Hypertext markup language HTML

BWA Slides #3

Internet and World Wide Web

"The **Internet** is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to link several billion devices worldwide. It is an international network of networks…"
//Wikipedia

"The **World Wide Web** (abbreviated as WWW or W3,[1] commonly known as the Web) is a system of interlinked hypertext documents [or other resources] that are accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks."
// Wikipedia

Hypertext Documents

Hypertext documents created with Hypertext Markup Language (HTML)

Basically
- Similar to XML but …
- Fixed set of elements (tags), designed to describe page content (no namespaces)
- Default rendering style
- Well formedness not mandatory. If not well formed browser tries to fix it (one major reason for incompatibilities)
- Mostly case insensitive (string comparison case sensitive)
- Don't preserve whitespace
- A possible character encoding declaration (HTML 4 > must support UTF-8)

Hyperlinks

We only use HTML Links

Basically
- Using the **anchor** element with attribute **href**, <a href=URL…>
- Where URL "… is a valid URL potentially surrounded by spaces if, after stripping leading and trailing whitespace from it, it is a valid URL." (referencing WHATWG spec. of URL)
- When clicked, browser sends a GET request for the resource

URL

Confusion URL vs URI, they are different but use is very messy.

- We don't differentiate (slides, code samples uses both)
- A URL is a universal identifier (often a location the "L"). URL Living Standard (WHATWG)
- Must be an absolute or relative URL
- Search engines like URLs < 2048 chars

Input	protocol	host	hostname	port	pathname	search	hash
http://example.com/carrot#question%3f	http:	example.com	example.com	(empty string)	/carrot	(empty string)	#question%3f
https://www.example.com:4443?	https:	www.example.com:4443	www.example.com	4443	/	?	(empty string)

Absolute and Relative URLs

If scheme part missing should be interpreted as a relative URL (simplified)

- Interpreted as relative to the base URL of the document (base URL = URL to document itself)
- If leading / relative to **serverroot**

Example

```
Base URL: http://www.server.example/xyz/bar/zap.html
Relative URL in page zap.html: href="foo.html"

Result (as interpreted by server):  http://www.server.
example/xyz/bar/foo.html
```

(X)HTML Versions

First drafts by Tim Berners-Lee at CERN in late 1991
- Then versions up to HTML4.01
- Then XHTML1.0 as a stricter alternative to HTML 4.01 (XHTML is an XML document)
- Then XHTML2.0 and HTML5 was announced
- XHTML2.0 cancelled (not backward compatible)
- Then XHTML5, which is an update to XHTML1.x, defined alongside HTML5 in the HTML5 drafts.
  - W3C draft at http://dev.w3.org/html5/spec/Overview.html
  - WHATWG draft at http://www.whatwg.org/specs/web-apps/current-work/ (Apple, Mozilla, Opera)
  - Similar but not identical

Situation a bit confusing…

HTML4 vs HTML5

Simplified: HTML5 is “cooler” (many more features)
- Will make Flash, MS Silverlight, JavaFX (all plugin based) obsolete?

The details (see New API)

We use HTML 5 (5.1)
- NetBeans wizard will generate

HTML5 vs XHTML5

Simplified: XHTML5 is HTML5 written in XML
- Also use the W3C spec.

Many details

We will use both HTML5 and XHTML5 , see code samples
- Possible to select in NetBeans

There's also polyglot HTML5, use if you like…

The HTML5.1 Specification

Hard to read (though there are examples)

- Mixes HTML, XHTML and "APIs for interacting with in-memory representations of resources", processing models, types, URLs, forms, … amazing number of definitions

Possible useful: Inspecting HTML structure with an HTML DOM inspector

HTML 5.1 Elements Overview

Categories

4.1 The root element
4.2 Document metadata
4.3 Scripting
4.4 Sections
4.5 Grouping content (common tags)
4.6 Text-level semantics (common tags)
4.7 Edits
4.8 Embedded content
4.9 Tabular data (common tags)
4.10 Forms (common tags)
4.11 Interactive elements
4.12 Links

Global Attributes

Attribute for any element

- Global attributes

Attributes id and class important

- Used in conjunction with style sheets and JavaScript (which elements are affected)
- id's must be unique in page

HTML Rendering

"User agents are not required to present HTML documents in any particular way. However, this section provides a set of suggestions for rendering HTML documents that, if followed, are likely to lead to a user experience that closely resembles the experience intended by the documents' authors."

"In general, user agents are expected to support CSS, and many of the suggestions in this section are expressed in CSS terms."
// HTML 5.1

We'll use CSS more to come...

HTML Limitations

HTML is static no way to insert dynamic content

HTML is monolithic no way to compose pages

We later see how to overcome…

Internet Media types (MIME)

Standard identifier used on the Internet to indicate the type of data a message contains. MIME Specification

Media type in HTTP headers
- Accept: text/html, … (field in request header)
- Content type: image/gif (field in response header)
- Optional parameter for character encoding

HTML primary media type: text/html
XHTML primary media type: application/xhtml+xml

List of registered media types

HTML Form Elements

"A form is a component of a Web page that has form controls, such as text fields, buttons, checkboxes, range controls, or color pickers. A user can interact with such a form, providing data that can then be sent to the server for further processing (e.g. returning the results of a search or calculation). No client-side scripting is needed in many cases, though an API is available so that scripts can augment the user experience or use forms for purposes other than submitting data to a server.

Writing a form consists of several steps, which can be performed in any order: writing the user interface, implementing the server-side processing, and configuring the user interface to communicate with the server. " // HTML 5.1/4.10.1

Form Controls

Input controls have the name attribute (name of a value)

Each input control has both an initial value and a current value (strings)
- Initial may be specified with value attribute
- Current replaces the initial (at user input)

If form is reset, each control's current value is reset to its initial value

At form submission (request to server) some name attributes are paired with their current value and included in the submission (aka successful controls)

Basic HTML Form

```
<form action="..." method="..." >
   <!-- Controls -->
   <input type="text" name="   key" value="default"
/>
   <input type="submit" value="Submit" />
</form>
```

action = form processing agent URI (something on server side handling the request)
method = HTTP method POST or GET
content-type = Media type of request body (post only), default application/x-www-form-urlencoded, for binary data (file upload) multipart/form-data. Using default above.

Can't nest forms (many on same page ok)

Form Submission and Response

User clicks submit button...the Browser...

1. Identifies successful controls
2. Builds a form "data set" with "name=value"-pairs
3. Encode data set depending on content type
4. Sends the data set

User agents should render the response from the HTTP "get" and "post" transactions.

So it will render a "post"...i.e. we will see some page in the browser after the post, more to come ...

Server Side Form Handling

Possible to extract the incoming form values (=parameters) sent from browser by name (string)

```
// Typical style, getting parameters
// In HTML form <input ... name="name" value=”pelle”/>

// Server side Java
String value = request.getParameter("name");

// value is “pelle”
```

More to come …

Get vs Post in Forms

Post is the recommended write operation. Use post in forms

Post

- Data invisible
- Data in message body

Get (Bad)

- Data appended to URI as a query string (?a=1&b=2...) also hidden data (input type="hidden")! Visible in address field
- Limited URI length, 255 bytes, only accept ASCII
- Possible need URLEncoding ('=' encoded as %3D)

Types for Input Element

```
<input type="..." />
```

For decades just a handful, then came HTML5... (scroll down a bit)

HTML5 also moved (some?) formvalidation into the browser (previously had to use JavaScript)
- And also attribute "required" (i.e. must fill in input)

Nice article

Type "hidden"

```
<input type="hidden" .../>
```

Special. Not for user input

Used by page author to add hidden form parameter
- Useful at server side, some extra information
- Extracted from request like before

Common input element attributes

In the specification <input.../>

Other Form Controls

Quit a few tags

- Button, select, datalist, optgroup, option, textarea, keygen, output, progress, meter

See spec. 4.10.6-4.10.18