

# **Comp 388/424 - Client-Side Web Design**

---

Spring Semester 2016 - Week 2

Dr Nick Hayward

# Contents

---

- Course resources
- Group projects
- HTML5
  - *intro*
  - *elements*
  - *semantic elements*
  - *semantic elements and structure*
  - *page structure*
  - *extra elements*
- HTML basics
- DOM basics
- HTML <head> element
- CSS basics

## Course resources

---

- Course website = <http://csteach424.github.io/>
  - *weekly assignments*
  - *bibliography*
  - *weekly notes*
  - *useful links and resources*
- Course GitHub account = <https://github.com/csteach424>
  - *source code examples*
  - *backup of notes*
- Course Trello group = <https://trello.com/comp424spring2016luc>
  - *weekly assignments and submissions*
- Course Slack group = <https://comp424-spring2016.slack.com/>
  - *course announcements*
  - *discussions*
  - *post questions, chat with colleagues, organise group chats...*

**Sakai is not used for this course**

## Group projects

---

- add project details to Trello group
  - *Comp 424 Trello*
- create channels on Slack for group communication
  - *also use 'final-project' channel to ask general questions about projects...*
- start working on an idea for your project
- start planning weekly development up to and including DEV Week
  - *7th to 14th March 2016*
  - *DEV week presentation due on 14th March 2016 @ 4.15pm*

## HTML5 - intro

---

- finally became a standard in October 2014
- introduces many new features to HTML standard
- additional features include, for example
  - *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 webpage 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 I



## HTML5 - Comments

---

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

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

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

# Image - HTML5 Sample Rendering I

---

Rendering of demo I

## Our first web page

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

Source - Demo I

# HTML5 - Semantic elements - part I

---

- new semantic elements added for HTML5
- known as **block-level** elements
- include 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>
<!-- navigation elements, links... -->
<h1>Our first web page</h1>
</header>
<nav>Option 1</nav>
<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 2

## Image - HTML5 Sample Rendering 2

---

Rendering of demo 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...

Source - Demo 2

## 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

---

- 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*
    - eg: `<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...



# 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*
  - eg:

```
<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>

---

## ■ <section>

- defines a section of a page or document
- W3C 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.*

## Source - W3C Documentation

## ■ a site can be sub-divided into multiple <section> groupings

- eg: 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
  - eg: 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 - <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 3

## HTML5 - Semantic elements and structure - <footer>

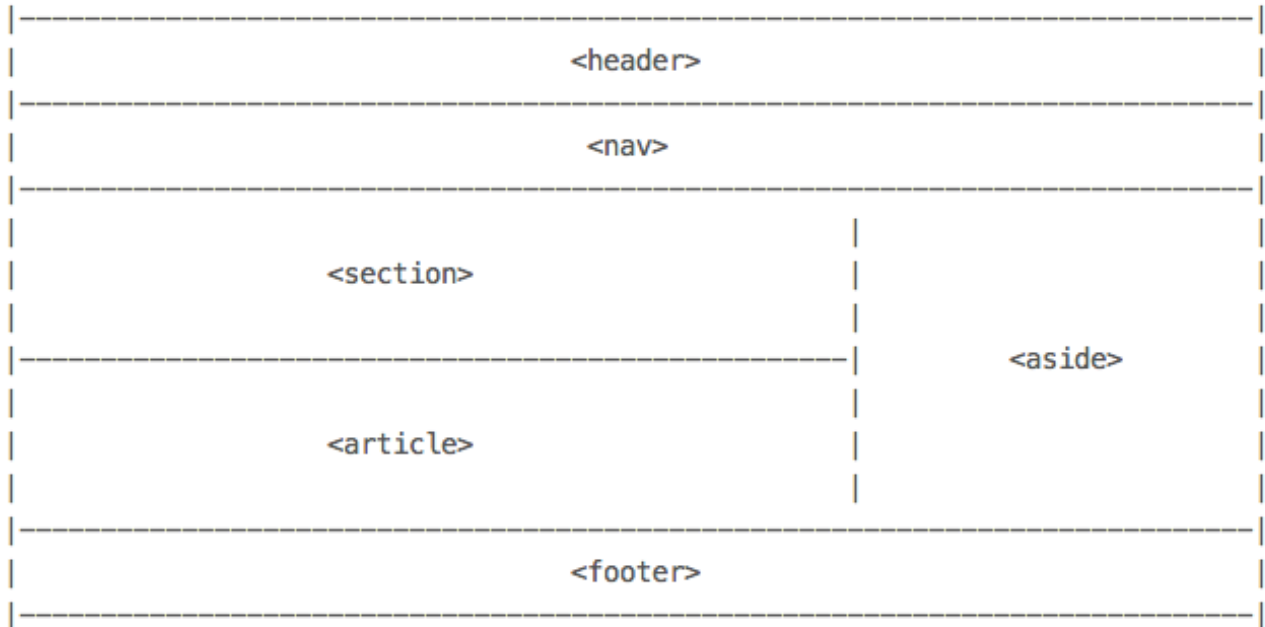
---

- <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 or 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 standard usage for <footer>

# Image - HTML5 Page Structure - Part I

---

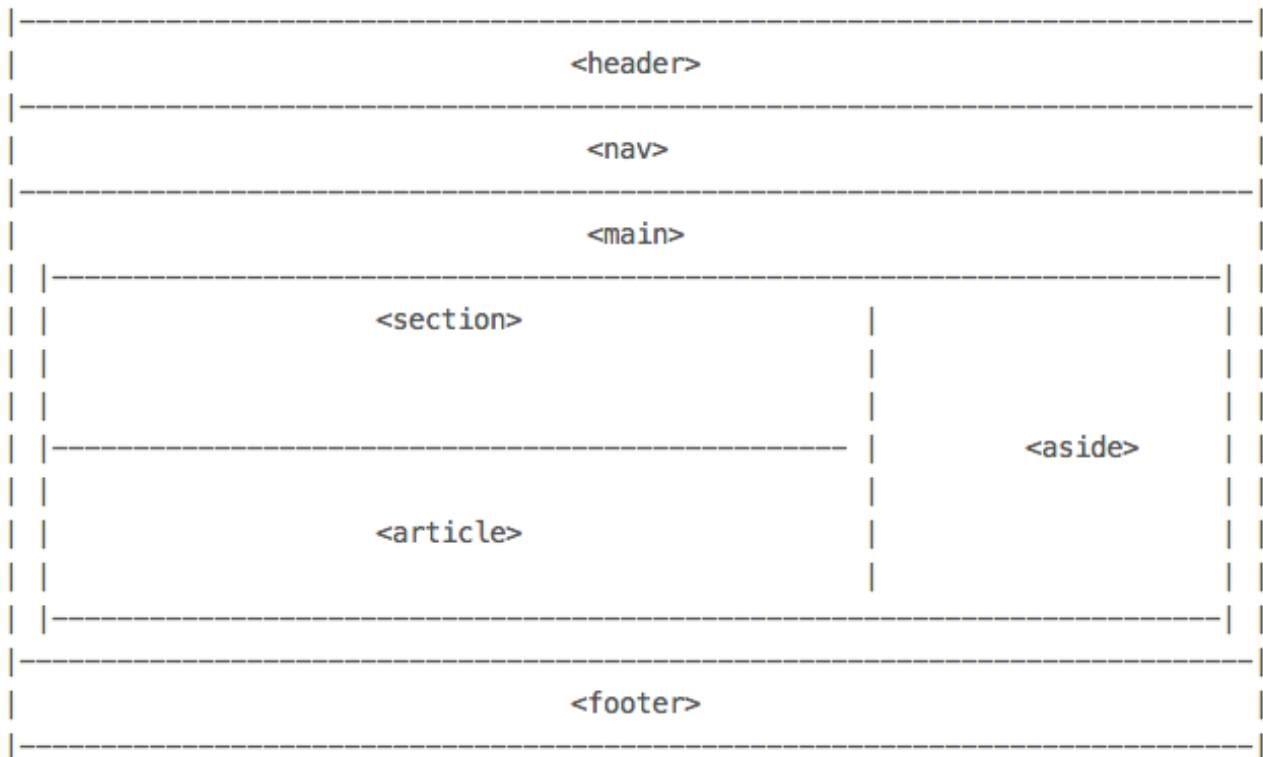
## HTML5 Semantic elements - Part I



## Image - HTML5 Page Structure - Part 2

---

### HTML5 Semantic elements - Part 2



## 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 move `<nav>` into the `<header>` division at the top
  - *not always necessary and not compulsory*
- 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\_\_\_\_\_?
  - eg: *Can I Use Drag and Drop?*



# HTML5 - Extra elements - media - part I

---

## <video> element

- until HTML5, video playback reliant on plugins
  - eg: *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> - 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 4 - HTML5 Video playback

## HTML5 - Extra elements - media - part 3

---

### <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> - 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 5 - HTML5 Audio playback

# HTML5 - Extra elements - graphics - part I

---

- 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...
  - *eg: draw a rectangle on the canvas*

## HTML5 - Extra elements - graphics - part 2

---

<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 6 - HTML5 Canvas - Rectangle

## HTML5 - Extra elements - graphics - part 3

---

A cube can be created as follows,

```
<script type="text/javascript">
function draw() {
  /*black cube*/
  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 7 - HTML5 Canvas - Cube

## HTML5 - Extra elements - graphics - part 4

---

- modify drawing for many different shapes and patterns
  - *simple lines, circles, gradients, images...*
    1. shows different rendered shapes on a canvas.
- Demo 8 - HTML5 Canvas - Assorted Shapes
  2. little retro games
- Demo 9 - HTML5 Canvas - Retro Breakout Game



## HTML Basics - <body> - part I

---

- to define the main body of the web page we use the <body> element
- headings can be created using variants of
  - <h1>, <h2>.....<h6>
- we can now add some simple text in a <p> element

```
<p>...</p>
```

- add a line break using the <br /> element
- <hr /> element adds a horizontal line
- comments can also be added through our HTML

```
<!-- comment... -->
```

### Linking in HTML

- linking is an inevitable part of web design and HTML usage
- can be considered within three different contexts
  - *linking to an external site*
  - *linking to another page within the same site*
  - *linking different parts of the same page*
- add links to text and images within the HTML
- <a> element for links plus required attributes
  - `<a href="http://www.google.com/">Google</a>` or
  - `<a href="mailto:name@email.com">Email</a>`
  - `<a href="/another_page.html">another page</a>`
  - `<a name="anchor">Internal Anchor</a>` or
  - `<a id="anchor">Anchor</a>`
  - `<a href="#anchor">Visit Internal Anchor</a>` or
  - `<a href="/another_page.html#anchor">Visit External Anchor</a>`
- Demo 10 - HTML - Internal Anchor

## HTML Basics - <body> - part 3

---

### Linking in HTML - continued

- standard attributes supported by <a> element include
  - *class, id, lang, style, title...*
- optional attributes are available for <a> element including
  - *target, href, name...*
- target attribute specifies where the link will be opened relative to the current browser window
- possible attribute values include

```
_blank  
_self  
_parent  
_top
```

## HTML Basics - <body> - part 4

---

### Working with images

- <img> allows us to embed an image within a web page
- <img> element requires a minimum `src` attribute

```

```

- other optional attributes include
  - *class, id, alt, title, width, height...*
- use images as links
- image maps

```
<map name="textmap">  
  <area shape="rect" coords="..." alt="Quote 1" href="notes1.html" />  
</map>
```

# Demos

---

- Demo 1 - Our first web page
- Demo 2 - New elements added
- Demo 3 - Semantic structuring
- Demo 4 - HTML5 Video playback
- Demo 5 - HTML5 Audio playback
- Demo 6 - HTML5 Canvas - Rectangle
- Demo 7 - HTML5 Canvas - Cube
- Demo 8 - HTML5 Canvas - Assorted Shapes
- Demo 9 - HTML5 Canvas - Retro Breakout Game
- Demo 10 - HTML - Internal Anchor

## References

---

- [HTML5 Audio formats](#)
- [HTML5 Test](#)
- [MDN - HTML developer guide](#)
- [Block-level elements](#)
- [Content categories](#)
- [Inline elements](#)
- [W3 Schools - HTML Block and Inline Elements](#)
- [W3C HTML5 Documentation](#)
- [W3Schools - HTML5 Semantic Elements](#)