Courtesy to w3schools.com

By The Name of Allah the Most Gracious the Most Merciful

Learning from Scratch (Review): HTML

(It is assumed that you are familiar with HTML. Then, you may know many of these, so am I. But, I used to forget 'the right thing at the right time'. That's why I'm just taking note…)

---Muhammad Mushfiqur Rahman

Contents

SOME BORING THEORIES (BASIC THINGS TO REMEMBER)....3

WHAT IS HTML?.... 3
HTML DOCUMENTS = WEB PAGES.... 3
HTML TAGS.... 3
HTML ELEMENTS.... 3
HTML ATTRIBUTES.... 4
Web Browsers.... 4

THE WORLD WIDE WEB CONSORTIUM (W3C) RECOMMENDATIONS....4

HTML 4.01 / XHTML 1.0 TAG REFERENCE....5

ORDERED BY FUNCTION.... 5
Basic.... 5
Formatting.... 5
Forms.... 6
Frames.... 6
Images.... 6
Links.... 6
Lists.... 6
Tables.... 7
Styles.... 7
Meta Info.... 7
Programming.... 7

HTML / XHTML STANDARD ATTRIBUTES....8

CORE ATTRIBUTES.... 8
LANGUAGE ATTRIBUTES.... 8
KEYBOARD ATTRIBUTES.... 8
*STANDARD EVENT ATTRIBUTES.... 8
<body> and <frameset> Events.... 8
Form Events.... 9
Image Events.... 9
Keyboard Events.... 9
Mouse Events.... 9

HTML/XHTML ELEMENTS AND VALID DTDS....9

A LITTLE EXPLANATION ABOUT SOME IMPORTANT ELEMENTS....10

HEADINGS ARE IMPORTANT.... 10
HTML FORMATTING TAGS.... 10
Definition and Usage.... 10
HTML FORM TAGS.... 11
HTML TABLES.... 12
HTML BLOCK OR BLOCK LEVEL ELEMENTS.... 13
The HTML <div> Element.... 13
HTML INLINE ELEMENTS.... 13
The HTML <span> Element.... 13
WEBSITE LAYOUTS.... 13
HTML Layout - Useful Tips.... 14

Some boring Theories (Basic Things to Remember)

But, important for any Quiz or Interview!

What is HTML?

HTML is a language for describing web pages.

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- It is not a programming language, it is a **markup language**
- A markup language is a set of **markup tags**
- The purpose of the tags are to **describe page content**

HTML Documents = Web Pages

- HTML documents **describe web pages**
- HTML documents **contain HTML tags** and plain text
- HTML documents are also **called web pages**

HTML Tags

HTML markup tags are usually called HTML tags

- HTML **tags are keywords** (tag names)
- surrounded by **angle brackets** like <html>
- normally **come in pairs** like <b> and </b>
- named as **start tag** or **opening tag &** the **end tag** or **closing tag**

HTML Elements

"HTML tags" and "HTML elements" are often used to describe the same thing.

But strictly speaking, an HTML **element is everything between the start tag and the end tag, including the tags.**

HTML Element: <p>This is a paragraph.</p>

Description:

Start tag *	Element content	End tag *
<p>	This is a paragraph	</p>
<a href="default.htm">	This is a link	</a>
<br />		

- Most HTML elements **can be nested** (can contain other HTML elements).
- Some HTML elements have **empty content**
- Empty elements are **closed in the start tag***
- Most HTML elements **can have attributes**

HTML Attributes

- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in name/value pairs like: **name="value"**

Example:

HTML links are defined with the <a> tag. The link address is specified in the **href attribute**:

```
<a href="http://www.w3schools.com">This is a link</a>
```

Web Browsers

The purpose of a web browser (Chrome, Internet Explorer, Firefox) is to parse (read) HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

The World Wide Web Consortium (W3C) recommendations

1. All elements must be closed.*

Tip-1: Adding a slash inside the start tag, like <br />, is the proper way of closing empty elements in XHTML (and XML)

2. Use Lowercase Tags

Tip-2: HTML tags are not case sensitive: <P> means the same as <p>. Many web sites use uppercase HTML tags.

W3Schools use lowercase tags because the World Wide Web Consortium (W3C) **recommends** lowercase in HTML 4, and **demands** lowercase tags in XHTML.

3. Always Quote Attribute Values

Tip-3: Double style quotes are the most common, but single style quotes are also allowed.

💡 In some rare situations, when the attribute value itself contains quotes, it is necessary to use single quotes: name='John "ShotGun" Nelson'

4. Use Lowercase Attributes

Tip-4: Attribute names and attribute values are case-insensitive.

However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

Newer versions of (X)HTML will demand lowercase attributes.

HTML 4.01 / XHTML 1.0 Tag Reference

Ordered by Function

DTD: indicates in which HTML 4.01 / XHTML 1.0 DTD the tag is allowed. S=Strict, T=Transitional, and F=Frameset

Tag	Description	DTD
Basic		
<!DOCTYPE>	Defines the document type	STF
<html>	Defines an HTML document	STF
<body>	Defines the document's body	STF
<h1> to <h6>	Defines HTML headings	STF
<p>	Defines a paragraph	STF
<br />	Inserts a single line break	STF
<hr />	Defines a horizontal line	STF
<!--...-->	Defines a comment	STF
Formatting		
<acronym>	Defines an acronym	STF
<abbr>	Defines an abbreviation	STF
<address>	Defines contact information for the author/owner of a document	STF
<b>	Defines bold text	STF
<bdo>	Overrides the current text direction	STF
<big>	Defines big text	STF
<blockquote>	Defines a long quotation	STF
<center>	Deprecated. Defines centered text	TF
<cite>	Defines a citation	STF
<code>	Defines a piece of computer code	STF
<del>	Defines text that has been deleted from a document	STF
<dfn>	Defines a definition term	STF
<em>	Defines emphasized text	STF
<font>	Deprecated. Defines font, color, and size for text	TF
<i>	Defines italic text	STF
<ins>	Defines text that has been inserted into a document	STF
<kbd>	Defines keyboard input	STF
<pre>	Defines preformatted text	STF
<q>	Defines a short quotation	STF
<s>	Deprecated. Defines strikethrough text	TF
<samp>	Defines sample output from a computer program	STF

<small>	Defines smaller text	STF
<strike>	Deprecated. Defines strikethrough text	TF
<strong>	Defines strong text	STF
<sub>	Defines subscripted text	STF
<sup>	Defines superscripted text	STF
<tt>	Defines teletype text	STF
<u>	Deprecated. Defines underlined text	TF
<var>	Defines a variable	STF
<xmp>	Deprecated. Defines preformatted text	

Forms

<form>	Defines an HTML form for user input	STF
<input />	Defines an input control	STF
<textarea>	Defines a multiline input control (text area)	STF
<button>	Defines a clickable button	STF
<select>	Defines a drop-down list	STF
<optgroup>	Defines a group of related options in a drop-down list	STF
<option>	Defines an option in a drop-down list	STF
<label>	Defines a label for an <input> element	STF
<fieldset>	Groups related elements in a form	STF
<legend>	Defines a caption for a <fieldset> element	STF

Frames

<frame />	Defines a window (a frame) in a frameset	F
<frameset>	Defines a set of frames	F
<noframes>	Defines an alternate content for users that do not support frames	TF
<iframe>	Defines an inline frame	TF

Images

<img />	Defines an image	STF
<map>	Defines an image-map	STF
<area />	Defines an area inside an image-map	STF

Links

<a>	Defines an anchor	STF
<link />	Defines the relationship between a document and an external resource	STF

Lists

<ul>	Defines an unordered list	STF
<ol>	Defines an ordered list	STF

<li>	Defines a list item	STF
<dir>	Deprecated. Defines a directory list	TF
<dl>	Defines a definition list	STF
<dt>	Defines an item in a definition list	STF
<dd>	Defines a description of an item in a definition list	STF
<menu>	Deprecated. Defines a menu list	TF

Tables

<table>	Defines a table	STF
<caption>	Defines a table caption	STF
<th>	Defines a header cell in a table	STF
<tr>	Defines a row in a table	STF
<td>	Defines a cell in a table	STF
<thead>	Groups the header content in a table	STF
<tbody>	Groups the body content in a table	STF
<tfoot>	Groups the footer content in a table	STF
<col />	Defines attribute values for one or more columns in a table	STF
<colgroup>	Defines a group of columns in a table for formatting	STF

Styles

<style>	Defines style information for a document	STF
<div>	Defines a section in a document	STF
<span>	Defines a section in a document	STF

Meta Info

<head>	Defines information about the document	STF
<title>	Defines the document title	STF
<meta>	Defines metadata about an HTML document	STF
<base />	Specifies the base URL/target for all relative URLs in a document	STF
<basefont />	Deprecated. Specifies a default color, size, or font for all the text in a document	TF

Programming

<script>	Defines a client-side script	STF
<noscript>	Defines an alternate content for users that do not support client-side scripts	STF
<applet>	Deprecated. Defines an embedded applet	TF
<object>	Defines an embedded object	STF
<param />	Defines a parameter for an object	STF

HTML / XHTML Standard Attributes

The attributes listed below are standard, and can be used in nearly all HTML/XHTML tags.

Core Attributes

Valid in all elements, except: <base>, <head>, <html>, <meta>, <param>, <script>, <style>, and <title>.

Attribute	Description
class	Specifies one or more classnames for an element (refers to a class in a style sheet)
id	Specifies a unique id for an element
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element

Language Attributes

Valid in all elements, except: <base>, <br>, <frame>, <frameset>, <hr>, <iframe>, <param>, and <script>.

Attribute	Description
dir	Specifies the text direction for the content in an element
lang	Specifies the language of the element's content
xml:lang	Specifies the language of the element's content (for XHTML documents)

Keyboard Attributes

Attribute	Description
accesskey	Specifies a shortcut key to activate/focus an element
tabindex	Specifies the tabbing order of an element

*Standard Event Attributes

HTML 4 added the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element.
To learn more about programming events, please visit our JavaScript tutorial and our DHTML tutorial.
Below is the standard event attributes that can be inserted into HTML / XHTML elements to define event actions.

<body> and <frameset> Events

The two attributes below can only be used in <body> or <frameset>:

Attribute	Value	Description
onload	*script*	Script to be run when a document load
onunload	*script*	Script to be run when a document unload

Form Events

The attributes below can be used in form elements:

Attribute	Value	Description
onblur	*script*	Script to be run when an element loses focus
onchange	*script*	Script to be run when an element changes
onfocus	*script*	Script to be run when an element gets focus
onreset	*script*	Script to be run when a form is reset
onselect	*script*	Script to be run when an element is selected
onsubmit	*script*	Script to be run when a form is submitted

Image Events

The attribute below can be used with the img element:

Attribute	Value	Description
onabort	*script*	Script to be run when loading of an image is interrupted

Keyboard Events

Valid in all elements except base, bdo, br, frame, frameset, head, html, iframe, meta, param, script, style, and title.

Attribute	Value	Description
onkeydown	*script*	Script to be run when a key is pressed
onkeypress	*script*	Script to be run when a key is pressed and released
onkeyup	*script*	Script to be run when a key is released

Mouse Events

Valid in all elements except base, bdo, br, frame, frameset, head, html, iframe, meta, param, script, style, and title.

Attribute	Value	Description
onclick	*script*	Script to be run on a mouse click
ondblclick	*script*	Script to be run on a mouse double-click
onmousedown	*script*	Script to be run when mouse button is pressed
onmousemove	*script*	Script to be run when mouse pointer moves
onmouseout	*script*	Script to be run when mouse pointer moves out of an element
onmouseover	*script*	Script to be run when mouse pointer moves over an element
onmouseup	*script*	Script to be run when mouse button is released

HTML/XHTML Elements and Valid DTDs

THIS LINK shows the lists all HTML/XHTML elements, and shows what Doctype (DTD) each element appears in.

A Little explanation about some IMPORTANT ELEMENTS

Headings Are Important

- ✓ Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.
- ✓ **Search engines use your headings to index** the structure and content of your web pages.
- ✓ Since users may skim your pages by its headings, it is important to use headings to show the document structure.
- ✓ H1 headings should be used as main headings, followed by H2 headings, then the less important H3 headings, and so on.

HTML Formatting Tags

HTML uses tags like <b> and <i> for formatting output, like **bold** or *italic* text.
These HTML tags are called formatting tags (look at the bottom of this page for a complete reference).

Often <strong> renders as <b>, and <em> renders as <i>.
However, there is a difference in the meaning of these tags:
<b> or <i> defines bold or italic text only.
<strong> or <em> means that you want the text to be rendered in a way that the user understands as "important". Today, all major browsers render strong as bold and em as italics. However, if a browser one day wants to make a text highlighted with the strong feature, it might be cursive for example and not bold!

Definition and Usage

The <tt>, <i>, <b>, <big>, and <small> tags are all font-style tags. Font-style tags are defined in HTML4, but it is strongly recommended to use CSS styling instead.
Note: <tt> and <big> are removed from the HTML5 version. Avoid using them.

On the other hand, The <em>, <strong>, <dfn>, <code>, <samp>, <kbd>, <var>, and <cite> tags are all phrase tags. They are not deprecated, but it is possible to achieve richer effect with CSS.

Tag	**Description**
<em>	Renders as emphasized text
<strong>	Renders as strong (highlighted) text
<dfn>	Defines a definition term
<code>	Defines a piece of computer code
<samp>	Defines sample output from a computer program
<kbd>	Defines keyboard input
<var>	Defines a variable part of a text
<cite>	Defines a citation

HTML Form Tags

The differences between

1. `<input type='submit'/>`
2. `<input type='button' />` and
3. `<button type="submit">…</button>`
4. `<button type="button">…</button>` :

- ✓ The ques is wired, it should be either dif between 'tags' <input/>and<button> or 'attribute values 'submit' and 'button'
- ➢ `<input type="submit">` buttons will submit the form they are in when the user clicks on them, unless you specifiy otherwise with JavaScript.

  `<input type="button" />` buttons will not submit a form - they don't do anything by default. They're generally used in conjunction with JavaScript as part of an AJAX application. To be used like: <input type="button" value="Click me" onclick="msg()" />

Attribute Values

Value	**Description**
submit	Defines a submit button. A submit button sends form data to a server
button	Defines a clickable button (mostly used with a JavaScript to activate a script)
image	Defines an image as a submit button <input type="image" src="submit.gif" alt="Submit" />
reset	Defines a reset button. A reset button resets all form fields to their initial values
text	Defines a one-line input field that a user can enter text into. Default width is 20 characters
password	Defines a password field. The characters in this field are masked
file	Defines an input field and a "Browse..." button, for file uploads
hidden	Defines a hidden input field
checkbox	Defines a checkbox
radio	Defines a radio button

> Buttons created with the BUTTON element function just like buttons created with the INPUT element, but they offer richer rendering possibilities: the BUTTON element may have content. For example, a BUTTON element that contains an image functions like and may resemble an INPUT element whose type is set to "image", but the BUTTON element type allows content.
>
> *The Button Element - W3C*

```
<button type="submit">(html content)</button>
```

IE will submit all text for this button between the tags, other browsers will only submit the value. Using`<button>` gives you more layout freedom over the design of the button. In all its intends and purposes, it seemed excellent at first, but the browser quirks make it hard to use at times.

In the example, IE will send `text` to the server, while most other browsers will send nothing. To make it cross-browser compatible, use

```
<button type="submit" value="text">text</button>
```

Better yet: don't use the value, because if you add HTML it becomes rather tricky what is received on server side. Instead, if you must send an extra value, use a hidden field.

```
<input type="button" />
```

By default, this does next to nothing. It will not even submit your form. You can only place text on the button, give it a size and a border by means of CSS. It's original (and current) intend was to execute a script without the need to submit the form to the server.

```
<input type="submit" />
```

Like the former, but actually submits the surrounding form.

```
<input type="image" />
```

Like the former (submit), it will also submit a form, but you can use any image. This used to be the preferred way to use images as buttons when a form needed submitting.
For more control, `<button>` is now used. This can also be used for *server side image maps* but that's a rarity these days. You use use the usemap-attribute and (with or without that attribute), the browser will send the coordinates of you clicking the image to the server. If you just ignore these extras, it is nothing more than a submit button.
There are some subtle differences between browsers, but all will submit the value-attribute, unless for the `<button>` tag as explained above.

On contrary: A bit in favour of <button>

When using `<button>` always specify the type, since browsers default to different types.
This will work consistently across all browser:

- `<button type="submit">...</button>`
- `<button type="button">...</button>`

This way you gain all of `<button>`'s goodness, no downsides

HTML Tables

With border="10" cellspacing="10" cellpadding="10"

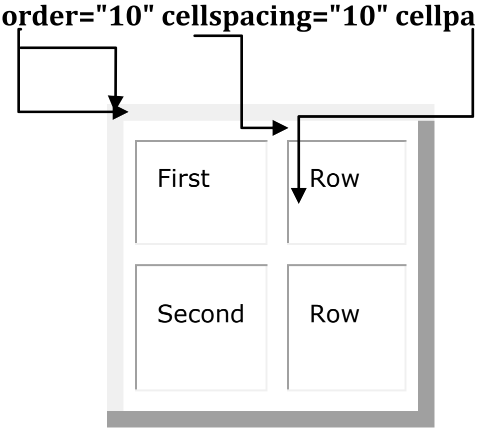

Table cells that span more than one row or one column- e.g. colspan=2, rowspan=2

Cell that spans two columns:

Name	Telephone	
Bill Gates	555 77 854	555 77 855

Cell that spans two rows:

First Name:	Bill Gates
Telephone:	555 77 854
	555 77 855

Most HTML elements are defined as **block level** elements or as **inline** elements.

HTML Block or Block Level Elements

Block level elements normally start (and end) with a new line when displayed in a browser.

Examples: <h1>, <p>, <ul>, <table>

The HTML <div> Element

The HTML <div> element is a block level element that can be used as a container for grouping other HTML elements.

The <div> element has no special meaning. Except that, because it is a block level element, the browser will display a line break before and after it.

When used together with CSS, the <div> element can be used to set style attributes to large blocks of content.

Another common use of the <div> element, is for document layout. It replaces the "old way" of defining layout using tables. Using tables is not the correct use of the <table> element. The purpose of the <table> element is to display tabular data.

HTML Inline Elements

Inline elements are normally displayed without starting a new line.

Examples: <b>, <td>, <a>, <img>

The HTML <span> Element

The HTML <span> element is an inline element that can be used as a container for text.

The <span> element has no special meaning.

When used together with CSS, the <span> element can be used to set style attributes to parts of the text.

Website Layouts

Most websites have put their content in multiple columns (formatted like a magazine or newspaper).

Multiple columns are created by using <div> or <table> elements. CSS are used to position elements, or to create backgrounds or colorful look for the pages.

Even though it is possible to create nice layouts with HTML tables, tables were designed for presenting tabular data - NOT as a layout tool!

HTML Layout - Useful Tips

Tip: The biggest advantage of using CSS is that, if you place the CSS code in an external style sheet, your site becomes MUCH EASIER to maintain. You can change the layout of all your pages by editing one file. To learn more about CSS, study our CSS tutorial.

Tip: Because advanced layouts take time to create, a quicker option is to use a template. Search Google for free website templates (these are pre-built website layouts you can use and customize).

HTML5 Tag Reference

Ordered Alphabetically

New : New tags in HTML5.

Tag	Description
<!--...-->	Defines a comment
<!DOCTYPE>	Defines the document type
<a>	Defines a hyperlink
<abbr>	Defines an abbreviation
<acronym>	Not supported in HTML5
<address>	Defines contact information for the author/owner of a document/article
<applet>	Not supported in HTML5
<area>	Defines an area inside an image-map
<article>New	Defines an article
<aside>New	Defines content aside from the page content
<audio>New	Defines sound content
<b>	Defines bold text
<base>	Specifies the base URL/target for all relative URLs in a document
<basefont>	Not supported in HTML5
<bdi>New	Isolates a part of text that might be formatted in a different direction from other text outside it

<bdo>	Overrides the current text direction
<big>	Not supported in HTML5
<blockquote>	Defines a section that is quoted from another source
<body>	Defines the document's body
<br>	Defines a single line break
<button>	Defines a clickable button
<canvas>New	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<caption>	Defines a table caption
<center>	Not supported in HTML5
<cite>	Defines the title of a work
<code>	Defines a piece of computer code
<col>	Specifies column properties for each column within a <colgroup> element
<colgroup>	Specifies a group of one or more columns in a table for formatting
<command>New	Defines a command button that a user can invoke
<datalist>New	Specifies a list of pre-defined options for input controls
<dd>	Defines a description of an item in a definition list
<del>	Defines a text that has been deleted from a document
<details>New	Defines additional details that the user can view or hide
<dfn>	Defines a definition term
<dir>	Not supported in HTML5
<div>	Defines a section in a document
<dl>	Defines a definition list
<dt>	Defines a term (an item) in a definition list
<em>	Defines emphasized text
<embed>New	Defines a container for an external application or interactive content (a plug-in)
<fieldset>	Groups related elements in a form
<figcaption>New	Defines a caption for a <figure> element
<figure>New	Specifies self-contained content
<font>	Not supported in HTML5
<footer>New	Defines a footer for a document or section
<form>	Defines an HTML form for user input
<frame>	Not supported in HTML5
<frameset>	Not supported in HTML5
<h1> to <h6>	Defines HTML headings
<head>	Defines information about the document

<header>New	Defines a header for a document or section
<hgroup>New	Groups heading (<h1> to <h6>) elements
<hr>	Defines a thematic change in the content
<html>	Defines the root of an HTML document
<i>	Defines a part of text in an alternate voice or mood
<iframe>	Defines an inline frame
<img>	Defines an image
<input>	Defines an input control
<ins>	Defines a text that has been inserted into a document
<keygen>New	Defines a key-pair generator field (for forms)
<kbd>	Defines keyboard input
<label>	Defines a label for an input element
<legend>	Defines a caption for a <fieldset>, <figure>, or <details> element
<li>	Defines a list item
<link>	Defines the relationship between a document and an external resource (most used to link to style sheets)
<map>	Defines a client-side image-map
<mark>New	Defines marked/highlighted text
<menu>	Defines a list/menu of commands
<meta>	Defines metadata about an HTML document
<meter>New	Defines a scalar measurement within a known range (a gauge)
<nav>New	Defines navigation links
<noframes>	Not supported in HTML5
<noscript>	Defines an alternate content for users that do not support client-side scripts
<object>	Defines an embedded object
<ol>	Defines an ordered list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<output>New	Defines the result of a calculation
<p>	Defines a paragraph
<param>	Defines a parameter for an object
<pre>	Defines preformatted text
<progress>New	Represents the progress of a task
<q>	Defines a short quotation
<rp>New	Defines what to show in browsers that do not support ruby annotations
<rt>New	Defines an explanation/pronunciation of characters (for East Asian

	typography)
<ruby>New	Defines a ruby annotation (for East Asian typography)
<s>	Defines text that is no longer correct
<samp>	Defines sample output from a computer program
<script>	Defines a client-side script
<section>New	Defines a section in a document
<select>	Defines a drop-down list
<small>	Defines smaller text
<source>New	Defines multiple media resources for media elements (<video> and <audio>)
<span>	Defines a section in a document
<strike>	Not supported in HTML5
<strong>	Defines important text
<style>	Defines style information for a document
<sub>	Defines subscripted text
<summary>New	Defines a visible heading for a <details> element
<sup>	Defines superscripted text
<table>	Defines a table
<tbody>	Groups the body content in a table
<td>	Defines a cell in a table
<textarea>	Defines a multiline input control (text area)
<tfoot>	Groups the footer content in a table
<th>	Defines a header cell in a table
<thead>	Groups the header content in a table
<time>New	Defines a date/time
<title>	Defines a title for the document
<tr>	Defines a row in a table
<track>New	Defines text tracks for media elements (<video> and <audio>)
<tt>	Not supported in HTML5
<u>	Defines text that should be stylistically different from normal text
<ul>	Defines an unordered list
<var>	Defines a variable
<video>New	Defines a video or movie
<wbr>New	Defines a possible line-break

Elements introduced in HTML5

There has been a few elements introduced in HTML5. Some of them were imported from XHTML2, others from large statistical surveys on made by Google, Opera and other people. The analysis of **class** names and **id** names helped defining what were the most common semantic patterns. These new elements are not yet completely imported in all browsers at this time.

- article
- aside
- audio
- canvas
- command
- datalist
- details
- embed
- figcaption
- figure
- footer
- header
- hgroup
- keygen
- mark
- math
- meter
- nav
- output
- progress
- rp
- rt
- ruby
- section
- source
- summary
- svg
- time
- track
- video
- wbr

Note also that the input element has been extended with several new types as well