

AJAX Essentials 2

Basics

- Asynchronous
- JavaScript
- And
- XML
- Browser won't hang waiting for a response
- When server response is received, then will process and update
- AJAX, by definition, must involve a request and a response to a server

Ahead

- DOM & XML
- XML HTTP Requests
- Previewing Links with Ajax
- Creating Dynamic Forms
- Dynamic Maps
- AJAX Frameworks
- YUI Animation widget (UI Library)

Ajax Development Toolkit

- Text Editor
- Browser
- Web Server
- HTML, CSS JavaScript

Ajax Development Workflow

- PreReq
- Static HTML site
- Site with some scripting
- Site with server-side code, but not client side
- Make sure it is all working

Flow


- Edit
- Move file to the server
- Test
- Repeat

Text Editors

- Mac
 - BBEdit
 - Textmate
- Windows
 - UltraEdit
 - E Text Editor
- X-Platform
 - Eclipse
 - Komodo



Browser: FireFox

- FireBug

**Firebug 1.12.8**Wednesday, July 16, 2014

Web Development Evolved. Firebug is free and open source softwa... [More](#)



-
- Web Developer

 **Web Developer will be installed after you restart Firefox.** [Restart now](#) [Undo](#)

Web Developer 1.2.5Wednesday, July 16, 2014

Adds a menu and a toolbar with various web developer tools. [More](#)

-
- DOM Inspector

 **DOM Inspector will be installed after you restart Firefox.** [Restart now](#) [Undo](#)

DOM Inspector 2.0.14

Inspects the structure and properties of a window and its contents. [More](#)

-

HTML

```
<html>
<head>
  <title>Make Your Own Bingo Card</title>
  <link rel="stylesheet" rev="stylesheet" href="script.css" />
  <script type="text/javascript" src="script.js">
  </script>
</head>
<body>
<h1>Create A Bingo Card</h1>
<table>
  <tr>
    <th width="20%">B</th>
    <th width="20%">I</th>
    <th width="20%">N</th>
    <th width="20%">G</th>
    <th width="20%">O</th>
  </tr>
  <tr>
    <td id="square0">&nbsp;</td>
    <td id="square1">&nbsp;</td>
    <td id="square2">&nbsp;</td>
    <td id="square3">&nbsp;</td>
    <td id="square4">&nbsp;</td>
  </tr>
  <tr>
    <td id="square5">&nbsp;</td>
    <td id="square6">&nbsp;</td>
    <td id="square7">&nbsp;</td>
    <td id="square8">&nbsp;</td>
    <td id="square9">&nbsp;</td>
  </tr>
  <tr>
    <td id="square10">&nbsp;</td>
    <td id="square11">&nbsp;</td>
    <td id="free">Free</td>
    <td id="square12">&nbsp;</td>
    <td id="square13">&nbsp;</td>
  </tr>
  <tr>
    <td id="square14">&nbsp;</td>
    <td id="square15">&nbsp;</td>
    <td id="square16">&nbsp;</td>
    <td id="square17">&nbsp;</td>
    <td id="square18">&nbsp;</td>
  </tr>
  <tr>
    <td id="square19">&nbsp;</td>
    <td id="square20">&nbsp;</td>
    <td id="square21">&nbsp;</td>
    <td id="square22">&nbsp;</td>
    <td id="square23">&nbsp;</td>
  </tr>
</table>
<p><a href="script.html" id="reload">Click here</a> to create a new card</p>
</body>
</html>
```

Create A Bingo Card

| B | I | N | G | O |
|----|----|------|----|----|
| 13 | 27 | 34 | 58 | 61 |
| 2 | 16 | 44 | 54 | 71 |
| 9 | 22 | Free | 52 | 63 |
| 5 | 23 | 31 | 47 | 64 |
| 3 | 17 | 37 | 48 | 65 |

[Click here](#) to create a new card

```

window.onload = newCard;
var usedNums = new Array(76);

function newCard() {
    if (document.getElementById) {
        for (var i=0; i<24; i++) {
            setSquare(i);
        }
    }
    else {
        alert("Sorry, your browser doesn't support this script");
    }
}

function setSquare(thisSquare) {
    var currSquare = "square" + thisSquare;
    var colPlace = new Array(0,1,2,3,4,0,1,2,3,4,0,1,3,4,0,1,2,3,4,0,1,2,3,4);
    var colBasis = colPlace[thisSquare] * 15;
    var newNum;

    do {
        newNum = colBasis + getNewNum() + 1;
    }
    while (usedNums[newNum]);

    usedNums[newNum] = true;
    document.getElementById(currSquare).innerHTML = newNum;
}

function getNewNum() {
    return Math.floor(Math.random() * 15);
}

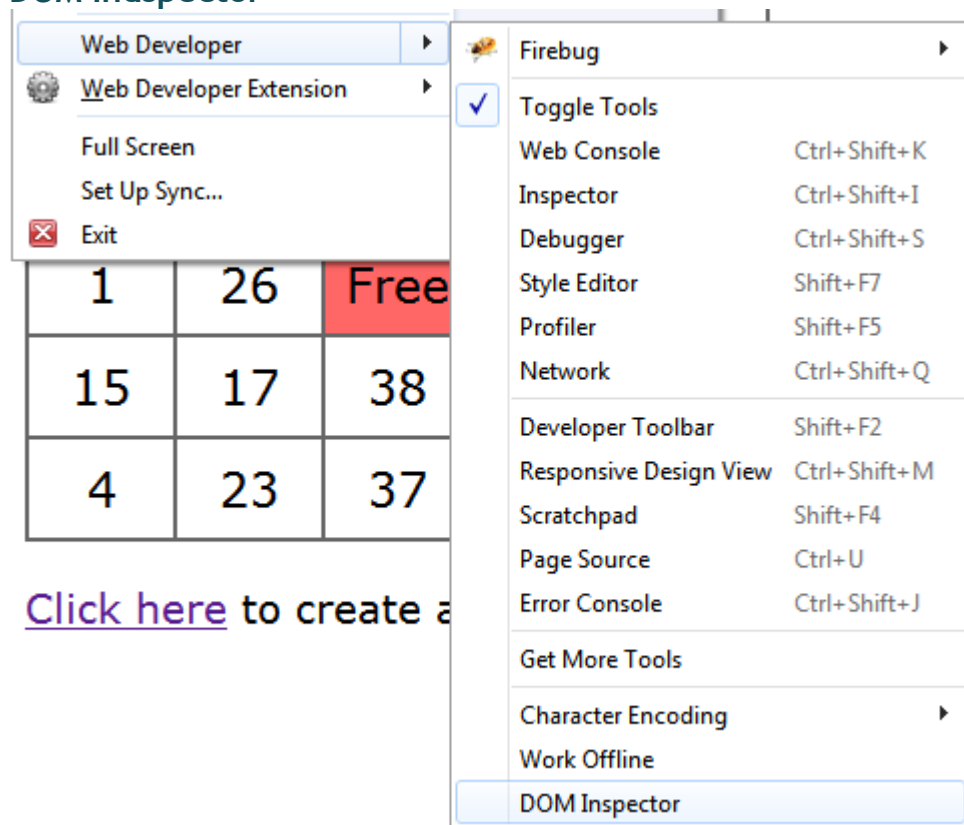
```

DOM

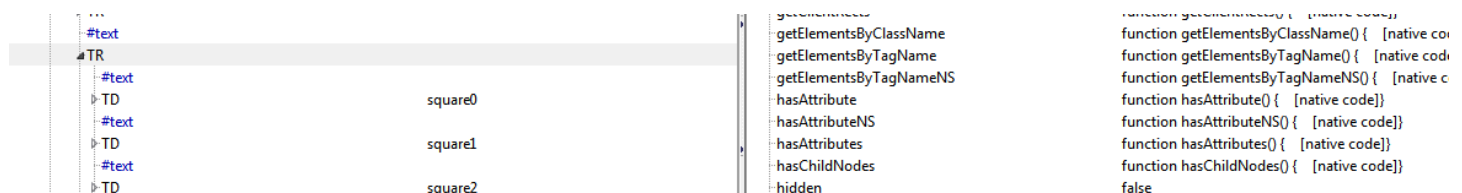
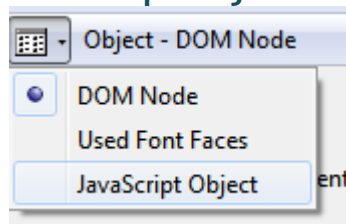
Document Object Window
A model of a web page

Firefox Tools

DOM Indspector



JavaScript Object Inspector



JavaScript DOM

XML

Extensible Markup Language

Human-digestible

XMLHttpRequest Object

Javascript is CASE Sensitive

PROPERTIES

- onreadystatechange
- readyState
 - 0 - Uninit: no data
 - 1 - Loading: object is currently loading its data
 - 2 - Loaded object has finished loading its data:
 - 3 - Interactive: User may interact with the object even though it's not fully loaded
 - 4 - Complete: Object has finished initializing
- .responseText
- .responseXML
- status
 - 200 - OK
 - 400 - Bad Request
 - 401 - Unauthorized
 - 403 - Forbidden
 - 404 - Not Found
 - 500 - Internal Server Error
- .statusText

METHODS

- abort()
- getAllResponseHeaders()
- getResponseHeader("header")
- open("method","url",async,[“user”,“pword”])
 - open(“GET”, “myFile.xml, true)
- send(content)
 - send(null)
 - send(xmlText)
- setRequestHeader("header","value")
- string.getResponseHeader(string header)

Basic Page Structure

```
<html>
<head>
    <title>Page Title</title>
    <script src="script.js" type="text/javascript"></script>
</head>
<body>
    <div id="updateArea">&nbsp;</div>
</body>
</html>
```

```
window.onload = makeRequest;
var xhr = false;
```

xhr.onreadystatechange & status

```
window.onload = makeRequest;
var xhr = false; // global var
```

```
/*
Create the XMLHttpRequest Object based on browser type
*/
function makeRequest() {
    // if this exist, then do - NOT IE6
    if (window.XMLHttpRequest) {
        xhr = new XMLHttpRequest();
    }
    else {
        // OR...if this exist, then do - IE6
        if (window.ActiveXObject) {
            // use try/catch to eliminate runtime error OR capture error info
            try {
                xhr = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (e) { }
        }
    }

    // if the XMLHttpRequest object exists...
    if (xhr) {
        xhr.onreadystatechange = showState;
        xhr.open("GET", "colors.xml", true);
        xhr.send(null);
    }
    else {
        document.getElementById("updateArea").innerHTML = "Sorry, but I couldn't create an XMLHttpRequest";
    }
}

function showState() {
    var currMsg = document.getElementById("updateArea").innerHTML;
    document.getElementById("updateArea").innerHTML = currMsg + "<p>The current state is " +
    xhr.readyState + " and the status is " + xhr.status + "</p>";
}
```


Anonymous Functions

- JavaScript makes extensive use of anonymous functions, namely for Event Handlers

Setup

Element

`<p>Request an XML file as XML</p>`

Script

```
document.getElementById("id_name").event_handler = function() {}
```

- The HTML file creates two `<a>` tags with ids `requestXML` & `requestText`

```
<html>
<head>
  <title>A Simple Ajax Script</title>
  <script src="script.js" type="text/javascript">
  </script>
</head>
<body>
  <p><a id="requestXML" href="#">Request an XML file as XML</a></p>
  <p><a id="requestText" href="#">Request an XML file as text</a></p>
  <div id="updateArea">&nbsp;</div>
</body>
</html>
```

- The JavaScript file gets the `<a>` elements ById and creates 'onclick' handlers with anonymous functions

```
function initAll() {
  document.getElementById("requestText").onclick = function() {
    textRequest = true;
    makeRequest();
    return false;
  }
  document.getElementById("requestXML").onclick = function() {
    textRequest = false;
    makeRequest();
    return false;
  }
}
```