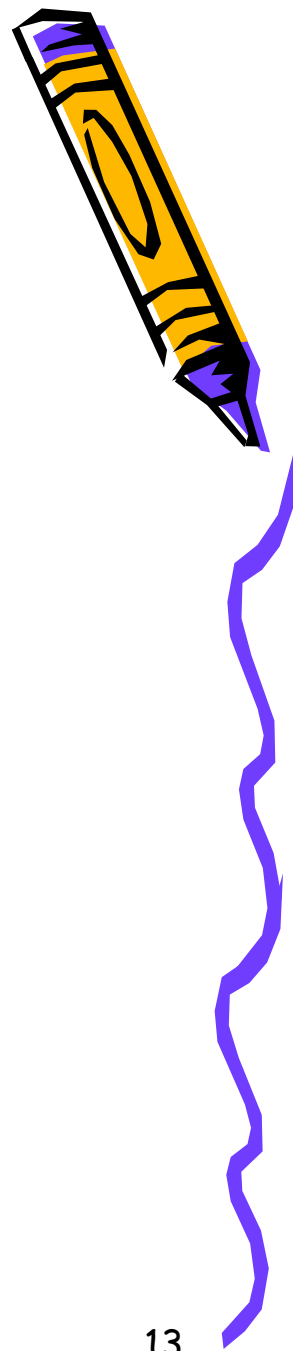
To develop Ajax

1. Download JQuery library

Or

2. Google Ajax Library API

3. Create the html code

A. Mousavi

Communication with the server

- **XMLHTTPRequest object**
  - asynchronous GETs + POSTs interchange with the server
  - Simultaneous multiple XMLHTTPRequest
  - Specify a handler method for state changes
  - Handler notified when request is:
    - Initialised
    - Started
    - In the process of being returned
    - Completed
- **IFRAME**
  - A "mini-browser" window in an HTML document
  - Can be hidden (0 width, 0 height)
  - Can call a URL
  - JavaScript can read the contents of the IFRAME
  - User sees messages on status bar
  - Event listener handles submits request
  - Slower than XMLHTTPRequest

A. Mousavi

XMLHttpRequest Object: Methods
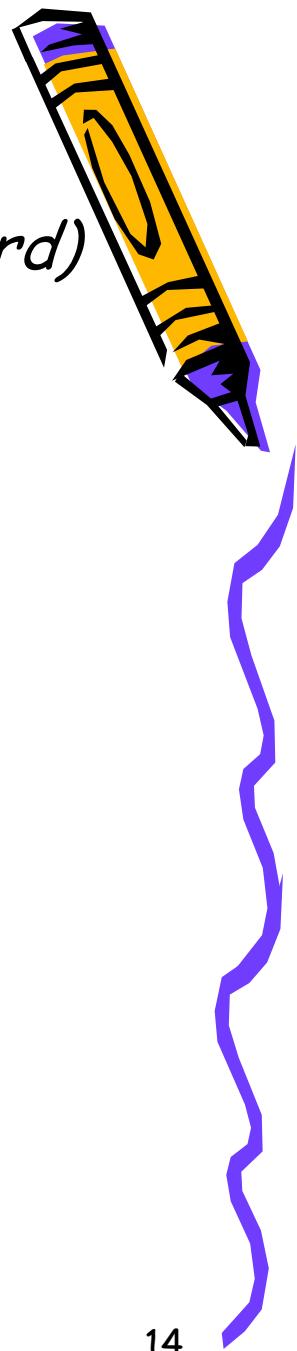
- *open("method", "URL", async, username, password)*
  - Assigns destination URL, method, etc.
- *send(content)*
  - Sends request with *post* method for string or DOM object data
- *abort( )*
  - Terminates current request
- *getAllResponseHeaders( )*
  - Returns headers (labels and values) in string format
- *getResponseHeader("header")*
  - Returns value of a given header
- *setRequestHeader("label","value")*
  - Sets Request Headers before sending

XMLHttpRequest Properties
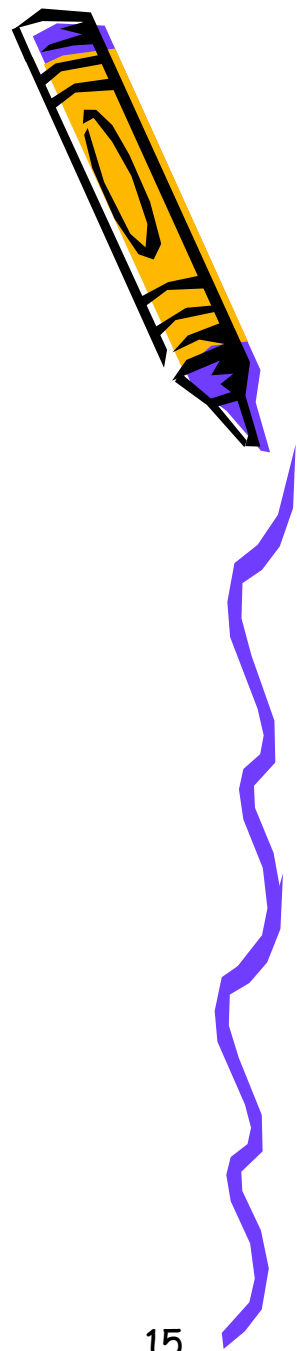
- **onreadystatechange**
  - fires at each state change via this event handler
  - implement your own function that handles this request
- **readyState – current status of request**
  - 0 = uninitialised
  - 1 = loading
  - 2 = loaded
  - 3 = interactive (some data has been returned)
  - 4 = complete
- **status**
  - HTTP Status returned from server: 200 = OK
- **responseText**
  - String version of data returned from server
- **responseXML**
  - XML DOM document of data returned
- **statusText**
  - Status text returned from server

A. Mousavi

Google example

| adv		Advanced Search Language Tools
adventure quest	23,800,000 results	
advent	22,400,000 results	
adventure quest worlds	17,800,000 results	
adventure games	39,500,000 results	
advfn	1,050,000 results	
advent calendar	5,150,000 results	
advanced system care	95,900,000 results	
adventureland	17,800,000 results	
adventure holidays	49,900,000 results	
adverts	12,700,000 results	
	close	

How It Works

- **Fires roughly every keystroke**

  - Uses a timer to determine when to send a request to the server

- **Creates a hidden DIV that is shown when you start typing**

- **DIV is populated with the data from the server**

- **Text field is set to include the next word from the list in the server**

- **Results cached; if you backspace, server is not called again**

- **Has a timer adjustment to increase or slow down server calls**
  - Slow dialup sessions hit server fewer times
  - Fast broadband hits a lot

A. Mousavi

AJAX Patterns
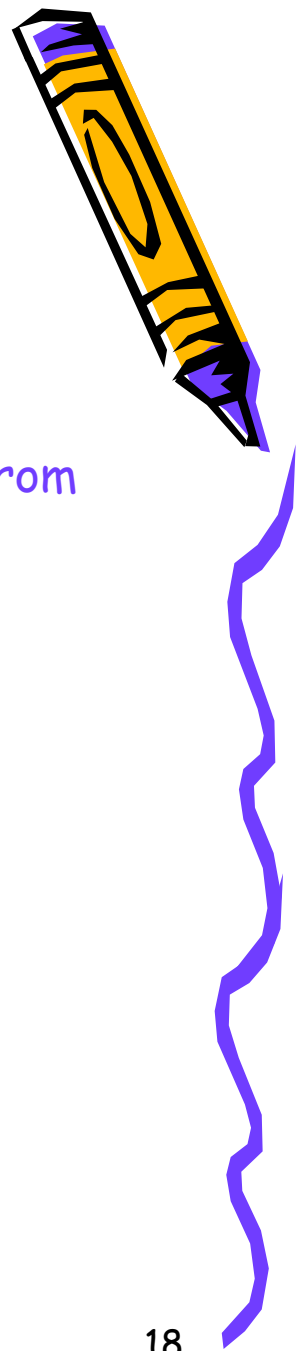
- **Change Browser Content with the DOM**
  - Return HTML or XML and change named items
    - Usually DIVs or SPANs
- **Remote Scripting**
  - Makes the execution of specified functions on the server from the client-side possible
- **Events & Scheduling**
  - Application of timers to ping server at intervals
- **Specialized Functions to perform common UI tasks**
  - Populate SELECT lists
  - Pass target DIV to-from server
  - Instantaneous Validation of the document
  - SAVE instead of SUBMIT
  - CANCEL instead of BACK
- **DHTML Techniques**
  - CSS, Drag & Drop
- **Server-Side Code Generation**
  - Return Javascript instigated functions to the client

A. Mousavi

AJAX Frameworks
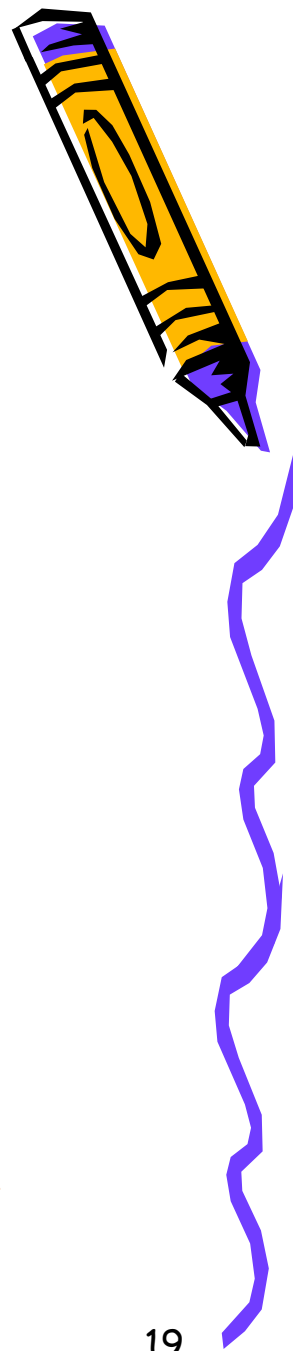
- **Pure JavaScript Framework**
  - Infrastructure
    - Provides basic piping & portable browser abstractions
    - Content up to developer
    - Typical Functionality:
      - <u>Wrapper around XMLHttpRequest</u>
      - <u>XML manipulation & interrogation</u>
      - <u>DOM manipulations based on responses from XMLHttpRequest</u>

  - Application Framework

- **Server-Side Framework**
  - HTML/JavaScript Generation
    - Server provides complete HTML/JavaScript code and browser ←→ server coordination
    - Customisation (Browser side)
  - Remote Invocation
    - JavaScript calls routed directly to server-side functions (Java Methods) and returned back to Javascript callback handlers

Also see clearnova literature. http://www.ajaxcoded.com/

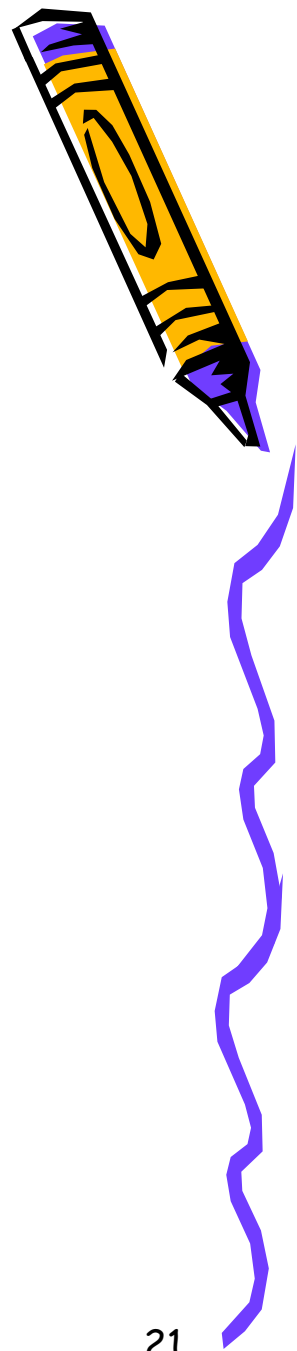A. Mousavi

Some to choose from
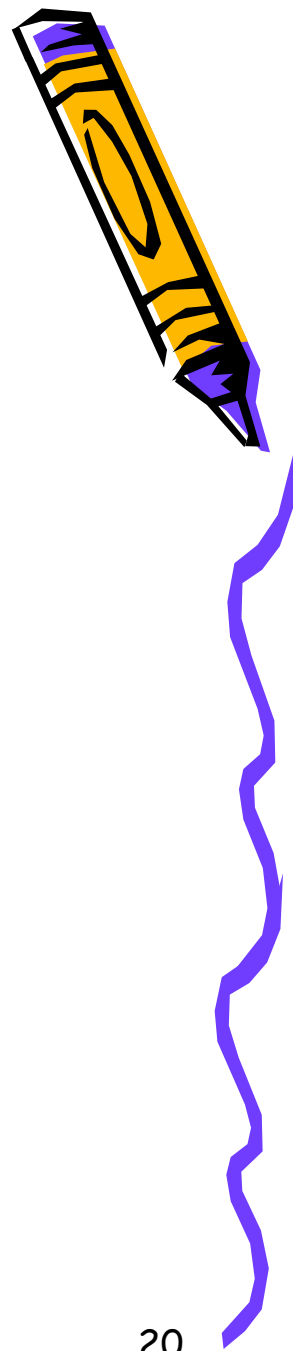
- **Pure JavaScript**
    - DOJO (9/04)
    - Prototype (2001)
    - Open Rico (5/05)
    - Qooxdoo (5/05)
- **Pure JavaScript Infrastructural**
    - AjaxJS (5/05)
    - HTMLHttpRequest (2001)
    - Interactive Website Framework (5/05)
    - LibXMLHttpRequest (6/03)
    - RSLite
    - Sack (5/05)
    - Sarissa (2/03)
    - XHConn (4/05)
- **Server-Side (Multi Language)**
    - Cross-Platform Asynchronous Interface Toolkit (5/05)   also see: **www.ajaxpatterns.org**
    - SAJAX (3/05)
    - Javascript Object Notation (JSON) & JSON-RPC
    - Javascript Remote Scripting (2000)
- **Server-Side (Java)**
    - Echo2 (3/05)
    - Direct Web Remoting (DWR) (2005)]
    - ThinkCAP Minerva (04/2005)
- **Server-Side (Lisp)**
    - CL-Ajax
- **Server-Side (.NET)**
    - Ajax.NET (305)
- **Server-Side (PHP)**
    - AjaxAC (4/05)
    - JPSpan
    - XAJAX
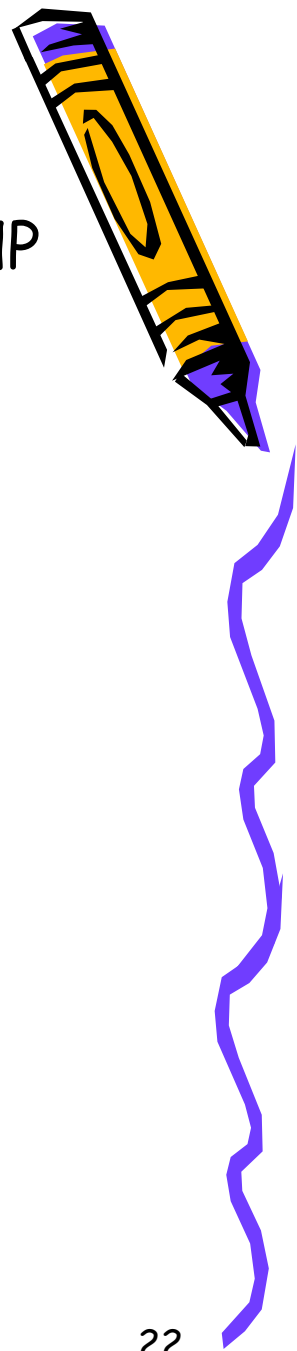- **Server-Side (Ruby)**
    - Ruby-On-Rails (3/05)

A. Mousavi

Examples

- http://www.chazzuka.com/blog/?p=88

- Gmail: **http://mail.google.com**

- EcoXChange website: http://eco-xchange.com/

- Google maps: http://maps.google.co.uk/

- Flickr.com

Further Reading

- Sams Teach Yourself Ajax, JavaScript, and PHP All in One by Phil Ballard & Michael Mancur (2008)

- Ajax on Rails by Scot Raymond, O'Reily Media (2007)

- Ajax Programming for the Absolute Beginner
  By Jerry Ford, Delmar 2008

- Learning jQuery 1.3 by Jonathan Chaffer, Karl Swedberg ISBN 1847196705 , 2009

A. Mousavi