snippet nuggets

HTML and HTML5 Cheat sheets

HTML and HTML5 Cheat sheets for tags of Text formatting, Section Division, Images, Links, Lists, Frames, Forms, Special Characters and much more

Topics	Page No.
Document Summary and Information Tags	1
Document Structuring Tags	3
Text Formatting Tags	4
Anchor and Image Tags	5
List and Form Tags	6
Input Tag	7
Select Tag	8
Table	9
Object and iframe	10
List of HTML5 New Tags	11
List of Character Objects	13

Developed by **SnippetNuggets.com**

Document Summary Tags

<!DOCTYPE html>

This tag is not an HTML tag, but a declaration declares that document is HTML5. It is DTD(Document Type Declaration)
Modern browsers interprets HTML5 without DTD but still it is best practice to start webpage with <!DOCTYPE html>.
Nuggets:

- It is not HTML tag.
- Case-insensitive.
- Helps browser to display webpage correctly.

<html> … <html>

This tag is root element/tag for page.
Nuggets:

- Must be used as parent or root tag.
- Presence of <html> … <html> tag confirms that document is HTML page.
- Must contain <head> … </head>, <title> … </title> and <body> … </body>.

<head> … </head>

This tag is used to specify meta-data about HTML page and dependencies files. Some meta-datas like: description, keywords, author, viewport and http-equiv etc. It also contains tags like link and script for including dependencies files for CSS, JavaScript and font-usages etc.
Nuggets:

- Must be first child of HTML tag.
- <head> … </head> is the first element loaded by browser, hence all essential linking/dependencies and sourcing of JavaScript must be included inside this tag.

<title> … </title>

This tag contains name/title of the page. It is the name visible at the title's bar or tab bar of browser.
Nuggets:

- Search Engines use title as one of the most important parameter for indexing webpage. Hence, a relevant and informative title helps improving ranking of your webpage.
- Assigning precise yet descriptive title to webpage is essential and useful for user, search engine as well as web developer.
- While saving a webpage into local disk, the default name given to file is title.

<body> … </body>

This tag contains all elements which are intended to be displayed on page. It is the parent container for all contents or elements of the page.
Nuggets:

- Elements in webpage are rendered in the order of their existence inside <body> unless altered using CSS or JavaScript.
- Modularization of elements inside <body> tag is an art and differentiates professional web developer with others. Modularisation using <section>, <header>, <footer> and customized containers is good practice.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Document Title</title>
        <meta name="keywords" content="Tutorial, reference,
cheat sheet ">
        <link rel="stylesheet" href="./css/libs/font-
awesome/4.6.3/css/font-awesome.min.css">
        <script src="./js/libs/jquery.min_3.3.1.js"
type="text/javascript"></script>
    </head>
    <body>
        Body of webpage.
    </body>
</html>
```

Document Information Tags

<base> … </base>

This tag is used to specify base URL/target of webpage for all relative URLs of a webpage. It also acts as default URL and a default target for the links in webpage in case there is broken link.

Nuggets:

- Using <base> tag will make website design easy to maintain and lesser prone to link mismatch.
- Should be inside <head> tag.
- Presence of <base> tag ensures a default landing page for your webpage.

```
<head>
      <base
href=http://snippetnuggets.com/cheatsheets/index.html“ target
=”_blank”>
</head>
```

<meta> … </meta>

This tag is for providing information/data about the webpage. Modern webpages need some additional information which are not rendered in UI, but are useful for Browser and Search Engines to understand webpage in a better way.

Nuggets:

- Should be inside <head> tag.
- Passed as name/value pairs.
- 5 meta-tags are: ‘keywords’, ‘description’, author’, ‘viewport’ and ‘http-equiv’.
- For setting character encoding for webpage we use <meta charset=”UTF-8”> or <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">

```
<head>
      <meta name="keywords" content="Tutorial, reference, cheat
sheet ">
      <meta name="description" content="Contains quick reference
for HTML tags ">
      <meta name="author" content="Prakash Sao">
      <meta name="viewport" content="width=device-width,
initial-scale=1.0">
      <meta name="http-equiv " content="30">
</head>
```

<link> … </link>

This tag is used to link resources like CSS and fonts which are external to the webpage.

Nuggets:

- Using <base> tag will make website design easy to maintain and lesser prone to link mismatch.

```
<head>
      <link rel="stylesheet" type="text/css" href="theme.css">
</head>
```

<style> … </style>

This tag is used when you don’t want to use <link> tag to add external resources into your webpage, instead you want to place CSS inside your HTML page.

Nuggets:

- When there are only few CSS properties needed, it is adviced to use in-file CSS using <style> tag instead of maintaining an external CSS file and linking it using <link> tag

```
<head>
      <style>
            h1{ color: blue;}
            .paragraph_class{font-color: grey;}
      </style>
</head>
```

<script> … </script>

This tag is used to define a client-side JavaScript. Webpage either contains JavaScript inside <script> tag or uses <script> tag with ‘src’ attribute to point external JavaScript file.

Nuggets:

- JavaScript code are used to manipulate webpage, form validation and alteration of element based on User’s choice. It enables us to access any of the DOM elements of webpage and undertake varieties of actions on them. It is always a good practice to maintain an external JavaScript unless webpage contains just few JavaScript statements which can better be included using <script> tag in webpage itself.
- Better if included inside <head> so that JavaScript statements get loaded quickly.

```
<head>
      <script>
document.getElementById("demo").innerHTML = "Hello
JavaScript!";
      </script>
</head>
```

<noscript> … </noscript>

This tag is used to provide an alternate message or content for users if the browser doesn’t support client-side JavaScripts.

Nuggets:

- It is advised to include <noscript> messages for informing user about their browser settings instead of showing them broken page.

```
<head>
      <script>
document.getElementById("demo").innerHTML = "Hello
JavaScript!";
      </script>
      <noscript>Your browser does not support
JavaScript!</noscript>
</head>
```

Document Structuring Tags

<h1..h6> … </h1..h6>

HTML supports six levels of headings from h1(largest) to h6 (smallest). One can further customize their size/color/font through CSS.

Nuggets:

- Heading is given importance during indexing of webpage by Search Engines. Also, headings are used to categorize webpage in a better way so that User find it easy to browse through document structure. Hence, Heading should be given due importance.

```
<div class="myClass">
      <h1>Largest heading</h1>
      <h6>Smallest heading</h6>
      <p>This is paragraph displaying fruits color:
      <span style="color: red">red apple</span> and   <span
style="color: yellow;"> yellow mango              </span>
      </p>
      </hr>
      </br>
</div>
```

<div> … </div>

This tag divides webpage into different blocks or sections by enclosing element(s) inside <div> which has to be shown as one block.

Nuggets:

- <div> tag is used to design an efficient layout of webpage. Assigning an unique id and class to a <div> and providing style for it is common practice.
- All browser treats <div> as a block and places it after line break. However, this behavior can be changed using desirable CSS.

<span> … </span>

Used to group inline-elements in webpage, unlike <div> tag which by default places line break before it.

Nuggets:

- <span> tag is mostly used with text data like, formatting different part of paragraph text in different format.
- <span> tag doesn't offer any visual properties in itself. One has to provide visual properties using CSS.

<p> … </p>

This tag is used to display plain text as it is.

Nuggets:

- Browsers automatically put some space/margin before and after each <p> element. This behavior of browser can be modified using CSS margin-left/right.

<br/>

This tag inserts a single line break wherever placed. It is an empty tag with no closing tag. In HTML/HTML5 <br> is sufficient but in XHTML, properly closed <br/> is expected. <br/> tag doesn't display a line, instead use <hr/> tag

<hr/>

This tag displays a single line break wherever placed. It is an empty tag with no closing tag.

Example

```
<p>
      Text with different color:
      <span style="color: red"> red apple </span> and       <span
style="color: yellow;"> yellow mango   </span>
</p>
```

Text Formatting Tags

Following are phrase or emphasis tags.

<strong> … </strong>

Used to make enclosed text bold. Better output with effect can be achieved using CSS, hence <strong> is rarely used in modern web development.

<small> … </small>

Used to make enclosed text smaller. Better output with effect can be achieved using CSS, hence <small> is rarely used in modern web development.

<em> … </em>

Displays texts in italic

<code> … </code>

Displays computer code.

<samp > … </samp>

Used to display sample output of computer code.

<kbd > … </kbd>

Defines keyboard input.

<var > … </var>

Defines a variable.

<i> … </i>

Displays text in italic, but doesn't emphasis text like <em> does.

<b> … </b>

Displays enclosed text in bold font. It is similar or same as <strong>.

<tt> … </tt>

Formatting for typewriter-like text. No longer supported in HTML5.

<strike> … </strike>

Used to strike out a text at the center.

<cite> … </cite >

Used to cite author, reference or document or a quote.

<del > … </del>

Used to define that text has been deleted from a document. It is opposite of <ins> tag.

<ins> … </ins>

Used to define that text is inserted into the document. It is opposite of <del> tag.

<blockquote> … </blockquote>

Used to specify a section which is quoted from other source, reference or documents whose hyperlink is provided to 'cite' attribute in <blockquote> tag. *Refer example.*

<q> … </q>

Used as one liner short quotation. Browsers insert double quotes around the quotation text.

<abbr> … </abbr>

Used to define an abbreviation or an acronym. It is used with attribute 'title' whose value is shown when acronym is mouse-hovered. *Refer example.*

<address> … </address>

Used to display address/contact of owner/author of document/webpage/article.

<dfn> … </dfn>

Used to define a term when used for the first time in document. Nearest parent to <dfn> tag must contain complete definition or explaination for the term inside <dfn> tag. *Refer example.*

<sub> … </sub>

Used for writing a subscript(smaller font of height half and below the normal line. It is used to write chemical formula like H_2O)

<sup> … </sup>

Used for writing superscript(smaller font of height half and above the normal line. It is used to write footnotes or references like USA[1])

Example

```
<abbr title=”United States of America”> USA </abbr> consists of 50 states.

<blockquote cite=”https://en.wikipedia.org/wiki/United_States”> USA consists of 50 states and consists of North America and Alaska </blockquote>

<p><dfn>USA</dfn> consists of 50 state</p>
```

Anchor Tag

<a href=””> … </a>

Used for define a destination hyperlink, which gets loaded when anchor text/image or element is clicked. Most important attribute of <a> tag is ‘href’. By default, browser displays(unless modified by custom CSS):

1. An unvisited link as underlined and blue.
2. A visited link as underlined and purple.
3. An active or clicked link as underlined and red.

Nuggets:

- Following Attributes are supported in <a> tag only with ‘href’ tag: download, hreflang, media, rel, target and type.
- A linked page mentioned in ‘href’ attribute is loaded in the same browser tab unless ‘target’ attribute is specified for other target.
- ‘href’ tag can also contain ‘mailto’, ‘tel’ and ‘#name’ as explained below.

<a href=”mailto:contact@snippetnuggets.com”> … </a>

Used initiating the mail sending to specified mail-id when anchor link is clicked.

<a href=”tel://###,#####”> … </a>

Used for initiating a phone call to number specified. It is useful when webpage is accessed from mobile phone and link is clicked.

<a name=”name”> … </a>

Used for navigate to different section of webpage when anchor link is clicked.

<a href=”#class_name”> … </a>

Used for navigate to DOM element having class name ‘class_name’ when anchor link is clicked.

Image Tag

<img> … <img/>

Used to insert and display image in webpage with desired height, width, position. List of attributes:

1. src=’url’
   URL or physical path of image.
2. alt=’text’
   Text which get displayed on mouse-hovering over image. If image link/path is broken, this text will be displayed in place of image. Also, this alt text is used by Search Engines while indexing the webpage for what the image is about.
3. height=’’ and width=’’
   defines height and width in pixels or percentage.
4. align=’top|bottom|middle|left|right’
   defines relative alignment of an image.
5. border=’xxpx’
   specifies width of border around the image. Not supported in HTML5
6. longdesc=*URL*
   specifies a URL to a detailed description for image.
7. usemap=*#mapname*
   specifies an image as client-side image-map

<map name=””> … </map>

designs an interactive(clickable and different mouse-hover title for different region) image. Required attribute ‘name’ in the <map> element is associated with <img> element’s ‘usemap’ attribute and creates relationship between image <img> and map <map>.

<area/>

Used for specifying <img>’s image map area. Following attributes goes with <area> element.

1. shape=’default|rect|poly|circle’
   shape of area.

2. cords=’’
   specifies coordinates of area in comma separated format.

```
<img src="flower.gif" width="125" height="126" alt="Planets"
usemap="#flowermap">
      <map name="flowermap">
      <area shape="rect" coords="0,0,82,126" href="petal.htm"
alt="petal">
      <area shape="circle" coords="90,58,3" href="center.htm"
alt="center">
      <area shape="circle" coords="124,58,8" href="bud.htm"
alt="bud">
      </map>
```

List Tag

<ul> … </ul>

Defines an unordered or bulleted/dotted list. Following attributes are used in <ul> element:

1. type=’disc|square|circle’
   Specifies the kind of marker to use in the list. It is not supported in HTML5. Hence, same result is achieved using CSS property ‘list-style-type’ with value: ‘disc|square|circle’

<ol> … </ol>

Defines an ordered list. Following attributes are used in <ol> element:

1. reversed:
   Specifies that list ordering should be in reverse order of number like(5,4,3,2,1 or e, d, c, b, a).
   Example: <ol reversed> … </ol>
2. start=’<number>’
   Specifies start value of numbering.
3. type=’1|A|a|I|i’
   Specifies the kind of marker used in ordered list like numeral, Uppercase alphabetic, Lowercase alphabetic, Uppercase roman and Lowercase roman

<li> … </li>

Lists an individual items inside <ul> or <ol> element.

<dl> … </dl>

Like <ul> and <ol> lists unordered and ordered lists respectively. <dl> is description list where it lists item which has to be displayed with definition/description. Inside <dl> element we need to provide list items using <dt> and <dd>.

<dt> … </dt>

Defines a term/name in description list. It is used in conjunction with <dl> (defines a description list) and <dd> (describes each term/name).

<dd> … </dd>

Describes a term/name in description list. It is used in conjunction with <dl> (defines a description list) and <dd> (describes each term/name).

```
<ol type="1" start="50" reversed>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

<dl>
  <dt>Coffee</dt>
  <dd>Black hot drink</dd>
  <dt>Milk</dt>
  <dd>White cold drink</dd>
</dl>
```

Form Tag

<form> … </form>

Used for creating HTML form for user to input. Form elements contain several elements(like input box, textarea, button, select menu etc) to let user input the data. All such elements for user inputs are enclosed inside <form> element.

Following attributes can be used in <form> element:

1. accept-charset=”IS-8859-1|utf-8”
   Specifies the character encodings when form is submitted. But if webpage is already interpreted by browser as being ‘utf-8’, then having ‘accept-charset’ attribute is not required. Character encoding for webpage can be set using <meta charset=”UTF-8”>

2. action=”*URL*”
   URL where the form data will be submitted after user fills form and submit.
3. autocomplete=”on|off”
   Specifies whether a form should have autocomplete on or off.
4. enctype=” application|x-www-form-urlencoded|multipart|form-data|text|plain”
   Specifies how form-data should be encoded when submitting it to the server(only for method=”post”)
5. name=”*text*”
   Specifies the name of a form
6. method=”get|post”
   Specifies HTTP method to use while sending form-data over network.
7. novalidate
   Specifies that form should not be validated while submitting.
8. target=”_blank|_self|_parent|_top”
   Specifies where to display the response which is received after submitting the form.

Other child elements of <Form>

<fieldset> … </fieldset>

Used to group related elements inside form as one entity and enclose them inside box.

<legend> … </legend>

Specifies caption to the <fieldset> element.

<label> … </label>

Used for assigning label to a field in the form.

Example

```
<form>
      <fieldset>
     <legend>Personalia:</legend>

     Name: <input type="text"><br>

     Email: <input type="text"><br>

     Date of birth: <input type="text">
     </fieldset>
</form>
```

<input> … </input>

Input Tag

Used to create input elements inside the form. Based on the value of ‘type’ attribute, Input element changes behavior to button, checkbox, color, data, datetime-local, file, email, radio, range etc. Given below are different attributes of <input> element.

1. accept=”*file_extension*|audio/*|video/*|image/*|*media_type*”
   Specifies the types of files that server accepts(only for type=”file”)
2. alt=”*text*”
   Specifies an alternate text for images(only when type=”image” is used in <input>). ‘alt’ attribute is important for Indexing by Search Engines.
3. autocomplete=”on|off”
   Specifies, if <input> should have autocomplete enabled or disabled.
4. autofocus
   Specifies if <input> element should get autofocus when page loads.
5. checked
   Specifies the state of <input> element type=”checkbox” or type=”radio”
6. disabled
   Specifies that an <input> element should be disabled.
7. form=”*form_id*”
   Specifies the id of <form> element to which <input> element belongs to.
8. formaction=”*URL*”
   Specifies URL of the file that will process the input control when form is submitted(only for type=”submit” and type=”image”).
9. formenctype=”application/x-www-form-urlencoded | multipart/form-data | text/plain”
   Specifies how the form data should be encoded when submitting it to the server(only for type=”submit” and type=”image”).
10. formmethod=”get|post”
    Defines the HTTP method for sending data to action URL(only for type=”submit” and type=”image”)
11. formnovalidate
    Defines that form elements should not be validated when submitted.

12. formtarget="_blank|_self|_parent|_top|*frame name*"
    Specifies where to display the response that is received after submitting the form(only for type="submit" and type="image")
13. height="<in pixels>"
    Specifies height of an <input> element(only for type="image"). For other types of <input> element, modifying height through CSS is preferred.
14. list="*datalist_id*"
    Specifies to <datalist> element that contains pre-defined options for an <input> element.
15. max="*number|data*"
    Specifies the maximum value for an <input> element.
16. maxlength="*number*"
    Specifies the maximum number of characters allowed in <input> element.
17. min=="*number|data*"
    Specifies a minimum value for an <input> element.
18. multiple
    Specifies that a user can enter more than one value in an <input> element.
19. name="*text*"
    Specifies the name of an <input> element.
20. pattern="<regexp>"
    Specifies a regular expression that an <input> element's value is checked against before accepting input as correct.
21. placeholder="*text*"
    Specifies a short hint that describes the expected value of an <input> element.
22. readonly
    Specifies a short hint that describes the expected value of an <input> element.
23. required
    Specifies that an input field must be filled before submitting the form. Such elements are shown with an asterisk symbol before it.
24. size="*number*"
    Specifies the width, in characters, of an <input> element.
25. src="*URL*"
    Specifies the URL of the image to use as a submit button(only for type="image").
26. step="*number*"
    Specifies the legal number intervals for an input field.
27. type="<type of input>"
    Following are the values an type attribute can take

type values

- button
- checkbox
- color
- data
- datetime-local
- email
- file
- hidden
- image
- month
- number
- password
- radio
- range
- reset
- search
- submit
- tel
- text
- time
- url
- week

28. value="*text*"
    Specifies the value of an <input> element.
29. width="<pixels>"
    Specifies width of an <input> element(only for type="image").

Select Tag

<select> … </select>

Used to create drop-down list with several options to select. The <option> tags inside <select> element creates list of options.

<option> … </option>

Used to create list of options for <select> element.

Nuggets:

- The <select> element is a form control and can be used in a form to collect user input.
- <select> and <option> elements also supports both global and event attribute.
- <option> element(s) go inside <select> or <datalist> element
- The <option> tag can be used without any attributes, but you usually need the value

attribute, which indicates what is sent to the server.

- If you have a long list of options, you can group and sub-group related options with the <optgroup> tag.

Example

```
<select>
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
```

List of Attributes for <select> element

1. name=”*text*”
   Specifies name of drop-down list.
2. size=”*number*”
   Specifies number of visible options in drop-down list.
3. autofocus
   Specifies that, drop-down should get by default focus when webpage loads.
4. disabled
   Specifies that drop-down list should be disabled.
5. form=”*form_id*”
   Defines one or more forms the <select> element belongs to.
6. multiple
   Specifies, if multiple options can be selected.
7. required
   Specifies if user is required to select an option before submittimg form.

List of Attributes for <option> element

1. disabled
   Specifies that an option in <select> has to be disabled.
2. label=”*text*”
   Specifies shorter label for an option.
3. selected
   Specifies that the option should be pre-selected on page load.
4. value=”*text*”
   Specifies the value which is sent to server.

Example

```
<form>
  <fieldset>
  <legend>Personalia:</legend>
  Name: <input type="text"><br>
  Email: <input type="text"><br>
  Date of birth: <input type="date"><br/>
  Occupation:
    <select>
       <option value="student">Student</option>
      <option value="professional">Professional</option>
      <option value="retired">Retired</option>                <option
value="others" disabled>Others</option>
    </select><br/>
    <textarea name="comments">Comments</textarea><br/>
       <input type="submit" value="Submit">
 </fieldset>
```

Table Tag

<table> … </table>

Used to create table in webpage. Inside <table> element there are one or more <th>(table header), <tr>(table row) and <td>(table data) elements. A more complex HTML table may also include <caption>, <col>, <colgroup>, <thead>, <tfoot>, and <tbody> elements.

Nuggets:

- HTML5 specifications have made <table> element more precise and suggested to use CSS structuring instead of using tag’s attributes. Hence, "align", "bgcolor", "border", "cellpadding", "cellspacing", "frame", "rules", "summary", and "width" attributes are not supported in HTML5.

<caption> … </caption>

Used to provide description of table. <caption> tag must be inserted immediately after <table> tag.

Nuggests:

- Only one caption per table is allowed.
- By default, a table caption will be center-aligned above a table. However, the CSS properties

'text-align' and 'caption-side' can be used to align and place the caption.

<thead> … </thead>

Used to group header content in HTML table.

Nuggets:

- The <thead> element is used in conjunction with the <tbody> and <tfoot> elements to specify each part of a table (header, body, footer). The <thead> tag must be used in the following context: As a child of a <table> element, after any <caption>, and <colgroup> elements, and before any <tbody>, <tfoot>, and <tr> elements.

<tbody> … </tbody>

Used to define the body of table where data is held.

<tfoot> … </tfoot>

Used to create the footer of the table.

<tr> … </tr>

Used to create single row of the table.

<th> … </th>

Used to create heading of the table's column.

<td> … </td>

Used to create a single cell in table which contains actual data. Following are the attributes for <td> element:

- colspan="<integer>"
  colspan attribute defines the number of columns a cell should span.
- rowspan="<integer>"
  rowspan attribute specifies the number of rows a cell should span.

<colgroup> … </colgroup>

Used for grouping column together.

<col> … </col>

Denotes a column inside the table.

<object> … </object>

Used to embed object(multimedia) within an HTML document. Multimedia objects like audio, video, Java applets, ActiveX, PDF, and Flash are embedded in web pages using <object> element.

```
<table>
  <caption>Monthly savings</caption>
  <tr>
    <td colspan="2">Table containing month savings</td>
  </tr>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
```

Object and iFrame Tags

<object> element also supports both global and event attributes.

Nuggets:

- You can also use the <object> tag to embed another webpage into your HTML document.
- You can use the <param> tag to pass parameters to plugins that have been embedded with the <object> tag.
- An <object> element must appear inside the <body> element. The text between the <object> and </object> is an alternate text, for browsers that do not support this tag.
- For images use the <img> tag instead of the <object> tag.
- At least one of the "data" or "type" attribute MUST be defined.

Following attributes are used in <object> element:

1. data="*URL*"
   Specifies the URL of the resource to be used by the object
2. height="<pixels value>"
   Specifies the height of object.

Example

3. form="form_id"
   Specifies one or more forms the object belongs to.

4. name=”*name*”
   Specifies name of object.
5. width=”<pixels value>”
   Specifies the width of object.
6. type=”*media_type*”
   Specifies the media type of data specified in the data attribute
7. usemap=”#mapname”
   Specifies the name of a client-side image map to be used with the object.

<iframe> … </iframe>

Used to embed an inline frame. An inline frame is used to embed another document within current HTML document.

Nuggets:

- To deal with browsers that do not support <iframe>, add a text between the opening <iframe> tag and the closing </iframe> tag.
- Use CSS to style the <iframe> (even to include scrollbars).

Attributes for <iframe> element:

1. name=”*name*”
   Specifies name of an <iframe> element.
2. height=”<pixels value>”
   Specifies the height of object.
3. width=”<pixels value>”
   Specifies the width of object.
4. src=”*URL*”
   Source URL/path of the multimedia object to be held inside the iFrame.
5. srcdoc=”*HTML_code*”
   Specifies the HTML content of the page to show in the <iframe>
6. sandbox=”allow-forms | allow-pointer-lock | allow-popups | allow-same-origin | allow-scripts | allow-top-navigation”
   Enables an extra set of restrictions for the content in an <iframe>.

<param/>

For iFrame customization. This includes additional parameters to go along with the content.

<embed> … </embed>

Used to embed an external object, like plugins(e.g. flash video). <embed> tag defines a container for an external application or interactive content (a plug-in). Attributes of <embed> element:

1. height=”<in pixels>”
   Specifies height of embedded item.
2. width=”in pixels”
   Specifies width of embedded item.
3. type=”*media_type*”
   Specifies the media type of content of embedded content.
4. src=”*URL/Path*”
   The URL/Path of the embedded item.

Example

```
<object width="500" height="500">
</object>
<iframe src="link_to_other_webpage.html" width="500"
height="500"> </iframe>
<embed src="some_video.swf" width="250" height="250"></embed>
```

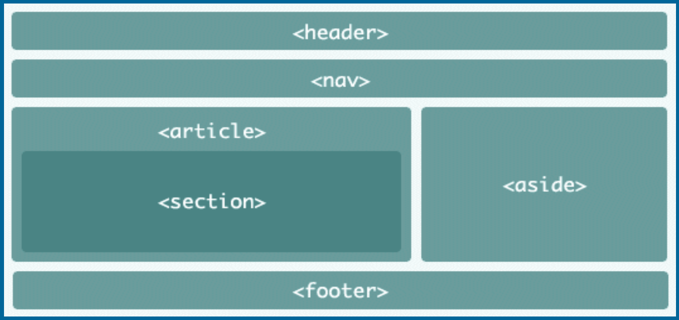
List of HTML5 new Tags

The W3C HTML5 Specification introduced many new tags for purposes like Semantic/Structural elements, Text-formatting, Form elements, input types, Graphics and Media Elements.

New Semantic/Structural Elements for better documenting structure:

Tags	Description
<article>	Defines an article in document.
<aside>	Defines content aside of page content
<bdi>	Defines text which has to be displayed bi-directional
<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
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<main>	Defines main content of document.

Tags	Description
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within known range(a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters(for East Asian typography)
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element.
<time>	Defines a date/time
<wbr>	Defines a possible line-break



New Form Elements

Tags	Description
<datalist>	Specifies a list of pre-defined options for input controls
<output>	Defines the result of a calculation

New Input types and Attributes

New Input Types	New Input Attributes
color	autocomplete
date	autofocus
datetime	form
datetime-local	formaction
email	formenctype
month	formmethod
number	formnovalidate
range	formtarget
search	height and width
tel	list
time	min and max
url	multiple
week	pattern(regex)
	placeholder
	required
	step

New Attributes syntax

HTML5 allows four different syntaxes for attributes.
Example with <input> tag:

Type	Example
Empty	<input type="text" value="John" **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 four syntaxes may be used, depending on what is needed for the attribute.

New HTML5 Graphics Elements

HTML5 allows four different syntaxes for attributes.
Example with <input> tag:

Type	Example
<canvas>	The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.
<svg>	Draw scalable vector graphics. SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

New Media Elements

HTML5 allows four different syntaxes for attributes.
Example with <input> tag:

Type	Example
<audio>	Defines sound content
<embed>	Defines a container for an external (non-HTML) application
<source>	Defines multiple media resources for media elements (<video> and <audio>)
<track>	Defines text tracks for media elements (<video> and <audio>)
<video>	Defines video or movie

List of Character Objects

Many mathematical, technical and currency symbols are not present in normal keyboard. For displaying/adding such symbols in webpage we use HTML entity name. If no entity name exists, you can use an entity number, a decimal, or hexadecimal reference.

Char	Number	Entity	Description
©	&#169;	&copy;	Copyright symbol
€	&#8364;	&euro;	Euro symbol
™	&#8482;	&trade;	Trademark
“	&#34;	&quot;	Quotation mark
<	&#60;	&lt;	Less than sign
>	&#62;	&gt;	Greater than sign
	&#160;	&nbsp;	Non-breaking space
&	&#38;	&amp;	Ampersand
@	&#64;	&Uuml;	@ symbol
®	&#174;	&reg;	Registered symbol