

Part One: HTML

1. Introduction.....	1
2. HTML Editors	2
3. HTML Elements.....	3
4. HTML Attributes.....	5
5. HTML Headings.....	6
6. HTML Paragraphs.....	8
7. HTML Styles	9
8. HTML Text Formatting Elements	11
9. HTML Quotation and Citation Elements	12
10. HTML Computer Code Elements.....	13
11. HTML Comments	14
12. HTML Links.....	14
13. HTML Images	16
14. HTML Tables	18
15. HTML Lists	22
16. HTML Block Elements.....	25
17. HTML Layouts	26
18. HTML Multimedia	29
19. HTML Forms and Input.....	30
20. HTML5 Input Types	33
21. HTML Input Attributes	37
22. HTML5 Attributes.....	38
23. HTML Iframes.....	42
24. HTML Scripts	43
25. HTML Head	43
26. HTML Entities.....	45
27. HTML Symbols	46
28. Complete HTML Tags	48

1. Introduction

1.1. What is HTML?

- HTML is a **markup** language for **describing** web documents (web pages).
- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- A markup language is a set of **markup tags**
- HTML documents are described by **HTML tags**
- Each HTML tag **describes** different document content

HTML Example: Small HTML document

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Example Explained:

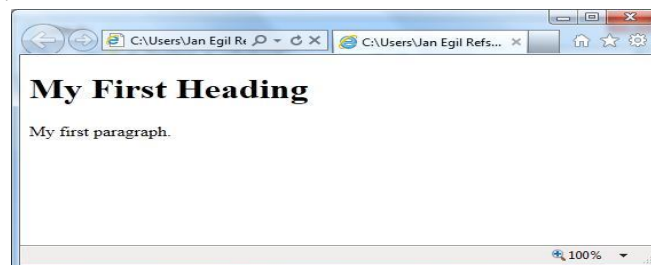
- The **DOCTYPE** declaration defines the **document type**
- The text between **<html>** and **</html>** describes the **web document**
- The text between **<body>** and **</body>** describes the **visible page content**
- The text between **<h1>** and **</h1>** describes a **heading**
- The text between **<p>** and **</p>** describes **paragraph**

1.2. HTML Tags

- HTML tags are **keywords** (tag names) surrounded by **angle brackets**:
Syntax: `<tag_name>content</tag_name>`
- HTML tags normally come **in pairs** like `<p>` and `</p>`
- The first tag in a pair is the **start tag**, the second tag is the **end tag**
- The end tag is written like the start tag, but with a **slash** before the tag name
- The start tag is often called the **opening tag**. The end tag is often called the **closing tag**

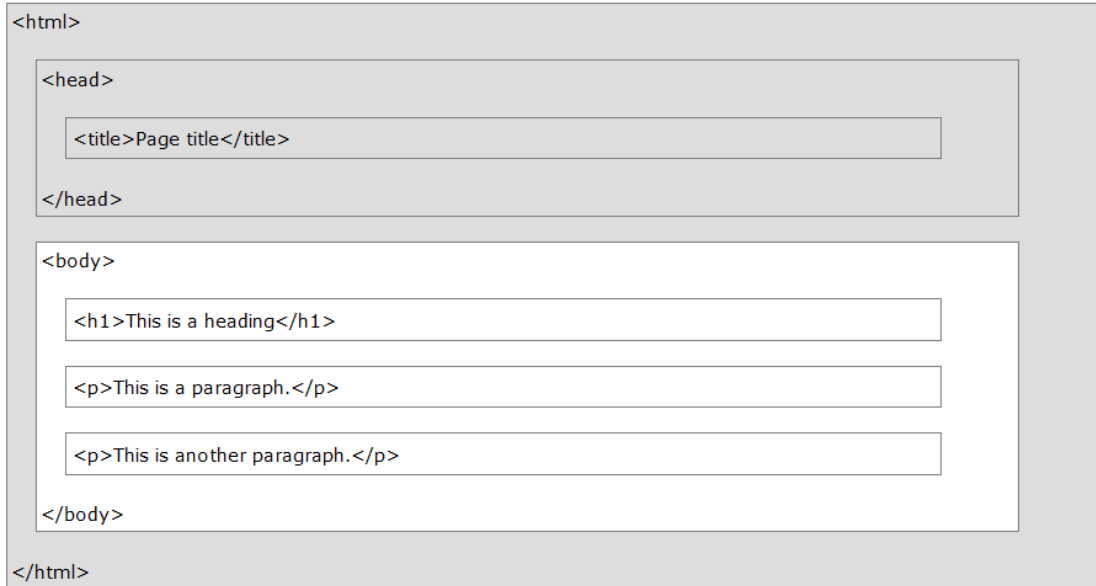
1.3. Web Browsers

- The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them.
- The browser does not display the HTML tags, but uses them to determine how to display the document:



1.4. HTML Page Structure

- Below is a visualization of an HTML page structure:



1.5. HTML Versions

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

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

The <! DOCTYPE> Declaration

- The <! DOCTYPE> declaration helps the browser to display a web page correctly.
- There are many different documents on the web, and a browser can only display an HTML page correctly if it knows the HTML version and type.
- All examples in this course use HTML5.**

2. HTML Editors

Write HTML Using Notepad or TextEdit

HTML can be edited by using a professional HTML editor like:

- Adobe Dreamweaver
- Microsoft WebMatrix
- CoffeeCup HTML Editor

However, for learning HTML we recommend a text editor like Notepad (PC) or TextEdit (Mac).

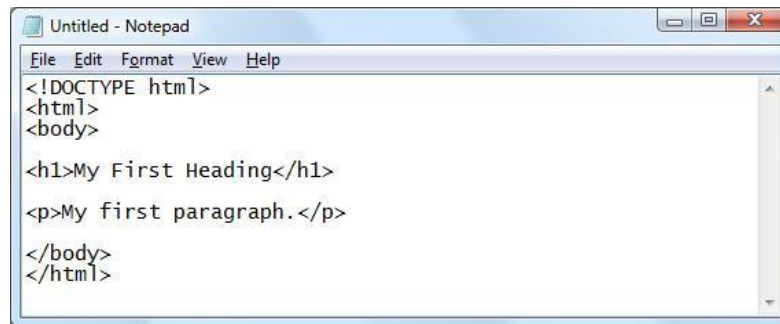
- We believe using a simple text editor is a good way to learn HTML.
- Follow the 4 steps below to create your first web page with Notepad.

Step 1: Open Notepad

Step 2: Write Some HTML

Write or copy some HTML into Notepad.

Example:



Step 3: Save the HTML Page

- Save the file on your computer.
- When saving an HTML file, use either the **.htm** or the **.html** file extension. There is no difference; it is entirely up to you. The only difference between the two is that **.htm** is used as an alternate to **.html** by few servers that do not accept four character extensions.

Step 4: View HTML Page in Your Browser

- Double-click your saved HTML file, and the result will look much like this:



3. HTML Elements

- HTML **documents** are made up by HTML **elements**.
-
- HTML elements are written with a **start** tag, with an **end** tag, with the **content** in between:
`<tag_name>content</tag_name>`
 - The HTML **element** is everything from the start tag to the end tag:
`<p>My first paragraph.</p>`
-

Nested HTML Elements

- HTML elements can be nested (elements can contain elements).
- All HTML documents consist of nested HTML elements.
- This example contains 4 HTML elements:

Example

```
<!DOCTYPE html>
<html>
<body>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</body>
</html>
```

Example Explained

- The **<html>** element defines the **whole document**.
- It has a **start** tag **<html>** and an **end** tag **</html>**.
- The element **content** is another HTML element (the **<body>** element).

```
<html>
<body>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</body>
</html>
```

- The **<body>** element defines the **document body**.
- It has a **start** tag **<body>** and an **end** tag **</body>**.
- The element **content** is two other HTML elements (**<h1>** and **<p>**).

```
<body>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</body>
```

- The **<h1>** element defines a **heading**.
- It has a **start** tag **<h1>** and an **end** tag **</h1>**.
- The element **content** is: My First Heading.

```
<h1>My First Heading</h1>
```

Don't Forget the End Tag

- Some HTML elements will display correctly, even if you forget the end tag:

Example

```
<html>
<body>
  <p>This is a paragraph
  <p>This is a paragraph
</body>
</html>
```

- The example above works in all browsers, because the closing tag is considered optional.
 - **Never rely on this.** It might produce unexpected results and/or errors if you forget the end tag.
-

Empty HTML Elements

- HTML elements with no content are called empty/void elements.
- `
` is an empty element without a closing tag (the `
` tag defines a line break).
- Empty element can be "closed" in the opening tag like this: `
`.
- HTML5 does not require empty elements to be closed. But if you need stricter validation, and make your document readable by XML parsers, please close all HTML elements.

HTML Tip: Use Lowercase Tags

- HTML tags are not case sensitive: `<P>` means the same as `<p>`.

4. HTML Attributes

- Attributes provide additional information **about HTML elements**.

-
- HTML elements can have **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"**
-

The *lang* Attribute

- The document language can be declared in the `<html>` tag.
- The language is declared in the **lang** attribute.
- Declaring a language is important for accessibility applications (screen readers) and search engines:

Example

```
<!DOCTYPE html>
<html lang="en-US">
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

- The first two letters specify the language (en). If there is a dialect, use two more letters (US).

The *title* Attribute

- HTML paragraphs are defined with the `<p>` tag.
- In this example, the `<p>` element has a **title** attribute. The value of the attribute is **"About W3Schools"**:
- When you move the mouse over the element, the title will be displayed as a tooltip

Example

```
<p title="About W3Schools">
W3Schools is a web developer's site.
It provides tutorials and references covering
many aspects of web programming,
including HTML, CSS, JavaScript, XML, SQL, PHP, ASP, etc.
</p>
```

The *href* Attribute

- HTML links are defined with the `<a>` tag. The link address is specified in the **href** (hypertext reference) attribute:

Example

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

Size Attributes

- HTML images are defined with the `` tag.
- The filename of the source (**src**), and the size of the image (**width** and **height**) are all provided as **attributes**:

Example

```

```

- The image size is specified in pixels: width="104" means 104 screen pixels wide.
-

The *alt* Attribute

- The **alt** attribute specifies an alternative text to be used, when an HTML element cannot be displayed.
- The value of the attribute can be read by "screen readers". This way, someone "listening" to the webpage, i.e. a blind person, can "hear" the element.

Example

```

```

HTML Tip: Always Use Lowercase Attributes

HTML Tip: Always Quote Attribute Values

- The HTML5 standard does not require quotes around attribute values.
- The **href** attribute, demonstrated above, can be written as:

Example

```
<a href=http://www.w3schools.com>
```

- Sometimes it is **necessary** to use quotes. This will not display correctly, because it contains a **space**:

Example

```
<p title=About W3Schools>
```

Single or Double Quotes?

- Double style quotes are the most common in HTML, but single style can also be used.
- In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

Example

```
<p title='John "ShotGun" Nelson'>
```

Or vice versa:

Example

```
<p title="John 'ShotGun' Nelson">
```

5. HTML Headings

- Headings are important in HTML documents.
- Headings are defined with the `<h1>` to `<h6>` tags.

- `<h1>` defines the **most important heading**. `<h6>` defines the **least important heading**.

Example

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

Note: Browsers automatically add some empty space (a margin) before and after each heading.

- 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.
- Users skim your pages by its headings. It is important to use headings to show the document structure.
- h1 headings should be main headings, followed by h2 headings, then the less important h3, and so on.

HTML Horizontal Rules

- The `<hr>` tag creates a horizontal line in an HTML page.
- The hr element can be used to separate content:

Example

```
<p>This is a paragraph.</p>
<hr>
<p>This is a paragraph.</p>
<hr>
<p>This is a paragraph.</p>
```

The HTML `<head>` Element

- The HTML `<head>` element has nothing to do with HTML headings.
- The HTML `<head>` element only contains **meta data**.
- The HTML `<head>` element is placed between the `<html>` tag and the `<body>` tag:
- Meta data means data **about** data. HTML meta data is data **about** the HTML document

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>
<body>
  .
  .
  .
```

The HTML `<title>` Element

- The HTML `<title>` element is **meta data**.
- It defines the HTML document's title. It will not be displayed in the document.
- However, it might be displayed in one of the **browser tabs**.

The HTML `<meta>` Element

- The HTML `<meta>` element is meta data.

- It defines the character set used in the HTML document.

HTML Tip - How to View HTML Source

- Have you ever seen a Web page and wondered "Hey! How did they do that?"
- To find out, right-click in the page and select "View Page Source" (in Chrome) or "View Source" (in IE), or similar in another browser. This will open a window containing the HTML code of the page.

6. HTML Paragraphs

- HTML documents are divided into paragraphs.
- The HTML `<p>` element defines a **paragraph**.
- Browsers automatically add an empty line before and after a paragraph

Example

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

HTML Display

- You cannot be sure how HTML will be displayed.
- Large or small screens, and resized windows will create different results.
- With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code.
- The browser will remove extra spaces and extra lines when the page is displayed.
- Any number of spaces, and any number of new lines, counts as **only one space**.

Example

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>

<p>
This paragraph
contains    a lot of spaces
in the source    code,
but the    browser
ignores it.
</p>
```

HTML Line Breaks

- The HTML `
` element defines a **line break**.
- Use `
` if you want a line break (a new line) without starting a new paragraph:

Example

```
<p>This is<br>a para<br>graph with line breaks</p>
```

- The `
` element is an empty HTML element. It has no end tag.

The Poem Problem

Example

```
<p>This poem will display as one line:</p>
<p>
  My Bonnie lies over the ocean.
  My Bonnie lies over the sea.
  My Bonnie lies over the ocean.
  Oh, bring back my Bonnie to me.
</p>
```

The HTML `<pre>` Element

- The HTML `<pre>` element defines a block of **pre-formatted** text, with structured spaces and lines.
- To display anything, with right spacing and line-breaks, you must wrap the text in a `<pre>` element:

Example

```
<p>This will display as a poem:</p>
<pre>
  My Bonnie lies over the ocean.
  My Bonnie lies over the sea.
  My Bonnie lies over the ocean.
  Oh, bring back my Bonnie to me.
</pre>
```

7. HTML Styles

I am Red
I am Blue

HTML Styling

- Every HTML element has a **default style** (background color is **white**, text color is **black** ...)
- Changing the default style of an HTML element, can be done with the **style attribute**.
- The **bgcolor** attribute, supported in older versions of HTML, *is not valid in HTML5*

Example

```
<body style="background-color: lightgrey">
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

The HTML Style Attribute

- The HTML style attribute has the following **syntax**:
`style="property:value"`
 - The **property** is a CSS property. The **value** is a CSS value.
-

HTML Text Color

- The **color** property defines the text color to be used for an HTML element:

Example

```
<!DOCTYPE html>
<html>
<body>
  <h1 style="color:blue">This is a heading</h1>
  <p style="color:red">This is a paragraph.</p>
</body>
</html>
```

HTML Text Fonts

- The **font-family** property defines the font to be used for an HTML element:
- The **** tag, supported in older versions of HTML, *is not valid in HTML5*

Example

```
<!DOCTYPE html>
<html>
<body>
  <h1 style="font-family:verdana">This is a heading</h1>
  <p style="font-family:courier">This is a paragraph.</p>
</body>
</html>
```

HTML Text Size

- The **font-size** property defines the text size to be used for an HTML element:

Example

```
<!DOCTYPE html>
<html>
<body>
  <h1 style="font-size:300%">This is a heading</h1>
  <p style="font-size:160%">This is a paragraph.</p>
</body>
</html>
```

HTML Text Alignment

- The **text-align** property defines the horizontal text alignment for an HTML element:
- The **<center>** tag, supported in older versions of HTML, *is not valid in HTML5*

Example

```
<!DOCTYPE html>
<html>
<body>
  <h1 style="text-align:center">Centered Heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

8. HTML Text Formatting Elements

This text is bold

This text is italic

This is ^{superscript}

HTML also defines special **elements**, for defining text with a special **meaning**. HTML uses elements like `` and `<i>` for formatting output, like **bold** or *italic* text. Formatting elements were designed to display special **types of text**:

- Bold text
- Important text
- Italic text
- Emphasized text
- Marked text
- Deleted text
- Inserted text
- Subscripts
- Superscripts

HTML Bold Formatting

- The HTML `` element defines **bold** text, without any extra importance.

Example

```
<p>This text is normal.</p>
<p><b>This text is bold</b>.</p>
```

- The HTML `` element defines **strong** text, with added semantic "strong" importance.

Example

```
<p>This text is normal.</p>
<p><strong>This text is bold</strong>.</p>
```

HTML *Italic* Formatting

- The HTML `<i>` element defines *italic* text, without any extra importance.

Example

```
<p>This text is normal.</p>
<p><i>This text is italic</i>.</p>
```

- The HTML `` element defines *emphasized* text, with added semantic importance.

Example

```
<p>This text is normal.</p>
<p><em>This text is emphasized</em>.</p>
```

- Browsers display `` as ``, and `` as `<i>`.
- However, there is a difference in the meaning of these tags: `` and `<i>` defines bold and italic text, but `` and `` means that the text is "important"

HTML Marked Formatting

- The HTML `<mark>` element defines **marked** or highlighted text:

Example

```
<h2>HTML <mark>Marked</mark> Formatting</h2>
```

HTML Deleted Formatting

- The HTML `` element defines **deleted** (removed) of text.

Example

```
<p>My favorite color is <del>blue</del> red.</p>
```

HTML Inserted Formatting

- The HTML `<ins>` element defines **inserted** (added) text.

Example

```
<p>My favorite <ins>color</ins> is red.</p>
```

HTML Subscript Formatting

- The HTML `<sub>` element defines **subscripted** text.

Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

HTML Superscript Formatting

- The HTML `<sup>` element defines **superscripted** text.

Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

9. HTML Quotation and Citation Elements

HTML Short Quotations

- The HTML `<q>` element defines a **short quotation**.
- Browsers usually insert **quotation marks** around the `<q>` element.

Example

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>
```

HTML Long Quotations

- The HTML `<blockquote>` element defines a quoted section.
- Browsers usually **indent** `<blockquote>` elements.

Example

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>
```

HTML Quotations, Citations, and Definition Elements

Tag	Description
<code><abbr></code>	Defines an abbreviation or acronym
<code><address></code>	Defines contact information for the author/owner of a document
<code><bdo></code>	Defines the text direction
<code><blockquote></code>	Defines a section that is quoted from another source
<code><q></code>	Defines an inline (short) quotation
<code><cite></code>	Defines the title of a work
<code><dfn></code>	Defines a definition term d

10.HTML Computer Code Elements

Computer Code

```
var person =
{
  firstName:"John",
  lastName:"Doe",
  age:50,
  eyeColor:"blue"
}
```

HTML Computer Code Formatting

- Normally, HTML uses **variable** letter size, and variable letter spacing.
- This is not wanted when displaying examples of **computer code**.
- The `<kb>`, `<samp>`, and `<code>` elements all support **fixed** letter size and spacing.

HTML Keyboard Formatting

- The HTML `<kbd>` element defines **keyboard input**:

Example

```
<p>To open a file, select:</p>
<p><kbd>File | Open...</kbd></p>
```

HTML Sample Formatting

- The HTML `<samp>` element defines a **computer output sample**:

Example

```
<samp>
demo.example.com login: Apr 12 09:10:17
Linux 2.6.10-grsec+gg3+e+fhs6b+nfs+gr0501+++p3+c4a+gr2b-reslog-v6.189
</samp>
```

HTML Code Formatting

- The HTML `<code>` element defines **programming code sample**:

Example

```
<code>
var person = { firstName:"John", lastName:"Doe", age:50, eyeColor:"blue" }
</code>
```

- The `<code>` element does **not** preserve extra **whitespace** and **line-breaks**:

Example

```
<p>Coding Example:</p>
<code>
var person = {
  firstName:"John",
  lastName:"Doe",
  age:50,
  eyeColor:"blue"
}
</code>
```

- To fix this, you must wrap the code in a `<pre>` element:

Example

```
<p>Coding Example:</p>
<code>
<pre>
var person = {
  firstName:"John",
  lastName:"Doe",
  age:50,
  eyeColor:"blue"
}
</pre>
</code>
```

HTML Variable Formatting

- The HTML **<var>** element defines a **mathematical variable**:

Example

```
<p>Einstein wrote:</p>
<p><var>E = m c<sup>2</sup></var></p>
```

11.HTML Comments**HTML Comment Tags**

- You can add comments to your HTML source by using the following syntax:
- **<!--** Write your comments here **-->**
- **Note:** There is an exclamation point (!) in the opening tag, but not in the closing tag.
- Comments are not displayed by the browser, but they can help document your HTML.
- With comments you can place notifications and reminders in your HTML:

Example

```
<!-- This is a comment -->
<p>This is a paragraph.</p>
<!-- Remember to add more information here -->
```

- Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

Example

```
<!-- Do not display this at the moment

-->
```

12.HTML Links**HTML Links - Hyperlinks**

- A hyperlink is an element, a text, or an image that you can click on, and jump to another document.

HTML Links - Syntax

- In HTML, links are defined with the **<a>** tag:

Link Syntax:

```
<a href="url">link text</a>
```

Example:

```
<a href="http://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

- The **href** attribute specifies the destination address (<http://www.w3schools.com/html/>)
- The **link text** is the visible part (Visit our HTML tutorial).
- Clicking on the link text, will send you to the specified address.
- The link text does not have to be text. It can be an HTML image or any other HTML element

Local Links

- The example above used an absolute URL (A full web address).
- A local link (link to the same web site) is specified with a relative URL (without <http://www....>).

Example:

```
<a href="html_images.php">HTML Images</a>
```

HTML Links - Colors and Icons

- When you move the mouse cursor over a link, two things will normally happen:
 1. The mouse arrow will turn into a little hand
 2. The color of the link element will change

By default, links will appear as this in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red
- You can change the default colors, using styles:

HTML Links - The target Attribute

- The **target** attribute specifies where to open the linked document.
- This example will open the linked document in a new browser window or in a new tab:

Example

```
<a href="http://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

Target Value	Description
<code>_blank</code>	Opens the linked document in a new window or tab
<code>_self</code>	Opens the linked document in the same frame as it was clicked (this is default)
<code>_parent</code>	Opens the linked document in the parent frame
<code>_top</code>	Opens the linked document in the full body of the window
<code>framename</code>	Opens the linked document in a named frame

- If your webpage is locked in a frame, you can use `target="_top"` to break out of the frame:

Example

```
<a href="http://www.w3schools.com/html/" target="_top">HTML5 tutorial!</a>
```

HTML Links - Image as Link

- It is common to use images as links:
- `border:0` is added to prevent IE9 (and earlier) from displaying a border around the image

Example

```
<a href="default.asp">
   </a>
```

13.HTML Images

HTML Images Syntax

- In HTML, images are defined with the **** tag.
- The **** tag is empty, it contains attributes only, and does not have a closing tag.
- The **src** attribute defines the url (web address) of the image:

```

```

The alt Attribute

- The **alt** attribute specifies an alternate text for the image, if it cannot be displayed.
- The value of the alt attribute should describe the image in words:

Example

```

```

- The alt attribute is **required**. A web page will not validate correctly without it.

HTML Screen Readers

- Screen readers are software programs that can read what is displayed on a screen.
- Used on the web, screen readers can "reproduce" HTML as text-to-speech, sound icons, or braille output.
- Screen readers are used by people who are blind, visually impaired, or learning disabled.
- Screen readers can read the **alt** attribute.

Image Size - Width and Height

- Always specify image size. If the size is unknown, the page will flicker while the image loads.
- You can use the **style** attribute to specify the **width** and **height** of an image.
- The values are specified in pixels (use px after the value):

Example

```

```

- Alternatively, you can use width and height **attributes**.
- The values are specified in pixels (without px after the value):

Example

```

```

Width and Height or Style?

- Both the width, the height, and the style attributes, are valid in the latest HTML5 standard.
- We suggest you use the style attribute. It prevents styles sheets from changing the default size of images:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
  img { width:100%; }
</style>
</head>
<body>


</body>
</html>
```

Images in another Folder

- If not specified, the browser expects to find the image in the same folder as the web page.
- However, it is common on the web, to store images in a sub-folder, and refer to the folder in the image name:

Example

```

```

- If a browser cannot find an image, it will display a broken link icon:

Example

```

```

Images on another Server

- Some web sites store their images on image servers.
- Actually, you can access images from any web address in the world:

Example

```

```

Animated Images

- The GIF standard allows animated images:

Example

```

```

- Note that the syntax of inserting animated images is no different from non-animated images.
-

Image Maps

- For an image, you can create an image map, with clickable areas:

Example

```

<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>
```

Image Floating

- You can let an image float to the left or right of a paragraph:

Example

```
<p>

A paragraph with an image. The image floats to the left of the text.
</p>
```

- Loading images takes time. Large images can slow down your page. Use images carefully

14.HTML Tables

HTML Table Example:

Number	First Name	Last Name	Points
1	Helen	Bezabih	94
2	Yohannes	Girma	80
3	Alem	G/mariam	67
4	Tigist	Worku	50

Defining HTML Tables:

Example:

```
<table style="width:100%">
<tr>
<td>Helen</td>
<td>Bezabih</td>
<td>94</td>
</tr>
<tr>
<td>Yohannes</td>
<td>Girma</td>
<td>80</td>
</tr>
</table>
```

Example explained:

- Tables are defined with the **<table>** tag.
- Tables are divided into **table rows** with the **<tr>** tag.
- Table rows are divided into **table data** with the **<td>** tag.
- A table row can also be divided into **table headings** with the **<th>** tag.
- Table data **<td>** are the data containers of the table
- They can contain all sorts of HTML elements like text, images, lists, other tables, etc

An HTML Table with a Border Attribute:

- If you do not specify a border for the table, it will be displayed without borders.
- A border can be added using the border attribute:

Example

```
<table border="1" style="width:100%">
  <tr>
    <td>Helen</td>
    <td>Bezabih</td>
    <td>94</td>
  </tr>
  <tr>
    <td>Yohannes</td>
    <td>Girma</td>
    <td>80</td>
  </tr>
</table>
```

- The border attribute is on its way out of the HTML standard! It is better to use CSS
- To add borders, use the **CSS border** property:

Example:

```
table, th, td
{
  border: 1px solid black;
}
```

- Remember to define borders for both the table and the table cells.

An HTML Table with Collapsed Borders

- If you want the borders to collapse into one border, add **CSS border-collapse**:

Example:

```
table, th, td
{
  border: 1px solid black;
  border-collapse: collapse;
}
```

An HTML Table with Cell Padding

- Cell padding specifies the space between the cell content and its borders.
- If you do not specify padding, the table cells will be displayed without padding.
- To set the padding, use the **CSS padding** property:

Example:

```
table, th, td
{
  border: 1px solid black;
  border-collapse: collapse;
}
th,td {
  padding: 15px;
}
```

HTML Table Headings

- Table headings are defined with the **<th>** tag.
- By default, all major browsers display table headings as bold and centered:

Example

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

- To left-align the table headings, use the **CSS text-align** property:

Example:

```
Th
{
  text-align: left;
}
```

An HTML Table with Border Spacing

- Border spacing specifies the space between the cells.
- To set the border spacing for a table, use the **CSS border-spacing** property:

Example

```
Table
{
  border-spacing: 5px;
}
```

- If the table has collapsed borders, border-spacing has no effect

Table Cells that Span Many Columns

- To make a cell span more than one column, use the **colspan** attribute:

Example:

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>555 77 854</td>
    <td>555 77 855</td>
  </tr>
</table>
```

Table Cells that Span Many Rows

- To make a cell span more than one row, use the **rowspan** attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>First Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>555 77 854</td>
  </tr>
  <tr>
    <td>555 77 855</td>
  </tr>
</table>
```

An HTML Table with a Caption

- To add a caption to a table, use the **<caption>** tag:

Example:

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

- The **<caption>** tag must be inserted immediately after the **<table>** tag

Different Styles for Different Tables

- Most of the examples above use a style attribute (width="100% ") to define the width of each table.
- This makes it easy to define different widths for different tables.
- The styles in the **<head>** section, however, define a style for all tables in a page.
- To define a special style for a special table, add an **id attribute** to the table:

Example

```
<table id="t01">
  <tr>
    <th>Firstname</th>
```

```

<th>Lastname</th>
<th>Points</th>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
</table>

```

Now you can define a different style for this table:

```

table#t01
{
width: 100%;
background-color: #f1f1c1;
}

```

And add more styles:

```

table#t01 tr:nth-child(even)
{
background-color: #eee;
}
table#t01 tr:nth-child(odd)
{
background-color: #fff;
}
table#t01 th
{
color: white;
background-color: black;
}

```

15.HTML Lists

HTML can have Unordered Lists, Ordered Lists, or Description Lists:

Unordered HTML List	Ordered HTML List	HTML Description List
<ul style="list-style-type: none"> • The first item • The second item • The third item • The fourth item 	<ol style="list-style-type: none"> 1. The first item 2. The second item 3. The third item 4. The fourth item 	<p>The first item Description of item</p> <p>The second item Description of item</p>

Unordered HTML Lists

- An unordered list starts with the **** tag. Each list item starts with the **** tag.
- The list items will be marked with bullets (small black circles).

Unordered List:

```

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

```

Unordered HTML Lists - The Style Attribute

- A **style** attribute can be added to an **unordered list**, to define the style of the marker:

Style	Description
list-style-type:disc	The list items will be marked with bullets (default)
list-style-type:circle	The list items will be marked with circles
list-style-type:square	The list items will be marked with squares
list-style-type:none	The list items will not be marked

Disc:

```
<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea
  <li>Milk</li>
</ul>
```

Circle:

```
<ul style="list-style-type:circle">
  <li>Coffee</li>
  <li>Tea
  <li>Milk</li>
</ul>
```

Square:

```
<ul style="list-style-type:square">
  <li>Coffee</li>
  <li>Tea
  <li>Milk</li>
</ul>
```

None:

```
<ul style="list-style-type:none">
  <li>Coffee</li>
  <li>Tea
  <li>Milk</li>
</ul>
```

- Using a type attribute `<ul type="disc">`, instead of `<ul style="list-style-type:disc">`, also works
- But in **HTML5**, the type attribute is not valid in unordered lists, only in ordered list

Ordered HTML Lists

- An ordered list starts with the `` tag. Each list item starts with the `` tag.
- The list items will be marked with numbers.

Ordered List:

```
<ol>
  <li>Coffee</li>
  <li>Milk</li>
</ol>
```

Ordered HTML Lists - The Type Attribute

- A **type** attribute can be added to an **ordered list**, to define the type of the marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Numbers:

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

Upper Case:

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea
  <li>Milk</li>
</ol>
```

Lower Case:

```
<ol type="a">
  <li>Coffee</li>
  <li>Tea
  <li>Milk</li>
</ol>
```

Roman Upper Case:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea
  <li>Milk</li>
</ol>
```

Roman Lower Case:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea
  <li>Milk</li>
</ol>
```

HTML Description Lists

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

Description List:

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

Nested HTML Lists

- List can be nested (lists inside lists).

Nested Lists:

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

```

    <li>Green tea</li>
  </ul>
</li>
<li>Milk</li>
</ul>

```

- List items can contain new list, and other HTML elements, like images and links, etc

Horizontal Lists

- HTML lists can be styled in many different ways with CSS.
- One popular way, is to style a list to display horizontally:

Horizontal List:

```

<!DOCTYPE html>
<html>
<head>
<style>
ul#menu li
{
  display:inline;
}
</style>
</head>
<body>
<h2>Horizontal List</h2>
<ul id="menu">
  <li>Apples</li>
  <li>Bananas</li>
  <li>Lemons</li>
  <li>Oranges</li>
</ul>
</body>
</html>

```

16. HTML Block Elements

- Most HTML elements are defined as **block level** elements or **inline** elements.
- Block level elements normally start (and end) with a new line, when displayed in a browser.
 - **Examples:** <h1>, <p>, , <table>
- Inline elements are normally displayed without line breaks.
 - **Examples:** , <td>, <a>,

The HTML <div> Element

- The HTML <div> element is a **block level element** that can be used as a container for other HTML elements.
- The <div> element has no special meaning. It has no required attributes, but **style** and **class** are common.
- Because it is a block level element, the browser will display line breaks before and after it.
- When used together with CSS, the <div> element can be used to style **blocks of content**.

The HTML Element

- The HTML `` element is an **inline element** that can be used as a container for text.
- The `` element has no special meaning. It has no required attributes, but **style** and **class** are common.
- Unlike `<div>`, which is formatted with line breaks, the `` element does not have any automatic formatting.
- When used together with CSS, the `` element can be used to style **parts of the text**:

Example

```
<h1>My <span style="color:red">Important</span>Heading</h1>
```

17.HTML Layouts

- Websites often display content in multiple columns (like a magazine or newspaper).

HTML Layout using <div> Elements

- The `<div>` element is often used as a layout tool, because it can easily be positioned with CSS
- This example uses 4 `<div>` elements to create a multiple column layout:

Example

```
<body>
<div id="header">
<h1>City Gallery</h1>
</div>
<div id="nav">
London<br>
Paris<br>
Tokyo<br>
</div>
<div id="section">
<h1>London</h1>
<p>
London is the capital city of England. It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.
</p>
<p>
Standing on the River Thames, London has been a major settlement for two millennia,
its history going back to its founding by the Romans, who named it Londinium.
</p>
</div>
<div id="footer">
Copyright © W3Schools.com
</div>
</body>
```

The CSS:

```
<style>
#header
{
background-color:black;
color:white;
text-align:center;
```

```

padding:5px;
}
#nav
{
line-height:30px;
background-color:#eeeeee;
height:300px;
width:100px;
float:left;
padding:5px;
}
#section
{
width:350px;
float:left;
padding:10px;
}
#footer
{
background-color:black;
color:white;
clear:both;
text-align:center;
padding:5px;
}
</style>

```

Website Layout Using HTML5

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



header	Defines a header for a document or a section
nav	Defines a container for navigation links
section	Defines a section in a document
article	Defines an independent self-contained article
aside	Defines content aside from the content (like a sidebar)
footer	Defines a footer for a document or a section
details	Defines additional details
summary	Defines a heading for the details element

- This example uses <header>, <nav>, <section>, and <footer> to create a multiple column layout:

Example:

```

<body>
<header>

```

```

<h1>City Gallery</h1>
</header>
<nav>
London<br>
Paris<br>
Tokyo<br>
</nav>
<section>
<h1>London</h1>
<p>
London is the capital city of England. It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.
</p>
<p>
Standing on the River Thames, London has been a major settlement for two millennia,
its history going back to its founding by the Romans, who named it Londinium.
</p>
</section>
<footer>
Copyright © W3Schools.com
</footer>
</body>

```

The CSS:

```

<style>
header {
  background-color:black;
  color:white;
  text-align:center;
  padding:5px;
}
nav {
  line-height:30px;
  background-color:#eeeeee;
  height:300px;
  width:100px;
  float:left;
  padding:5px;
}
section {
  width:350px;
  float:left;
  padding:10px;
}
footer {
  background-color:black;
  color:white;
  clear:both;
  text-align:center;
  padding:5px;
}

```

</style>

18.HTML Multimedia

- Multimedia comes in many different formats. It can be almost anything you can hear or see. Examples: Images, music, sound, videos, records, films, animations, and more.
- Web pages often contain multimedia elements of different types and formats.

HTML5 Video

- Before HTML5, a video could only be played in a browser with a plug-in (like flash).
- The HTML5 **<video>** element specifies a standard way to embed a video in a web page.
- Only **MP4**, **WebM**, and **Ogg** video are supported by the HTML5 standard.

Example

```
<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.
- It is a good idea to always include **width** and **height** attributes. If height and width are not set, the page might flicker while the video loads.
- 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.

HTML **<video>** Autoplay

- To start a video automatically use the **autoplay** attribute:

Example

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

HTML Video - Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML5 Audio

- Before HTML5, audio files could only be played in a browser with a plug-in (like flash).
- The HTML5 **<audio>** element specifies a standard way to embed audio in a web page.
- Only **MP3**, **WAV**, and **Ogg** audio are supported by the HTML5 standard

Example

```
<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>

HTML Audio - How It Works

- The **controls** attribute adds 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 only be displayed in browsers that do not support the **<audio>** element.

HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

19.HTML Forms and Input

HTML Forms

- HTML Forms are used to select different kinds of user input
- HTML forms are used to pass data to a server.
- An HTML form can contain input elements like **text fields**, **checkboxes**, **radio-buttons**, **submit buttons** and more. A form can also contain **select lists**, **textarea**, **fieldset**, **legend**, and **label** elements.
- The **<form>** tag is used to create an HTML form:

<form>

input elements

</form>

HTML Forms - The Input Element

- The most important form element is the **<input>** element.
- The **<input>** element is used to select user information.
- An **<input>** element can vary in many ways, depending on the type attribute. An **<input>** element can be of type **text field**, **checkbox**, **password**, **radio button**, **submit button**, and more.
- The most common input types are described below.

Text Fields

- **<input type="text">** defines a one-line input field that a user can enter text into:

<form>

First name: <input type="text" name="firstname">

Last name: <input type="text" name="lastname">

</form>

- How the HTML code above looks in a browser:

First name:

Last name:

Note: The form itself is not visible. Also note that the default size of a text field is 20 characters.

Password Field

- `<input type="password">` defines a password field:

```
<form>
```

```
Password: <input type="password" name="pwd">
```

```
</form>
```

How the HTML code above looks in a browser:

Password:

Note: The characters in a password field are masked (shown as asterisks or circles).

Radio Buttons

- `<input type="radio">` defines a radio button.
- Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<form>
```

```
<input type="radio" name="sex" value="male">Male<br>
```

```
<input type="radio" name="sex" value="female">Female
```

```
</form>
```

- How the HTML code above looks in a browser:



Male



Female

Checkboxes

- `<input type="checkbox">` defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices.

```
<form>
```

```
<input type="checkbox" name="vehicle" value="Bike">I have a bike<br>
```

```
<input type="checkbox" name="vehicle" value="Car">I have a car
```

```
</form>
```

- How the HTML code above looks in a browser:



I have a bike



I have a car

Submit Button

- `<input type="submit">` defines a submit button.
- A submit button is used to send form data to a server.
- The data is sent to the page specified in the form's action attribute.
- The file defined in the action attribute usually does something with the received input:


```
<form name="input" action="demo_form_action.php" method="get">
Username: <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

- How the HTML code above looks in a browser:

Username:

- If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "demo_form_action.php". The page will show you the received input.

Input Type Reset

- **<input type="reset">** defines a **reset button** that will reset all form values to their default values:

Example

```
<form action="/action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey"><br>
Last name:<br>
<input type="text" name="lastname" value="Mouse"><br><br>
<input type="submit" value="Submit">
<input type="reset">
</form>
```

Textarea example

```
<textarea rows="4" cols="50">
At w3schools.com you will learn how to make a website. We offer free tutorials in all web
development technologies.
</textarea>
```

Fieldset Example

```
<form>
<fieldset>
<legend>Personalia:</legend>
Name: <input type="text"><br>
Email: <input type="text"><br>
Date of birth: <input type="text">
</fieldset>
</form>
```

Label example

```
<form action="demo_form.asp">
<label for="male">Male</label>
<input type="radio" name="sex" id="male" value="male"><br>
<label for="female">Female</label>
<input type="radio" name="sex" id="female" value="female"><br>
<input type="submit" value="Submit">
</form>
```

Select example

```
<select>
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
```

```
<option value="mercedes">Mercedes</option>
<option value="audi">Audi</option>
</select>
```

Optgroup 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>
</select>
```

Datalist example

```
<input list="browsers">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

20.HTML5 Input Types

HTML5 added several new input types:

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

N.B 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

```
<form>
  Select your favorite color:
```

```
<input type="color" name="favcolor">
</form>
```

Input Type Date:

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

Example

```
<form>
  Birthday:
  <input type="date" name="bday">
</form>
```

- You can also add restrictions to dates:

Example

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

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

```
<form>
  Birthday (date and time):
  <input type="datetime-local" name="bdaytime">
</form>
```

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.
- Some smartphones recognize the email type, and adds ".com" to the keyboard to match email input.

Example

```
<form>
  E-mail:
  <input type="email" name="email">
</form>
```

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.

Example

```
<form>
  Birthday (month and year):
```

```
<input type="month" name="bdaymonth">
</form>
```

Input Type Number:

- The `<input type="number">` defines a **numeric** input field.
- You can also set restrictions on what numbers are accepted.
- The following example displays a numeric input field, where you can enter a value from 1 to 5:

Example

```
<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
</form>
```

Input Restrictions

- Here is a list of some common input restrictions (some are new in HTML5):

Attribute	Description
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

- The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

Example

```
<form>
  Quantity:
  <input type="number" name="points" min="0" max="100" step="10" value="30">
</form>
```

Input Type Range:

- The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes:

Example

```
<form>
  <input type="range" name="points" min="0" max="10">
</form>
```

Input Type Search:

- The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

Example

```
<form>
  Search Google:
  <input type="search" name="googlesearch">
</form>
```

Input Type Tel:

- The `<input type="tel">` is used for input fields that should contain a telephone number.
- The tel type is currently supported only in Safari 8.

Example:

```
<form>
  Telephone:
  <input type="tel" name="usrtel">
</form>
```

Input Type Time:

- The `<input type="time">` allows the user to select a time (no time zone).
- Depending on browser support, a time picker can show up in the input field.

Example:

```
<form>
  Select a time:
  <input type="time" name="usr_time">
</form>
```

Input Type Url:

- The `<input type="url">` is used for input fields that should contain a URL address.
- Depending on browser support, the url field can be automatically validated when submitted.
- Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

Example

```
<form>
  Add your homepage:
  <input type="url" name="homepage">
</form>
```

Input Type Week:

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

Example

```
<form>
  Select a week:
  <input type="week" name="week_year">
</form>
```

21.HTML Input Attributes

The value Attribute

- The **value** attribute specifies the initial value for an input field:

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John">
</form>
```

The readonly Attribute

- The **readonly** attribute specifies that the input field is read only (cannot be changed):

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" readonly>
</form>
```

The disabled Attribute

- The **disabled** attribute specifies that the input field is disabled.
- A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form:

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" disabled>
</form>
```

The size Attribute

- The **size** attribute specifies the size (in characters) for the input field:

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" size="40">
</form>
```

The maxlength Attribute

- The **maxlength** attribute specifies the maximum allowed length for the input field:

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" maxlength="10">
</form>
```

- With a maxlength attribute, the input field will not accept more than the allowed number of characters.
- The maxlength attribute does not provide any feedback. If you want to alert the user, you must write JavaScript code.

Note: Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input.

- To safely restrict input, it must be checked by the receiver (the server) as well!

22.HTML5 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 (regex)
- placeholder
- required
- step
- spellcheck
- contenteditable

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 complete the input values based on values that the user has entered before.

Tip: 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**.

Example

```
<form action="/action_page.php" autocomplete="on">
  First name:<input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  E-mail: <input type="email" name="email" autocomplete="off"><br>
  <input type="submit">
</form>
```

Tip: In some browsers you may need to activate the autocomplete function for this to work.

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.php" 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.

Example

- Let the "First name" input field automatically get focus when the page loads:

First name: <input type="text" name="fname" autofocus>

The form Attribute

- The **form** attribute specifies one or more forms an <input> element belongs to.

Tip: 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"><br>
  <input type="submit" value="Submit">
</form>
Last name: <input type="text" name="lname" form="form1">
```

The formation Attribute

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

Example

- An HTML form with two submit buttons, with different actions:

```
<form action="/action_page.php">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit"><br>
  <input type="submit" formation="/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".

Example

- The second submit button overrides the HTTP method of the form:

```
<form action="/action_page.php" method="get">
  First name: <input type="text" name="fname"><br>
```



```
Last name: <input type="text" name="lname"><br>
<input type="submit" value="Submit">
<input type="submit" formmethod="post" formaction="action_page_post.php"
value="Submit using POST">
</form>
```

The formnovalidate Attribute

- The **formnovalidate** attribute overrides the novalidate attribute of the <form> element.
- The formnovalidate attribute can be used with type="submit".

Example

- A form with two submit buttons (with and without validation):
<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 formtarget Attribute

- The **formtarget** attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.
- The formtarget attribute overrides the target attribute of the <form> element.
- The formtarget attribute can be used with type="submit" and type="image".

Example

- A form with two submit buttons, with different target windows:
<form action="/action_page.php">
First name: <input type="text" name="fname">

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

<input type="submit" value="Submit as normal">
<input type="submit" formtarget="_blank" value="Submit to a new window">
</form>

The list Attribute

- The **list** attribute refers to a <datalist> element that contains pre-defined options for an <input> element.

Example

- An <input> element with pre-defined values in a <datalist>:
<input list="browsers">
<datalist id="browsers">
<option value="Internet Explorer">
<option value="Firefox">
<option value="Chrome">
<option value="Opera">

```
<option value="Safari">
</datalist>
```

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.

Example

Select images: `<input type="file" name="img" multiple>`

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.

Tip: Use the global [title](#) attribute to describe the pattern to help the user.

Tip: Learn more about [regular expressions](#).

Example

- An input field that can contain only three letters (no numbers or special characters):

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.

Example

`<input type="text" name="fname" placeholder="First name">`

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.

Example

Username: `<input type="text" name="username" required>`

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.

Tip: 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.

Example

```
<input type="number" name="points" step="3">
```

HTML spellcheck Attribute

- The spellcheck attribute specifies whether the element is to have its spelling and grammar checked or not.

Example

- An editable paragraph with spellcheck:

```
<p contenteditable="true" spellcheck="true">This is a paragraph.</p>
```

The following can be spellchecked:

- Text values in input elements (not password)
- Text in <textarea> elements
- Text in editable elements

HTML contenteditable Attribute

- The contenteditable attribute specifies whether the content of an element is editable or not.

Example

- An editable paragraph

```
<p contenteditable="true">This is an editable paragraph.</p>
```

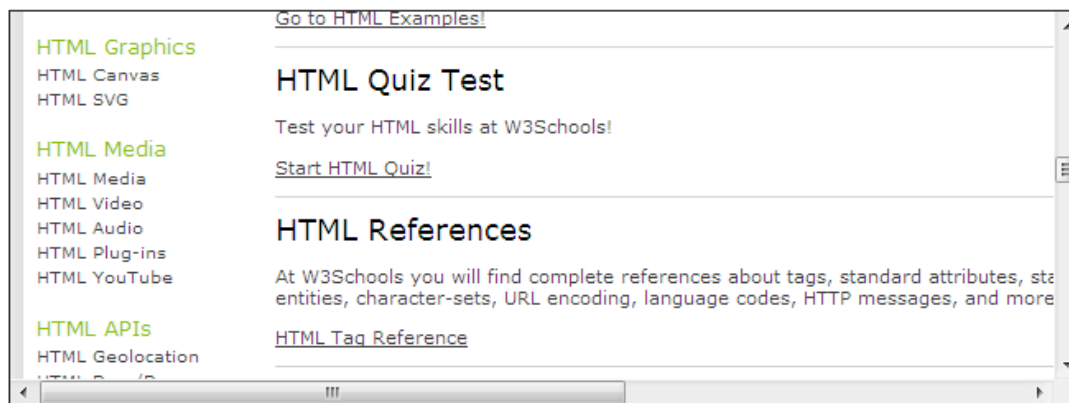
23.HTML Iframes

- An iframe is used to display a web page within a web page.

Iframe Syntax

- The syntax for adding an iframe is:

```
<iframe src="URL"></iframe>
```
- The **src** attribute specifies the URL (web address) of the iframe page.



Iframe - Set Height and Width

- Use the **height** and **width** attributes to specify the size.
- The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

Example

```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

Iframe - Remove the Border

- The **frameborder** attribute specifies whether or not to display a border around the iframe.
- Set the attribute value to "0" to remove the border:

Example

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

Use iframe as a Target for a Link

- An iframe can be used as the target frame for a link.
- The **target** attribute of the link must refer to the **name** attribute of the iframe:

Example

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
<p><a href="http://www.w3schools.com"
target="iframe_a">W3Schools.com</a></p>
```

24.HTML Scripts

- JavaScripts make HTML pages more dynamic and interactive.
- The `<script>` tag is used to define a client-side script, such as a JavaScript.
- The `<script>` element either contains scripting statements or it points to an external script file through the `src` attribute.
- Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.
- The script below writes Hello JavaScript! into an HTML element with `id="demo"`:

Example

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

The HTML `<noscript>` Tag

- The `<noscript>` tag is used to provide an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support client-side scripting.
- The `<noscript>` element can contain all the elements that you can find inside the `<body>` element of a normal HTML page.
- The content inside the `<noscript>` element will only be displayed if scripts are not supported, or are disabled in the user's browser:

Example

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

25.HTML Head

The HTML `<head>` Element

- The `<head>` element is a container for all the head elements. Elements inside `<head>` can include scripts, instruct the browser where to find style sheets, provide meta information, and more.

- The following tags can be added to the head section: **<title>**, **<style>**, **<meta>**, **<link>**, **<script>**, **<noscript>**, and **<base>**.
-

The <title> element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine results

A simplified HTML document:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>
<body>
The content of the document.....
</body>
</html>
```

The HTML <base> Element

- The <base> tag specifies the base URL/target for all relative URLs in a page:

Example

```
<head>
<base href="http://www.w3schools.com/images/" target="_blank">
</head>
```

The HTML <link> Element

- The <link> tag defines the relationship between a document and an external resource.
- The <link> tag is most used to link to style sheets:

Example

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

The HTML <style> Element

- The <style> tag is used to define style information for an HTML document.
- Inside the <style> element you specify how HTML elements should render in a browser:

Example

```
<head>
<style>
body {
background-color:yellow;
}
p {
color:blue;
}
</style>
</head>
```

The HTML <meta> Element

- The <meta> tag provides metadata about the HTML document.
- Metadata will not be displayed on the page, but will be machine parsable.
- Meta elements are typically used to specify **page description, keywords, author of the document, last modified, and other metadata**.
- The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.
- <meta> tags always go inside the <head> element.

<meta> Tags - Examples of Use

- Define keywords for search engines:

Example

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

- Define a description of your web page:

Example

```
<meta name="description" content="Free Web tutorials on HTML and CSS">
```

- Define the author of a page:

Example

```
<meta name="author" content="Selama G.">
```

- Refresh document every 30 seconds:

Example

```
<meta http-equiv="refresh" content="30">
```

26.HTML Entities

- **Reserved characters** in HTML must be replaced with character entities.
 - Characters, **not present on your keyboard**, can also be replaced by entities.
-

HTML Entities

- If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags.
 - Character entities are used to display reserved characters in HTML.
 - A character entity looks like this:
&entity_name;
OR
&#entity_number;
 - To display a less than sign we must write: **<** or **<**;
 - The advantage of using an entity name, instead of a number, is that the name is easier to remember.
 - The disadvantage is that browsers may not support all entity names, but the support for numbers is good.
-

Non Breaking Space

- A common character entity used in HTML is the non breaking space ().
 - Remember that browsers will always truncate spaces in HTML pages.
 - If you write 10 spaces in your text, the browser will remove 9 of them. To add real spaces to your text, you can use the ** ** character entity.
-

Some Other Useful HTML Character Entities

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

- Entity names are case sensitive

27.HTML Symbols

HTML Symbol Entities

- Many mathematical, technical, and currency symbols, are not present on a normal keyboard.
- To add these symbols to an HTML page, you can use an HTML **entity name**.
- If no entity name exists, you can use an entity number; a **decimal (or hexadecimal)** reference.
- If you use an HTML entity name or a hexadecimal number, the character will always display correctly
- This is independent of what character set (encoding) your page uses!

Example:

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>
```

Will display as:

I will display €
 I will display €
 I will display €

Some Mathematical Symbols Supported by HTML

Char	Number	Entity	Description
∀	∀	∀	FOR ALL
∂	∂	∂	PARTIAL DIFFERENTIAL
∃	∃	∃	THERE EXISTS
∅	∅	∅	EMPTY SETS
∇	∇	∇	NABLA
∈	∈	∈	ELEMENT OF
∉	∉	∉	NOT AN ELEMENT OF
⊃	∋	∋	CONTAINS AS MEMBER
∏	∏	∏	N-ARY PRODUCT
∑	∑	∑	N-ARY SUMMATION

Some Greek Letters Supported by HTML

Char	Number	Entity	Description
A	Α	Α	GREEK CAPITAL LETTER ALPHA
B	Β	Β	GREEK CAPITAL LETTER BETA
Γ	Γ	Γ	GREEK CAPITAL LETTER GAMMA
Δ	Δ	Δ	GREEK CAPITAL LETTER DELTA
E	Ε	Ε	GREEK CAPITAL LETTER EPSILON
Z	Ζ	Ζ	GREEK CAPITAL LETTER ZETA

Some Other Entities Supported by HTML

Char	Number	Entity	Description
©	©	©	COPYRIGHT SIGN
®	®	®	REGISTERED SIGN
€	€	€	EURO SIGN
™	™	™	TRADEMARK
←	←	←	LEFTWARDS ARROW
↑	↑	↑	UPWARDS ARROW
→	→	→	RIGHTWARDS ARROW
↓	↓	↓	DOWNWARDS ARROW
♠	♠	♠	BLACK SPADE SUIT
♣	♣	♣	BLACK CLUB SUIT
♥	♥	♥	BLACK HEART SUIT
♦	♦	♦	BLACK DIAMOND SUIT

28. Complete HTML Tags

5= New in HTML5.

R= Removed from HTML 5

Tag	Description
<!--...-->	Defines a comment
<!DOCTYPE>	Defines the document type
<a>	Defines a hyperlink
<abbr>	Defines an abbreviation
<acronym> R	Not supported in HTML5. Use <abbr> instead. Defines an acronym
<address>	Defines contact information for the author/owner of a document
<applet> R	Not supported in HTML5. Use <object> instead. Defines an embedded applet
<area>	Defines an area inside an image-map
<article> 5	Defines an article
<aside> 5	Defines content aside from the page content
<audio> 5	Defines sound content
	Defines bold text
<base>	Specifies the base URL/target for all relative URLs in a document
<basefont> R	Not supported in HTML5. Use CSS instead. Specifies a default color, size, and font for all text in a document
<bdi> 5	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> R	Not supported in HTML5. Use CSS instead. Defines big text
<blockquote>	Defines a section that is quoted from another source
<body>	Defines the document's body

	Defines a single line break
<button>	Defines a clickable button
<canvas> 5	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<caption>	Defines a table caption
<center> R	Not supported in HTML5. Use CSS instead. Defines centered text
<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
<datalist> 5	Specifies a list of pre-defined options for input controls
<dd>	Defines a description/value of a term in a description list
	Defines text that has been deleted from a document
<details> 5	Defines additional details that the user can view or hide
<dfn>	Defines a definition term
<dialog> 5	Defines a dialog box or window
<dir> R	Not supported in HTML5. Use instead. Defines a directory list
<div>	Defines a section in a document

<u><dl></u>	Defines a description list
<u><dt></u>	Defines a term/name in a description list
<u></u>	Defines emphasized text
<u><embed></u> 5	Defines a container for an external (non-HTML) application
<u><fieldset></u>	Groups related elements in a form
<u><figcaption></u> 5	Defines a caption for a <figure> element
<u><figure></u> 5	Specifies self-contained content
<u></u> R	Not supported in HTML5. Use CSS instead. Defines font, color, and size for text
<u><footer></u> 5	Defines a footer for a document or section
<u><form></u>	Defines an HTML form for user input
<u><frame></u> R	Not supported in HTML5. Defines a window (a frame) in a frameset
<u><frameset></u> R	Not supported in HTML5. Defines a set of frames
<u><h1> to <h6></u>	Defines HTML headings
<u><head></u>	Defines information about the document
<u><header></u> 5	Defines a header for a document or section
<u><hgroup></u> 5	Defines a group of headings
<u><hr></u>	Defines a thematic change in the content
<u><html></u>	Defines the root of an HTML document
<u><i></u>	Defines a part of text in an alternate voice or mood
<u><iframe></u>	Defines an inline frame
<u></u>	Defines an image
<u><input></u>	Defines an input control
<u><ins></u>	Defines a text that has been inserted into a document
<u><kbd></u>	Defines keyboard input
<u><keygen></u> 5	Defines a key-pair generator field (for forms)
<u><label></u>	Defines a label for an <input> element
<u><legend></u>	Defines a caption for a <fieldset> element
<u></u>	Defines a list item
<u><link></u>	Defines the relationship between a document and an external resource (most used to link to style sheets)
<u><main></u> 5	Specifies the main content of a document
<u><map></u>	Defines a client-side image-map
<u><mark></u> 5	Defines marked/highlighted text
<u><menu></u>	Defines a list/menu of commands
<u><menuitem></u> 5	Defines a command/menu item that the user can invoke from a popup menu
<u><meta></u>	Defines metadata about an HTML document
<u><meter></u> 5	Defines a scalar measurement within a known range (a gauge)
<u><nav></u> 5	Defines navigation links
<u><noframes></u> R	Not supported in HTML5. Defines an alternate content for users that do not support frames
<u><noscript></u>	Defines an alternate content for users that do not support client-side scripts
<u><object></u>	Defines an embedded object
<u></u>	Defines an ordered list

<code><optgroup></code>	Defines a group of related options in a drop-down list
<code><option></code>	Defines an option in a drop-down list
<code><output></code> ⁵	Defines the result of a calculation
<code><p></code>	Defines a paragraph
<code><param></code>	Defines a parameter for an object
<code><pre></code>	Defines preformatted text
<code><progress></code> ⁵	Represents the progress of a task
<code><q></code>	Defines a short quotation
<code><rp></code> ⁵	Defines what to show in browsers that do not support ruby annotations
<code><rt></code> ⁵	Defines an explanation/pronunciation of characters (for East Asian typography)
<code><ruby></code> ⁵	Defines a ruby annotation (for East Asian typography)
<code><s></code>	Defines text that is no longer correct
<code><samp></code>	Defines sample output from a computer program
<code><script></code>	Defines a client-side script
<code><section></code> ⁵	Defines a section in a document
<code><select></code>	Defines a drop-down list
<code><small></code>	Defines smaller text
<code><source></code> ⁵	Defines multiple media resources for media elements (<code><video></code> and <code><audio></code>)
<code></code>	Defines a section in a document
<code><strike></code> ^R	Not supported in HTML5. Use <code></code> instead. Defines strikethrough text
<code></code>	Defines important text
<code><style></code>	Defines style information for a document
<code><sub></code>	Defines subscripted text
<code><summary></code> ⁵	Defines a visible heading for a <code><details></code> element
<code><sup></code>	Defines superscripted text
<code><table></code>	Defines a table
<code><tbody></code>	Groups the body content in a table
<code><td></code>	Defines a cell in a table
<code><textarea></code>	Defines a multiline input control (text area)
<code><tfoot></code>	Groups the footer content in a table
<code><th></code>	Defines a header cell in a table
<code><thead></code>	Groups the header content in a table
<code><time></code> ⁵	Defines a date/time
<code><title></code>	Defines a title for the document
<code><tr></code>	Defines a row in a table
<code><track></code> ⁵	Defines text tracks for media elements (<code><video></code> and <code><audio></code>)
<code><tt></code> ^R	Not supported in HTML5. Use CSS instead. Defines teletype text
<code><u></code>	Defines text that should be stylistically different from normal text
<code></code>	Defines an unordered list
<code><var></code>	Defines a variable
<code><video></code> ⁵	Defines a video or movie
<code><wbr></code> ⁵	Defines a possible line-break