

Announcements

- Acid Tests
 - <http://acid3.acidtests.org/>
- Maxthon Browser
 - <http://www.maxthon.com/>
- <http://caniuse.com/>

Private Variables

- **Example:** Private.html

Private Variables/Functions

- Closures – functions that can access variables from another functions' scope
 - Usually defined by creating a function inside of another function
 - A scope chain makes possible accessing functions/variables
- **Example:** PrivateVarFunctions.html

Nonextensible and Sealed Obj

- In JavaScript you can add properties and methods to an object any time (extensible)
- You can restrict this behavior by using `Object.preventExtensions()`
- What if you don't want properties deleted as well?
- **Example:** ExtensibleSealed.html

Freeze

- Strictest protection
- Not extensible, sealed and data properties can not be modified
- **Example:** Freeze.html

JavaScript Console

- Allow us to view JavaScript errors
- **console** object functions
 - log → General message
 - info → Informational message
 - error → Error message
 - warn → Warning message
- In Chrome
 - Tools → JavaScript console
 - Notice that different icons are used
- **Example:** consoleEx.html

Errors

- **Example:** Errors1.html
- Error types
 - Error → Base type for errors
 - URIError
 - TypeError
 - EvalError → Error while using eval()
 - RangeError
 - SyntaxError
 - ReferenceError
- **Example:** Errors2.html

throw

- Can be used to throw custom errors
 - throw “NOOOO!”;
 - throw { type1Error: “Done” }
- You can customize your errors
- **Example:** Errors3.html

Strict Mode

- Allows for error checking both globally or within a function
- Use the strict mode pragma
 - “use strict”;
- If pragma used outside of a function it applies to all the script
- It can appear in a function

```
function computeAvg() {  
  “use strict”;  
}
```
- **Example:** Strict1Var.html
 - We need to use var
 - Look at the console message (ReferenceError)
 - Cannot use reserved words (interface, package, private, ...)

Strict Mode

- Objects → Errors reported
 - Trying to add property to nonextensible object
 - Defining property twice
- Other restrictions in strict mode
 - with statement removed
 - with statement affects how identifiers are resolved
 - Octal literal removed
 - Function parameter's name must be unique
 - Others...
- **Example:** Strict2Obj.html

Array Functions

- concat → returns copy of joined arrays
- indexOf → returns position of element in array
- join → returns string with all elements in the array
- pop → removes & returns last element
- push → adds to the end (returns length)
- reverse → reverses the array
- shift → removes & returns first element
- unshift → adds new element to the beginning
- slice → selects part of an array
- split → splits string into an array

Regular Expressions

- `var pat1 = new RegExp(pat, modifiers)`
- Alternative
- `var pat1 = /pat/modifiers/;`
- Modifiers
 - `i` → case-insensitive matching
 - `g` → global match
- **Example:** `Regexp.html`

JSON

- JSON – JavaScript Object Notation
 - Syntax for serializing objects, arrays, numbers, booleans, and null
 - Based on JavaScript syntax, but distinct from it
 - Some JavaScript is not JSON and some JSON is not JavaScript
 - See <http://timelessrepo.com/json-isnt-a-javascript-subset>
- Lightweight data-interchange format
- Alternative to XML
- Derived from JavaScript but it is language independent
- JSON Example: <http://json.org/example.html>
- Methods
 - `JSON.parse()` → parse a string as JSON (returns the Object corresponding to the JSON text)
 - `JSON.stringify()` → returns a string corresponding to the specified value
- Examples and information: <http://www.w3schools.com/json/>
- References: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON
- **Example:** JSONExample.html

HTML5 Video

- HTML5 <video> element allows embedding of video in a web page
- Example:

```
<video width="400" height="400" controls>  
  <source src="mystory.mp4" type="video/mp4">  
  HTML5 video element not supported by browser  
</video>
```

- You can add or replace controls with **autoplay**
- Several files can be listed and the browser will use the first format recognized
- Example:
http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_video_all

iframe

- Format:
`<iframe src="URL_TO_DISPLAY"></iframe>`
- Additional attributes
 - width and height
 - To remove border use `style="border:none"`
- Playing youtube videos with iframe
 - http://www.w3schools.com/html/html_youtube.asp
 - You can customize the video interaction by adding options after the link.
 - `?controls=0` → player controls does not display
 - `?loop=1` → video will loop (forever)

HTML5 Audio

- HTML5 <audio> element allows playing audio files
- Example:
- <audio **controls**>
 <source src="macarena.mp3" type="audio/mpeg">
 HTML5 video element not supported by browser
</audio>
- You can add or replace controls with **autoplay**
- Several files can be listed and the browser will use the first format recognized
- Example:
http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_audio_all

HTML5 Canvas

- Canvas
 - Container for graphics
 - It is a rectangular area on the page
 - You use JavaScript to draw on the fly
 - Suited for game applications
 - You can store the image as a jpg or png file
- Reference and Examples:
 - http://www.w3schools.com/html/html5_canvas.asp
- **Example:** DrawingPointer.html
- How would you create an animation of a drawing you have done?

HTML5 SVG

- SVG – Scalable Vector Graphics
 - Language use to describe 2D graphics
 - Pure XML
 - SVG graphics do not lose any quality when resized or zoomed
 - Best suited for applications with large rendering areas
 - Not appropriate for games
 - You can embed in your HTML
- <svg> element → container for SVG Graphics
- Reference, Examples and comparison table available at:
 - http://www.w3schools.com/html/html5_svg.asp
- Open-source vector graphics editor
 - <https://inkscape.org/en/>

HTML5 Form Elements

- **Example:** HTML5FormElements.html

HTML5 Geolocation API

- **Example:** Geolocation.html
- **Example (in CS server):**
<http://www.cs.umd.edu/~nelson/classes/utilities/Geolocation.html>
- **Reference:**
 - http://www.w3schools.com/html/html5_geolocation.asp

HTML5 Local Storage API

- **localStorage** → stores data with no expiration date
- To store data:
 - `localStorage.setItem("name", "Mary");`
- To retrieve data:
 - `localStorage.getItem("name");`
- You can only store strings. You store objects you could use `JSON.stringify`
- **sessionStorage** → equivalent to `localStorage`, but data is deleted when the browser is closed
- **Example:** `ToDoList.html`
- To clean `localStorage`
 - Right click on page in Chrome
 - Select "Inspect element"
 - Select "Console"
 - Type `"window.localStorage.clear()"`
- Reference: http://www.w3schools.com/html/html5_webstorage.asp