HTML

Sujata Batra

Overview

1. Advance version of HTML.
2. In 2008, the first HTML5 public draft was released
3. HTML5 W3C Final Recommendation was released 28. October 2014.
4. New elements, attributes, and behaviors were introduced.
5. It helps to create more powerful website and interactive web applications.
6. HTML5 comes with XML syntax.
7. HTML5 is to compete with Flash and Silverlight.
8. Empowering Mobile devices.

Sujata Batra

Technical Advantages Over Previous Version.

1. Audio and Videos are integral part of HTML5 specifications e.g. <audio> and<video> tags.
2. Vector graphics is integral part of HTML5 e.g. SVG and canvas.
3. JS GeoLocation API in HTML5 helps identify location of user browsing any website (provided user allows it).
4. Full duplex communication channels can be established with Server using Web Sockets.
5. Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5.
6. Application Cache, Web SQL database and Web storage is available as client side storage.
7. Retain Backward Compatibility with previous versions of HTML5.

HTML5 Technology Functions

Sujata Batra

Semantics: allowing you to describe more precisely what your content is.

Connectivity: allowing you to communicate with the server in new and innovative ways.

Offline & Storage: allowing webpages to store data on the client-side locally and operate offline more efficiently.

Multimedia: making video and audio first-class citizens in the Open Web.

2D/3D Graphics & Effects: allowing a much more diverse range of presentation options.

Performance & Integration: providing greater speed optimization and better usage of computer hardware.

Device Access: allowing for the usage of various input and output devices.

Styling: letting authors write more sophisticated themes.

Elements removed in HTML5

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

Sujata Batra

HTML5 New Tags and Elements

HTML5 Indroduces 28 New Elements, Some of them are mentioned here.

Navigation:

<article>
<aside>
<header>
<hgroup>
<footer>
<figure>
<figcaption>
<nav>
<section>

Multimedia/Interactivity:

<audio>
<canvas>
<embed>
<source>
<track>
<video>

New <input> types:

color
date
datetime
datetime-local
email
month
number
range
search
tel
time
url
week

Miscellaneous:

<bdi>
<command>
<datalist>
<details>
<mark>
<meter>
<output>
<progress>
<summary>
<rp>
<rt>
<ruby>
<time>
<wbr>

Suja ta Batra

Defining HTML5 Documents

Remember the DOCTYPE declaration-

```
<!DOCTYPE html>
```

Again, HTML5 simplifies this line:

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

The default character encoding (charset) declaration

```
<meta charset="UTF-8">
```

Semantic Elements

A semantic element clearly describes its meaning to both the browser and the developer.

Eg. of non-semantic elements: ** ** and **** - Tells nothing about its content.

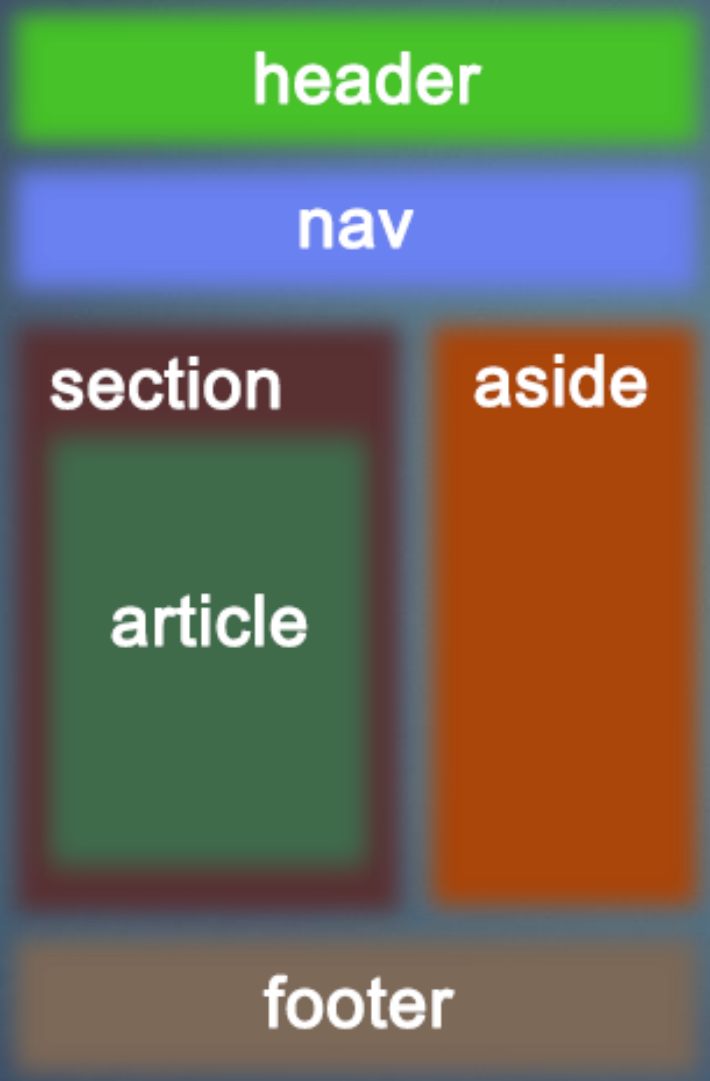
Eg. of semantic elements: **<form>, <table>,** and **<img>** - Clearly defines its content.

Many web sites contain HTML code like: ** ** to indicate navigation, header, and footer.

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

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

Semantic Elements

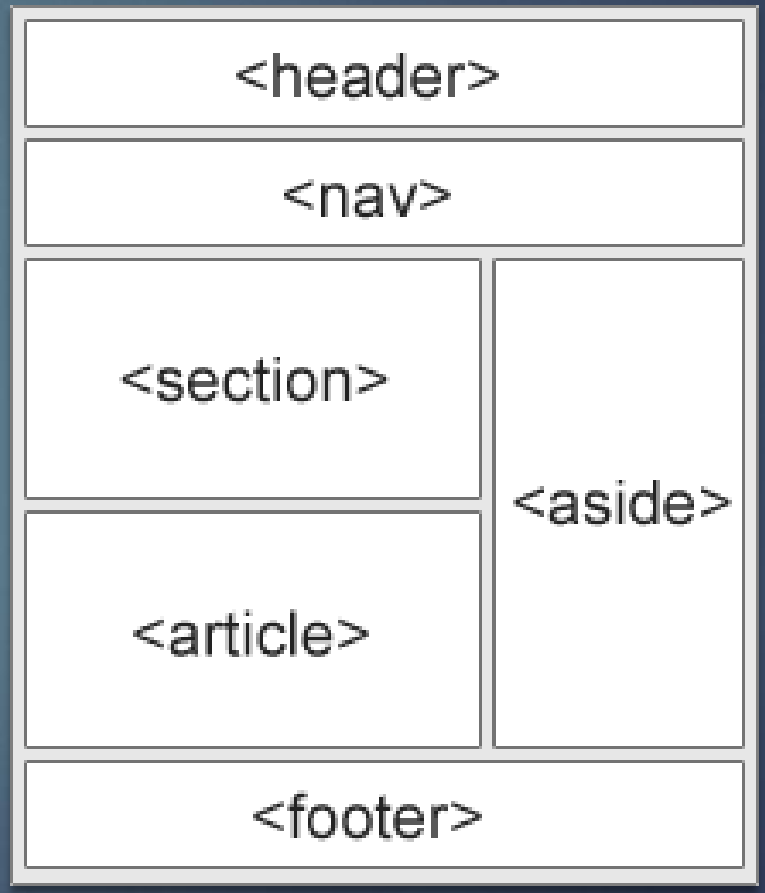



```
<header>
    <h1>title</h1>
    <form>Search</form>
    <nav>
        <ul>Site navigation</ul>
    </nav>
</header>
<section>
    <article>
        <h1>Article title</h1>
        <p>Summary</p>
    </article>
</section>
<aside>
    <section>
        <h2>Blogroll…</h2>
    </section>
</aside>
<footer>
    <h2>Footer</h2>
</footer>
```

Sujata Batra

New 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
<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



Sujata Batra

HTML5 <section> Element

- A section is a thematic grouping of content, typically with a heading.
- A home page could normally be split into sections for introduction, content, and contact information.

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

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.

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 may have several <footer> elements in one document.

HTML5 <nav> Element

- The <nav> element defines a set of navigation links.
- NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

HTML5 <aside> Element

- The <aside> element defines some content aside from the content it is placed in (like a sidebar).

- The <aside> content should be related to the surrounding content.

HTML5 <figure> and <figcaption> Elements

- The purpose of a figure caption is to add a visual explanation to an image.
- In HTML5, an image and a caption can be grouped together in a <figure> element:

- **Example:**

```
<figure>
  <img src="pic_trulli.jpg" alt="Trulli">
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>
</figure>
```

HTML <summary> Tag

- The <summary> tag defines a visible heading for the <details> element. The heading can be clicked to view/hide the details.
- Browser Support

Element	Chrome	Internet Explorer	Mozilla Firefox	Apple Safari	Opera
<summary>	12.0	Not Supported	49.0	6.0	15.0

HTML <details> Tag

- The <details> tag can be used to create an interactive widget that the user can open and close.
- The content of a <details> element should not be visible unless the open attribute is set.
- Browser Support

Element	Chrome	Internet Explorer	Mozilla Firefox	Apple Safari	Opera
<details>	12.0	Not Supported	49.0	6.0	15.0

Example


```
<details>
  <summary>Copyright 1999-2018.</summary>
  <p> - by Refsnes Data. All Rights Reserved.</p>
  <p>All content and graphics on this web site are the property of the
company Refsnes Data.</p>
</details>
```

HTML <main> Tag

- The <main> tag specifies the main content of a document.
- The content inside the <main> element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.
- Note: There must not be more than one <main> element in a document. The <main> element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.

Element	Chrome	Internet Explorer	Mozilla Firefox	Apple Safari	Opera
<main>	6.0	12.0	4.0	5.0	11.1

HTML <mark> Tag

- Use the <mark> tag if you want to highlight parts of your text.

Element	Chrome	Internet Explorer	Mozilla Firefox	Apple Safari	Opera
<mark>	6.0	9.0	4.0	5.0	11.1

HTML <time> Tag

Sujata Batra

- The <time> tag defines a human-readable date/time.
- This element can also be used to encode dates and times in a machine-readable way so that user agents can offer to add birthday reminders or scheduled events to the user's calendar, and search engines can produce smarter search results.

Element	Chrome	Internet Explorer	Mozilla Firefox	Apple Safari	Opera
<time>	6.0	9.0	4.0	5.0	11.1

- Attribute

Attribute	Value	Description
datetime	*datetime*	Represent a machine-readable date/time of the <time> element

Migration from HTML4 to HTML5

HTML4	HTML5
	<header>
	<nav>
	<section>
	<article>
	<footer>

Sujata Batra

HTML 5 Form Elements

HTML5 Form Elements

- HTML5 added the following form elements:

- <datalist>
- <output>

HTML5 <datalist> Element

- <datalist> element specifies a list of pre-defined options for an <input> element.
- Users will see a drop-down list of the pre-defined options as they input data.
- The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.
- <form action="/action_page">
  <input **list="browsers"**>
  <datalist **id="browsers"**>
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
  </datalist>
  </form>

HTML5 <output> Element

- <output> element represents the result of a calculation (like one performed by a script).

```
function showResult() {
x = document.forms["myform"]["newinput"].value;
document.forms["myform"]["result"].value = x;
  }
<form action = "/cgi-bin/html5.cgi" method = "get" name = "myform">
 Enter a value : <input type = "text" name = "newinput" />
<input type = "button" value = "Result"  onclick = "showResult();" />
<output name = "result"></output>
</form>
```

HTML 5 Input Types

HTML5 Input Types

- HTML5 added several new input types:
- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Note : New input types that are not supported by older web browsers, will behave as <input type="text">.

- **Input Type Color**
  - The <input type="color"> is used for input fields that should contain a color.
  - Depending on browser support, a color picker can show up in the input field.

Example : <input type="color" name="favcolor" value="#ff0000">

- **Input Type Date**
  - The <input type="date"> is used for input fields that should contain a date.
  - You can also use the min and max attributes to add restrictions to dates.
  - Depending on browser support, a date picker can show up in the input field.

Example : Enter a date before 1980-01-01:
<input type="date" name="bday" max="1979-12-31">
Enter a date after 2000-01-01:
<input type="date" name="bday" min="2000-01-02">

- **Input Type Datetime-local**
  - The <input type="datetime-local"> specifies a date and time input field, with no time zone.
  - Depending on browser support, a date picker can show up in the input field.

Example : <input type="datetime-local" name="bdaytime">

- **Input Type Email**
  - The <input type="email"> is used for input fields that should contain an e-mail address.
  - Depending on browser support, the e-mail address can be automatically validated when submitted.

- **Input Type File**
  - The <input type="file"> defines a file-select field and a "Browse" button for file uploads.

- **Input Type Month**
  - The <input type="month"> allows the user to select a month and year.
  - Depending on browser support, a date picker can show up in the input field.

- **Input Type Number**
  - The <input type="number"> defines a numeric input field.
  - You can also set restrictions on what numbers are accepted.

Example: <input type="number" name="quantity" min="1" max="5">

HTML 5 Input Attributes

HTML5 added the following attributes for <input>:

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

and the following attributes for <form>:

- autocomplete
- novalidate

- **The autocomplete Attribute**
  - The autocomplete attribute specifies whether a form or input field should have autocomplete on or off.
  - When autocomplete is on, the browser automatically completes the input values based on values that the user has entered before.
  - It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.
  - The autocomplete attribute works with <form> and the following <input> types: text, search, url, tel, email, password, datepickers, range, and color.

- **The novalidate Attribute**
  - The novalidate attribute is a <form> attribute.
  - When present, novalidate specifies that the form data should not be validated when submitted.
- Example :
```
<form action="/action_page" novalidate>
  E-mail: <input type="email" name="user_email">
  <input type="submit">
</form>
```

- **The autofocus Attribute**
  - The autofocus attribute specifies that the input field should automatically get focus when the page loads.

- **The form Attribute**
  - The form attribute specifies one or more forms an <input> element belongs to.
  - To refer to more than one form, use a space-separated list of form ids.
- Example :<form action="/action_page.php" id="form1">
  First name: <input type="text" name="fname">
  <input type="submit" value="Submit">
  </form>

  Last name: <input type="text" name="lname" form="form1">

- **The formaction Attribute**
  - The formaction attribute specifies the URL of a file that will process the input control when the form is submitted.
  - The formaction attribute overrides the action attribute of the <form> element.
  - The formaction attribute is used with type="submit" and type="image".

Example :<form action="/action_page.php">
First name: <input type="text" name="fname">
Last name: <input type="text" name="lname">
<input type="submit" value="Submit">
<input type="submit" formaction="/action_page2.php"
value="Submit as admin">
</form>

- **The formmethod Attribute**
  - The formmethod attribute defines the HTTP method for sending form-data to the action URL.
  - The formmethod attribute overrides the method attribute of the <form> element.
  - The formmethod attribute can be used with type="submit" and type="image".

- **The formnovalidate Attribute**
  - The formnovalidate attribute overrides the novalidate attribute of the <form> element.
  - The formnovalidate attribute can be used with type="submit".
- Example : <form action="/action_page.php">
  E-mail: <input type="email" name="userid">
  <input type="submit" value="Submit">
  <input type="submit" formnovalidate value="Submit without validation">
  </form>

- **The height and width Attributes**
  - The height and width attributes specify the height and width of an <input type="image"> element.
  - Always specify the size of images. If the browser does not know the size, the page will flicker while images load.

- **The min and max Attributes**
  - The min and max attributes specify the minimum and maximum values for an <input> element.
  - The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

- **The multiple Attribute**
  - The multiple attribute specifies that the user is allowed to enter more than one value in the <input> element.
  - The multiple attribute works with the following input types: email, and file.

- **The pattern Attribute**
  - The pattern attribute specifies a regular expression that the <input> element's value is checked against.
  - The pattern attribute works with the following input types: text, search, url, tel, email, and password.
  - Use the global title attribute to describe the pattern to help the user.
- Example : Country code: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">

- **The placeholder Attribute**
  - The placeholder attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format).
  - The hint is displayed in the input field before the user enters a value.
  - The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

- **The required Attribute**
  - The required attribute specifies that an input field must be filled out before submitting the form.
  - The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

- **The step Attribute**
  - The step attribute specifies the legal number intervals for an <input> element.

Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

  - The step attribute can be used together with the max and min attributes to create a range of legal values.
  - The step attribute works with the following input types: number, range, date, datetime-local, month, time and week.

HTML 5 Media

HTML Audio Tag

- HTML audio tag is used to define sounds such as music and other audio clips.
- Currently there are three supported file format for HTML 5 audio tag.
  - mp3
  - wav
  - ogg

Element	Chrome	Internet Explorer	Mozilla Firefox	Apple Safari	Opera
<audio>	4.0	9.0	3.5	4.0	10.5

Browser Support Audio File Format

Sujata Batra

Browser	mp3	wav	ogg
Internet Explorer	yes	no	no
Google Chrome	yes	yes	yes
Mozilla Firefox	yes	yes	yes
Opera	no	yes	yes
Apple Safari	yes	yes	no

Attributes of HTML Audio Tag

Attribute	Description
controls	It defines the audio controls which is displayed with play/pause buttons.
autoplay	It specifies that the audio will start playing as soon as it is ready.
loop	It specifies that the audio file will start over again, every time when it is completed.
muted	It is used to mute the audio output.
preload	It specifies the author view to upload audio file when the page loads.
src	It specifies the source URL of the audio file.

Audio Tag Example


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

HTML Video Tag

- The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.
- Three video formats supported for HTML video tag:
  - mp4
  - webM
  - ogg

Element	Chrome	Internet Explorer	Mozilla Firefox	Apple Safari	Opera
<video>	4.0	9.0	3.5	4.0	10.5

Browser Support Video File Format

Browser	mp4	webM	ogg
Internet Explorer	yes	no	no
Google Chrome	yes	yes	yes
Mozilla Firefox	yes	yes	yes
Opera	no	yes	yes
Apple Safari	yes	no	no

Attributes of HTML Video Tag

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	pixels	Sets the height of the video player
loop	loop	Specifies that the video will start over again, every time it is finished
muted	muted	Specifies that the audio output of the video should be muted
poster	URL	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
src	URL	Specifies the URL of the video file
width	pixels	Sets the width of the video player

Sujata Batra

Video Tag Example


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

HTML5 Plug-ins

To extend the functionality of the HTML browser. Plug-ins are also known as Helper Applications. Popular example of plug-ins are Java applets. Plug-ins can be added to web pages with the **<object>** tag or the **<embed>** tag.

1. **<object> Element –** It is used to embed plug-ins (like Java applets, PDF readers, Flash Players) in web pages.

   Eg.: <object width="200" height="60" data="SujataTraining.swf"></object>

2. **<embed> Element –** Used to embed object within an HTML document. It does not have closing tag. It can not contain alternative text.

   Eg.: <embed width="200" height="60" src="SujataTraining.swf">

Playing a YouTube Video in HTML

- Upload the video to YouTube
- Take a note of the video id
- Define an <iframe> element in your web page
- Let the src attribute point to the video URL
- Use the width and height attributes to specify the dimension of the player.

Example:

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY">
</iframe>
```

Note: YouTube <object> and <embed> were deprecated from January 2015. You should migrate your videos to use <iframe> instead.

YouTube Autoplay


```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1">
</iframe>
```

- Value 0 (default): The video will not play automatically when the player loads.
- Value 1: The video will play automatically when the player loads.

YouTube Controls


```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=0">
</iframe>
```

- Value 0: Player controls does not display.
- Value 1 (default): Player controls display.