Lecture 4b: New elements in HTML5
Sectioning elements

HTML 5 introduces a lot of sectioning elements.

Meant to give more meaning to your pages.

People talk of semantic meaning (standardized, understandable by computers and humans.)

Many of these replace the `<div>` element.
Sectioning elements

<main> Used to contain the main content of your webpage. This element should be used only once.

It should not be placed within <article>, <aside>, <header>, <footer>, or <nav> elements.
Sectioning elements

<article> is used for individual entries that one might syndicate.

Examples: News articles, blog posts, or user comments.

<article> elements may be nested. It is implied that the nested elements are related to the outer <article> element.
Sectioning elements

Much like `<article>`, `<section>` element is used to encapsulate a group of related content.

Main difference is that `<section>` does not need to make sense out of context.

Example: `<article>` might be a post and the `<section>` elements within the `<article>` might be sub-topics discussed in the post.
Sectioning elements

<aside> element is used for content that tangentially relates to the material being discussed.

Think a sidebar or a footnote in a book.

<header> element is used as an introductory content for an article. It might contain an <h1> heading and the date of the article.
Structural elements

`<nav>` is used to enclose navigation menu or multiple links.

`<footer>` is used for anything that might go into the footer: Copyright stuff, links or any closing slogans etc.
What is div then?

<div> element still has its place in HTML5. It is now used as a container for other elements.

- Any group of elements to be styled together.
- Whole sidebar can be a div.
- The whole webpage can be inside a div

With CSS one can do things like set widths, center webpage and in general style these larger blocks.

When content does not seem to belong to any of the structural elements discussed before, it can go inside a div.

They are also used in JavaScript to create boxes, popups and containers and to create some "animation effects". They can be used to create a clickable area.
Other HTML5 elements: `<audio>` element

```html
<audio controls autoplay loop>
  <source src = "karmapolicel.ogg" />
  <source src = "karmapolicel.wav" />
  <source src = "karmapolicel.mp3" />
  <p>Your browser does not support the audio element.</p>
</audio>
```

controls: Displays the standard HTML5 controls for the audio on the web page.
autoplay : Makes the audio play automatically.
loop : Make the audio repeat (loop) automatically.

Example:
http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_audio_all

controls, autoplay and loop are boolean attributes in HTML5 they need not come in name-value pairs
Other HTML5 elements

**<video> Element**

```
<video width="320" height="240" controls="controls">
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

Example:
http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_video_all
Other HTML5 elements

<canvas> Element

<canvas width="200" height="100" />

Canvas element is a rectangular element used for drawing. By default it has no content and no border.

All the drawing is done via JavaScript. We will talk about this more when we get to JS.

Example:
http://www.w3schools.com/html/html5_canvas.asp
Syntax differences between HTML 5 and XHTML

In XHTML all tags and attributes are in lowercase. HTML5 is case insensitive.

In XHTML all tags need to have a matching closing tag. In HTML5 Most table elements, including `<td>`, `<tr>` and `<th>`, do not need to be closed. Neither do `<body>`, `<dd>`, `<dt>`, `<head>`, `<li>`, `<p>`, even `<html>`

It is even more messy in HTML5 since some times `<p>` elements need to be closed, but under right circumstances they need not be.

In HTML5 you can enclose multiple elements in a link.

In HTML5 input elements in form need not be inside a fieldset, div or a p
Ok so which is better?

No conclusive answer.

The advantages and benefits depend on how you use the languages.

For content that uses lot of media elements use HTML5.

If web content is mainly for computer screens XHTML is a better choice.