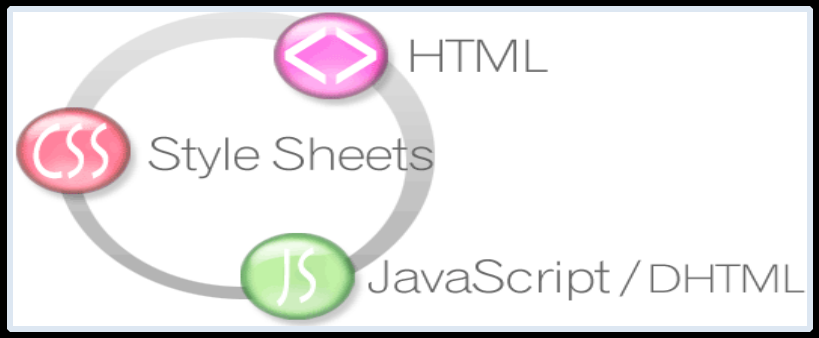


Semantic Web

How to Use HTML Elements Properly?

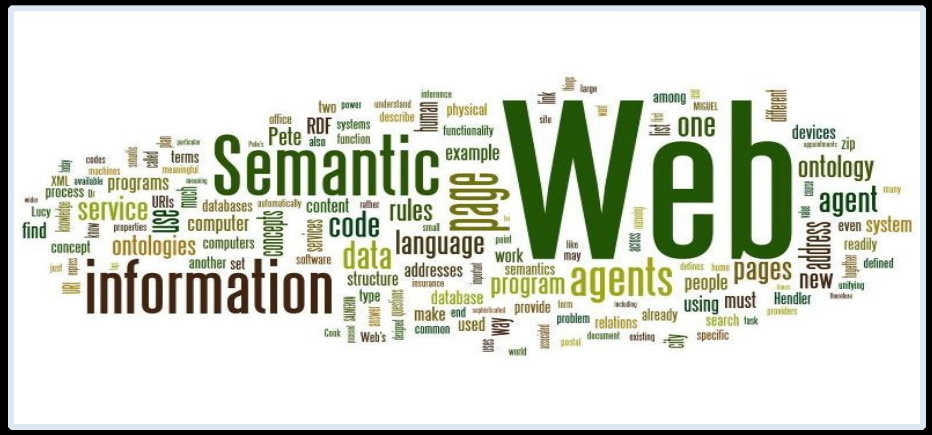


Table of Contents

Web Page

The Elements of a Web Page

- A Web page consists of:
  - HTML markup
  - CSS rules
  - JavaScript code
    - JS libraries
  - Images
  - Other resources
    - Fonts, audio, video, Flash, Silverlight, etc…

The Elements of a Web Page: HTML Markup

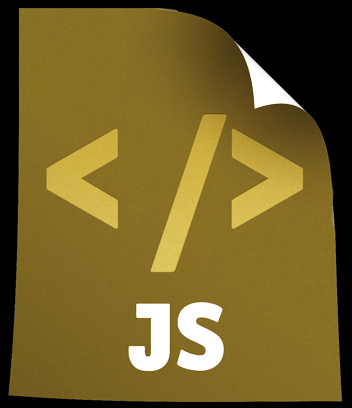
- The HTML is used to define the content of a Web page
  - Not the layout
  - Not the decorations
- HTML's role is to present the information in a meaningful manner
  - Like a paper document
  - Define headers, paragraphs, textboxes, etc…
  - Not define size, color and/or positioning

The Elements of a Web Page: CSS Rules

- Cascading Style Sheets (CSS) is the way to make a Web page look pretty
  - Define styling rules
    - Fonts, colors, positioning, etc.
  - Define the layout of the elements
  - Define the presentation
- The CSS files are attached to a web page and the browser applies these styles to elements

The Elements of a Web Page: JavaScript Code

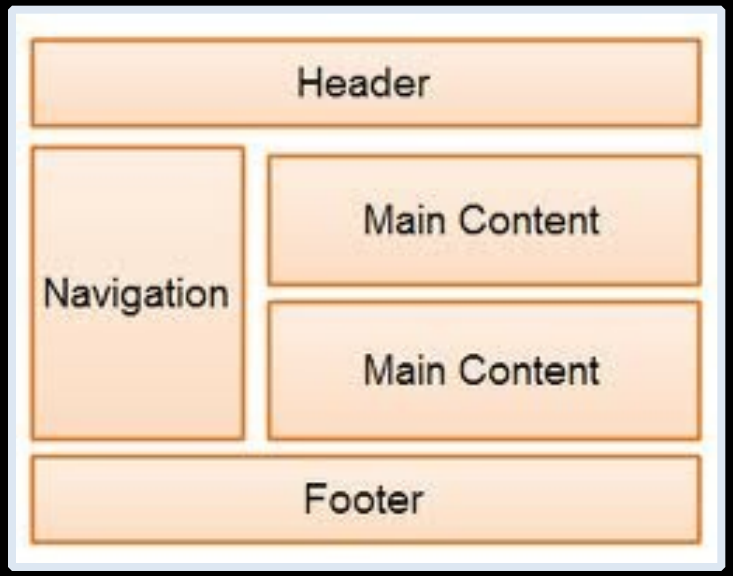
- JavaScript is the programming language for the Web
  - Makes the Web pages dynamic
  - Dynamically adding / removing HTML elements, applying styles, etc.
  - Modern JavaScript UI libraries provide UI components like dialog boxes, grids, tabs, etc.
- Like CSS the JavaScript files are attached to a web page



The Elements of a Web Page: Other Resources

- Other resources are needed for a Web page to run properly
  - Images, fonts (glyph icons), audio, video files
  - Flash / Silverlight / ActiveX objects

The Semantic HTML



Semantic HTML

- Semantic HTML is:
  - The use of HTML markup to reinforce the semantics of the information in Web pages
    - Make the content understandable for computers
  - Rather than merely to define its presentation
  - A kind of metadata about the HTML content
- Semantic HTML is processed by regular Web browsers and other user agents
  - CSS is used to suggest its presentation to human users

Why Use Semantic HTML?

- Semantic HTML is:
  - Easier to read by developers, parsers, bots, machines, AIs
  - A way to show the search engines the correct content

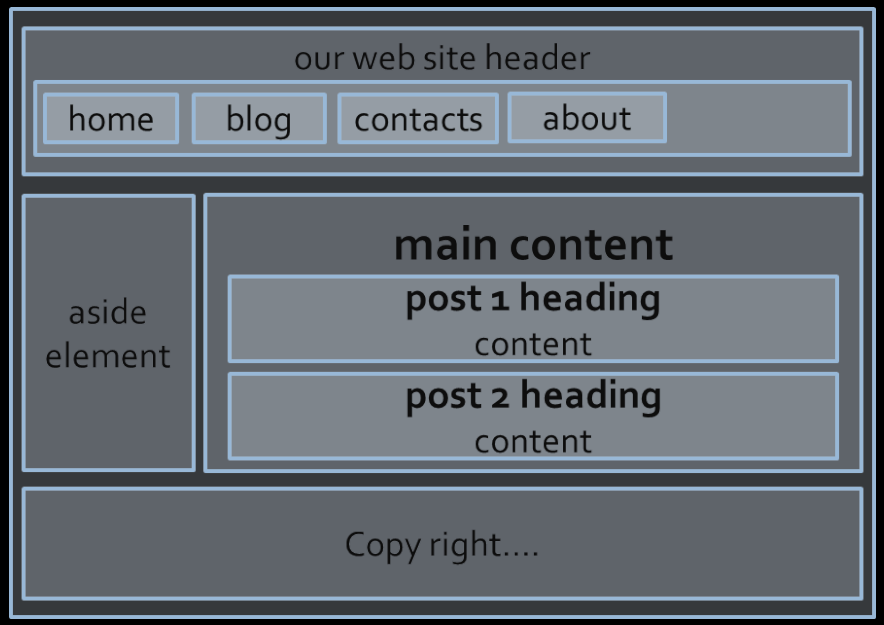
How To Write Semantic HTML?

- Just follow some guidelines when creating a Web site
  - Use HTML5 semantic tags
    - `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`
  - Use Headings when you need to structure the content into sub-headings
    - In increasing order, staring with `<h1>`
  - Do not use empty tags
    - Like a clearing `<div>`

HTML5 Semantic Tags

HTML5 Semantic Tags

- HTML5 introduces semantic structure tags
  - Imagine the following site:



  - This is a common Web page structure
    - Used in 90% of the web sites

HTML5 Semantic Tags (2)

- This can be created using all kind of HTML elements
  - ** **, ****, even **<p>**
  - Browsers will render invalid / wrong / pseudo valid HTML
- The correct way: use the HTML 5 semantic tags:


```
<header> … </header>
<nav> … </nav>
<main> … </main>
<article> … </article>
<section> … </section>
<aside> … </aside>
<footer> … </footer>
```

HTML5 Structure Tags

- **`<main>`**
  - Specifies the main content of a document (info)
  - There must not be more than one <main> element in a document
- **`<header>`**
  - Site header or section header or article header
  - Could include navigation (<nav>)
- **`<footer>`**
  - Site footer (sometime can be a section footer)
  - Providing author, copyright data, etc.

HTML5 Structure Tags (2)

- `<nav>`
  - Defines a set of navigation links.
  - E.g. site navigation (usually in the header)
- `<aside>`
  - Content slightly related to primary content
  - E.g. sidebar (usually on the left or on the right)
- `<section>`
  - Grouping of content usually with a heading, similar to chapters
  - Site section (e.g. news, comments, links, …)

HTML5 Content Tags

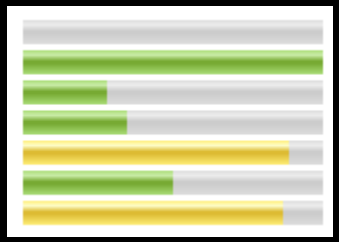
- **`<article>`**
  - Independent content such as blog post or an article (e.g. news item)
- **`<details>` + `<summary>`**
  - Specifies additional details that the user can view or hide on demand (accordion-like widget)
- **`<time>`**
  - Specifies date / time (for a post / article / news)
- **`<mark>`**
  - Defines marked/highlighted text

HTML5 Content Tags (2)

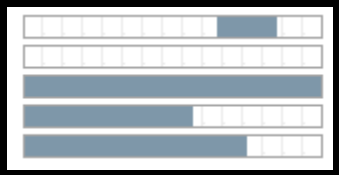
- **`<figure>`**
  - Grouping stand-alone content (video or image)
  - Figure (a figure, e.g. inside an article)
- **`<figcaption>`**
  - A caption of a figure (inside the **`<figure>`** tag)
- **`<video>`** (info)
  - Video element (uses the built-in player)
- **`<audio>`** (info)
  - A standard for playing audio files (built-in player)

HTML5 Content Tags (3)

- **`<dialog>`**
  - Defines a dialog box or window



- **`<meter>` / `<progress>`**
  - Defines a scalar measurement within a known range (a gauge) or task progress



- **`<output>`**
  - Defines the result of a calculation
- **`<wbr>`**
  - Defines a possible line-break

More info

Other Semantics

Headings, ems, strongs

Other Semantics

- Headings
  - Always use headings (`<h1>` – `<h6>`) when you need a heading or title
    - Like in a MS Word document
    - Google uses it to mark important content
- Strong `<strong>` vs. Bold `<b>`
  - `<b>` does not mean anything
    - It just makes the text bolder
  - `<strong>` marks the text is "stronger" than the other, surrounding text

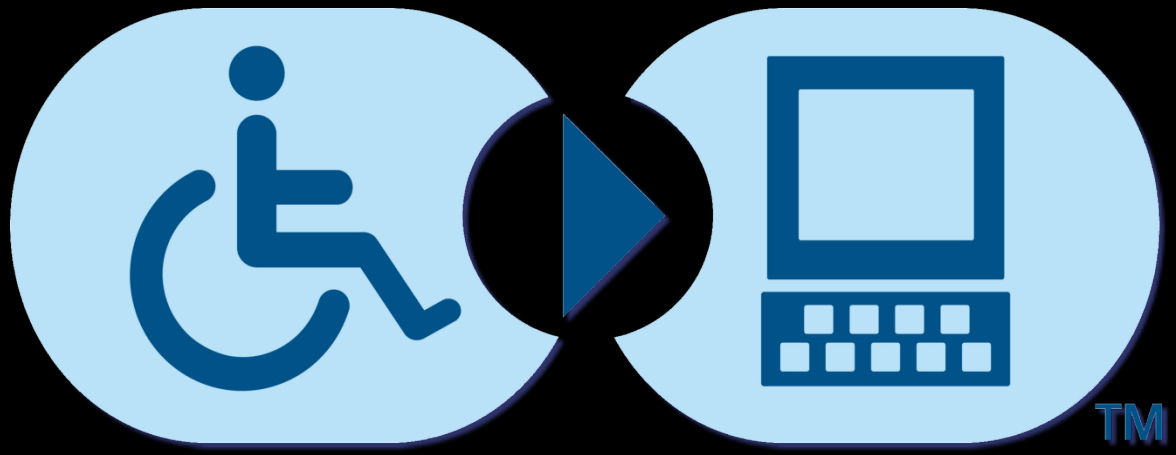
Other Semantics (2)

- Emphasis `<em>` vs. Italic `<i>`
  - Emphasis does not always mean, that the code should be italic
    - It could be bolder, italic and underlined
  - The styles for the emphasis text should be set with CSS
    - Not by HTML
- Old browsers (like IE6)?
  - Use **Modernizr** or **HTML5shiv**

Accessibility

"A person's a person,
no matter how small"

Dr. Seuss



Accessibility

- Craft content minding disabled users
  - Blind - include text equivalents of images, use labels in forms
  - Colorblind - do not convey information using color only
  - Visually impaired - avoid small font sizes
  - Epileptic - avoid flashing content (3Hz or more)
  - Physical disabilities - avoid functionality that relies only on the mouse or keyboard

Accessibility (2)

- Why implement accessibility?
  - Some accessibility features are mandatory for government sites in some countries (US, NL, SW)
  - "Everyone gets visited by a very important blind user, named Google"
  - Some SEO and accessibility considerations overlap

Accessibility (3)

- Standards
  - Web Content Accessibility Guidelines (WCAG) - http://www.w3.org/WAI/intro/wcag
  - Section 508 - http://www.section508.gov
- Tools
  - Will never replace manual testing, but may help
  - WAVE - http://wave.webaim.org/

Search Engine Optimization

Getting ahead in search engines

Search Engine Optimization

- Search engines use so-called "crawlers" to get the content of the page and index it
- The crawlers weigh the data on the page
  - `<title>`, page URL and headings have great weight
  - Links from highly valued pages to your page increase its value (Google Page Rank)
  - Add alt text to images
  - Use relevant keywords in the content and `<meta>` tags
- No SEO technique will replace good content

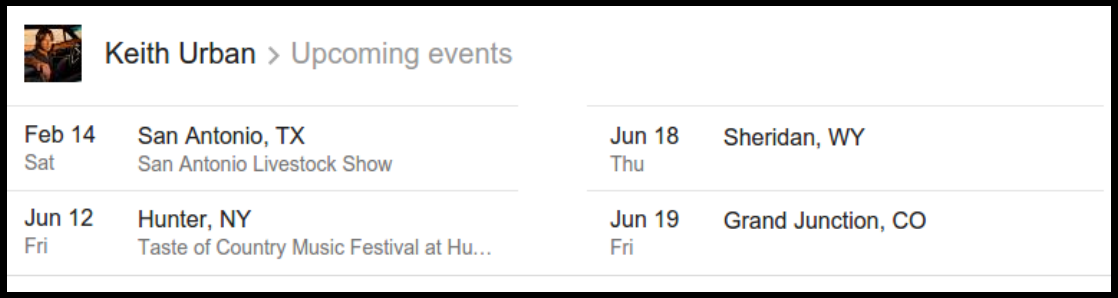
Structured Data Markup

Annotate your content so machines can understand it

Structured Data Markup

- A standard way to annotate your content so machines can understand it
- Google (and other search engines) can
  - use that data to index your content better
  - present it more prominently in search results

  - Provide answers from the Knowledge Graph



Structured Data Markup (2)

- Three alternative formats:
  - Microdata and RDFa
    - Define new HTML attributes
      - More info: http://schema.org/docs/gs.html#microdata_how
  - JSON-LD
    - Newest and simplest markup format
    - Embed a block of JSON data inside a script tag
      - Specification: http://www.w3.org/TR/json-ld/
      - Examples: http://json-ld.org/playground/

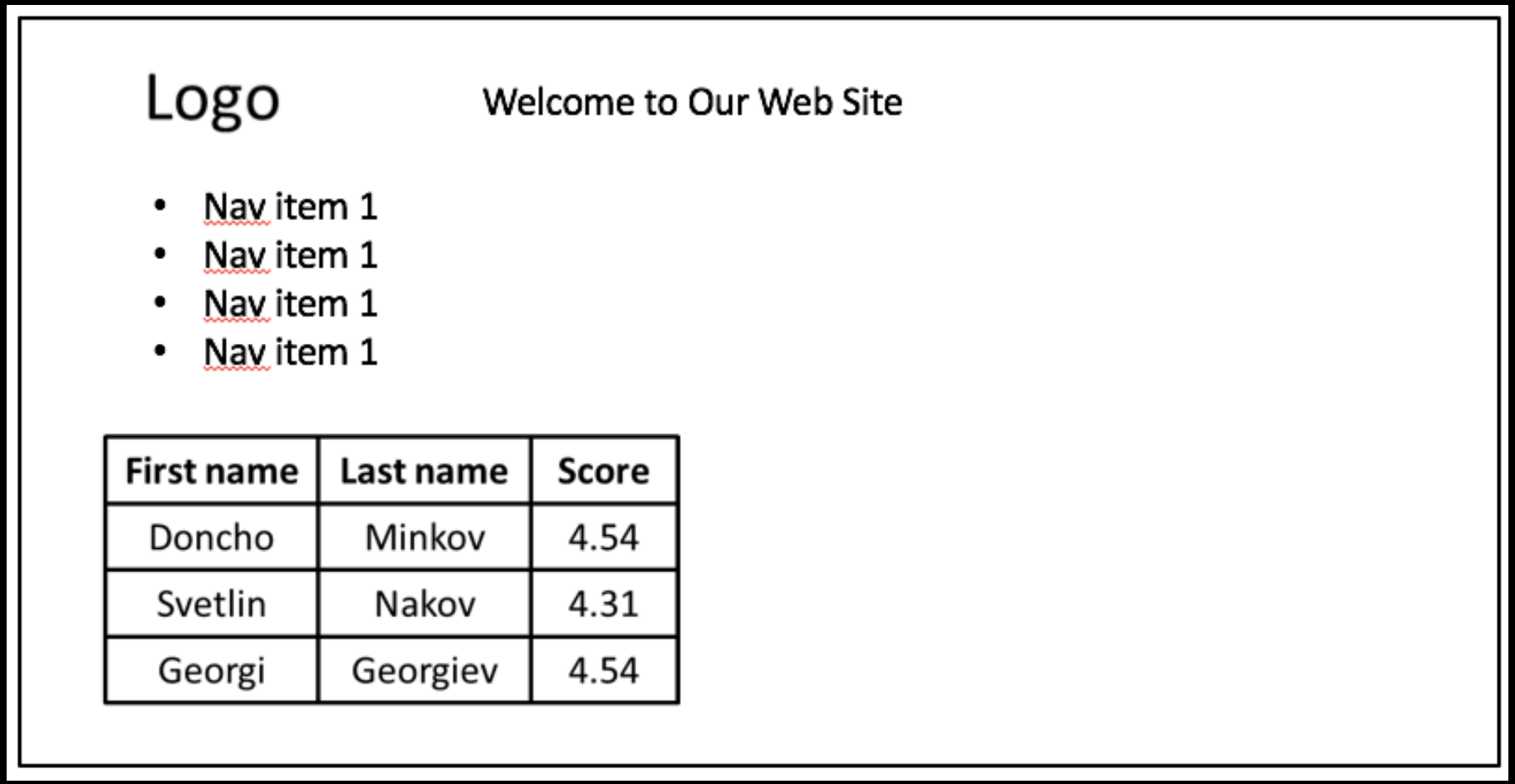
Questions?

Homework

1. Refactor the refactoring-homework.html web page and make its HTML semantic
2. Create a web page using semantic HTML by the design in homework-design-1.png
3. Create a web page using semantic HTML by the design in homework-design-2.png
   - *Note: do not try to make the same styles in Exercise 2 and Exercise 3
     - Implement just the content with its semantics
   - Use some kind of approach to support old (non-HTML5-compatible) Web browsers like IE6-IE8

Homework

1. homework-design-1.png



Homework

2. homework-design-2.png