

Chapter 2

Static HTML (Hyper Text Markup Language) (Two week)

HTML document structure and html tag format

HTML

❖ What is an HTML File?

- HTML stands for **Hyper Text Markup Language**
 - An HTML file is a text file containing small **markup tags**
 - The markup tags tell the Web browser **how to display** the page
 - An HTML file must have an **htm** or **html** file extension
 - An HTML file can be created using a **simple text editor**
-

❖ Do You Want to Try It?

If you are running Windows, start Notepad (or start SimpleText if you are on a Mac) and type in the following text:

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

Save the file as "mypage.htm".

Start your Internet browser. Select "Open" (or "Open Page") in the File menu of your browser. A dialog box will appear. Select "Browse" (or "Choose File") and locate the HTML file you just created - "mypage.htm" - select it and click "Open". Now you should see an address in the dialog box, for example

"C:\MyDocuments\mypage.htm". Click OK, and the browser will display the page.

❖ **Example Explained**

The first tag in your HTML document is `<html>`. This tag tells your browser that this is the start of an HTML document. The last tag in your document is `</html>`. This tag tells your browser that this is the end of the HTML document.

The text between the `<head>` tag and the `</head>` tag is header information. Header information is not displayed in the browser window.

The text between the `<title>` tags is the title of your document. The title is displayed in your browser's caption.

The text between the `<body>` tags is the text that will be displayed in your browser.

The text between the `` and `` tags will be displayed in a bold font.

❖ **HTM or HTML Extension?**

When you save an HTML file, you can use either the `.htm` or the `.html` extension. We have used `.htm` in our examples. It might be a bad habit inherited from the past when some of the commonly used software only allowed three letter extensions.

With newer software we think it will be perfectly safe to use `.html`.

❖ **Note on HTML Editors:**

You can easily edit HTML files using a WYSIWYG (what you see is what you get) editor like FrontPage, Dreamweaver, Claris Home Page, or Adobe PageMill instead of writing your markup tags in a plain text file.

But if you want to be a skillful Web developer, we strongly recommend that you use a plain text editor to learn your primer HTML.

HTML Elements

HTML documents are text files made up of HTML elements.

HTML elements are defined using HTML tags.

❖ HTML Tags

- HTML tags are used to mark-up HTML **elements**
 - HTML tags are surrounded by the **two characters < and >**
 - The surrounding characters are called **angle brackets**
 - HTML tags normally **come in pairs** like and
 - The first tag in a pair is the **start tag**, the second tag is the **end tag**
 - The text between the start and end tags is the **element content**
 - HTML tags are **not case sensitive**, means the same as
-

❖ HTML Elements

Remember the HTML example from the previous page:

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

This is an HTML element:

```
<b>This text is bold</b>
```

The HTML element starts with a **start tag**: ``
The **content** of the HTML element is: This text is bold
The HTML element ends with an **end tag**: ``

The purpose of the `` tag is to define an HTML element that should be displayed as bold.

This is also an HTML element:

```
<body>  
This is my first homepage. <b>This text is bold</b>  
</body>
```

This HTML element starts with the start tag `<body>`, and ends with the end tag `</body>`.

The purpose of the `<body>` tag is to define the HTML element that contains the body of the HTML document.

❖ Why do We Use Lowercase Tags?

We have just said that HTML tags are not case sensitive: `` means the same as ``. When you surf the Web, you will notice that most tutorials use uppercase HTML tags in their examples. We always use lowercase tags. Why?

If you want to prepare yourself for the next generations of HTML you should start using lowercase tags. The World Wide Web Consortium (W3C) recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) demands lowercase tags.

❖ Tag Attributes

Tags can have attributes. Attributes can provide additional information about the HTML elements on your page.

This tag defines the body element of your HTML page: `<body>`. With an added `bgcolor` attribute, you can tell the browser that the background color of your page should be red, like this: `<body bgcolor="red">`.

This tag defines an HTML table: <table>. With an added border attribute, you can tell the browser that the table should have no borders: <table border="0">

Attributes always come in name/value pairs like this: name="value".

Attributes are always added to the start tag of an HTML element.

❖ Quote Styles, "red" or 'red'?

Attribute values should always be enclosed in quotes. Double style quotes are the most common, but single style quotes are also allowed.

In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes:

```
name='John "ShotGun" Nelson'
```

Basic HTML Tags

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

The best way to learn HTML is to work with examples.

❖ Headings

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading. <h6> defines the smallest heading.

```
<h1>This is a heading</h1>  
<h2>This is a heading</h2>  
<h3>This is a heading</h3>  
<h4>This is a heading</h4>  
<h5>This is a heading</h5>  
<h6>This is a heading</h6>
```

HTML automatically adds an extra blank line before and after a heading.

❖ Paragraphs

Paragraphs are defined with the `<p>` tag.

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

HTML automatically adds an extra blank line before and after a paragraph.

❖ Line Breaks

The `
` tag is used when you want to end a line, but don't want to start a new paragraph. The `
` tag forces a line break wherever you place it.

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

The `
` tag is an empty tag. It has no closing tag.

❖ Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment will be ignored by the browser. You can use comments to explain your code, which can help you when you edit the source code at a later date.

```
<!-- This is a comment -->
```

Note that you need an exclamation point after the opening bracket, but not before the closing bracket.

❖ Basic Notes - Useful Tips

When you write HTML text, you can never be sure how the text is displayed in another browser. Some people have large computer displays, some have small. The text will be reformatted every time the

user resizes his window. Never try to format the text in your editor by adding empty lines and spaces to the text.

HTML will truncate the spaces in your text. Any number of spaces count as one. Some extra information: In HTML a new line counts as one space.

Using empty paragraphs `<p>` to insert blank lines is a bad habit. Use the `
` tag instead. (But don't use the `
` tag to create lists. Wait until you have learned about HTML lists.)

You might have noticed that paragraphs can be written without the closing tag `</p>`. Don't rely on it. The next version of HTML will not allow you to skip ANY closing tags.

HTML automatically adds an extra blank line before and after some elements, like before and after a paragraph, and before and after a heading.

We use a horizontal rule (the `<hr>` tag), to separate sections by a horizontal line.

❖ Basic HTML Tags

Tag	Description
<code><html></code>	Defines an HTML document
<code><body></code>	Defines the document's body
<code><h1></code> to <code><h6></code>	Defines header 1 to header 6
<code><p></code>	Defines a paragraph
<code>
</code>	Inserts a single line break
<code><hr></code>	Defines a horizontal rule
<code><!--></code>	Defines a comment

Hyper-Links, Tables, Frames, Forms, Form Elements, Image Maps, Special Characters, Meta Data...

HTML Text Formatting

HTML defines a lot of elements for formatting output, like bold or italic text.

Below are a lot of examples that you can try out yourself:

❖ **How to View HTML Source**

Have you ever seen a Web page and wondered "How do they do that?"

To find out, simply click on the VIEW option in your browsers toolbar and select SOURCE or PAGE SOURCE. This will open a window that shows you the actual HTML of the page.

❖ **Text Formatting Tags**

Tag	Description
	Defines bold text
<big>	Defines big text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text
	Defines strong text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
	Defines deleted text

❖ **"Computer Output" Tags**

Tag	Description
<code>	Defines computer code text
<kbd>	Defines keyboard text
<samp>	Defines sample computer code

<tt>	Defines teletype text
<var>	Defines a variable
<pre>	Defines preformatted text

❖ Citations, Quotations, and Definition Tags

Tag	Description
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines an address element
<bdo>	Defines the text direction
<blockquote>	Defines a long quotation
<q>	Defines a short quotation
<cite>	Defines a citation
<dfn>	Defines a definition term

HTML Character Entities

Some characters like the < character, have a special meaning in HTML, and therefore cannot be used in the text.

To display a less than sign (<) in HTML, we have to use a character entity.

❖ Character Entities

Some characters have a special meaning in HTML, like the less than sign (<) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in the HTML source.

A character entity has three parts: an ampersand (&), an entity name or a # and an entity number, and finally a semicolon (;).

To display a less than sign in an HTML document we must write: **<** or **<**

The advantage of using a name instead of a number is that a name is easier to remember. The disadvantage is that not all browsers support the newest entity names, while the support for entity numbers is very good in almost all browsers.

Note that the entities are case sensitive.

❖ **Non-breaking Space**

The most common character entity in HTML is the non-breaking space.

Normally HTML will truncate spaces in your text. If you write 10 spaces in your text HTML will remove 9 of them. To add spaces to your text, use the ` ` character entity.

❖ **The Most Common Character Entities:**

Result	Description	Entity Name	Entity Number
	non-breaking space	<code>&nbsp;</code>	<code>&#160;</code>
<code><</code>	less than	<code>&lt;</code>	<code>&#60;</code>
<code>></code>	greater than	<code>&gt;</code>	<code>&#62;</code>
<code>&</code>	ampersand	<code>&amp;</code>	<code>&#38;</code>
<code>"</code>	quotation mark	<code>&quot;</code>	<code>&#34;</code>
<code>'</code>	apostrophe	<code>&apos;</code>	<code>&#39;</code>

❖ **Some Other Commonly Used Character Entities:**

Result	Description	Entity Name	Entity Number
¢	cent	<code>&cent;</code>	<code>&#162;</code>
£	pound	<code>&pound;</code>	<code>&#163;</code>
¥	yen	<code>&yen;</code>	<code>&#165;</code>
§	section	<code>&sect;</code>	<code>&#167;</code>
©	copyright	<code>&copy;</code>	<code>&#169;</code>
®	registered trademark	<code>&reg;</code>	<code>&#174;</code>
×	multiplication	<code>&times;</code>	<code>&#215;</code>
÷	division	<code>&divide;</code>	<code>&#247;</code>

HTML Links (Hyperlinks)

HTML uses a hyperlink to link to another document on the Web.

❖ The Anchor Tag and the href Attribute

HTML uses the <a> (anchor) tag to create a link to another document.

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.

The syntax of creating an anchor:

```
<a href="url">Text to be displayed</a>
```

The <a> tag is used to create an anchor to link from, the href attribute is used to address the document to link to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

This anchor defines a link to the ITU website:

```
<a href="http://www.aau.edu.int/">Link to AAU.</a>
```

The line above will look like this in a browser:

[Link to AAU.](http://www.aau.edu.int/)

❖ The Target Attribute

With the target attribute, you can define **where** the linked document will be opened.

The line below will open the document in a new browser window:

```
<a href="http://www.aau.edu.et/"  
target="_blank">Visit AAU!</a>
```

❖ The Anchor Tag and the Name Attribute

The name attribute is used to create a named anchor. When using named anchors we can create links that can jump directly into a specific section on a page, instead of letting the user scroll around to find what he/she is looking for.

Below is the syntax of a named anchor:

```
<a name="label">Text to be displayed</a>
```

The name attribute is used to create a named anchor. The name of the anchor can be any text you care to use.

The line below defines a named anchor:

```
<a name="tips">Useful Tips Section</a>
```

You should notice that a named anchor is not displayed in a special way.

To link directly to the "tips" section, add a # sign and the name of the anchor to the end of a URL, like this:

```
<a href="http://www.aau.edu.et/IPCourse.html#tips">  
Jump to the Useful Tips Section</a>
```

A hyperlink to the Useful Tips Section from WITHIN the file "IPCourse.html" will look like this:

```
<a href="#tips">Jump to the Useful Tips Section</a>
```

❖ Basic Notes - Useful Tips

Always add a trailing slash to subfolder references. If you link like this: href="http://www.aau.edu.et/html", you will generate two HTTP requests to the server, because the server will add a slash to the address and create a new request like this: href="http://www.aau.edu.et/html/"

Named anchors are often used to create "table of contents" at the beginning of a large document. Each chapter within the document is given a named anchor, and links to each of these anchors are put at the top of the document.

If a browser cannot find a named anchor that has been specified, it goes to the top of the document. No error occurs.

❖ Link Tags

Tag	Description
<a>	Defines an anchor

HTML Frames

With frames, you can display more than one Web page in the same browser window.

❖ Frames

With frames, you can display more than one HTML document in the same browser window. Each HTML document is called a frame, and each frame is independent of the others.

The disadvantages of using frames are:

- The web developer must keep track of more HTML documents
 - It is difficult to print the entire page
-

❖ The Frameset Tag

- The <frameset> tag defines how to divide the window into frames
 - Each frameset defines a set of rows **or** columns
 - The values of the rows/columns indicate the amount of screen area each row/column will occupy
-

❖ The Frame Tag

- The <frame> tag defines what HTML document to put into each frame

In the example below we have a frameset with two columns. The first column is set to 25% of the width of the browser window. The second column is set to 75% of the width of the browser window. The HTML

document "frame_a.htm" is put into the first column, and the HTML document "frame_b.htm" is put into the second column:

```
<frameset cols="25%,75%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
</frameset>
```

❖ Basic Notes - Useful Tips

If a frame has visible borders, the user can resize it by dragging the border. To prevent a user from doing this, you can add `noresize="noresize"` to the `<frame>` tag.

Add the `<noframes>` tag for browsers that do not support frames.

❖ Frame Tags

Tag	Description
<code><frameset></code>	Defines a set of frames
<code><frame></code>	Defines a sub window (a frame)
<code><noframes></code>	Defines a noframe section for browsers that do not handle frames
<code><iframe></code>	Defines an inline sub window (frame)

HTML Tables

With HTML you can create tables.

❖ Tables

Tables are defined with the `<table>` tag. A table is divided into rows (with the `<tr>` tag), and each row is divided into data cells (with the `<td>` tag). The letters `td` stands for "table data," which is the content

of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

❖ Tables and the Border Attribute

If you do not specify a border attribute the table will be displayed without any borders. Sometimes this can be useful, but most of the time, you want the borders to show.

To display a table with borders, you will have to use the border attribute:

```
<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

❖ Headings in a Table

Headings in a table are defined with the `<th>` tag.

```
<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How it looks in a browser:

Heading	Another Heading
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

❖ Empty Cells in a Table

Table cells with no content are not displayed very well in most browsers.

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td></td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
---------------	---------------

row 2, cell 1	
---------------	--

Note that the borders around the empty table cell are missing.

To avoid this, add a non-breaking space () to empty data cells, to make the borders visible:

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>&nbsp;</td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	

❖ Basic Notes - Useful Tips

The <thead>, <tbody> and <tfoot> elements are seldom used, because of bad browser support. Expect this to change in future versions of XHTML.

❖ Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<colgroup>	Defines groups of table columns

<col>	Defines the attribute values for one or more columns in a table
<thead>	Defines a table head
<tbody>	Defines a table body
<tfoot>	Defines a table footer

❖ Others

Colspan / rowspan
width/height
caption

cellpadding/ cellspacing
Align

bgcolor
th (table header)

HTML Lists

HTML supports ordered, unordered and definition lists.

❖ Unordered Lists

An unordered list is a list of items. The list items are marked with bullets (typically small black circles).

An unordered list starts with the tag. Each list item starts with the tag.

```
<ul>  
<li>Coffee</li>  
<li>Milk</li>  
</ul>
```

Here is how it looks in a browser:

- Coffee
- Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

❖ Ordered Lists

An ordered list is also a list of items. The list items are marked with numbers.

An ordered list starts with the `` tag. Each list item starts with the `` tag.

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

Here is how it looks in a browser:

1. Coffee
2. Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

❖ Definition Lists

A definition list is **not** a list of items. This is a list of terms and explanation of the terms.

A definition list starts with the `<dl>` tag. Each definition-list term starts with the `<dt>` tag. Each definition-list definition starts with the `<dd>` tag.

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

Here is how it looks in a browser:

Coffee
 Black hot drink
Milk
 White cold drink

Inside a definition-list definition (the <dd> tag) you can put paragraphs, line breaks, images, links, other lists, etc.

❖ List Tags

Tag	Description
	Defines an ordered list
	Defines an unordered list
	Defines a list item
<dl>	Defines a definition list
<dt>	Defines a definition term
<dd>	Defines a definition description

HTML Forms and Input

HTML Forms are used to select different kinds of user input.

❖ Forms

A form is an area that can contain form elements.

Form elements are elements that allow the user to enter information (like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.

A form is defined with the <form> tag.

```
<form>
  <input>
  <input>
</form>
```

❖ Input

The most used form tag is the `<input>` tag. The type of input is specified with the type attribute. The most commonly used input types are explained below.

❖ Text Fields

Text fields are used when you want the user to type letters, numbers, etc. in a form.

```
<form>
First name:
<input type="text" name="firstname">
<br>
Last name:
<input type="text" name="lastname">
</form>
```

How it looks in a browser:

First name:
Last name:

Note that the form itself is not visible. Also note that in most browsers, the width of the text field is 20 characters by default.

❖ Radio Buttons

Radio Buttons are used when you want the user to select 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 it looks in a browser:

Male
 Female

Note that only one option can be chosen.

❖ Checkboxes

Checkboxes are used when you want the user to select one or more options of a limited number of choices.

```
<form>
<input type="checkbox" name="bike">
I have a bike
<br>
<input type="checkbox" name="car">
I have a car
</form>
```

How it looks in a browser:

- I have a bike
 - I have a car
-

❖ The Form's Action Attribute and the Submit Button

When the user clicks on the "Submit" button, the content of the form is sent to another file. The form's action attribute defines the name of the file to send the content to. The file defined in the action attribute usually does something with the received input.

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

How it looks in a browser:

Username:

If you type some characters in the text field above, and click the "Submit" button, you will send your input to a page called "html_form_action.asp". That page will show you the received input.

❖ Form Tags

Tag	Description
<form>	Defines a form for user input
<input>	Defines an input field
<textarea>	Defines a text-area (a multi-line text input control)
<label>	Defines a label to a control
<fieldset>	Defines a fieldset
<legend>	Defines a caption for a fieldset
<select>	Defines a selectable list (a drop-down box)
<optgroup>	Defines an option group
<option>	Defines an option in the drop-down box
<button>	Defines a push button

HTML Images

With HTML you can display images in a document.

❖ The Image Tag and the Src Attribute

In HTML, images are defined with the tag.

The tag is empty, which means that it contains attributes only and it has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page.

The syntax of defining an image:

```

```

The URL points to the location where the image is stored. An image named "boat.gif" located in the directory "images" on "www.aau.edu.et" has the URL:
<http://www.aau.edu.et/images/boat.gif>.

The browser puts the image where the image tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

❖ The Alt Attribute

The alt attribute is used to define an "alternate text" for an image. The value of the alt attribute is an author-defined text:

```

```

The "alt" attribute tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image. It is a good practice to include the "alt" attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers.

❖ Basic Notes - Useful Tips

If an HTML file contains ten images - eleven files are required to display the page right. Loading images take time, so my best advice is: Use images carefully.

❖ Image Tags

Tag	Description
	Defines an image
<map>	Defines an image map
<area>	Defines an area inside an image map

❖ Image example

```
 ->  
displays the coordinates on the status bar as the mouse moves  
over the image.
```

❖ Image map example

```
  
<map id="planetmap" 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="mercury.htm">
<area shape="circle" coords="124,58,8"
alt="Venus"href="venus.htm">
</map>
```

HTML Backgrounds

A good background can make a Web site look really great.

❖ Backgrounds

The `<body>` tag has two attributes where you can specify backgrounds. The background can be a color or an image.

Bgcolor

The `bgcolor` attribute sets the background to a color. The value of this attribute can be a hexadecimal number, an RGB value, or a color name.

```
<body bgcolor="#000000">
<body bgcolor="rgb(0,0,0)">
<body bgcolor="black">
```

The lines above all set the background color to black.

Background

The `background` attribute sets the background to an image. The value of this attribute is the URL of the image you want to use. If the image is smaller than the browser window, the image will repeat itself until it fills the entire browser window.

```
<body background="clouds.gif">
<body background="http://www.aau.edu.et/clouds.gif">
```

The URL can be relative (as in the first line above) or absolute (as in the second line above).

Note: If you want to use a background image, you should keep in mind:

- Will the background image increase the loading time too much?
Tip: Image files should be maximum 10k
- Will the background image look good with other images on the page?
- Will the background image look good with the text colors on the page?
- Will the background image look good when it is repeated on the page?
- Will the background image take away the focus from the text?

❖ **Basic Notes - Useful Tips**

The bgcolor, background, and the text attributes in the <body> tag are deprecated in the latest versions of HTML (HTML 4 and XHTML). The World Wide Web Consortium (W3C) has removed these attributes from its recommendations. In future versions of HTML, style sheets (CSS) will be used to define the layout and display properties of HTML elements.

Few of the most visited web sites use background images.

The most used background colors are: white, black and gray.

HTML Layout

Everywhere on the Web you will find pages that are formatted like newspaper pages using HTML columns.

❖ **HTML Layout - Using Tables**

One very common practice with HTML, is to use HTML tables to format the layout of an HTML page.

An HTML <table> is used to divide a part of this Web page into two columns.

A part of this page is formatted with two columns, like a newspaper page.

The trick is to use a table without borders, and maybe a little extra cell-padding.

As you can see at this page, there is a left column and a right column.

No matter how much text you add to this page, it will stay inside its column borders.

This text is displayed in the left column.

❖ Same Layout - Color Added

One very common practice with HTML, is to use HTML tables to format the layout of an HTML page.

An HTML `<table>` is used to divide a part of this Web page into two columns.

A part of this page is formatted with two columns, like a newspaper page.

This text is displayed in the right column.

As you can see at this page, there is a left column and a right column.

The trick is to use a table without borders, and maybe a little extra cell-padding.

No matter how much text you add to this page, it will stay inside its column borders.

HTML Fonts

The `` tag in HTML is deprecated. It is supposed to be removed in a future version of HTML.

Even if a lot of people are using it, you should try to avoid it, and use styles instead.

❖ The HTML `` Tag

With HTML code like this, you can specify both the size and the type of the browser output :

```
<p>
<font size="2" face="Verdana">
This is a paragraph.
</font>
</p>
<p>
<font size="3" face="Times">
This is another paragraph.
</font>
</p>
```

❖ Font Attributes

Attribute	Example	Purpose
size="number"	size="2"	Defines the font size
size="+number"	size="+1"	Increases the font size
size="-number"	size="-1"	Decreases the font size
face="face-name"	face="Times"	Defines the font-name
color="color-value"	color="#eeff00"	Defines the font color
color="color-name"	color="red"	Defines the font color

❖ The Tag Should NOT be Used

The tag is deprecated in the latest versions of HTML (HTML 4 and XHTML).

The World Wide Web Consortium (W3C) has removed the tag from its recommendations. In future versions of HTML, style sheets (CSS) will be used to define the layout and display properties of HTML elements.

HTML Styles

With HTML 4.0 all formatting can be moved out of the HTML document and into a separate style sheet.

❖ How to Use Styles

When a browser reads a style sheet, it will format the document according to it. There are three ways of inserting a style sheet:

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section.

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

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section with the <style> tag.

```
<head>
<style type="text/css">
body {background-color: red}
p {margin-left: 20px}
</style>
</head>
```

Inline Styles

An inline style should be used when a unique style is to be applied to a single occurrence of an element.

To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

```
<p style="color: red; margin-left: 20px">
This is a paragraph
```

```
</p>
```

❖ Style Tags

Tag	Description
<style>	Defines a style definition
<link>	Defines a resource reference
<div>	Defines a section in a document
	Defines a section in a document

HTML Head

❖ The Head Element

The head element contains general information, also called meta-information, about a document. Meta means "information about".

You can say that meta-data means information about data, or meta-information means information about information.

❖ Information Inside the Head Element

The elements inside the head element should not be displayed by a browser.

According to the HTML standard, only a few tags are legal inside the head section. These are: <base>, <link>, <meta>, <title>, <style>, and <script>.

Look at the following illegal construct:

```
<head>  
  <p>This is some text</p>  
</head>
```

In this case the browser has two options:

- Display the text because it is inside a paragraph element
- Hide the text because it is inside a head element

If you put an HTML element like `<h1>` or `<p>` inside a head element like this, most browsers will display it, even if it is illegal.

Should browsers forgive you for errors like this? We don't think so. Others do.

❖ Head Tags

Tag	Description
<code><head></code>	Defines information about the document
<code><title></code>	Defines the document title
<code><base></code>	Defines a base URL for all the links on a page
<code><link></code>	Defines a resource reference
<code><meta></code>	Defines meta information

Tag	Description
<code><!DOCTYPE></code>	Defines the document type. This tag goes before the <code><html></code> start tag.

HTML Meta

❖ The Meta Element

As we explained in the previous chapter, the head element contains general information (meta-information) about a document.

HTML also includes a meta element that goes inside the head element. The purpose of the meta element is to provide meta-information about the document.

Most often the meta element is used to provide information that is relevant to browsers or search engines like describing the content of your document.

❖ Keywords for Search Engines

Some search engines on the WWW will use the name and content attributes of the meta tag to index your pages.

This meta element defines a description of your page:

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

This meta element defines keywords for your page:

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

The intention of the name and content attributes is to describe the content of a page.

However, since too many webmasters have used meta tags for spamming, like repeating keywords to give pages a higher ranking, some search engines have stopped using them entirely.

❖ **Unknown Meta Attributes**

Sometimes you will see meta attributes that are unknown to you like this:

```
<meta name="security" content="low">
```

Then you just have to accept that this is something unique to the site or to the author of the site, and that it has probably no relevance to you.

HTML Uniform Resource Locators

❖ **HTML Links**

When you click on a link in an HTML document like this: [Last Page](#), an underlying <a> tag points to a place (an address) on the Web with an href attribute value like this: Last Page.

The Last Page link in the example is a link that is relative to the Web site that you are browsing, and your browser will construct a full Web address like <http://www.aau.edu.et/html/lastpage.htm> to access the page.

❖ Uniform Resource Locators

Something called a Uniform Resource Locator (URL) is used to address a document (or other data) on the World Wide Web. A full Web address like this: <http://www.aau.edu.et/html/lastpage.htm> follows these syntax rules:

scheme://host.domain:port/path/filename

The **scheme** is defining the **type** of Internet service. The most common type is **http**.

The **domain** is defining the Internet **domain name** like aau.edu.et.

The **host** is defining the domain host. If omitted, the default host for http is **www**.

The **:port** is defining the **port number** at the host. The port number is normally omitted. The default port number for http is **80**.

The **path** is defining a **path** (a sub directory) at the server. If the path is omitted, the resource (the document) must be located at the root directory of the Web site.

The **filename** is defining the name of a document. The default filename might be default.asp, or index.html or something else depending on the settings of the Web server.

❖ URL Schemes

Some examples of the most common schemes can be found below:

Schemes	Access
file	a file on your local PC
ftp	a file on an FTP server
http	a file on a World Wide Web Server
gopher	a file on a Gopher server
news	a Usenet newsgroup
telnet	a Telnet connection
WAIS	a file on a WAIS server

❖ Accessing a Newsgroup

The following HTML code:

```
<a href="news:alt.html">HTML Newsgroup</a>
```

creates a link to a newsgroup like this [HTML Newsgroup](#).

❖ Downloading with FTP

The following HTML code:

```
<a href="ftp://www.aau.edu.et/ftp/winzip.exe">Download WinZip</a>
```

creates a link to download a file like this: [Download WinZip](#).

(The link doesn't work. Don't try it. It is just an example. AAU doesn't really have such ftp directory.)

❖ Link to your Mail system

The following HTML code:

```
<a href="mailto:someone@aau.edu.et">someone@aau.edu.et</a>
```

creates a link to your own mail system like this:

[someone@aau.edu.et](#)

HTML Scripts

Add scripts to HTML pages to make them more dynamic and interactive.

❖ Insert a Script into HTML Page

A script in HTML is defined with the `<script>` tag. Note that you will have to use the `type` attribute to specify the scripting language.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```

The script above will produce this output:

Hello World!

❖ How to Handle Older Browsers

A browser that does not recognize the `<script>` tag at all, will display the `<script>` tag's content as text on the page. To prevent the browser from doing this, you should hide the script in comment tags. An old browser (that does not recognize the `<script>` tag) will ignore the comment and it will not write the tag's content on the page, while a new browser will understand that the script should be executed, even if it is surrounded by comment tags.

Example

JavaScript:

```
<script type="text/javascript">
<!--
document.write("Hello World!")
//-->
</script>
```

VBScript:

```
<script type="text/vbscript">
<!--
document.write("Hello World!")
'-->
</script>
```

❖ The <noscript> Tag

In addition to hiding the script inside a comment, you can also add a <noscript> tag.

The <noscript> tag is used to define an alternate text if a script is NOT executed. This tag is used for browsers that recognize the <script> tag, but do not support the script inside, so these browsers will display the text inside the <noscript> tag instead. However, if a browser supports the script inside the <script> tag it will ignore the <noscript> tag.

Example

JavaScript:

```
<script type="text/javascript">
<!--
document.write("Hello World!")
//-->
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

VBScript:

```
<script type="text/vbscript">
<!--
document.write("Hello World!")
'-->
</script>
<noscript>Your browser does not support VBScript!</noscript>
```

❖ Script Tags

Tag	Description
<script>	Defines a script
<noscript>	Defines an alternate text if the script is not executed
<object>	Defines an embedded object
<param>	Defines run-time settings (parameters) for an object

HTML Software

There are several HTML softwares that aid in building a web page.

FrontPage and Dreamweaver are some of html editing softwares. FrontPage comes integrated with windows operating systems while Dreamweaver is separate software.

Both facilitate the design of web pages by letting the designer to insert code from a code library, view the design from time to time, debug the design for possible errors, and update the design as required.

Browser compatibility issues

What is cross browser compatibility?

There are at least four popular browsers being used regularly on the internet. Netscape and Internet Explorer are forever enhancing their products.

As the new versions become adapted, their older counterparts lose popularity. It generally takes a year or more for the implementation of the new browser. And, you guessed it, by that time there is yet another version. This is good for us as users, but difficult for designers and webmasters.

While web pages will almost never appear EXACTLY the same in all browser versions, good design can eliminate many viewing problems. While we try to make everything as browser-neutral as possible, the truth is, tailoring a site for cross browser compatibility is a pain.

If pages (especially those with forms) are visually distorted or hard to view please use 800x600 or higher resolution, 16bit or higher color, and IE 5.0 or later.

In the early days of the Internet, many sites were advertising as "Best viewed with Netscape" or "Best viewed with Internet Explorer" or the like.

These days, such labels seem to be rarer. And no wonder. Webmasters today put an inordinate amount of effort to promote their sites on the search engines and elsewhere, and it's unlikely that they'll want to turn away any visitor just because he/she uses a different browser.

Along with this principle of catering to the widest possible audience is the principle of designing your page for compatibility with different browsers, operating systems and hardware.

Next, we shall discuss four compatibility issues. They are by no means exhaustive, but they are at least starting points to designing a site that will be viewable by more visitors.

1. Screen Resolution Issues

Earlier, majority of visitors of the web used a display resolution of 800x600 and others used 1024x768, and only a small number using 640x480. This situation has changed, and today, the bulk of my visitors are roughly divided between using a screen resolution of 800x600 and 1024x768 (with insignificant numbers using other resolutions).

If you design your site with fixed widths, you need to be aware of the above. For example, if you design for the 1024x768 screen resolution, you will force almost half your visitors to scroll their screens horizontally to see the entire page. Horizontal scrolling irritates a number of people (particularly if they have to scroll left and right continually just to read your sentences), hence many sites try to avoid it. They do this by either designing with a fluid design, where the page automatically fits whatever screen resolution the visitor uses, or by designing it for a worst case screen resolution (either 800x600 or 640x480).

If fixed width is your cup of tea, you should not design your pages to require more than a 800x600 display resolution. Doing so will probably displease at least half your visitors, who will have to repeatedly scroll horizontally (back and forth) just to read the sentences on your page. If, despite this, you feel that you need a fixed canvas larger than 800x600, you can use a trick just like some sites: they put optional material (like advertisements) in the rightmost column of their web page. This allows visitors who have smaller screens to ignore the rightmost column; they can read the main content without having to scroll horizontally.

Of course, if you only use relative widths with percentages like 100%, 80%, your page already caters to different screen resolutions (unless you put graphic images on your page that exceed the widths of the columns).

2. Colour Limitations

It may or may not come as a surprise to you that a colour code like "#F2C3BE" results in different colours on different systems, depending on the number of colours in your visitors' colour palettes, their monitors, etc.

For example, if you choose a colour that looks good on your 24-bit colour system ("True Color" on Windows, "millions of colors" on Mac), and your visitor goes to your site using a 256 colour setting on his system, your colour will be dithered to fit into the more limited number of colours in his palette.

What looks to you like beautiful shades of colours may thus turn out to be ugly combinations on a different system.

Of course, few people use 256 colours on their system anymore. As such, many webmasters are abandoning the old technique of only using colour combinations that are multiples of the hexadecimal "33" (which are supposed to be safe to use in that they display fairly similarly across the main platforms).

However, if you think that you can now use the next lowest denominator, the 16-bit colour setting ("High Color" on Windows, "thousands of colors" on Mac) safely, think again. Contrary to what you might expect, the 16-bit colour palette is not a subset of the 24-bit colour palette: apart from black and white, the colours in the two palettes are not identical. If you design the colour scheme of your site while working from one palette, be sure to switch to the other colour setting to make sure that your colour scheme blends well in the other setting as well.

3. Frames

Let's look at frames from two points of views here: screen resolution and people who browse your site using speech software.

One common complaint that people have against sites using frames is usually that webmasters tend to forget that visitors don't have the

same large screen resolution that the designers have when they created the web page. Sites with frames tend to provide a smaller area for people to view the main content of the site, since the outer frames occupy some of the screen real estate as well. If the web designer has not checked his site using a lower screen resolution like 800x600 and 640x460, and tested its usability with those lower resolutions, he may not be aware that the site is difficult to use in such situations. Visitors may have to scroll horizontally and vertically continually just to read the content. The situation is worse if the designer removed the scroll bar (because it looked fine without it on his high resolution screen), and visitors find they can no longer scroll left/right/up/down to read the content.

If you use frames on your site - be sure to check how it appears under lower resolutions. Try reading all the content on your site with those resolutions. If you find that the site is inconvenient to use under those resolutions, you may need to rethink of your design. Remember, your visitors don't have the same amount of patience with your site that you have.

Framed pages also pose certain difficulties for people who have to use speech software to access your pages, such as the visually impaired. Unlike people using a visual web browser, the speech software reads every item on your pages and frames serially. The person using such software does not have the ability to skip portions of the page because it appears irrelevant. Neither is he able to match what is displayed on one frame with the content appearing in another. Remember - matching the content of one frame with another requires ability to see the layout.

It does not mean that this necessarily precludes the use of frames on your site. What is needed, instead, if you think you really need to use frames on your site, is to plan carefully so that both the user with a low screen resolution and the person using speech software are able to access your site as you intend.

4. JavaScript Availability

Should you use JavaScript on your web page? While it is true that many people use browsers like Internet Explorer, Netscape, Mozilla (and derivatives) and Opera, all of which support it, it is also true that there is a small percentage of people who use browsers that do not support JavaScript. This small percentage is not confined to people using old browsers or those that have disabled JavaScript in their

browsers. People who use handheld devices may not be able to read the content generated using JavaScript on your site.

A good preference is not to rely entirely on JavaScript to get the job done. That is, whenever you use JavaScript in your pages, try to make the page work even if JavaScript is not available. "work", in this case, means that the visitor is still able to navigate the site and read the material on the page.

❖ **Conclusion**

Plan your website from the start to take into account compatibility issues. This way, you know that your site will benefit from the widest audience that you've strived so hard to obtain.