Comp 125 - Visual Information Processing

Spring Semester 2018 - week 11 - monday

Dr Nick Hayward

footer

- `<footer>`
  - *usually contains information about its containing element*

- example 1 - in a footer for an article
  - *might use this element to define and record*
  - *author of the article*
  - *publication date*
  - *suitable tags or metadata*
  - *associated documents...*

- example 2 - a footer simply placed at the **foot** of a page
  - *record copyright information*
  - *contextual links*
  - *contact information*
  - *small logos...*

- example 2 considered standard usage for `<footer>`
  - *continues from HTML4 and earlier generic usage...*

Image - HTML5 page structure - part 1
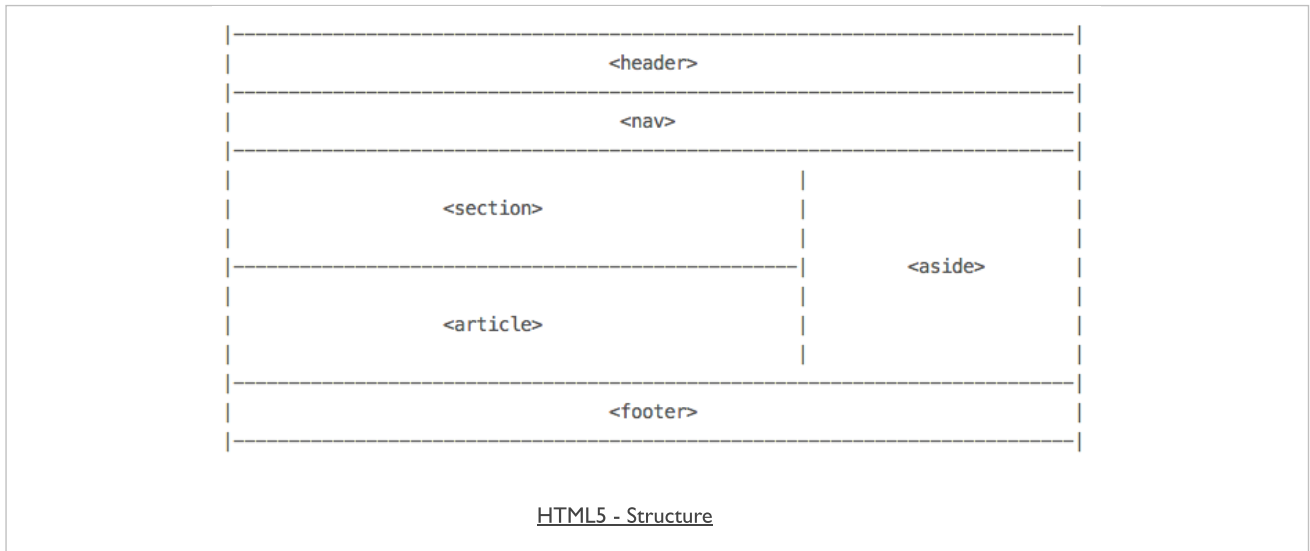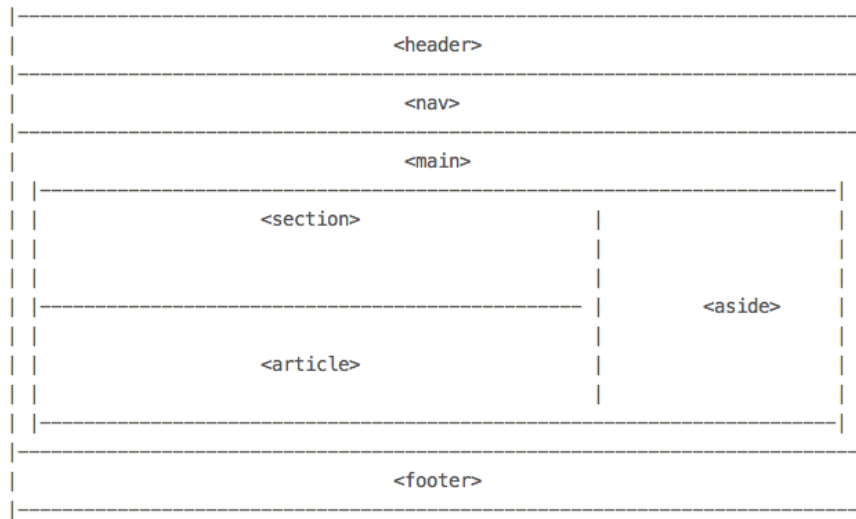
semantic elements

```
|----------------------------------------------------------------|
|                          <header>                              |
|----------------------------------------------------------------|
|                           <nav>                                |
|----------------------------------------------------------------|
|                                      |                         |
|              <section>               |                         |
|                                      |                         |
|--------------------------------------|       <aside>           |
|                                      |                         |
|              <article>               |                         |
|                                      |                         |
|----------------------------------------------------------------|
|                          <footer>                              |
|----------------------------------------------------------------|
```

HTML5 - Structure

Image - HTML5 page structure - part 2

semantic elements

```
|————————————————————————————————————————————————————|
|                        <header>                     |
|————————————————————————————————————————————————————|
|                         <nav>                       |
|————————————————————————————————————————————————————|
|                        <main>                       |
| |————————————————————————————————————————————| |    |
| |                 <section>          |        | |    |
| |                                    |        | |    |
| |                                    |        | |    |
| |————————————————————————————————————| <aside>| |    |
| |                                    |        | |    |
| |                 <article>          |        | |    |
| |                                    |        | |    |
| |————————————————————————————————————————————| |    |
|————————————————————————————————————————————————————|
|                        <footer>                     |
|————————————————————————————————————————————————————|

                   HTML5 - Structure
```

- not included `<html>` and `<body>` tags in diagrams
  - *required for all HTML documents*

- divided the page into four logical, semantic divisions
  - *header*
  - *nav*
  - *main*
  - *footer*

- we could also add a sidebar etc for further division of content

HTML5 - extra elements

intro

- many other interesting and useful new HTML5 elements
  - *in addition to semantic elements*

- some struggle for browser compatibility

- useful new elements such as
  - *graphics and media*

- HTML5 APIs introduced as well, including
  - *App Cache*
  - *Drag/Drop*
  - *Geolocation*
  - *Local Storage*
  - *...*

- again, check browser support and compatibility

Browser check

- Can I Use_____?
  - *e.g. Can I Use Drag and Drop?*

video

`<video>` element

- until HTML5, video playback reliant on plugins
  - *e.g. Adobe Flash*

- embed video using element tag `<video>`

- add attributes for
  - *height, width, controls...*

- not all web browsers support all video codecs

- option to specify multiple video sources

- best supported codecs include
  - *MP4 (or H.264), WebM, OGG...*

- good general support for `<video>` element

- check browser support for `<video>` element
  - *Can I use_____video?*

video example

`<video>` - a quick example might be as follows,

```
<video width="300" height="240" controls>
  <source src="media/video/movie.mp4" type="video/mp4">
  <source src="media/video/movie.webm" type="video/webm">
  Your browser does not support the video tag.
</video>
```

- Demo - HTML5 Video playback

audio

`<audio>` element

- HTML5 also supports standardised element for embedded audio
- supported codecs for `<audio>` playback include
  - *MP3 and mp4*
  - *WAV*
  - *OGG Vorbis*
  - *3GP*
  - *m4a*
- again, check browser support and compatibility
  - *Can I use_____audio?*
- fun test of codecs
  - *HTML5 Audio*

audio example

`<audio>` - a quick example might be as follows,

```
<audio controls>
  <source src="media/audio/audio.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

- Demo - HTML5 Audio playback

HTML5 - Extra elements - graphics - part 1

canvas

- graphics elements are particularly fun to use

- use them to create interesting, useful graphics renderings

- in effect, we can draw on the page

- `<canvas>` element acts as a placeholder for graphics
  - *allows us to draw with JavaScript*

- draw lines, circles, text, add gradients...
  - *e.g. draw a rectangle on the canvas*

canvas example

<canvas> will be created as follows,

```
<canvas id="canvas1" width="200" height="100">
  Your browser does not support the canvas element.
</canvas>
```

then use JavaScript to add a drawing to the canvas

```
<script type="text/javascript">
var can1 = document.getElementById("canvas1");
var context1 = can1.getContext("2d");
context1.fillStyle="#000000";
context1.fillRect(0,0,150,75);
</script>
```

Result is a rendered black rectangle on our web page.

- Demo - HTML5 Canvas - Rectangle

HTML5 - Extra elements - graphics - part 3

canvas example

A square can be created as follows,

```
<script type="text/javascript">
function draw() {
/*black square*/
var can1 = document.getElementById("canvas1");
var context1 = can1.getContext("2d");
context1.fillStyle="#000000";
context1.fillRect(0,0,50,50);
}
</script>
```

Again, we end up with the following rendered shape on our canvas.

- Demo - HTML5 Canvas - Square

HTML5 - Extra elements - graphics - part 4

canvas examples

- modify drawing for many different shapes and patterns
  - *simple lines, circles, gradients, images...*

  1. shows different rendered shapes on a canvas.

- Demo - HTML5 Canvas - Assorted Shapes

  2. little retro games

- Demo - HTML5 Canvas - Retro Breakout Game

References

- W3Schools - HTML5
- media elements
- canvas element