Comp 125 - Visual Information Processing

Spring Semester 2019 - Week 7 - Wednesday

Dr Nick Hayward

HTML - markup for tables

- great example of poor usage of HTML markup is `<table>` element
- main issue is use of nested tables and spacer elements, images...
- if used correctly in structured markup
  - *tables can be very useful structure*
  - *impart a sense of semantic organisation to data*
  - *creating various interpretive information*
- what is a table for?
  - *structuring data*
  - *data to impart curated information...*

HTML - markup for tables - example I

- simple table example - columns and rows for *presentation* purposes

```
<p>Travel Destinations</p>
<!-- basic table structure - minimal - rows and columns -->
<table>
  <tr>
    <td><b>Place</b></td>
    <td><b>Country</b></td>
    <td><b>Sights</b></td>
  </tr>
  <tr>
    <td>Nice</td>
    <td>France</td>
    <td>Cours Saleya</td>
  </tr>
  <tr>
    <td>Cannes</td>
    <td>France</td>
    <td>La Croisette</td>
  </tr>
  <tr>
    <td>Antibes</td>
    <td>France</td>
    <td>Picasso museum</td>
  </tr>
</table>
```

example

- example - basic table for presentation

HTML - markup for tables - example 2

- add semantic structure & elements to table caption - replace <p> with correct <caption> usage for a table...

```
<!-- basic table structure - minimal - add a caption -->
<table>
  <caption>Travel Destinations</caption>
  ...
```

- modern browsers style <caption> by default
  - *centred above the table*
- modify styling as required

example

- example - basic table caption

HTML - markup for tables - example 3

- add a summary attribute to the table

```
<!-- basic table structure - minimal - add summary attribute -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  ...
```

- add further meaning and structure to the table
  - *use of a summary attribute on the table element*
- processed by the browsers for semantics
- particularly useful for non-visual browsers

example

- example - basic table with summary

HTML - markup for tables - example 4

- add correct headers <th> to the table

```
<!-- basic table structure - minimal - add table headers -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <tr>
    <th>Place</th>
    <th>Country</th>
    <th>Sights</th>
  </tr>
  ...
```

Benefits include:

- remove need for presentational markup, bold elements
- visual browsers process structural and presentation qualities of headings
- such heading elements can also be useful for non-visual browsers

example

- example - basic table with headers

HTML - markup for tables - example 5

- table markup and accessibility markup...

```
<!-- basic table structure - accessibility - add ids and headers -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <tr>
    <th id="place">Place</th>
    <th id="country">Country</th>
    <th id="sights">Sights</th>
  </tr>
  <tr>
    <td headers="place">Nice</td>
    <td headers="country">France</td>
    <td headers="sights">Cours Saleya</td>
  </tr>
  ...
```

- creating a known relationship between the table's header, and its data
- a screen reader, for example, may read this table as follows,
  - *Place: Nice, Country: France, Sights: Cours Saleya*
- established a pattern to the output information for non-visual devices...

example

- example - basic table with accessibility

HTML - markup for tables - example 6

- add extra semantic markup for thead, tfoot, tbody...

```
<!-- basic table structure - add head, foot, body -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <thead>
    <tr>
      ...
    </tr>
  </thead>
  <tfoot>
    <tr>
      ...
    </tr>
  </tfoot>
  <tbody>
    <tr>
      ...
    </tr>
  </tbody>
</table>
```

- head and foot elements customarily go above the table body
  - *allows modern browsers, readers, &c. to load that data first*
  - *then render the main table content*

Benefits include:

- better underlying structure to data
- greater ease for styling a table due to clear divisions in data and information
- structural and presentational markup now working together correctly...

example

- example - basic table with head, foot, body

HTML - presentational vs structural

- consider *presentational* vs *structural*
  - *e.g.usage of quotations in markup*
  - *similar consideration to headings...*
- need to convey meaning and structure
- rather than a mere presentational instruction
- consider HTML's phrase elements
  - *e.g.* `<cite>`, `<code>`, `<abbr>`
- each phrase element imparts a sense of underlying meaning
  - *structure & then presentation...*

HTML - minimising markup

- noticeable benefit to creating sites with valid markup
  - *separation of structural from presentational*
  - *general reduction in required markup*
- simply conforming to the W3C's specifications
  - *does not inherently guarantee less code for your project*
  - *possible to include many unnecessary elements & retain valid markup*
  - *markup may still be valid*
- project issues may include:
  - *lack of efficiency*
  - *extraneous markup and code*
- to help minimise markup
  - *consider classes added to markup*
    - are there too many? are they all necessary? &c.
    - avoid class usage for unique reference
  - *avoid `<div>` usage for explicit block-level elements*

HTML & JS - Random Greeting Generator - variant I

---

example solution - project structure

- sample project structure
  - *project specific directory, e.g.* `myproject`
  - *project subdirectory for* `assets`
  - *assets include JS -* `greeting.js`

```
./
|-- assets
|   |-- js
|       |__ greeting.js
|-- index.html
```

HTML & JS - Random Greeting Generator - variant 1

example solution - HTML

- start with basic HTML template
  - *including metadata in the <head> element*
    - reference to JS script file at foot of <body> element

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!-- title -->
    <title>Random Greeting Generator</title>
  </head>
  <body>

        <!-- script files -->
        <script src="./assets/js/greeting.js"></script>
  </body>
</html>
```

HTML & JS - Random Greeting Generator - variant 1

---

example solution - HTML

- add application header and heading to top of <body>

```
<header>
  <h2>Create a random greeting...</h2>
</header>
```

- add <main> element to <body>

```
<main>
  ...
</main>
```

- add <footer> to end of <body>

```
<footer>
  ...
</footer>
```

HTML & JS - Random Greeting Generator - variant 1

example solution - HTML

- add first `<section>` with form, input, and button to `<main>`

```
<!-- elements for getting user input -->
<section id="generator">
  <header>
    <h3>Enter a name for the greeting</h3>
  </header>
  <form>
    <!-- player input for guessing a letter -->
    <input name="customName" placeholder="enter a name" type="text" autofocus id=
    <!-- send guess letter -->
    <button type="button" id="greetingBtn">create greeting</button>
  </form>
</section>
```

HTML & JS - Random Greeting Generator - variant 1

example solution - HTML

- add second <section> with <p> for output content to <main>

```
<!-- elements for outputting generated random greeting -->
<section id="output">
    <header>
        <h3>Greeting...</h3>
    </header>
    <p id="greeting"></p>
</section>
```

HTML & JS - Random Greeting Generator - variant 1

example solution - full HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!-- title -->
    <title>Random Greeting Generator</title>
  </head>
  <body>
    <header>
      <h2>Create a random greeting...</h2>
    </header>
    <main>
      <!-- elements for getting user input -->
      <section id="generator">
        <header>
          <h3>Enter a name for the greeting</h3>
        </header>
        <form>
          <!-- player input for guessing a letter -->
          <input name="customName" placeholder="enter a name" type="text" autofoc
          <!-- send guess letter -->
          <button type="button" id="greetingBtn">create greeting</button>
        </form>
      </section>
      <!-- elements for outputting generated random greeting -->
      <section id="output">
        <header>
          <h3>Greeting...</h3>
        </header>
        <p id="greeting"></p>
      </section>
    </main>
    <footer>
      <p>developed by ancientlives</p>
    </footer>
    <!-- script files -->
    <script src="./assets/js/greeting.js"></script>
  </body>
</html>
```