HTML5

Presentation by Uplatz

Contact Us: https://training.uplatz.com/

Email: info@uplatz.com

Phone:+44 7836 212635

Uplatz

Table Of Contents:

- What is HTML
- Difference between HTML and HTML5
- Doctypes
- Headings
- Paragraphs
- Text Formatting
- Anchors and Hyperlinks
- Tables
- Lists

What is HTML?

- **HTML** stands for *Hyper Text Markup Language*.
- It is used to design web pages using a markup language.
- HTML is the combination of Hypertext and Markup language.
- Hypertext defines the link between the web pages.
- A markup language is used to define the text document within tag which defines the structure of web pages.
- This language is used to annotate (at the note for computer) text so that a machine can understand it and manipulate text accordingly.
- Most of the markup (e.g. HTML) languages are human readable.

- The language uses tags to define what manipulation has to be done on the text.
- It is used for structuring and presenting the content on the web pages. HTML5 is the fifth version of HTML.
- Many elements are removed or modified from HTML5

Why HTML5?

- Cross Browser Compatibility
- New DOCTYPE declaration
- Bring improvements in usability and user experience
- Clean markup and Improved Code
- Offline Browsing
- Video and Audio Support
- Geolocation Support

Difference Between HTML and HTML5:

HTML	HTML5
It didn’t support audio and video without the use of flash player support.	It supports audio and video controls with the use of <audio> and <video> tags.
It uses cookies to store temporary data.	It uses SQL databases and application cache to store offline data
Does not allow JavaScript to run in browser.	Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5
Vector graphics is possible in HTML with the help of various technologies such as VML, Silver-light, Flash, etc.	Vector graphics is additionally an integral a part of HTML5 like SVG and canvas.

HTML	HTML5
It does not allow drag and drop effects.	It allows drag and drop effects
Not possible to draw shapes like circle, rectangle, triangle etc	HTML5 allows to draw shapes like circle, rectangle, triangle etc
It works with all old browsers.	It supported by all new browser like Firefox, Mozilla, Chrome, Safari, etc
It can not handle inaccurate syntax	It is capable of handling inaccurate syntax.
Attributes like charset, async and ping are absent in HTML.	Attributes of charset, async and ping are a part of HTML 5.

Uplatz

Element insight:

The tag represents a common paragraph.

Elements commonly have an opening tag and a closing tag.

- The opening tag contains the element's name in angle brackets (<p>).
- The closing tag is identical to the opening tag with the addition of a forward slash (/) between the opening bracket and the element's name (</p>)
- Content can then go between these two tags: <p>This is a simple paragraph</p>.

Creating a simple page:

- The following HTML example creates a simple "Hello World" web page.
- HTML files can be created using any text editor.

- The files must be saved with a .html or .htm[2] extension in order to be recognized as HTML files.
- Once created, this file can be opened in any web browser.

```
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="UTF-8">
 <title>Hello!</title>
 </head>
 <body>
 <h1>Hello World!</h1>
 <p>This is a simple paragraph.</p>
 </body>
</html>
:
```

Simple page break down

Tag **Meaning**

<!DOCTYPE> Defines the HTML version used in the document. In this case it is HTML5.

<html> Opens the page. No markup should come after the closing tag (). The lang attribute declares the primary language of the page using the ISO language codes (en for English.

<head> Opens the head section, which does not appear in the main browser window but mainly contains information about the HTML document, called metadata. It can also contain imports from external stylesheets and scripts. The closing tag is </head> .

<meta> Gives the browser some metadata about the document. The charset attribute declares the character encoding. Modern HTML documents should always use UTF-8, even though it is not a requirement. In HTML, the tag does not require a closing tag

<title> The title of the page. Text written between this opening and the closing tag (</title>) will be displayed on the tab of the page or in the title bar of the browser.

<body> Opens the part of the document displayed to users, i.e. all the visible or audible content of a page. No content should be added after the closing tag**</body>**

<h1> A level 1 heading for the page.

<p> Represents a common paragraph of text.

Doctypes

- Doctypes - short for 'document type' - help browsers to understand the version of HTML the document is written in

for better interpretability.

- Doctype declarations are not HTML tags and belong at the very top of a document. This explains the structure and declaration of various doctypes in HTML.

Adding the Doctype:

The <!DOCTYPE> declaration should always be included at the top of the HTML document, before the <html> tag.

HTML 5 Doctype:

HTML5 is not based on SGML (Standard Generalized Markup Language), and therefore does not require a reference to a DTD (Document Type Definition).

HTML 5 Doctype declaration:

<!DOCTYPE html>

Case Insensitivity

A DOCTYPE must consist of the following components, in this order:

1. A string that is an ASCII case-insensitive match for the string "<!DOCTYPE".

therefore the following DOCTYPEs are also valid:

<!doctype html>

<!dOCtyPe html>

<!DocTYpe html>

Headings

- HTML provides not only plain paragraph tags, but six separate header tags to indicate headings of various sizes and thicknesses.
- Enumerated as heading 1 through heading 6, heading 1 has the largest and thickest text while heading6 is the smallest and thinnest, down to the paragraph level.

Using Headings

Headings can be used to describe the topic they precede and they are defined with the <h1> to <h6> tags. Headings

support all the global attributes.

<h1> defines the most important heading.

<h6> defines the least important heading.

Defining a heading:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Correct structure matters

- **Search engines** and other **user agents** usually index page content based on heading elements, for example to create a table of contents, so using the correct structure for headings is important.
- In general, an article should have one h1 element for the main title followed by h2 subtitles – going down a layer if
- necessary.
- If there are h1 elements on a higher level they shoudn't be used to describe any lower level content.

Example document (extra intendation to illustrate hierarchy):

```
<h1>Main title</h1>
<p>Introduction</p>
 <h2>Reasons</h2>
```

```
<h3>Reason 1</h3>
 <p>Paragraph</p>
 <h3>Reason 2</h3>
 <p>Paragraph</p>
 <h2>In conclusion</h2>
 <p>Paragraph</p>
```

Paragraphs

Column	Column
<p>	Defines a paragraph
<br>	Inserts a single line break
<pre>	Defines pre-formatted text

Paragraphs are the most basic HTML element.

HTML Paragraphs

The HTML <p> element defines a paragraph:

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

<p>This is another paragraph.</p>

- DisplayYou 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 any extra spaces and extra lines when the page is displayed:

<p>This is another paragraph, extra spaces will be removed by browsers</p>

Text Formatting

- While most HTML tags are used to create elements, HTML also provides in-text formatting tags to apply specific text-related styles to portions of text.
- This topic includes examples of HTML text formatting such as highlighting,bolding, underlining, subscript, and stricken text.

Highlighting:

- The <mark> element is new in HTML5 and is used to mark or highlight text in a document "due to its relevance in another context“.
- The most common example would be in the results of a search were the user has entered a search query and results are shown highlighting the desired query.

<p>Here is some content from an article that contains the <mark>searched query</mark>

that we are looking for. Highlighting the text will make it easier for the user to

find what they are looking for.</p>

Bold, Italic, and Underline:

Bold Text

To bold text, use the <strong> or <b> tags:

<strong>Bold Text Here</strong>

or

<b>Bold Text Here</b>

- **What's the difference?** Semantics. <strong> is used to indicate that the text is fundamentally or semantically
- important to the surrounding text, while <b> indicates no such importance and simply represents text that should be bolded.
- If you were to use <b> a text-to-speech program would not say the word(s) any differently than any of the other words around it - you are simply drawing attention to them without adding any additional importance.

By using

<strong>, though, the same program would want to speak those word(s) with a different tone of voice to convey that the text is important in some way.

Italic Text

To italicize text, use the <em> or <i> tags:

<em>Italicized Text Here</em>

or

<i>Italicized Text Here</i>

- What's the difference? Semantics. <em> is used to indicate that the text should have extra emphasis that should be stressed, while <i> simply represents text which should be set off from the normal text around it.

Underlined Text

- While the <u> element itself was deprecated in HTMl 4, it was reintroduced with alternate semantic meaning in HTML 5 - to represent an unarticulated, non-textual annotation.

<p>This paragraph contains some <u>mispelled</u> text.</p>

Abbreviation

To mark some expression as an abbreviation, use <abbr> tag:

<p>I like to write <abbr title="Hypertext Markup Language">HTML</abbr>!</p>

- If present, the title attribute is used to present the full description of such abbreviation.

Inserted, Deleted, or Stricken:

To mark text as inserted, use the <ins> tag:

<ins>New Text</ins>

To mark text as deleted, use the <del> tag:

<del>Deleted Text</del>

To strike through text, use the <s> tag:

<s>Struck-through text here</s>

Superscript and Subscript

To offset text either upward or downward you can use the tags <sup> and <sub>.

To create superscript:

<sup>superscript here</sup>

To create subscript:

<sub>subscript here</sub>

Anchors and Hyperlinks:

- Anchor tags are commonly used to link separate webpages, but they can also be used to link between different places in a single document, often within table of contents or even launch external applications.
- This topic explains the implementation and application of HTML anchor tags in various roles.

Link to another site

- This is the basic use of the <a> (anchor element) element:

<a href="http://example.com/">Link to example.com</a>

- It creates a hyperlink, to the URL http://example.com/ as specified by the href (hypertext reference) attribute, with the anchor text "Link to example.com". It would look something like the following:

- To denote that this link leads to an external website, you can use the external link type:

<a href="http://example.com/" rel="external">example site</a>

- You can link to a site that uses a protocol other than HTTP.

For example, to link to an FTP site, you can do,

<a href="ftp://example.com/">This could be a link to a FTP site</a>

- In this case, the difference is that this anchor tag is requesting that the user's browser connect to example.com
- using the File Transfer Protocol (FTP) rather than the Hypertext Transfer Protocol (HTTP)

Link to an anchor:

Anchors can be used to jump to specific tags on an HTML page. The <a> tag can point to any element that has an id attribute.

To learn more about IDs, visit the documentation about Classes and IDs.

Anchors are mostly used to jump to a subsection of a page and are used in conjunction with header tags.

Suppose you've created a page (page1.html) on many topics:

<h2>First topic</h2>

<p>Content about the first topic</p>

<h2>Second topic</h2>

<p>Content about the second topic</p>

- Once you have several sections, you may want to create a Table of Contents at the top of the page with quick-links (or bookmarks) to specific sections.
- If you gave an id attribute to your topics, you could then link to them

```
<h2 id="Topic1">First topic</h2>
<p>Content about the first topic</p>
<h2 id="Topic2">Second topic</h2>
<p>Content about the second topic</p>
```

Now you can use the anchor in your table of contents:

```
<h1>Table of Contents</h1>
 <a href='#Topic1'>Click to jump to the First Topic</a>
 <a href='#Topic2'>Click to jump to the Second Topic</a>
```

These anchors are also attached to the web page they're on (page1.html).

- So you can link across the site from one
- page to the other by referencing the page and anchor name.Remember, you can always

<a href="page1.html#Topic1">look back in the First Topic</a>

Link to a page on the same site:

<a href="/example">Text Here</a>

- The above example would go to the file example at the root directory (/) of the server.
- If this link was on http://example.com, the following two links would bring the user to the same location

<a href="/page">Text Here</a>

<a href="http://example.com/page">Text Here</a>

Both of the above would go to the page file at the root directory of example.com.

Link that dials a number

- If the value of the href-attribute begins with tel:, your device will dial the number when you click it. This works on
- mobile devices or on computers/tablets running software – like Skype or FaceTime – that can make phone calls.

<a href="tel:11234567890">Call us</a>

- Most devices and programs will prompt the user in some way to confirm the number they are about to dial.

Link that runs JavaScript

Simply use the javascript: protocol to run the text as JavaScript instead of opening it as a normal link:

<a href="javascript:myFunction();">Run Code</a>

You can also achieve the same thing using the onclick attribute:

```
<a href="#" onclick="myFunction(); return false;">Run Code</a>
```

The return false; is necessary to prevent your page from scrolling to the top when the link to # is clicked.

Make sure to include all code you'd like to run before it, as returning will stop execution of further code.

- Also noteworthy, you can include an exclamation mark ! after the hashtag in order to prevent the page from
- scrolling to the top.
- This works because any invalid slug will cause the link to not scroll anywhere on the page,
- because it couldn't locate the element it references (an element with id="!").

- You could also just use any invalid slug (such as #scrollsNowhere) to achieve the same effect. In this case, return false; is not required:

<a href="#!" onclick="myFunction();">Run Code</a>

Link that runs email client :

Basic usage :

If the value of the href-attribute begins with mailto: it will try to open an email client on click:

<a href="mailto:example@example.com">Send email</a>

- This will put the email address example@example.com as the recipient for the newly created email.
- Cc and Bcc
- You can also add addresses for cc- or bcc-recipients using the following syntax:

<a href="mailto:example@example.com?cc=john@example.com&bcc=jane@example.com">Send email</a>

Subject and body text You can populate the subject and body for the new email as well:

<a href="mailto:example@example.com?subject=Example+subject&body=Message+text">Send email</a>

Those values must be URL encoded.

- Clicking on a link with mailto: will try to open the default email client specified by your operating system or it will ask you to choose what client you want to use.
- Not all options specified after the recipient's address are supported in all email clients.

Lists:

- HTML offers three ways for specifying lists: ordered lists, unordered lists, and description lists.
- Ordered lists use ordinal sequences to indicate the order of list elements, unordered lists use a define symbol such as a bullet to list elements in no designated order, and description lists use indents to list elements with their children.

Ordered List

- An ordered list can be created with the <ol> tag and each list item can be created with the <li> tag as in the example below:

```
<ol>
 <li>Item</li>
 <li>Another Item</li>
 <li>Yet Another Item</li>
</ol>
```

This will produce a numbered list (which is the default style):

1. Item

2. Another Item

3. Yet Another Item

Manually changing the numbers

- There are a couple of ways you can play with which numbers appear on the list items in an ordered list.
- The first way is to set a starting number, using the start attribute.
- The list will start at this defined number, and continue incrementing by one as usual.

```
<ol start="3">
 <li>Item</li>
 <li>Some Other Item</li>
 <li>Yet Another Item</li>
</ol>
```

This will produce a numbered list (which is the default style):

3. Item
4. Some Other Item
5. Yet Another Item

- You can also explicitly set a certain list item to a specific number.
- Further list items after one with a specified value will continue incrementing by one from that list item's value, ignoring where the parent list was at.

```
<li value="7"></li>
```

- It is also worth noting that, by using the value attribute directly on a list item, you can override an ordered list's existing numbering system by restarting the numbering at a lower value.
- So if the parent list was already up to value 7, and encountered a list item at value 4, then that list item would still display as 4 and continue counting
- from that point again.

```
<ol start="5">
 <li>Item</li>
 <li>Some Other Item</li>
 <li value="4">A Reset Item</li>
 <li>Another Item</li>
 <li>Yet Another Item</li>
</ol>
```

- So the example above will produce a list that follows the numbering pattern of 5, 6, 4, 5, 6 - starting again at a number lower than the previous and duplicating the number 6 in the list.
- Note: The start and value attributes only accept a number - even if the ordered list is set to display as Roman numerals or letters.
- You can reverse the numbering by adding reversed in your ol element:

```
<ol reversed>
 <li>Item</li>
 <li>Some Other Item</li>
 <li value="4">A Reset Item</li>
 <li>Another Item</li>
 <li>Yet Another Item</li>
</ol>
```

Unordered List :

- An unordered list can be created with the tag and each list item can be created with the tag as shown by the example below

```
<ul>
 <li>Item</li>
 <li>Another Item</li>
 <li>Yet Another Item</li>
</ul>
```

This will produce a bulleted list (which is the default style):

Item

Another Item

Yet Another Item

- You should use ul to display a list of items, where the order of the items is not important.
- If changing the order of the items makes the list incorrect, you should use <ol>.

Nested lists

You can nest lists to represent sub-items of a list item.

```
<ul>
 <li>item 1</li>
 <li>item 2
 <ul>
 <li>sub-item 2.1</li>
 <li>sub-item 2.2</li>
 </ul>
 </li>
 <li>item 3</li>
</ul>
```

Description List

- A description list (or definition list, as it was called before HTML5) can be created with the dl element.
- It consists of name-value groups, where the name is given in the dt element, and the value is given in the dd element.

```
<dl>
 <dt>name 1</dt>
 <dd>value for 1</dd>
 <dt>name 2</dt>
 <dd>value for 2</dd>
</dl>
```

A name-value group can have more than one name and/or more than one value (which represent alternatives):

```
<dl>
 <dt>name 1</dt>
 <dt>name 2</dt>
 <dd>value for 1 and 2</dd>
 <dt>name 3</dt>
 <dd>value for 3</dd>
 <dd>value for 3</dd>
</dl>
```

Tables

The HTML <table> element allows web authors to display tabular data (such as text, images, links, other tables, etc.) in a two dimensional table with rows and columns of cells.

Simple Table

```
<table>
 <tr>
 <th>Heading 1/Column 1</th>
 <th>Heading 2/Column 2</th>
 </tr>
 <tr>
 <td>Row 1 Data Column 1</td>
 <td>Row 1 Data Column 2</td>
 </tr>
 <tr>
 <td>Row 2 Data Column 1</td>
 <td>Row 2 Data Column 2</td>
 </tr> </table>
```

- This will render a <table> consisting of three total rows (<tr>): one row of header cells (<th>) and two rows of content cells (<td>).
- <th> elements are tabular headers and <td> elements are tabular data. You can put whatever you want inside a <td> or <th>.

Heading 1/Column 1 Heading 2/Column 2

Row 1 Data Column 1 Row 1 Data Column 2

Row 2 Data Column 1 Row 2 Data Column 2

Spanning columns or rows:

Table cells can span multiple columns or rows using the colspan and rowspan attributes. These attributes can be

applied to <th> and <td> elements.

<table>

<tr>

```
<td>row 1 col 1</td>
 <td>row 1 col 2</td>
 <td>row 1 col 3</td>
 </tr>
 <tr>
 <td colspan="3">This second row spans all three columns</td>
 </tr>
 <tr>
 <td rowspan="2">This cell spans two rows</td>
 <td>row 3 col 2</td>
 <td>row 3 col 3</td>
 </tr>
 <tr>
 <td>row 4 col 2</td>
 <td>row 4 col 3</td> </tr> </table>
```

Summary:

- □ What is HTML
- □ Difference between HTML and HTML5
- □ Doctypes
- □ Headings
- □ Paragraphs
- □ Text Formatting
- □ Anchors and Hyperlinks
- □ Tables
- □ Lists

Thank You……….

If you have any quries please write to info@uplatz.com".