HTML Basics

Outline

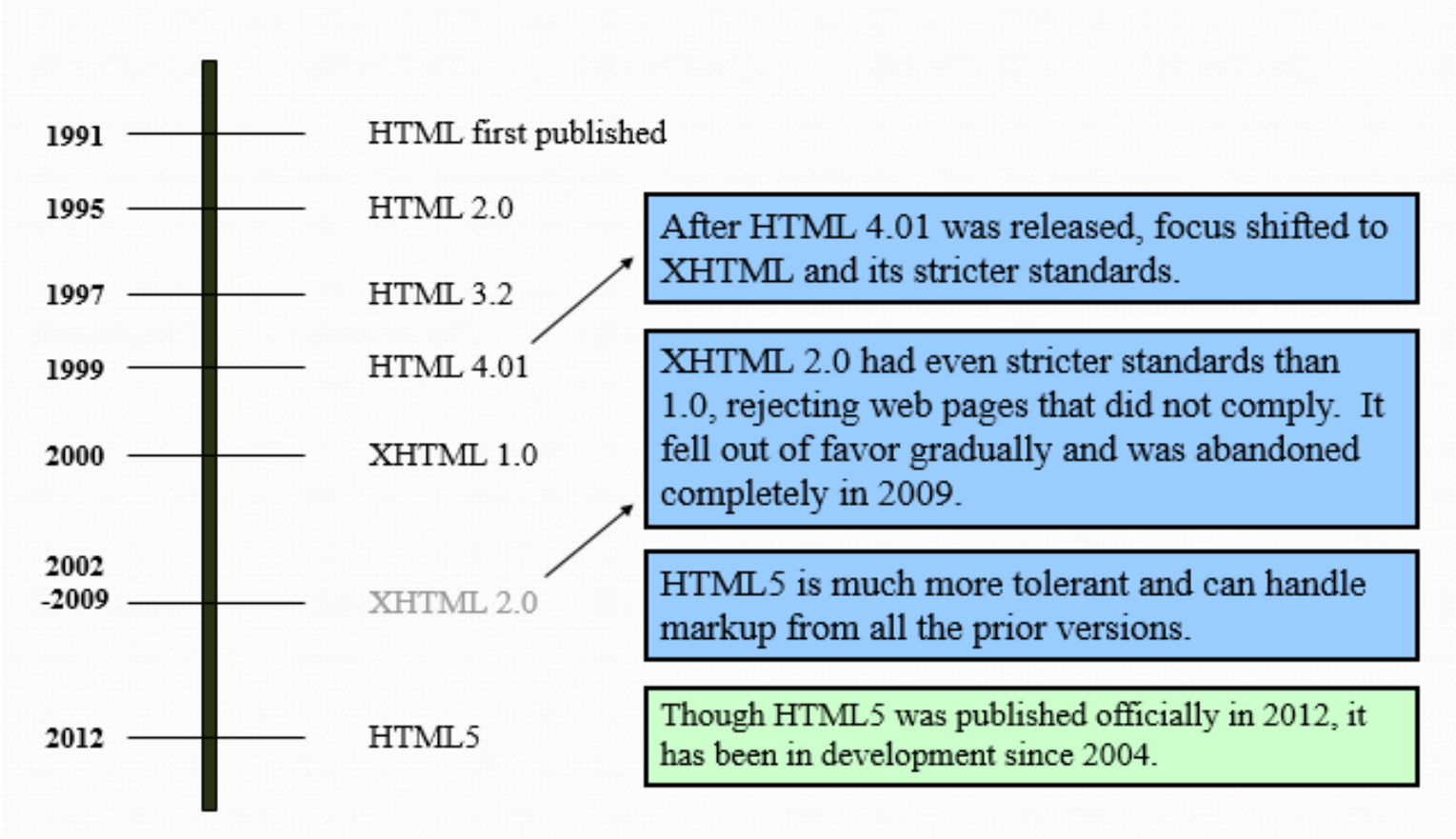
- Introduction to HTML5
- HTML Standards – Comparisons
- HTML Formatting Tags
- Semantic Tags
- HTML5 Content Model
- Working with Lists
- HTML Form
- Inline Content Editing
- Support for Audio and Video

Introduction

What is HTML

- HTML is a core technology Markup language of the Internet used for structuring and presenting content for the World Wide Web.
- HTML5 is the final and complete fifth revision of the HTML standard of the World Wide Web Consortium (W3C).
- The previous version, HTML 4, was standardized in 1999.
- HTML offers features that provide rich media support for creating web applications.

Evolution of HTML



Introduction to HTML5

- HTML 5 specification promised to provide in-built support for commonly used features
- HTML5 like any other HTML specification, has the same type of syntax. It defines few more tags and the DOM API associated with those tags
- In spite of removal of few tags that are part of HTML4, browsers will still support the usage of those tags for backward compatibility
- To test the support for HTML5 in a browser, individual features must be tested.
- It is still a work in progress. No browsers have full HTML5 support. It will be many years – perhaps not until 2018 or later - before being fully defined and supported

What is HTML5?

- Evolution of the browser into a programming platform
- Introduction to new semantic tag to add meaning to HTML document
- In-built support for embedding audio/video content to the document and many other features

HTML 5 Features

- The following features are part of HTML 5 specification
  - Audio/Video
  - Inline Editing
  - Drag & Drop
  - Offline Application
  - AJAX history
  - Selector API
  - Forms

What is not in HTML5?

- The following features were misunderstood to be a part of HTML 5 specification.
  - Canvas
  - Geo location
  - Messaging API
  - Web Socket
  - Web Worker
  - SVG
  - CSS3
- These are actually not a part of HTML 5 specification but are associated specifications developed individually

Getting Started

DOCTYPE Specification

- The DOCTYPE specification of a HTML page helps the browser determine the version of HTML used by the document
- The DOCTYPE specification must be the very first line of the document.
- Once the browser encounters the “DOCTYPE” tag based on the document type specified, it operates in the standards mode
- Without the document type specified, the browser will not check for any particular HTML syntax. This mode is called no-standard mode or quirks mode
- The DOCTYPE specification for HTML5 is as shown below

```
<!DOCTYPE html>
```

HTML tag

- The first tag that marks the start of any HTML document is the HTML tag
- The HTML tag is also redefined in HTML5. The HTML tag must be used as follows

```
<html lang="en">
```

- Lang attribute along with HTML tag specifies which language will be used by HTML document. As given above, it uses 'English'

Charset Attribute

- The character set declaration informs the browser about the character set used in a page.
- Based on the character set, the content of page is rendered
- Missing the character set declaration can lead to security vulnerabilities
- The character set specification is also simplified with HTML 5

```
<meta charset="utf-8" />
```

Meta Tag

- Meta tag is used to provide provide additional information about the page like
  - charset
  - Viewport
  - Keywords
- Meta tag uses the name of the property and the value for that property as given below:

```
<meta charset="UTF-8"/>
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<meta name="keywords" content="HTML5 Test Design"/>
```

Simple HTML Page

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Exploring an HTML document</title>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="keywords" content="Angular SPA Responsive web design" />
</head>
<body>
    <!--This is comment in HTML-->
    <h2>Exploring an HTML document-This is heading</h2>
    <div>
        <p>This is simple paragraph inside the div</p>
    </div>
</body>
</html>
```

HTML Formatting Tags

Text Structuring

- HTML's main job is to give text structure and meaning, so that a browser can display it correctly

- HTML can structure a page of text by adding headings and paragraphs, emphasizing words, creating lists, creating hyperlinks

Headings and Paragraphs

- Most structured text is comprised of headings and paragraphs
- Heading elements:
  - Six Heading elements - **<h1>, <h2>, <h3>, <h4>, <h5>, <h6>**
  - Each element represents a different level of content in document
  - <h1> represents the main heading, <h2> represents the subheading, <h3> represent the subheading and so on…
- Paragraph Element
  - **<p>**
    - <p> indicates the start of a new paragraph.
  - ** **
    - is a block-level element which defines a block of content in a page

Text Formatting Tags

Tag	Description	Example
	Defines bold text.	This is bold Text
	Defines italic text	This is Italic Text
<u></u>	Defines underline to the text	<u>This is underline Text<u>
<del>	Defines deleted text	This is ~~para~~graph
	Defines emphasized text	This text is emphasized
<ins>	Defines inserted text	This text is <ins>inserted</ins> to the document
<mark>	Defines highlighted text	<mark>This text is marked</mark>
<small>	Defines small text	This text is small
	Defines strong text	This text is strongly emphasized
<sub>	Defines subscripted text	This is _{subscript}
<sup>	Defines superscripted text	This is ^{Superscript}
<dfn>	Defines a definition term	This is <dfn>Defination</dfn> term.
<cite>	Defines a citation	<cite>Citation here...</cite>
<q>	Defines a short quotation	Virat says, <q>I love cricket.</q>

Text Formatting Tags

Tag	Description	Example
<bdo>	Defines a text direction	The sequence "1-2-3-4-5" was reversed as: <bdo dir="rtl">1-2-3-4-5</bdo>
<address>	Defines contact information for the author of a document	<address> Details </address>
<abbr>	Defines a abbrevation	<abbr title="Hyper Text Markup Language">HTML</abbr>
<code>	Defines computer code text	Regular text. <code>computer code</code> Regular text.
<kbd>	Defines keyboard text	Please, input "<kbd>Yes</kbd>" or "<kbd>No</kbd>"
<samp>	Defines a sample computer code	<samp>HTTP 404 - File not found</samp>,
<var>	Defines a variable	A simple equation: <var>x</var> = <var>y</var> + 2

<pre>	Defines preformatted text	<pre>
		The pre
		element preserves spaces,
		Line-breaks, tabs...
		</pre>

<blockquote>	Defines a long quotation	<blockquote> This is an example of a long quotation. </blockquote>

Block and Inline Elements

- Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline
- Block-level elements
  - A block-level element always starts on a new line and takes up the full width available
  - Examples
    -
    - <h1> - <h6>
    - <p>
    - <form>
- Inline Elements
  - An inline element does not start on a new line and only takes up as much width as necessary.
  - Example:
    -
    - <a>
    - <img>

Working with Images

- In html <img> tag is used to add image on the web page.
- Syntax:

  <img src="URL" alt="text“ width=“100” height=“100”>

- Image Maps
  - Use the <map> tag to define an image-map. An image-map is an image with clickable areas

Working with Images

```
<img src="planets.gif" alt="Planets" usemap="#planetmap" style="width:145px;height:126px;">

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>
```

Working with Hyperlinks – Building Navigation

- Hyperlink is used to navigate from one web resource to another in html or server anywhere in the world. Hyperlink in HTML is specified using the <a> tag.

  Syntax: Text link

- Example:

  the Mozilla homepage.

- Image as Link

  <img src="mozilla-image.png" alt="mozilla logo that links to the mozilla homepage">

- Email Links:

  Send email to nowhere

Semantic Tags

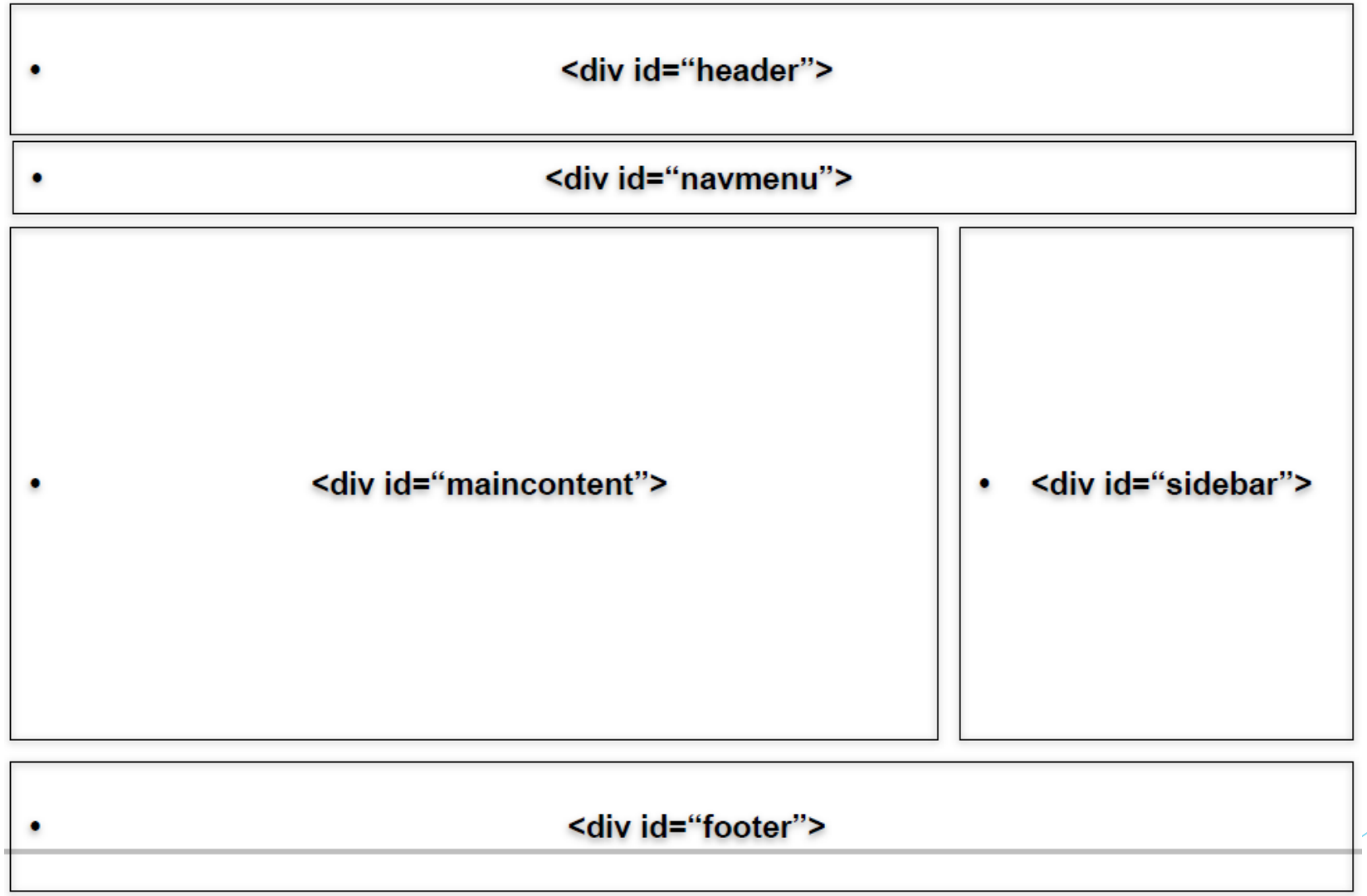
Nesting and Grouping Content

- In HTML4, div, p and span tags were used to structure the document.
- This structuring is typically to add suitable CSS classes. It added no semantic value to the document
- Semantic tags were added to HTML5 to add meaning to the structure of the document
- Semantic does not contain any presentation logic in them.
- They are interpreted in the same way as p tag

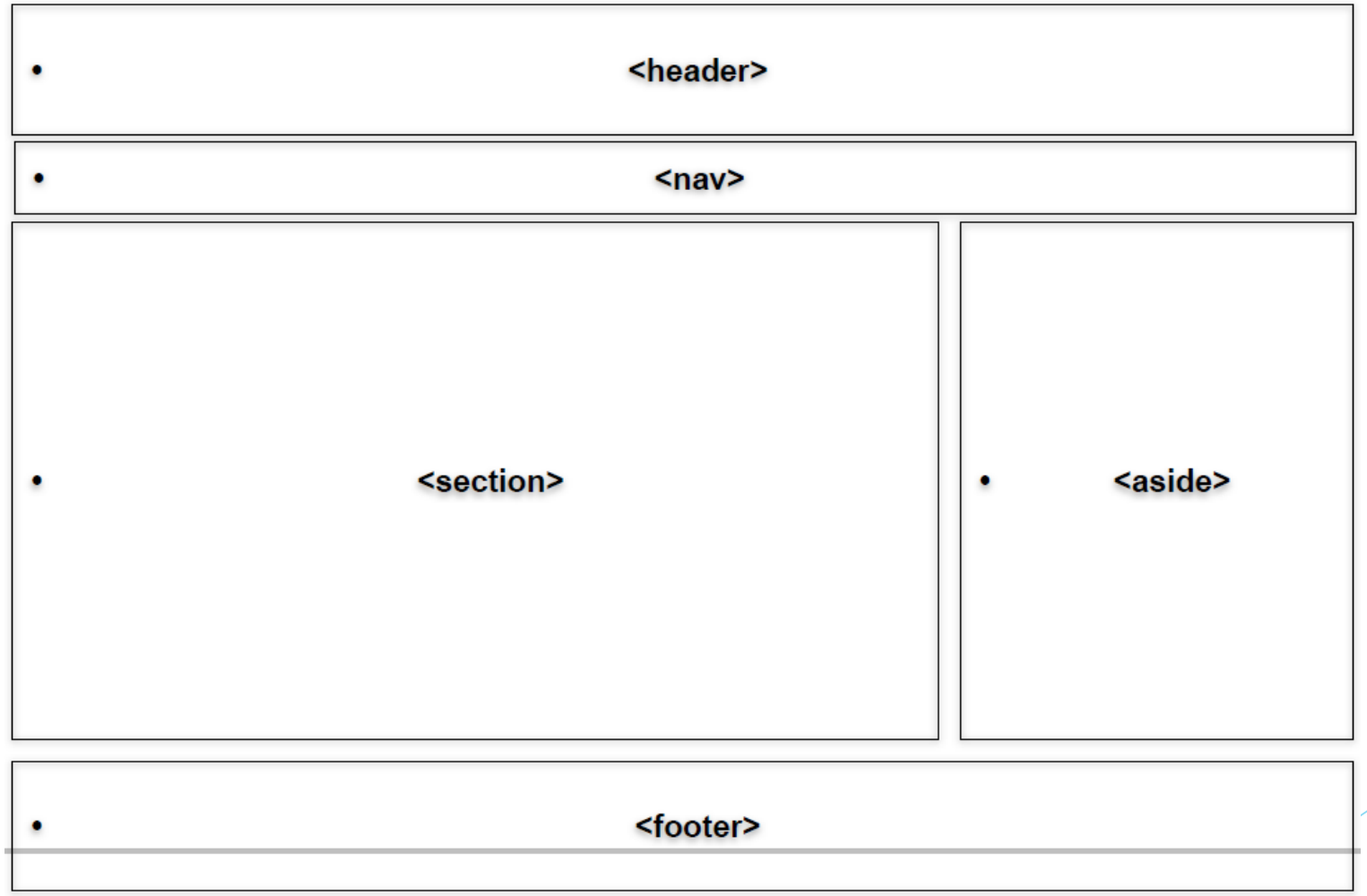
HTML5 – Semantic Tags

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
<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

Layout using



Layout using semantic tags

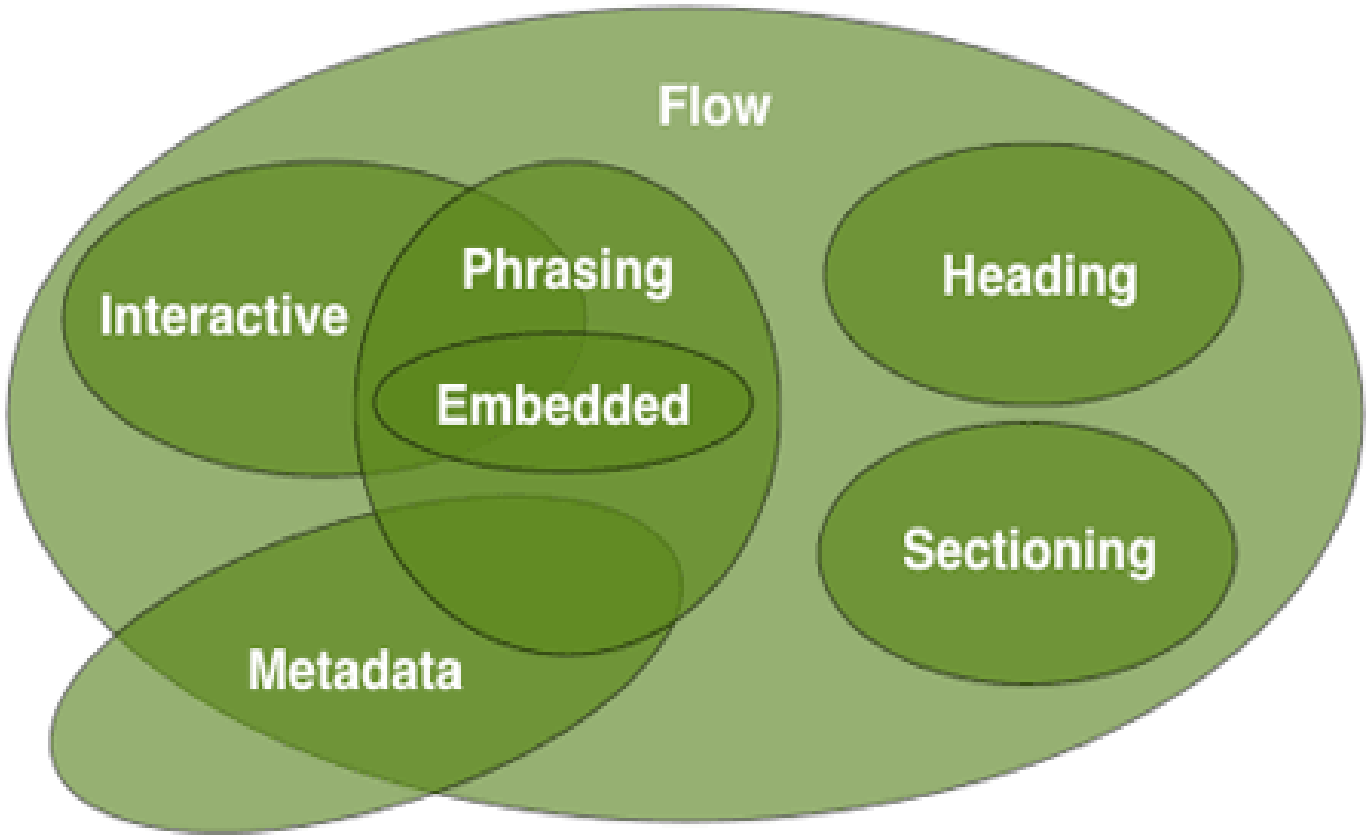


HTML5 Content Models

Content Model - Introduction

- According to HTML 4, content is always differentiated as block level or inline
- HTML 5 defines seven content model for proper structuring of the document
  - Metadata Content Model
  - Embedded Content Model
  - Interactive Content Model
  - Heading Content Model
  - Phrasing Content Model
  - Flow Content Model
  - Sectioning Content Model
- The names of the content model are self-explanatory to say what kind of content it can hold

HTML5 Content Model



Metadata Content

- It sets up the presentation of the document
- Metadata content also plays a major role in setting up relationship of the document with other document
- Tags that are part of this content model are:
  - base, command, link, meta, noscript, script, style and title

Flow Content

- Flow content model defines the body of the document
- Most of the semantic tags that are available in HTML 5 are part of flow content model
- Some tags that are part of flow content model are:
  - a, abbr, address, area, article, aside, embed, ….

Sectioning Content

- Sectioning content defines the scope of header and footer
- Tags that are part of sectioning content model are
  - article
  - aside
  - nav
  - section

Heading Content

- It defines the heading of a section
- The following tags are part of this content model
  - Headings h1 – h6
  - hgroup

Phrasing Content

- Phrasing content is the actual text of the document as well as elements that markup that text at the intra-paragraph level.

- Some tags that are part of this content model are
  - abbr,audion, button, canvas, cite, code, datalist, embed, iframe,…

Embedded Content

- Embedded content refers to resources that are imported into the document
- Resources like audio, video, SVG fall into this category
- Some tags that are part of this content model are
  - audio
  - video
  - svg
  - math

Interactive Content

- Contents that requires user interaction fall into this category
- Contents like JavaScript, Flash falls into this category

Tag Duplication

- Several tags are belong to two or more content models. This happens how and for what purpose they are using

- For example,
  - a video tag is in embedded content model.
  - If there have controls option at that time it will be Interactive content model also because user may control the video.
  - Also it will be phrasing when this included within a paragraph.

Working with Lists

HTML Lists

- The different types of available lists are:
  - Unordered List
  - Ordered List
  - Description List
  - Nested List
  - Horizontal List (discussed along with CSS later in CSS module)

Unordered List

- Syntax:

```
<ul type="disc">
    <li>Red</li>
    <li>Blue</li>
    <li>Green</li>
</ul>
```

Output

- Red
- Blue
- Green

- List Item Marker
  - Disc (default)
  - Circle
  - Square
  - None

Ordered List

- Syntax

```
<ol type="A">
    <li>Red</li>
    <li>Blue</li>
    <li>Green</li>
</ol>
```

Output

A. Red
B. Blue
C. Green

- The **type** attribute values
  - '1' : numbered with numbers
  - 'A' : numbered with uppercase letters
  - 'a' : numbered with lowercase letters
  - 'I' : numbered with uppercase roman numbers
  - 'i' : numbered with lowercase roman numbers

- The **start** attribute
  - To start the list numbering from somewhere

Description & Nested List

- **Description List**
  - A description list is a list of terms, with a description of each term.
  - The **<dl>** tag defines the description list, the **<dt>** tag defines the term (name), and the **<dd>** tag describes each term

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Coffee
- black hot drink

Milk
- white cold drink

- **Nested List**
  - List can be nested (lists inside lists)

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
  - Black tea
  - Green tea
- Milk

HTML5 Form

HTML Forms

- HTML Form allows user to send data to the web site. It is one of the main points of interaction between a user and a web site.
- An HTML Form is made up of one or more elements. These elements can be
  - Text field
  - Select boxes
  - Buttons
  - Checkboxes
  - Radio buttons
  - Drop Down
  - List Box
  - Submit button – used to submit the form and send data to server.
- Attributes of Form Tag
  - Action
    - The action attribute points to the server side script that handles form submission
  - Method (GET/POST)
    - The mechanism how the data will be sent to the server. In case of GET, the form values are passed as part of URL and in case of POST, the information is sent to the server as part of the data body

Form Elements

Input Element	Description	Example
Text Field	Input type text is used to create textbox that can be used to collect the name, email, phone number etc. from your web site visitors.	<input type="text" name="username" />
Password Field	characters in a password field are masked i.e. shown as asterisks or dots	<input type="password" name="user-password" id="user-pwd">
Radio Buttons	Radio buttons are used to let the user select exactly one option from a pre-defined set of options. It is created using an <input> element whose type attribute has a value of radio.	<input type="radio" name="sex" id="male"> <input type="radio" name="sex" id="female">
Checkboxes	Checkboxes allows the user to select one or more option from a pre-defined set of options. It is created using an <input> element whose type attribute has a value of checkbox.	<input type="checkbox" name="sports" id="soccer"> <input type="checkbox" name="sports" id="cricket">
Textarea	Textarea is a multiple-line text input control that allows a user to enter more than one line of text. Multi-line text input controls are created using an <textarea> element.	<textarea rows="3" cols="30" name="address" id="address">

Form Elements

Input Element	Description	Example
File Select box	The file fields allow a user to browse for a local file and send it as an attachment to the form data. It normally rendered as a text box with a button that enables the user to browse for a file.	<input type="file" name="upload" id="file-select">
Button	You can create buttons using the <button> element. Buttons created with the <button> element function just like buttons created with the input element, but they offer richer rendering possibilities.	<input type="button" value=”button”>
Submit Buttons	A submit button is used to send the form data to a web server. When submit button is clicked the form data is sent to the file specified in the form's action attribute to process the submitted data.	<input type="submit" value="Submit">
Reset Buttons	A reset button resets all the forms control to default values.	<input type="reset" value="Reset">

New input types in HTML5

Input Type	Description	Example
color	Allows the user to select a color from a drop-down color picker and returns the hex value for that color.	<input type="color" name="mycolor">
date	Allows the user to select a date from a drop-down calendar.	<input type="date" name="mydate">
datetime	Allows the user to select a date and time along with time zone.	<input type="datetime" name="mydatetime">
email	Allows the user to enter e-mail address. When used in combination with the required attribute, the browser may look for patterns to ensure a valid e-mail address should be entered.	<input type="email" name="myemail" required>
month	Allows the user to select a month and year from a drop-down calendar.	<input type="month" name="mymonth">
number	Can be used for entering a numerical value. You can also restrict the user to enter only acceptable values using the additional attributes min, max, and step.	<input type="number" value="1" min="1" max="10" step="0.5“ name="mynumber">

New input types in HTML5

Input Type	Description	Example
range	Can be used for entering a numerical value within a specified range.	<input type="range" value="1" min="1" max="10" step="0.5" name="mynumber">
search	Can be used for creating search fields.	<input type="search" name="mysearch">
tel	Can be used for entering a telephone number.	<input type="tel" name="mytelephone" required>
time	Can be used for entering a time.	<input type="time" name="mytime">
url	Can be used for entering web addresses i.e. URL's.	<input type="url" name="mywebsite" required>
Week	Allows the user to select a week and year from a drop-down calendar.	<input type="week" name="myweek">

Attributes added for HTML5 elements

- PlaceHolder
  - Placeholders are displayed when the input text box is empty. The text disappears when the focus is set to the control
  - Browsers that doesnot support the placeholder attribute will ignore it

```
<input type=”text” id=”firstName” placeholder=”First Name” />
```

- Autofocus
  - Autofocus is used to move the cursor automatically to an input element that user have to fill
  - Prior to HTML5, JavaScript was used to set the focus to an element automatically

```
<input type=”text” id=”firstName” placeholder=”First Name” autofocus />
```

HTML5 Inline Editing

Inline Editing - Introduction

- With evolution of Web 3.0, inline editing of web pages has become common feature in many web application like wiki, blog etc.
- HTML5 specification adds an inbuilt support for inline editing and spell checking
- Using HTML5 inline editing feature:
  - Visitors can be allowed to edit a particular part of the page
  - Visitors can be allowed to edit the whole page
  - Visitors can perform spellcheck

HTML5 Element – Inline Editing

- HTML5 introduces a global attribute name “contenteditable” which can be used with any element to make it editable
- HTML5 also provides a JavaScript API, using which the changes can be saved or a style can applied
- Inline editing for an article tag can be enabled as shown

```
<article class=“comment” contenteditable>
You Comment Goes Here
</article>
```

- HTML5 also allows enabling spell checking for a particular element or a document

```
<article class=“comment” contenteditable
spellcheck=“true”>
You Comment Goes Here
</article>
```

HTML5 Audio and Video

Adding Audio and Video

- With HTML4, the websites like YouTube are streaming video over the web.
- These video contents require one plugin or another to get rendered in a browser
- HTML5 specifications adds in-built support for audio and video.
- HTML5 specifies a standard way of adding videos to the web pages using the video tag and audio using audio tags

Supported Formats

- HTML5 specification has standardized three encoding for
  - video
    - MP4
    - OGG
    - WebM
  - Audio
    - MP3
    - OGG
    - Wav
- To make a audio/video compatible across browsers, the audio/video needs to be encoded in more than one format

Audio Tag

- Syntax

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

- How it works
  - **controls** attribute add audio controls like play, pause and volume
  - The **<source>** element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
  - The text between the <audio> and </audio> tags will be displayed in browsers that do not support the <audio> element

Video Tag

- Syntax

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

- How it works
  - The **controls** attribute adds video controls, like play, pause, and volume.
  - The **<source>** element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.
  - The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

Video Tag - Attributes

Attribute	Purpose
src	Defines the source of video file
poster	Takes any image file as argument. Fallback
width	Specify the width of the video
height	Specify the height of the video
preload	Option to enable or disable buffering (on page load)
autoplay	To enable or disable auto play on page load
controls	To show the controls

Example

```
<video width="420" src="mov_bbb.mp4" autoplay controls>
    Your browser does not support HTML5 video.
</video>
```

Video – JavaScript API

- HTML5 also has rich JavaScript API using which the video element can be managed

Method/Attribute	Description
canPlayType	To check whether the browser is capable of playing given video type
Play()	Starts playing the video
Pause()	Stops playing the video
Paused – attribute	Returns true/false based on whether the video is paused or playing