

## **Comp 324/424 - Client-side Web Design**

---

Fall Semester 2018 - Week 3

Dr Nick Hayward

## HTML5 - intro

---

- finally became a standard in October 2014
- introduces many new features to HTML standard
- additional features include, e.g.
  - *new canvas element for drawing*
  - *video and audio support*
  - *support for local offline storage*
  - *content specific elements*
  - *including article, footer, header, nav, section*
  - *form controls such as*
  - *calendar, date, time, email, url, search*
- new input type attribute values
  - *assigned to provide better input control*
- Check browser compatibility using [HTML5 Test](#)

# HTML5 - basic template

---

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

  </body>
</html>
```

# HTML5 - Elements - part I

---

- often known simply as **tags**
- elements allow us to add a form of metadata to our HTML page
- for example, we might add

```
<!-- a paragraph element -->  
<p>add some paragraph content...</p>  
<!-- a first heading element -->  
<h1>our first heading</h1>
```

- this metadata used to apply structure to a page's content

## HTML5 - Elements - part 2

---

- we can now add additional structure to our basic template

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!-- title for the web page appears in the window, tab heading... -->
    <title>Demo 1</title>
  </head>
  <body>
    <h1>Our first web page</h1>
    <p>
      As we build our web apps, more elements and content will be added...
    </p>
  </body>
</html>
```

- Demo - Our first web page

## HTML5 - Comments

---

- comments are simple and easy to add to HTML
- add to HTML code as follows,

```
<!-- a comment in html -->
```

- comment not explicitly visible to the user in the rendered page
- comment appears in the code for reference...

## Image - HTML5 sample rendering I

---

### Our first web page

As we build our web apps, more elements and content will be added to this template.

HTML - sample rendering of demo I

Source - Demo I

# HTML5 - semantic elements - part I

---

- new semantic elements added for HTML5
- known as **block-level** elements
  - *includes the following elements,*

```
<article>
<aside>
<details>
<figure>
<figcaption>
<footer>
<header>
<main>
<nav>
<section>
```

- better structure underlying documents
  - *add clear semantic divisions*



## HTML5 - semantic elements - part 2

---

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!-- our second demo with lots of new elements -->
    <title>Demo 2</title>
  </head>
  <body>
    <header>
      <h1>Our first web page</h1>
    </header>
    <!-- primary navigation elements, links... -->
    <nav>Option 1</nav>
    <!-- main content -->
    <main>
      <section>
        <p>
          As we build our web apps, more elements and content will be added...
        </p>
        <figure>
          
        </figure>
      </section>
      <aside>
        Temple at Philae in Egypt is Ptolemaic era of Egyptian history...
      </aside>
    </main>
    <footer>
      foot of the page...
    </footer>
  </body>
</html>
```

- Demo - New elements added

## Image - html5 sample rendering 2

---

### Our first web page

Option 1

As we build our web apps, more elements and content will be added to this template.



Temple at Philae in Egypt is Ptolemaic era of Egyptian history. Similar temples include Edfu...  
foot of the page...

HTML - sample rendering of demo 2

Source - Demo - New elements added

## HTML5 - semantic elements - part 3

---

- element tag `article` not used in previous demo
- `article` and `section` tag can both cause some confusion
- not as widely used as expected
- `div` element still widely seen in development
- HTML5 is supposed to have relegated `div`
  - *sectioning element of last resort...*
- `article` and `section`
  - *good analogy with a standard newspaper*
  - *different sections such as headlines, politics, health...*
  - *each section will also contain articles*
- HTML specification also states that an `article` element

*represents a self-contained composition in a document, page, application, or site and that is, in principle, independently distributable or reusable, e.g. in syndication.*

## HTML5 - semantic elements and structure - intro

---

- perceived issue or concern with HTML5 semantic elements
  - *how and when to add them to our document*
  - *where and when do we add them to our page?*
- non-semantic elements often considered simpler to apply
  - *generalised application and context for usage*

# HTML5 - semantic elements and structure

---

## header and nav

- `<header>`
  - *used to collect and contain introductory content*
  - *semantically appropriate for the head or top of a page*
  - *technically feasible and acceptable to include multiple `<header>` elements*
  - *e.g. `<header>` within main content, sidebar content, an article, a section...*
- `<nav>`
  - *short for navigation*
  - *stores and defines a set of links for internal or external navigation*
  - *not meant to define all page navigation links*
  - *often considered suitable for primary site links*
  - *additional links can be placed in*
  - *sidebar, footer, main content...*
  - *no need to consider a `<nav>` element for these links...*

# HTML5 - Semantic elements and structure

---

## main

- this element tag defines our **main** content
- traditionally the central content area of our page or document
- HTML4 often used a `<div>` element
  - *plus a class or id to define central content*
  - e.g.

```
<!-- e.g. HTML4 main content -->
<div id="main">
  ...
</div>
```

- HTML5 semantically defines and marks content as `<main>`
- `<main>` should not include any page features such as
  - *nav links, headers etc, that are repeated across multiple pages*
- cannot add multiple `<main>` elements to a single page
- must not be structured as a child element to
  - `<article>`, `<aside>`, `<footer>`, `<header>`, or `<nav>`

# HTML5 - Semantic elements and structure

---

## section, article, aside - part I

- `<section>`
  - defines a section of a page or document
  - W3C Documentation defines as follows,

*a section is a thematic grouping of content. The theme of each section should be identified, typically by including a heading as a child of the section element.*

- a site can be sub-divided into multiple `<section>` groupings
  - e.g. as we might consider a chapter or section break in a book...
- `<article>`
  - suitable for organising and containing independent content
  - include multiple `<article>` elements within a page
  - use to establish logical, individual groups of content
  - again, newspaper analogy is useful to remember
  - e.g. a blog post, story, news report...might be a useful article
  - key to using this element is often whether content can be used in isolation
- `<aside>`
  - used to define some content aside from containing parent content
  - normally used to help define or relate material to surrounding content
  - effectively acts as supporting, contextual material

# HTML5 - Semantic elements and structure

---

## section, article, aside - part 2

- MDN Documentation suggests,

*if it makes sense to separately syndicate the content of a `<section>` element, use an `<article>` element instead*

and

*do not use the `<section>` element as a generic container; this is what `<div>` is for, especially when the sectioning is only for styling purposes. A rule of thumb is that a section should logically appear in the outline of a document.*



# HTML5 - Semantic elements and structure

---

## figure, figcaption

- `<figure>` & `<figcaption>`
  - as with *print media*, we can logically group image and caption
  - `<figure>` acts as parent for image grouping
  - child elements include
  - `<img>` and `<figcaption>`

```
<figure>

<figcaption>Ptolemaic temple at Philae, Egypt</figcaption>
</figure>
```

- updated demo with figure grouping
  - Demo - Semantic structuring

# HTML5 - Semantic elements and structure

---

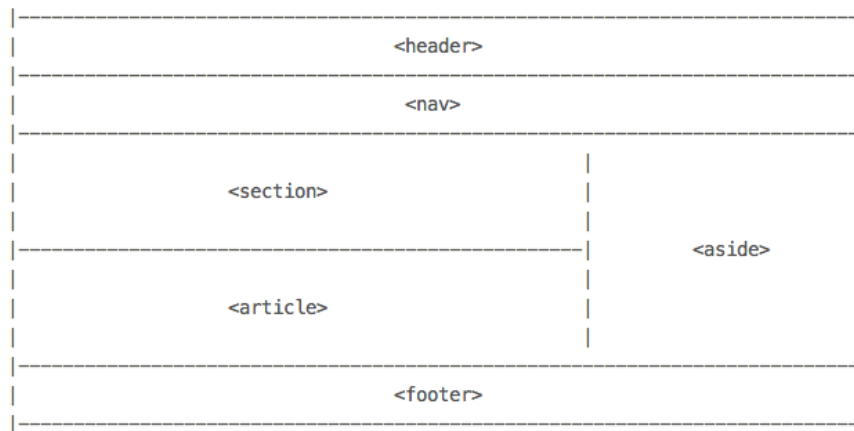
## footer

- `<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 I

---

## semantic elements

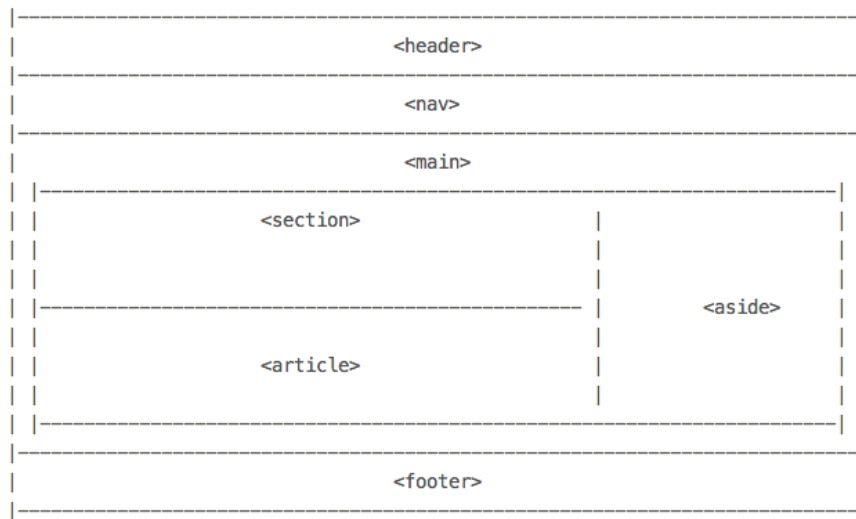


HTML5 - Structure

## Image - HTML5 page structure - part 2

---

### semantic elements



HTML5 - Structure

## HTML5 page structure - part 3

---

- 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?*

# HTML5 - Extra elements - media - part I

---

## 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?*

## HTML5 - Extra elements - media - part 2

---

### 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



## HTML5 - Extra elements - media - part 3

---

### 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*

## HTML5 - Extra elements - media - part 4

---

### 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 I

---

## 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*

## HTML5 - Extra elements - graphics - part 2

---

### 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

# HTML5 - Extra elements - graphics - part 5

---

## canvas examples - basics

- basic drawing - rectangle & staircase
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic/>
- Example - basic drawing - stepped pyramid
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic2/>
- Example - various colours
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic3/>
- Example - basic drawing - rectangle outlines
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic4/>
- Example - draw lines - line & pyramid
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic5/>
- Example - draw a stickman
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic6/>
- Example - fill paths
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic7/>

## HTML5 - Extra elements - graphics - part 6

---

### canvas examples - curves & circles

- Example - arcs and circles
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic8/>
- Example - Bézier curves - quadratic
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic9-quadratic/>
- Example - Bézier curves - cubic
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic9-cubic/>
- Example - arcs and circles - combine shapes to create an *ankh*
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic10-ankh/>
- Example - circle function
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic11-function-circles/>



# HTML5 - Extra elements - graphics - part 7

---

## canvas examples - animation & fun

- Example - horizontal animation
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-animation/animation1/>
- Example - animate size
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-animation/animation2/>
- Example - variant mouse colours
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-animation/animation3.1/>
- Example - variant mouse colours
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-animation/animation3.2/>
- Example - random movement and animation
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-animation/animation3.3/>

## HTML5 - Extra elements - graphics - part 8

---

### canvas examples - images & files

- Example - draw image to canvas from local file
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-image/basic1/>
- Example - draw image to canvas from local file - dw & dh
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-image/basic2/>
- Example - draw image to canvas from local file - dw & dh plus source crop
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-image/basic3/>

## HTML5 - Extra elements - graphics - part 9

---

### canvas examples - move & control

- Example - move ball with keyboard control
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-game/basic-ball-move1/>
- Example - update move ( ) to check canvas boundaries
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-game/basic-ball-move2/>
- Example - move ball on 4-point axis
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-game/basic-ball-move3/>
- Example - move sprite image
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-game/basic-sprite-move1/>
- Example - move sprite image
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-game/basic-ball-move4/>
- Example - check basic collision against blocks
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-game/basic-ball-move5/>
- Example - check basic collision against blocks - horizontal
  - <http://linode4.cs.luc.edu/teaching/cs/demos/I25/drawing/basic-game/basic-ball-move6/>

## HTML5 - Extra elements - graphics - part 10

---

### more fun canvas examples

Some fun examples of animations with HTML5 Canvas API.

- Destroy things in a video - <http://www.craftymind.com/factory/html5video/CanvasVideo.html>
- Particles - <https://codepen.io/eltonkamami/pen/ECrKd>
- Curtain - <https://codepen.io/dissimulate/pen/KrAwx>
- Jelly - <https://codepen.io/dissimulate/pen/dJgMaO>
- Canvas cycle - <http://www.effectgames.com/demos/canvascycle/>

# HTML5 - structure and layout

---

## fun exercise

Choose one of the following app examples,

- magazine or news reader
  - e.g. *a local newspaper, or perhaps a news aggregator*
- social media aggregator
  - *collect and display updates and news from various social media APIs*
- gaming portal for a community
  - *collect latest scores, news, comments, photos &c. for a chosen game*

Then, consider the following

- initial HTML5 structural design and layout
- required semantic markup for a chosen example page
  - e.g. *elements and nested usage, structure &c.*
- structural parts of the app with regular updates
  - e.g. *news and content update...*

~ 10 minutes

# Demos

---

## ***HTML5***

- Basic structure
- New elements added
- Semantic structuring
- HTML5 Video playback
- HTML5 Audio playback

## ***HTML5 Canvas demos***

- HTML5 Canvas - Rectangle
- HTML5 Canvas - Square
- HTML5 Canvas - Assorted Shapes
- HTML5 Canvas - Retro Breakout Game
- please see links in slides for further examples

## Resources

---

- Can I Use\_\_\_\_\_?
  - e.g. *Can I Use Drag and Drop?*
- Check browser compatibility using HTML5 Test
- HTML5 Audio & Codecs
- MDN Documentation
  - *Section element*
- W3C Documentation
  - *Section element*

### **Extras**

- HTML5 Canvas - fun examples
  - *Destroy things in a video*
  - *Particles*
  - *Curtain*
  - *Jelly*
  - *Canvas cycle*