

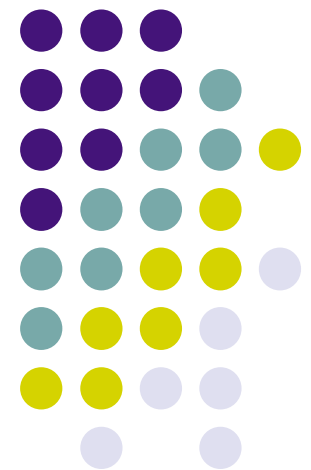
# Making The Leap to Web 2.0

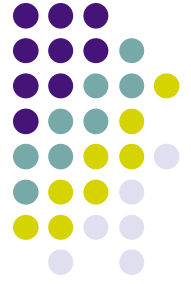
---

Jason Sliss

[plethora@optonline.net](mailto:plethora@optonline.net)

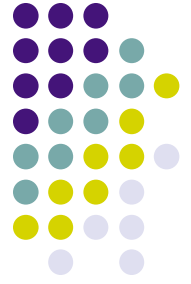
October 31, 2007





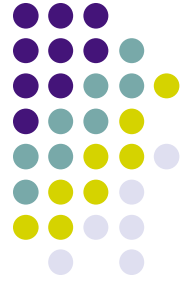
# What was Web 1.0?

- Page-for-Page Interaction (Browsing)
- Information Catalog Retrieval
- Isolated Storefronts
- Limited Application Functionality
- User Groups
- Email Communication to Customers
- Browser Wars



# What is Web 2.0?

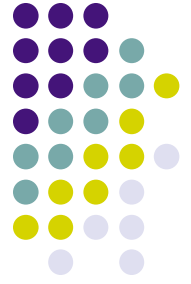
- Web as a Platform (Software as a Service)
- Context-Based Interaction (Search Results)
- Information Catalog Contribution (wiki)
- Storefront Cross-Selling (suggestions, advertising)
- Storefront Participation (reviews and feedback)
- Richer Application Functionality (mouse interactions, in-place page updates, mash-ups, AJAX, REST)
- Social Networking (personal/professional sites, forums, blogs, rating sites)
- Enhanced Real-Time Communication (Syndicated News and Information Feeds)
- Browser War Irrelevance through cross-browser compatible open-source rich-library creation (Javascript, Java, .NET, Ruby, PHP)



# What is Web 2.0?

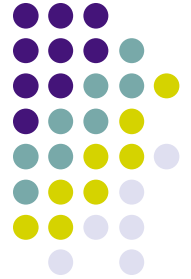
- Web as a Platform (Google Apps, eBay, YouTube, Flickr)
- Context-Based Interaction (Google, Yahoo, ask.com)
- Information Catalog Contribution (Wikipedia, MOC, SourceForge)
- Storefront Cross-Selling (Amazon, w00t!, kayak)
- Storefront Participation (Amazon.Com, eBay)
- Richer Application Functionality (Maps, ToolTips, Tabs, Pop-Ups, Expand/Collapse Panes, Animation)
- Social Networking (Facebook/LinkedIn, blogspot, digg)
- Enhanced Real-Time Communication (RSS/Atom traffic, weather, newest aluminum siding products)
- Browser War Irrelevance through community rich-library creation (Prototype, ASP.NET AJAX, GWT)

# How Did We Get to Web 2.0?

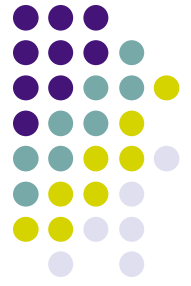


- Dot-Com Bust and Dissatisfaction with Web Applications
- Increased community collaboration and self-motivated drive to produce quality technology
- Increased information sharing amongst grass-roots groups
- New generation of technology users networking at earlier ages (email, texting, IM, video chat)
- Increased access to information only increased the appetite for more information
- More fast-track routes to Fame

# Leap to Web 2.0 NOW



- AJAX – Asynchronous Javascript And Xml
- Make Asynchronous Server Calls
- In-place Page Updates vs. Page Navigation

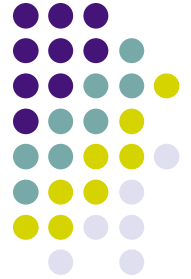


# What is Asynchronous?

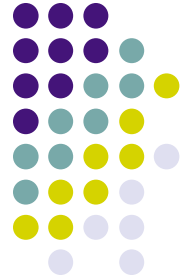
- A Server Request is made outside of the current thread of execution
- A **callback function** is specified to execute when the Request completes
- The callback function typically will update one or more parts of the rendered content, without reloading the page

# Start AJAX Today

- CSS
- DOM
- Javascript
- XMLHttpRequest



# Get The XMLHttpRequest



```
function initXMLHttpRequest() {
    var xRequest = null;
    if (window.XMLHttpRequest)
    {
        xRequest = new XMLHttpRequest(); <!-- Mozilla/Safari -->
    }
    else if (window.ActiveXObject)
    {
        xRequest = new ActiveXObject("Microsoft.XMLHTTP"); <!-- Internet
Explorer -->
    }
    return xRequest;
}
```

# Make the Request



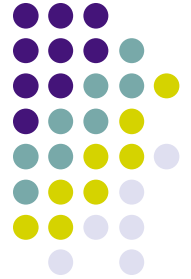
```
function sendRequest(url,params,HttpMethod)
{
    if (!HttpMethod)
    {
        HttpMethod = "GET";
    }

    req = initXMLHttpRequest();
    if (req)
    {
        req.onreadystatechange = onReadyState;
        req.open(HttpMethod, url, true);
        req.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
        req.send(params);
    }
}
```

# Define the Callback



```
function onReadyState() { <!-- Callback handler -->
  var ready = req.readyState;
  var data = null;
  if (ready == READY_STATE_COMPLETE)
  {
    data = req.responseText;
  }
  else
  {
    data = "loading..." + ready + "";
  }
  toConsole(data); <!-- Render Results -->
}
```



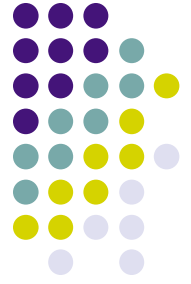
# Render the Results

- Using div and span Elements to hold callback-rendered content

```
<div id='console'></div>
```

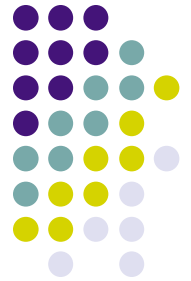
```
function toConsole(data)  <!-- Renders the content to the page -->
{
  if (console != null)
  {
    var newline = document.createElement("div");
    console.appendChild(newline);
    newline.innerText = data;
  }
}
```

```
window.onload = function()
{
  console = document.getElementById('console');
  sendRequest("http://www.cnn.com");  <!-- Actual server request -->
}
```



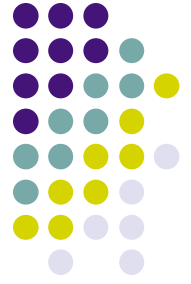
# AJAX Examples

- Asynchronous Request Page
- Request for Header Info
- Request for RSS Feed XML
- RSS Reader One-Page Application
- Microsoft Mapping and Yahoo Traffic Mash-up



# The Future is Now

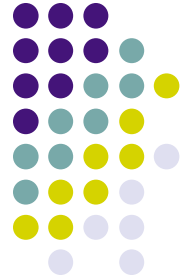
- Software as a Service – on-line data hosting and backup, .Mac, salesforce.com, Facebook, eBay, Gmail/Google Apps
- User-Created Applications without Programming –Visual Mash-Up Tools like Yahoo! Pipes and Microsoft Popfly
- Distributed Data Hosting (vudu)
- Microsoft Silverlight vs. Flash
- More on Facebook and MSFT at <http://msdn2.microsoft.com/en-us/express/bb510381.aspx>



# What To Explore Now

- AJAX and REST
- XML and JSON
- Use Open-Source Libraries (Prototype, jQuery, ASP.NET Ajax)
- Consume RSS/Atom Feeds
- Use Mapping Tools, Facebook Apps, Flickr/YouTube tools
- Create Mash-Ups
- Host and/or Contribute to a Wiki

# Bibliography



- *Ajax in Action*, Manning, 2006
- <http://www.manning.com/crane/>