HTML 5

Introduction

What is New in HTML5?

- The DOCTYPE declaration for HTML5 is very simple:
  - **<!DOCTYPE html>**
- The character encoding (charset) declaration is also very simple:
  - **<meta charset="UTF-8">**

What is New in HTML5?

HTML5 Example:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Title of the document</title>
    </head>
    <body>
        Content of the document……
    </body>
</html>
```

New HTML5 Elements

- The most interesting new elements are:

  - New semantic elements like
    `<header>`, `<footer>`, `<article>`, and `<section>`
  - New form controls like
    `number`, `date`, `time`, `calendar`, and `range`
  - New graphic elements
    `<svg>` and `<canvas>`
  - New multimedia elements
    `<audio>` and `<video>`

New HTML5 API's (Application Programming Interfaces)

- The most interesting new API's are:
  - HTML Geolocation
  - HTML Drag and Drop
  - HTML Local Storage
  - HTML Application Cache
  - HTML Web Workers
  - HTML SSE

Elements Removed in HTML5

- The following HTML4 elements have been removed from HTML5:

Element	Use instead
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
	CSS
<dir>	
<font>	CSS
<strike>	CSS
<tt>	CSS

HTML History

- Since the early days of the web, there have been many versions of HTML:

Version	Year
Tim Berners-Lee invented www	1989
Tim Berners-Lee invented HTML	1991
Dave Raggett drafted HTML+	1993
HTML Working Group defined HTML 2.0	1995
W3C Recommended HTML 3.2	1997
W3C Recommended HTML 4.01	1999
W3C Recommended XHTML 1.0	2000
HTML5 WHATWG First Public Draft	2008
HTML5 WHATWG Living Standard	2012
HTML5 W3C Final Recommendation	2014

HTML History

- Tim Berners-Lee invented the "World Wide Web" in 1989, and the Internet took off in the 1990s.
- From 1991 to 1998, HTML developed from version 1 to version 4.
- In 2000, the World Wide Web Consortium (W3C) recommended XHTML 1.0.
- The XHTML syntax was strict, and the developers were forced to write valid and "well-formed" code.
- In 2004, WHATWG (Web Hypertext Application Technology Working Group) was formed in response to slow W3C development, and W3C's decision to close down the development of HTML, in favor of XHTML.
- WHATWG wanted to develop HTML, consistent with how the web was used, while being backward compatible with older versions of HTML.
- In the period 2004-2006, the WHATWG initiative gained support by the major browser vendors.
- In 2006, W3C announced that they would support WHATWG.

HTML History

- In 2008, the first HTML5 public draft was released
- In 2012, WHATWG and W3C decided on a separation:
- **WHATWG will develop HTML as a "Living Standard"**.
- A living standard is never fully complete, but always updated and improved. New features can be added, but old functionality can not be removed.
- The WHATWG Living Standard was published in 2012, and is continuously updated.
- **W3C will develop a definitive HTML5 and XHTML5 standard**, as a "snapshot" of WHATWG.
- The W3C HTML5 recommendation was released 28. October 2014.

HTML5 Browser Support

- HTML5 is supported in all modern browsers.
- In addition, all browsers, old and new, automatically handle unrecognized elements as inline elements.
- Because of this, you can "teach" old browsers to handle "unknown" HTML elements.

Define HTML5 Elements as Block Elements

- HTML5 defines 8 new semantic HTML elements. All these are block level elements.
- To secure correct behavior in older browsers, you can set the CSS display property to block
- A block-level element always starts on a new line.
- A block-level element always takes up the full width available (stretches out to the left and right as far as it can).
- A block level element has a top and a bottom margin, whereas an inline element does not.

Example:

```
header, section, footer, aside, nav, main, article, figure
{
    display: block;
}
```

Define HTML5 Elements as Block Elements

- HTML Block Elements and Inline Elements
- Most HTML elements are defined as block level elements or inline elements.
- Block level elements normally start (and end) with a new line, when displayed in a browser.

Examples:

`<h1>`, `<p>`, `<ul>`, `<table>`

- Inline elements are normally displayed without line breaks.

Examples:

`<b>`, `<td>`, `<a>`, `<img>`

Inline Elements

- An inline element does not start on a new line.

- Here are the inline elements in HTML:

- <a > <abbr > <acronym > <b > <bdo > <big >
 <button > <cite > <code > <dfn > <i > <img > <input > <kbd > <label > <map > <object > <output > <q > <samp > <script > <select > <small >   <sub > <sup > <textarea > <time > <tt > <var>

Adding New Elements to HTML

- You can add any new element to HTML with a browser trick:
- This example adds a new element called <myHero> to HTML, and defines a display style for it:

Example:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Creating an HTML
        Element</title>
        <style>
            myHero {
                display: block;
                background-color: #ddd;
                padding: 50px;
                font-size: 30px;
            }
        </style>
    </head>
    <body>
        <h1>My First Heading</h1>
        <p>My first paragraph.</p>
        <myHero>My First Hero</myHero>
    </body>
</html>
```

New Elements in HTML5

- New Semantic/Structural Elements
- HTML5 offers new elements for better document structure:

Tag	Description
<article>	Defines an article in the document
<aside>	Defines content aside from the page content
<bdi>	Defines a part of text that might be formatted in a different direction from other text
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content, like illustrations, diagrams, photos, code listings, etc.

Example: Isolate the usernames from the surrounding text-direction settings:

```
<ul>
  <li>User <bdi>hrefs</bdi>: 60 points</li>
  <li>User <bdi>jdoe</bdi>: 80 points</li>
  <li>User <bdi> إيان </bdi>: 90 points</li>
</ul>
```

- User hrefs: 60 points
- User jdoe: 80 points
- User إيان : 90 points

- User hrefs: 60 points
- User jdoe: 80 points
- User 90 : إيان points

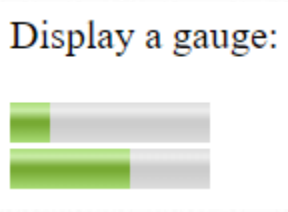
New Elements in HTML5

- New Semantic/Structural Elements
- HTML5 offers new elements for better document structure:

Tag	Description
<footer>	Defines a footer for the document or a section
<header>	Defines a header for the document or a section
<main>	Defines the main content of a document
<mark>	Defines marked or highlighted text
<menuitem>	Defines a command/menu item that the user can invoke from a popup menu
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links in the document
<progress>	Defines the progress of a task

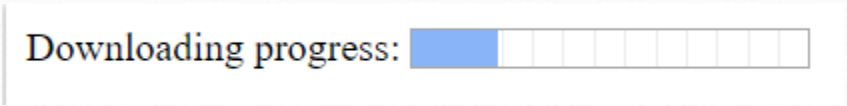
Example:

```
<p>Display a gauge:</p>
<meter value="2" min="0" max="10">2 out of 10</meter><br>
<meter value="0.6">60%</meter>
```



Example: Downloading progress:

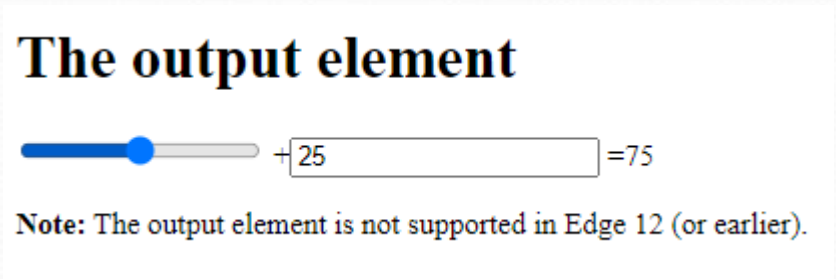
```
<progress value="22" max="100">
</progress>
```



New Form Elements

Tag	Description
<datalist>	Defines pre-defined options for input controls
<keygen>	Defines a key-pair generator field (for forms)
<output>	Defines the result of a calculation

```
<!DOCTYPE html>
<html> <body>
<h1>The output element</h1>
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
<input type="range" id="a" value="50">
+<input type="number" id="b" value="25">
=<output name="x" for="a b"></output>
</form>
<p><strong>Note:</strong> The output element is not supported in Edge 12 (or earlier).</p>
</body> </html>
```



https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_output

New Input Types

color	date	Datetime	datetime-local
Email	month	Number	range
Search	tel	Time	url
week			

New Input Attributes

Autocomplete	autofocus	form	formaction
Formenctype	formmethod	formnovalidate	formtarget
height	width	list	min and max
Multiple	pattern (regexp)	placeholder	required
step			

HTML5 - New Attribute Syntax

- HTML5 allows 4 different syntaxes for attributes.
- This example demonstrates the different syntaxes used in an <input> tag:

Type	Example:
Empty	`<input type="text" value="John Doe" disabled>`
Unquoted	`<input type="text" value=John>`
Double-quoted	`<input type="text" value="John Doe">`
Single-quoted	`<input type="text" value='John Doe'>`

- In HTML5, all 4 syntaxes may be used, depending on what is needed for the attribute.

New Media Elements

Tag	Description
<audio>	Defines sound or music content
<embed>	Defines containers for external applications (like plug-ins)
<source>	Defines sources for <video> and <audio>
<track>	Defines tracks for <video> and <audio>
<video>	Defines video or movie content

HTML5 Graphics

Tag	Description
<canvas>	Defines graphic drawing using JavaScript
<svg>	Defines graphic drawing using SVG

HTML5 Semantic Elements

Semantics means (from Ancient Greek), is the study of meaning.

Semantic elements are elements with a meaning.

What are Semantic Elements?

What are Semantic Elements?

- A semantic element clearly describes its meaning to both the browser and the developer.
- Examples of non-semantic elements:

`<div>` and `<span>`

Tells nothing about its content.

- Examples of semantic elements:

`<form>`, `<table>`, and `<img>`

Clearly defines its content.

Browser Support

- HTML5 semantic elements are supported in all modern browsers.
- In addition, you can "teach" older browsers how to handle "unknown elements".

New Semantic Elements in HTML5

- Many web sites contain HTML code like

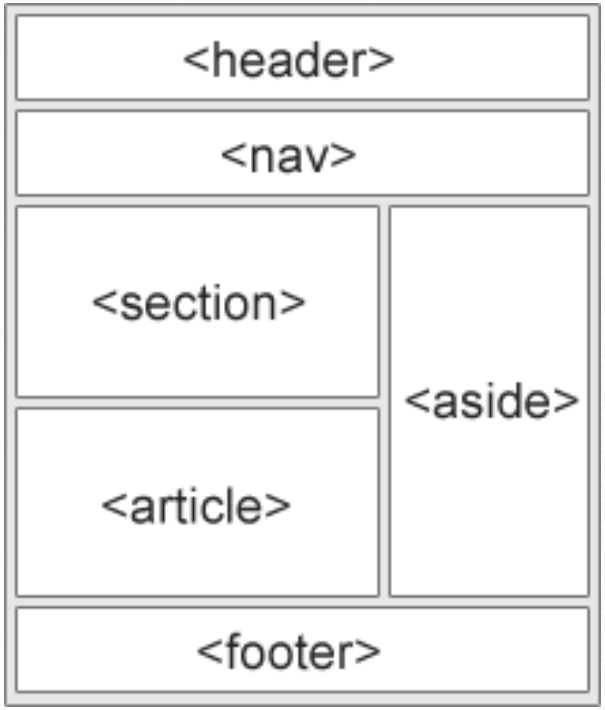
```
<div id="nav"> <div class="header"> <div id="footer">
```

to indicate navigation, header, and footer.

- HTML5 offers new semantic elements to define different parts of a web page:

New Semantic Elements in HTML5

- **<article>**
- **<aside>**
- **<details>**
- **<figcaption>**
- **<figure>**
- **<footer>**
- **<header>**
- **<main>**
- **<mark>**
- **<nav>**
- **<section>**
- **<summary>**
- **<time>**



HTML5 <section> Element

- The **<section>** element defines a section in a document.
- According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with a heading."
- A Web site's home page could be split into sections for introduction, content, and contact information.

Example:

```
<section>
    <h1>WWF</h1>
    <p>The World Wide Fund for Nature (WWF) is....</p>
</section>
```

HTML5 <article> Element

- The **<article>** element specifies independent, self-contained content.
- An article should make sense on its own, and it should be possible to read it independently from the rest of the web site.
- Examples of where an <article> element can be used:
  - Forum post
  - Blog post
  - Newspaper article

Example:

```
<article>
  <h1>What Does WWF Do?</h1>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,and build a future in which humans live in harmony with nature.</p>
</article>
```

Nesting Semantic Elements

- In the HTML5 standard, the <article> element defines a complete, self-contained block of related elements.
- The <section> element is defined as a block of related elements.
- Can we use the definitions to decide how to nest elements? No, we cannot!
- On the Internet, you will find HTML pages with <section> elements containing <article> elements, and <article> elements containing <sections> elements.
- You will also find pages with <section> elements containing <section> elements, and <article> elements containing <article> elements.

HTML5 <header> Element

- The <header> element specifies a header for a document or section.
- The <header> element should be used as a container for introductory content.
- You can have several <header> elements in one document.
- The following example defines a header for an article:

Example:

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
   <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.</p>
</article>
```

HTML5 <footer> Element

- The <footer> element specifies a footer for a document or section.
- A <footer> element should contain information about its containing element.
- A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
- You can have several <footer> elements in one document.

Example

```
<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a href="mailto:someone@example.com">
  someone@example.com</a>.</p>
</footer>
```

HTML5 <figure> and <figcaption> Elements

- In books and newspapers, it is common to have captions with images.
- The purpose of a caption is to add a visual explanation to an image.
- With HTML5, images and captions can be grouped together in <figure> elements:

Example:

```
<figure>
  <img src="pic_mountain.jpg" alt="The Pulpit Rock" width="304" height="228">
  <figcaption>Fig1. - The Pulpit Rock, Norway</figcaption>
</figure>
```

The <img> element defines the image, the <figcaption> element defines the caption.

Why Semantic HTML5 Elements?

- With HTML4, developers used their own favorite attribute names to style page elements:
- header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, ...
- This made it impossible for search engines to identify the correct web page content.
- With HTML5 elements like:

`<header> <footer> <nav> <section> <article>,`

this will become easier.

- According to the W3C, a Semantic Web:

"Allows data to be shared and reused across applications, enterprises, and communities."

Semantic Elements in HTML5

Tag	Description
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section

Semantic Elements in HTML5

Tag	Description
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time