HTML+CSS+JavaScript

• Three “little languages”
• No, four: CSS has two.
• No, five: JavaScript has two also.
• Intermingled in several ways.
It starts with Runoff

• Text formatting program developed for CTSS in 1964
• Interleaved *text lines* with *dot commands*
• Dot commands are a form of *markup*
• They are a simple *imperative language*. 
Many successors

• Many universities and companies copied it.
• And extended it, making richer languages.
• CMU saw document processing as interpreting the dot language and devised a compiler.
• Brian Reid did a PhD at CMU: he came up with the first declarative markup language.
Charles Goldfarb at IBM came up with GML (Generalised Markup Language)

Separated *structure* from *presentation*. One document could be formatted many ways.

Just like Scribe!

Standardised as Standard GML.

A trace of imperative left: <?...>
Core ideas of SGML:

- There is a meta-language in which you can specify document grammars, e.g.,
  ```xml
  <!ELEMENT html O O (head, body)>
  ```
- Documents mark up structure.
- The semantics of a document is a tree.
• Leaves of the tree can be text strings, called *parsed character data*.

• The principal other nodes are *elements*, having *generic identifiers* to say what kind, *attributes* to attach properties, and *children* which are text and elements.
SGML 3

- Elements are delimited by tags.
- `<name att1=”val1” ... attn=’valn’>` is a start tag. `name` is a generic identifier.
- `</name>` is an end tag.
- Meant for human entry, so abbreviations are allowed, e.g., `<item>xxx</>` and `<item/xxx/>`. Attributes may have defaults.
Special characters

- `< and >` are special.
- Use `<` and `>` to get them in text.
- So `&` is special too. Use `&` for that.
- `<gi a=""b='"'>`
- Can get any Unicode character by number
- HTML has long list of character names.
Central idea:

- Markup to indicate *structure* and *semantics*
- Not presentation (appearance).
- A document is a *tree*, a little data base,
- not an image.
Linking

• Documents often contain cross references

• Attribute types ID (unique identifier) for naming a node and IDREF for referring.

• `<!ATTLIST ref id IDREF #REQUIRED>`
See `<ref id=section-4>`.

• So it’s a cross-linked tree.
HTML applies SGML

• HTML was one SGML application of many.

• Had lots of presentation markup like:
  `<i>`  `<b>`  `<u>`  `<s>`  `<big>`  `<small>`  `<sup>` ...

• Has links to other documents; introduced URLs.

• In particular, lets you point to documents.
HTML4: back 2 future

- HTML4 tried to move away from presentation markup.
- Control over presentation was given to a new language, CSS.
- HTML lets you point to supporting material, include it in special elements, or include it in special attributes.
Styling

• `<span style="font-family: Times, serif">this should be in Times if you have it</span>`

• `<head><style type="text/css">.rouge {color: red}</style>... <p class=rouge>This is in red</p>`

• `<head><style src="some-URL">...</head>`
Two sublanguages

• First sublanguage: lists of property: value pairs. These can be used in style="..." attributes.

• Core idea: style sheet uses structure of tree and adds attributes to nodes.

• Second sublanguage says which nodes.
CSS selectors

- [http://www.w3schools.com/cssref/css_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)
- Specify what kind of element (p, h2)
- Or by class attribute .cls match class="cls"
- Or even by unique ID: #theid
CSS selectors 2

- Or by any attribute [attr=val]
- Or by tree structure: parent>child
- Language is frustratingly non-orthogonal
- Also fictitious things like :first-letter
Other style approaches

- DSSSL (Document Structure and Semantics Specification Language) is an enrichment of pure Scheme (a functional programming language) for SGML.

- XSL (XML Stylesheet Language) is a weird and difficult style language for general XML.

- CSS is meant to be simple for non-programmers; applicable to HTML & XML.
What CSS does

- A *declarative* notation for adding formatting *properties* to trees.
- Style sheets can be selected to suit devices and can be merged.
- It relies on the HTML trees and does not change them. Still *static* pages, just prettier.
Dynamic! Pages!

- HTML has forms.
- Wouldn’t it be nice to validate form entries before sending them to the server?
- And wouldn’t it be cool to have pages that change?
What didn’t take off

• Java `<APPLET>`s embedded in pages.

• The problem with applets (and Flash) is that it does not embrace HTML but replaces it. It doesn’t make HTML dynamic.

• Applets and Flash have a long sad history of security bugs.
JavaScript

• Netscape gave Brendan Eich ten days to design and implement a programming language that would
  • be interpreted by the web browser
  • have access to the page as a data structure
  • be able to change the page as data.
Deceitfully familiar

- JavaScript was given syntax that looks like Java, faithfully copying some of its flaws.
- But the semantics is utterly different.
- The semantics is more like Lisp and Self. There are no classes. Functions may be freely nested and nesting works. Objects inherit properties from objects, not classes.
JavaScript attributes

• onclick="code", ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout
• onkeypress, onkeydown, onkeyup
• onload, onunload (<body>)
• onfocus, onblur (<a>, <area>, <label>)
• onsubmit, onreset (<form>)
• Just like CSS, you can *point to* a file of JavaScript code, or you can *embed* <script>...code... </script>

• Unlike CSS, JavaScript was *not* designed to fit comfortably in attributes or elements.
Lexical problems:

• Java has && and &. Therefore so does JavaScript. But & is special!

• Java has < <= << > >= >> >>>. Therefore so does JavaScript. But < and > are special!

• Get your editor to automatically convert & < > to &amp; &lt; &gt;
The horror! The horror!

- SGML, hence HTML, has a solution. 
  `<![CDATA[ .... ]]>` is a sort of superquote.
- `& < >` are not special inside that, only `]]>`.
- (Beware of `a[p[i]]>b[q[i]]` !)
- People abused **comments** instead:
  `<!-- comment -->`
- (Beware of `x-->0` !)
Comment abuse

• In SGML, comments are comments. They are not part of the data structure that a document represents.

• Many HTML browsers broke that rule so that scripts could be hidden in comments.

• This is now official in XML.
The data structure

• An SGML document can be represented by a very simple space-efficient data structure.

• That’s not what we get. The Document Object Model (DOM) has a large and painfully clumsy API.

• Which still isn’t quite compatible between browsers.
Why jQuery?

• jQuery tries to hide browser differences.
• jQuery lets you use something very like CSS selectors to navigate the document.
• jQuery is much easier to use than the official DOM API.
• HTML originally did drawings using `<IMG>`
• Patent problems with GIF and JPEG, so PNG format devised for use with `<IMG>`.
• Scalable Vector Graphics (SVG) designed to be embedded in HTML and adapt to device
<canvas>

• <canvas> is a new element in HTML5.

• Its only use is to be drawn on by imperative code written in JavaScript.

• There are libraries like D3.js to make this much easier.

• jQuery and D3 use Lispy features of JS.