Markup Languages

SGML, XML, HTML & XHTML

Markup Languages

- Used to *mark up* documents via annotations in a given format
- Markup adds…
  - Structure
    - Formatting
  - Semantics
    - Classification
  - Presentation (to a lesser extent)
    - Style

Side Note: Programming vs Markup

- Markup: Annotating text
- Programming: Writing code
- HTML != Programming

It’s (Not) Just Semantics

- Writing markup is easy.
- Writing good markup is hard!
- *Semantic markup* is a matter of making the choices in hierarchy and component elements that best reflect the underlying meaning of the document

It's (Not) Just Semantics

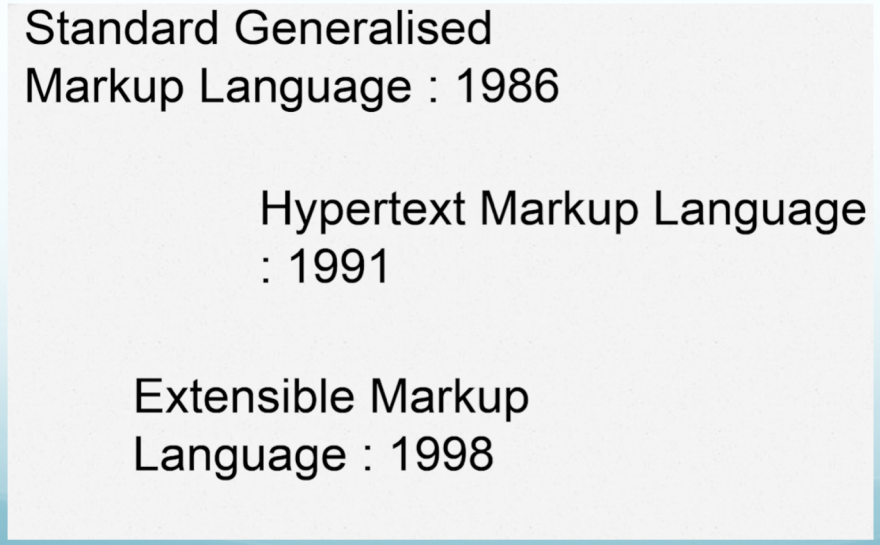
- Sometimes we *just* want the meaning of the document...
  - Screen readers interpret content differently based on the underlying markup
  - Search engines rank pages better or worse depending on the markup used in relation to the inferred meaning of the content
- Semantically correct documents are inherently easier to maintain

Markup vs Content?

- How do we separate content from markup?
  - Markup for structure, not for presentation…
- SGML is typically (but not always!) involved…

SGML?

- Standard Generalized Markup Language
- ISO standard
- Used for defining generalized markup languages
  - Yes, this is a markup language for markup languages!



Source: https://www.youtube.com/watch?v=vkGsjOin0KA

Brief History of SGML

- https://www.youtube.com/watch?v=vkGsjOin0KA&t=6

What HTML/SGML/XML have in common

```
<lg>
   <l n="1">Faith" is a fine invention</l>
   <l n="2">  When Gentlemen can  <hi
rend="hi">see</hi> </l>
   <l n="3"> But  <hi rend="underline"
>Microscopes</hi>
      are prudent </l>
    <l n="4">In an Emergency.</l>
 </lg>
```

Source: https://www.youtube.com/watch?v=vkGsjOin0KA

SGML Applications

- Fancy term for “something defined using SGML”
- Not an application in the usual sense of the word (meaning it is not a program or set of programs)
- Rather, its an application of the SGML markup rules…

Anatomy of an SGML Application

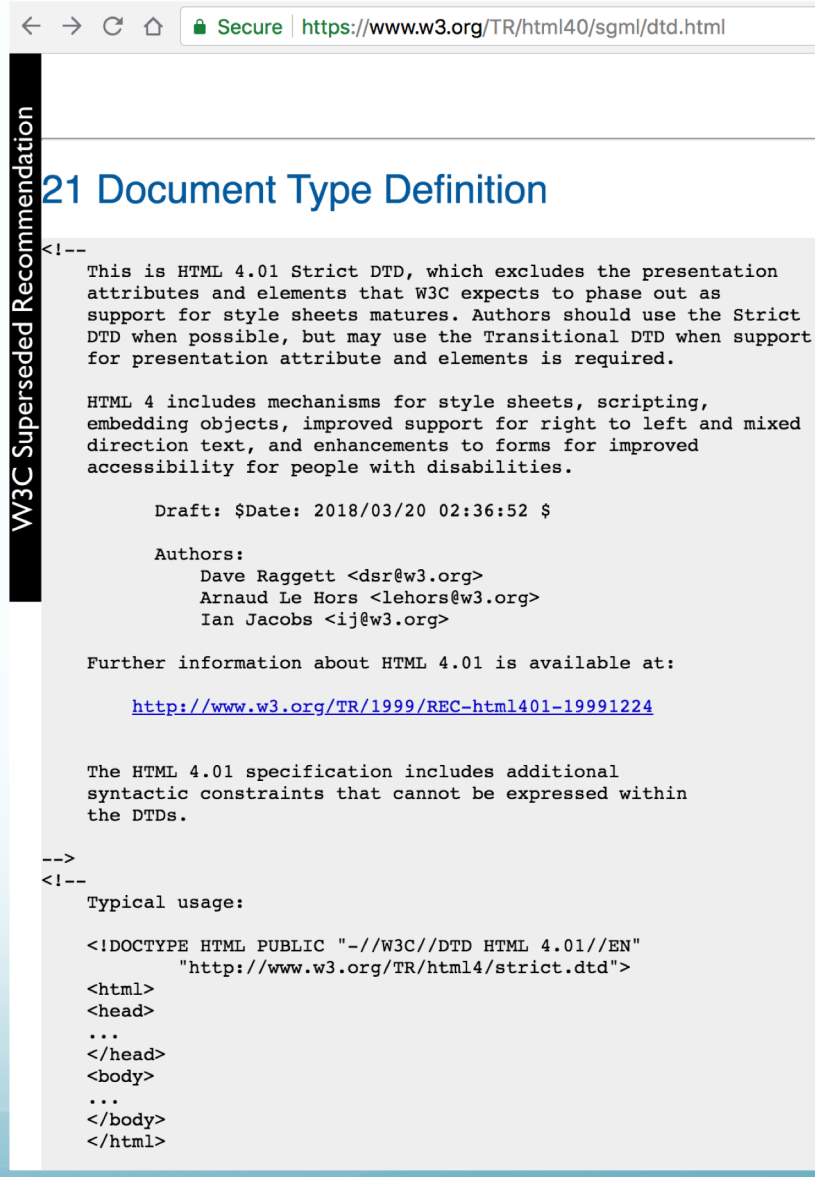
- SGML Declaration
- Document Type Definition (DTD)
- Specification
- Document instances

SGML Declarations

- Characters
- Delimiters
- Reserved Names
- Etc...

Document Type Definition (DTD)

- Defines the general syntax using those characters and delimiters
- May be more than one per declaration
- See: https://www.w3.org/TR/html40/sgml/dtd.html



Document Instances

- Refer to a DTD
- Contains…
  - Data
  - Markup structuring the data (elements: <tag></tag>)
- Most of the time, we only worry about document instances, and link to the right DTD
  - Ie http://www.w3.org/TR/html4/sgml/dtd.html
  - HTML 4 is a valid SGML Application and we use its DTD to validate our markup!
  - **Note: HTML5 does not require a the DTD declaration!**

Elements, Tags and Attributes

- Elements – everything between start- and end-tags
  - Tags – define the structure of the document instance
    - Attributes – provide additional information about an element and its content

Elements

- Consists of everything between the start and end tags
- Elements will be nested
- Example:

```
<html>
    <body>
        <p> content </p>
    </body>
</html>
```

Tags

- Format: <*tagname*>Data</*tagname*>
- *Tags can be nested:*

  <*strong*><*em*>*Hello World*</*em*></*strong*>
- Short-hand for empty tags: <*tagname /*>
  - *e.g.* <*br /*>
- Data within a tag is given a certain structural, semantic and/or presentational (<HTML5) meaning according to the tag used

Attributes

- Appear within an opening tag
- Values appear in quotes
- name=”value”
  - Space-delimited if more than one
- Ex:
  - <h1 class=“Websys” color=”red” >...</h1>

Comments

- <!– comment -->
- Do not include “--” inside of a comment, or weirdness may ensue!

Who Cares?

- A number of markup languages are based on (or have roots in) SGML
- You may have heard of a few…
  - XML
  - HTML
  - XHTML

Extensible Markup Language (XML)

What is XML?

- eXtensible Markup Language
- Multipurpose document markup language
- Human-readable and machine-readable
- Defined using SGML

XML Documents

- *Must* be well-formed
  - Valid XML syntax
- *May* be valid
  - Refers to a DTD
  - Adheres to the rules of that DTD
- Can be served as
  - application/xml
  - text/xml

Well-formedness : Tags

- Tags are case-sensitive, angle-brace delimited
- Start and end tags must match exactly, all tags must be closed
  - <tag>Hello</tag>
  - Empty tags: <tag attr="hi" />
    - is equivalent to <tag attr="hi"></tag>
- Tags and attribute names cannot include special characters

<aside> Predefined entities reference

- Used to substitute characters where they might otherwise be treated as markup
- XML defines five of them:
  - **>** >
  - **<** <
  - **&** &
  - **"** "
  - **'** ’

https://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references

Well-formedness: Hierarchy

- XML is a hierarchical (tree)
- Must be properly nested according to hierarchy

```
<!-- OK: -->
<class>
   <student>Jane Smith</student>
</class>

<!-- Wrong: -->
<student>
   <class>Jane Smith</student>
</class>
```

Well-formedness: Heirarchy

- Single root element

```
<!-- OK: -->
<classes>
   <class>ITWS-2110</class>
   <class>CSCI-4961</class>
</classes>

<!-- Wrong: -->
   <class>ITWS-2110</class>
   <class>CSCI-4961</class>
```

Validity

- Adheres to the DTD (if present)
- A DTD doesn't have to be used in order to be XML!

Anatomy of an XML document

- XML Prologue (Suggested)
- DTD (Required if validating against one)
  - XSL – XML Schema – a more sophisticated OO way to describe
- Root element with child elements

XML Prologue

- <?xml version=”1.0” encoding=”UTF-8”?>
- Defines the version of XML, the character encoding used, and potentially other information
- The XML Prologue is suggested, but not required—in some cases there are good reasons to leave it out…

Anatomy of an XML document

```
<?xml version=”1.0” encoding=”UTF-8”?>
<root>
   <branch>
      <leaf color=”red”>L1</leaf>
   </branch>
   <leaf color=”orange”>L2</leaf>
</root>
```

Why XML?

- Data structure is easy to parse/iterate through
- Human-readable and machine-readable
- Extensibility
  - Open-ended base specification
  - Format is robust when new tags are added
  - Given definition through DTDs, Schemas, Namespaces when needed (we'll go over this later)
- Documents have the meaning you give them
  - Or some previously agreed-upon meaning

XML in practice

- Really Simple Syndication (RSS)
  - Used to publish updated lists of content
- Document formats
  - OfficeXML – docx, xlsx, DocBook, DITA
- Basis of communications protocols
  - XMPP (Extensible Messaging and Presence Protocol) aka Jabber
- B2B applications/compliance
  - XBRL (eXtensible Business Reporting Language)

Hypertext Markup Language (HTML)

What is HTML?

- Defines the overall structure of a Web page
  - Structurally
  - Semantically
  - Visually (to a lesser extent, due to CSS)
- Served as
  - text/html
- Also defined in SGML

Semantics

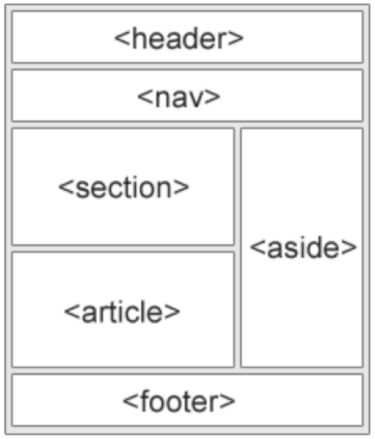
- The intended meaning of a document and its constituent parts
- Context-dependent

New Semantic Elements in HTML5

Many web sites contain HTML code like: <div id="footer" to indicate navigation, header, and footer.

HTML5 offers new semantic elements to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



See: https://www.w3schools.com/html/html5_semantic_elements.asp

(Visual) Presentation

- How the document is displayed to the user
- Browser defaults, modified by Cascading Style Sheets (CSS) included in the document
  - We'll go over this next week

The Goal

- Use semantically correct markup
  - Accessibility (screen readers for blind people)
  - Makes parsing easier (for search engine bots for the indexing purposes)
- Acceptable presentation in all supported browsers
  - Not the same as identical or pixel-perfect presentation!
- Separate presentation from semantics
  - Decouple meaning (HTML) from style (CSS)

Anatomy of an HTML document

- DTD declaration
- Nested HTML *elements*
  - Everything from start tag to end tag
  - Many may contain more elements and/or text
  - Some are empty tags that contain no content
  - Same basic rules as XML nesting

Anatomy of an HTML document

```
<!doctype >

<html>
  <head>
    <title>Title of document</title>
  </head>

  <body>
    <p>This is HTML!</p>
  </body>
</html>
```

<!DOCTYPE>

- Defines the rules used to structure the document
- DTD linked to from w3.org for HTML4
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
- **Not needed for HTML 5!**
  - **Just <!DOCTYPE html>**

Required elements

- <html>*document*...</html> - Defines the document's start/end
- <head>*head elements*...</head> - Defines document metadata and linked files (CSS, JS)
- <title>*title*...</title> - Document title
- <body>*body*...</body> - Document body

Other HTML head elements

- <meta> - Used to define metadata about the page and how it should be rendered
- <link> - Used to fetch a resource for use in rendering this document (usually CSS)
- <script> - Used to fetch a script for use with this document (usually JavaScript)

HTML Body elements

- Block-level elements
- Inline elements
- Full list available online, make sure you're using the most semantically correct element

Block-level elements

- Containers for other block-level or inline elements
- Begin on new lines when rendered

Block-level elements

-
  - Generic block-level element
  - No semantic meaning
- <p>
  - Paragraph
-   <dl>
  - Unordered list
  - Ordered list
  - Definition List
- <address>

Inline elements

- Contain inline elements or raw data
- Continue on same line while rendered

Inline elements

-
  - Generic inline element
  - No semantic meaning
- <img src="image.gif">
  - Image
-
  - Emphasis
  - Strong emphasis

About and

- No inherent semantic meaning
- Use only when no other container has a more fitting semantic meaning!

HTML Attributes

- Same as SGML attributes
- Tag-dependent
- Link
- <img src=”/images/logo.png”>
- Two very important attributes: id and class

#id

- *Unique* identifier for an element
- Only one ID of a given name may exist in a document
- Link

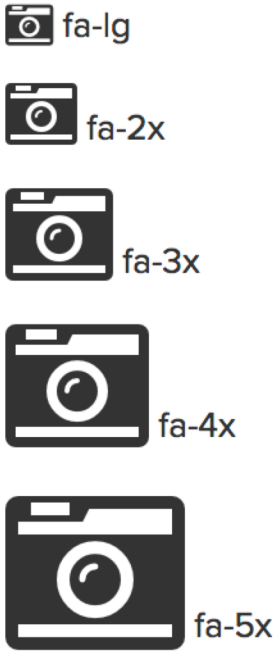
#id: Uses

- Fragment ID anchor in a document
  - Text
  - Back to top
- Target specific element with…
  - CSS
  - JavaScript
  - Other parsers

.class

- Assign one or more class (group) names to an element
  - Multiple classes are space-delimited
- May be shared by multiple elements
- Link
- Link

Example: multiple classes for the same attribute



To increase icon sizes relative to their container, use the `fa-lg` (33% increase), `fa-2x`, `fa-3x`, `fa-4x`, or `fa-5x` classes.

```
<i class="fa fa-camera-retro fa-lg"></i> fa-lg
<i class="fa fa-camera-retro fa-2x"></i> fa-2x
<i class="fa fa-camera-retro fa-3x"></i> fa-3x
<i class="fa fa-camera-retro fa-4x"></i> fa-4x
<i class="fa fa-camera-retro fa-5x"></i> fa-5x
```

If your icons are getting chopped off on top and bottom, make sure you have sufficient line-height.

See: https://fontawesome.com/v4.7.0/examples/

.class: Uses

- Target specific element with…
  - CSS
  - JavaScript
  - Other parsers

DTDs in HTML 4.01

- Three DTDs
  - HTML 4.01 Transitional
  - HTML 4.01 Frameset
  - HTML 4.01 Strict
- Seen on older sites

HTML 4.01 Transitional

- HTML 4.01 spec, plus deprecated elements are allowed. Framesets are not allowed.
- Examples:
  - - Centers text
  - <font> - Defines font style
  - <u> - Underline
  - <s> - Strikethrough
- Even if not deprecated, use of presentational markup is discouraged

Semantic vs Presentational Markup in HTML 4

- Use for text that needs emphasis
  - Not <i>
- Use for text that needs **strong** emphasis
  - Not <b>
- If you just want to apply style, you want CSS

HTML 4.01 Frameset

- As HTML 4.01, but framesets are allowed
  - Load multiple pages within frames of this one
- Don't bother. Websites that use frames are difficult...
  - to navigate
  - to bookmark
  - for search engines to process
  - for screen readers to process
- <iframe> tag (inline frames) does the same thing, doesn't require Frameset DTD
  - Used most often in advanced scripting

HTML 4.01 Strict

- Deprecated elements are not allowed.
- Reinforces best practices

Extensible Hypertext Markup Language (XHTML)

XHTML 1.0

- HTML 4.01 converted to XML
- XHTML versions of the same DTDs
  - XHTML 1.0 Transitional
  - XHTML 1.0 Frameset
  - XHTML 1.0 Strict
- Intended to be served as
  - application/xhtml+xml
  - application/xml
- Often served as text/html as older browsers – specifically IE7&8 don't support the former very well

In Short, XHTML

- All lowercase in tags/attributes
- Must have quoted attribute values
  - here
- Must be properly nested
  - here
- Must always be closed
  - OK
  - Not OK
  - Self-closing tags:
, not

In Short, XHTML

- Is well-formed XML
- Follows the XHTML DTDs
- Can therefore be extended as XML
  - Resource Description Framework in attributes (RDFa) is one example

Anatomy of an XHTML document

- Optional XML Prologue
- XHTML-compatible DTD
- <html> root element with sub-elements as described above

XHTML Gotchas

- IE7 does not support serving XHTML as anything but text/html
- Including the optional XML prologue may invoke different rendering behavior in older browsers

</class>

- Full references for HTML tags are available online. Semantic correctness matters!
- For this course, and for now, the following doctypes are acceptable unless otherwise specified:
  - HTML 4.01 Strict
  - XHTML 1.0 Strict
  - HTML5 (we'll go over this soon)
- Be prepared to defend your decisions on doctypes and semantic markup