Basic HTML & CSS

Junghoo Cho

cho@cs.ucla.edu

From Text to Structured Document

- We understand how to correctly send textual document to a
- But we see richly-structued documents
  - Complex layout, images, links, videos, …
- Q: How does the browser get the structure of a page from te

HTML (HyperText Markup Language)

- Document standard of the Web
- Specifies both the content and the structure of a Web page
- Document = Text (content) + Tags (structure)
    - Tags: special “markup symbols” enclosed in `< ... >`
    - Text: anything not enclosed in `< ... >`
    - Example

```
I <em>love</em> <strong>UCLA</strong>!!
```

HTML History (1)

- 1991: HTML(1)
  - Standardized by W3C (World Wide Web Consortium)
  - Designed by Tim-Berner’s Lee at CERNS
  - Based on SGML (Standard Generalized Markup Language)

HTML History (2)

- 1995: HTML2.0, 1997: HTML3.2, 1998: HTML4.01
- 2000: XHTML
- 2014: HTML5
    - Standardized by WHATWG (Web Hypertext Application Technology Group)
    - Standardization is both political and technical process

HTML5

- Starts with `<!DOCTYPE html>`
  - Earlier versions use different DOCTYPE
    - HTML4.01: `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
  - Remnants from SGML (Standard Generalized Markup Language)

HTML5: Bare Minimum

```html
<!DOCTYPE html>
<html>
<head><title>...</title></head>
<body>...</body>
</html>
```

HTML Element

- A single HTML entity enclosed in an opening and a closing ta
  - Example: `<p>paragraph</p>`
  - An open tag `<x>` must be followed by a matching closing tag `</x>`, e
    elements (a.k.a. empty elements)
    - Examples of void elements: `br`, `hr`, `img`, `input`

HTML Attributes

- *Attributes*: tags can have “attributes”
  - E.g., `<img src="...">`
  - Both single or double quotes can be used to enclose an attribute va
- *ID attribute*: its value should be unique in a document
  - Unique identifier of an element. Just like a “key”
- Tag and attribute names are case *insensitive*. (Lower case is recommended)

HTML Tags are for Structure

- Tags represent the document *structure*, not *formatting*
- In HTML5
  - Most formatting tags like `<font>`, `<center>`, `<u>`, `<s>`, `<tt>` have bee
    - Except `<b>` and `<i>` . They are just too popular
  - Many new “semantic elements” have been added
    - `<header>`, `<footer>`, `<nav>`, `<article>`, `<section>`, …

Formatting vs Structure Tags

- Use structure tags instead of (old) formatting tags
  - `<b>` → `<strong>`
  - `<i>` → `<em>`
  - `<tt>` → `<code>`
  - `<s>` → `<del>`
  - `<u>` → `<ins>`
- Use CSS (Cascading Style Sheet) to specify formatting
  - More on this later
- Q: Why do we want to separate structure from style?

Special Characters

- Multiple white spaces and line breaks are displayed as a sing space
- Q: How do we display multiple white spaces? line breaks? &

  &lt; &gt; &amps;
- Q: How do we include < or > in text? What about &?
- Q: Can we include comments? <!-- ... -->
- Comment tags cannot be nested

Embedding Rich Objects (1)

- Q: How do we embed links?
- A: `<a href="url">...</a>`
    - relative vs absolute URL
- Q: Can we embed another HTML page inside a page?
- A: `<iframe src="b.html" width="200" height="300">`

Embedding Rich Objects (2)

- Q: How can we embed an image, video, audio, etc?
- A:
    - Image: `<img src="...">` (void element)
    - Video: `<video src="video.mp4" controls></video>`
    - Audio: `<audio src="voice.mp3" controls></audio>`
    - Others: `<embed src="flash.swf">` (void element)

Embedding Rich Objects (3)

- Q: Where does the small icon next to title come?
- A: *Favicon* (favorite icon)
  - Small icon displayed next to title
  - Default favicon path: `/favicon.ico`
    - Can be explicitly specified via `<link rel="icon" href="path/to/favicon.p`
  - Details are complicated due to browser-specific extensions
    - Use one of many online “favicon generators”

More in HTML5

- Clearly defined logic to translate “ill-defined” documents anc errors
  - More consistent behavior among multiple browsers
- Programmable Javascript API
  - Canvas element for 2D drawing
  - Web Storage for local data storage
  - Offline Web Application for offline app support
  - Document Editing and Drag-and-Drop
  - …

XHTML

- XHTML is mostly the same as HTML, but much stricter forma
  - Tags and attributes MUST be lower case, not upper case.
  - ALL tags MUST have matching end tags. No empty elements. (e.g., <
  - Always use quotes around attribute values
- Failed to take off because it was just too strict without much developers or end users

CSS (Cascading Style Sheet)

- A set of rules for specifying document formatting and preser
- Rule = selector + declaration block
- Basic CSS demo

CSS Rules

- Rule = Selector + Declaration block
- Selector: tag, class, ID, *, …
- Declaration block:
  - Enclosed inside `{ ... }`
  - List of “property: value;” pairs

Adding CSS Rules to Page

- CSS can be specified either
  - directly inside `<style> ... </style>`
  - in a separate file via `<link rel="stylesheet" href="example.css`
- To format a particular part, add `<div>` or `<span>` tags if need

More CSS selectors (1)

```
[src] { /* has attribute named "src" */
    color: red;
}
[target="_blank"] { /* has attribute "target" with value "_blank
    color: blue;
}
div, p { /* multiple selectors separated by commas */
    background-color: grey;
}
div p { /* p is a descendent of div */
    background-color: yellow;
}
div > p { /* p is a direct child of div */
    background-color: green;
}
```

<div> .

<div> ... <se

More CSS selectors (2)

```
p.class1.class2 { /* p belonging to both class1 and class2 */
    background-color: yellow;
}
div + p { /* p is an immediately adjacent sibling of div */
    background-color: blue;
}
div ~ p { /* p is any sibling of div */
    background-color: red;
}
:hover { /* : "pseudo class" selector */
    color: #0000ff;
}
::first-letter { /* :: "pseudo element" selector */
    font-size: 2em;
}
```

<div> <

<div> .. </div> .. <p

<p> Abc </

- See CSS selectors for more detail

Inheritance

- CSS can be specified in three places:
  1. Browser default
  2. User preference
  3. Web page
- If not set in any of the three places, an element *inherits its pa properties*

Cascading Rule

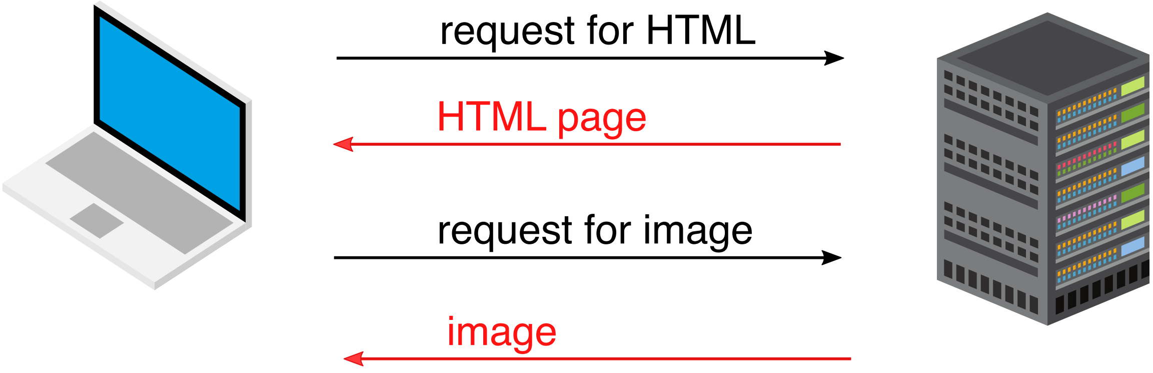
- *Cascading rule* dictates which CSS rule wins in case of conflic
  1. *Specificity*: more "specific" rule wins!
     - id > class > tag
     - more detailed specificity rule: https://www.w3.org/TR/css3-selectors/#specific
  2. Source order
     - if equal specificity, later rule wins
     - web page > user preference > browser default

HTML Validator

- Online HTML validators exist to check the standard compliar document
    - https://validator.w3.org/
- Check your HTML using an HTML validator

Retrieving Static Content

- Early Web was mainly designed to retrieve static content (HT images) from servers



Static Web Site

- Q: What should a Web site do to serve static contents for a re
- A:
  1. Retrieve the corresponding file
  2. Return it as a response
- Can be set up with
  - HTTP server (say, Apache) + filesystem
  - URL path to file mapping needed
    - Example: `DocumentRoot /var/www/html/` (Apache)
- Q: How can we specify a particular Web resource that we wa retrieve?

URL (Uniform Resource Locator)

- Unique ID of any object on the Web
  - Example: http://oak.cs.ucla.edu/classes/cs144/
- Format
  - `protocol://hostname/path?query#fragment_id`
- *Fragment identifier*: String behind # in URL
  - Points to the *HTML element* with the given id attribute
  - Example: http://a.com/a.html#g3
    - `id` attribute value must be unique in a page
- *Query*: "`name=value`" pairs
  - More on this later

What We Learned

- HTML element
- HTML tag, HTML attribute
- HTML tags are for structure
- Basic CSS rules for styling
- Static Web site
- URL standard

References

- HTML5
- HTML validator
- CSS
- URL