Introduction

Welcome to the second edition of very my first HTML tutorials e-book! This e-book is the same as the tutorials I have on the website (www.htmlonline.tk/tutorials/html/).

I regard this tutorial as intermediate-level HTML, as compared to my 'HTML for Starters' e-book, which is just a brief introduction to HTML. But after reading this e-book and practicing enough, you would be ready to make good-enough looking webpages and websites, trust me!

At this point, I would like to make mention of a plain-text editor that I developed to facilitate HTML learners. The editor is called ProTx™ and you can download it free at www.htmlonline.tk/tools/protx.php. Although at the time of writing of this material, ProTx™ is still the way it was in 2004 but I intend to improve it and release an upgraded and better version.

ProTx™ is not so different from Notepad or any other regular plain-text editor you may have access to, but ProTx™ has some extra features which you might find interesting and useful when learning HTML, and that is why I also recommend readers of my HTML lessons to use ProTx™.

I hope that this e-book gives you a proper starting point for website design and development.

Happy reading!

Kheme

About Me

My name is Okiemute Omuta, but I’m very well known as Kheme. I am a Nigerian freelance website designer and developer specialized in PHP & MySQL. I also have a good understanding of JavaScript, DHTML and CSS. I also have a considerable applicable knowledge of Flash, Action Scripts, AJAX, C/C++ and Oracle SQL.

I have gathered experience in designing websites commercially using PHP, MySQL, (X)HTML, CSS, JavaScript and DHTML since 2003. Hence, I am able to develop both static and dynamic websites, from small scale websites through medium scale to medium-large scale websites.

Besides developing websites, I run a free website design tutorials website where I have website design related lessons. The name of the website is HTML Online (www.htmlonline.tk); I also have a few free e-books on that website, such as this one you’re reading right now.

If you noticed any spelling or typographical errors, or you just like my tutorials, or you have problems or questions to ask, feel free to email me at Kheme@htmlonline.tk or visit my website at www.khemeonline.tk. Also, I consider it consolation for my efforts when my readers e-mail in to say a thing or two about my e-books, or simply just to say “Thanks!” :)

Feel free to share this e-book with you friends and those you feel could benefit from it.

Thanks for downloading one of my e-books.

Kheme

Table of Content

Introduction to HTML

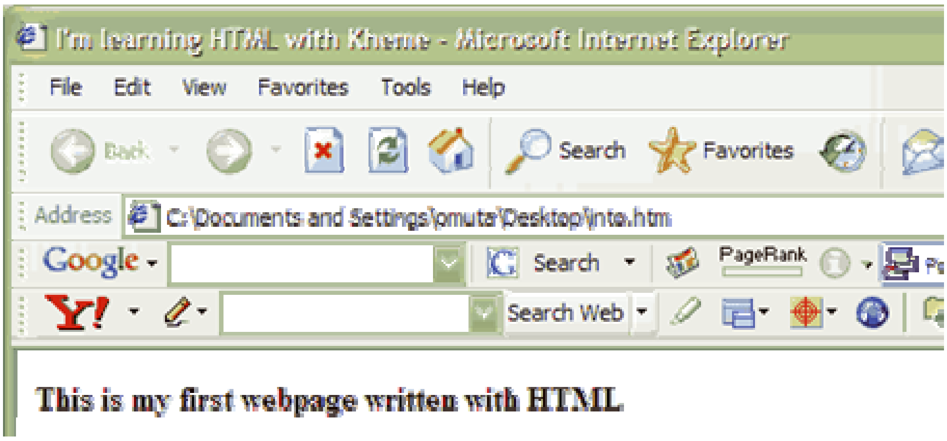
HTML means **H**yper**T**ext **M**arkup **L**anguage, it is the language used to tell a web browser *how* to display information, such as text, images, etc., on a webpage. A webpage is a single page on a website and a collection of these webpages linked together forms a website.

For a web browser to display a webpage using HTML, your HTML *tags* have to be saved in plain-text using a plain-text editor such as Notepad or ProTx™ and saved with the extension ".htm" or ".html" e.g. mypage.htm or mypage.html.

To begin writing webpages with HTML, copy and paste the following *tags* into your plain-text editor (Notepad or ProTx™) and save it as "intro.htm" on your desktop or any other location where you can easily access it.

```
<html>
<head>
<title>I'm learning HTML with Kheme</title>
</head>
<body>
<b>This is my first webpage written with HTML</b>
</body>
</html>
```

After saving the HTML file, open it by double-clicking on it and it should open up displaying something like this:


I'm learning HTML with Kheme - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
C:\Documents and Settings\omuta\Desktop\into.htm
Google
Search
PageRank
Search Web

This is my first webpage written with HTML

I will briefly explain what I have above because this is an introduction; when we go into the proper lessons, you will understand these things in details… don't worry.

The first line tells the web browser that it is viewing a HTML document and that it should be treated at HTML. The information placed within `<head> </head>` is usually not visible in the web browser (is used by web browsers and search engines). If you noticed the title bar of your web browser, it has the phrase "I'm learning HTML…" meaning that, the `<title> </title>` controls what is displayed as the title of the webpage on the title bar of the browser. Now, everything visible on a webpage is usually placed within `<body> </body>` including text, images, etc. The `<b> </b>` tells the browser to display the text within it as **bold**.

So, was that easy? You bet it was!

NOTE: Each time you change or add some new code into your HTML file with your plain-text editor, simply save your work (CTRL+S) and return to your opened webpage and refresh (F5) to see the new changes.
For ProTx™ users: you can quickly preview changes after saving by simply pressing ALT+Z.

With this introduction, let's begin proper definition of things… shall we?

HTML Tags

A HTML document or file is made of HTML tags called *elements.* HTML tags are usually written in the following format:

```
<tagname> </tagname>
```

So a typical HTML tag or HTML element would look like:

```
<b> </b>
```

This tag tells the browser to display the text within `<b> </b>` in **bold**.

Start & End Tags

By now you must have noticed that I wrote the tags in pairs (e.g. `<b> </b>`, `<marquee> </marquee>`, etc.); that's because we have to tell the browser (for example) where the bold text ‘starts’ and where it ‘ends'. So the first tag (e.g. `<b>`) is called the start tag while the second tag is called the end tag. So like I said earlier, only the text, etc. placed with the start and end tags will be affected by the tag; get it?

HTML Tag Attributes

HTML tag attributes add certain properties to the tag. HTML tag attribute are added to a tag like shown in the example below. In this example, the *size* attribute of `<font> </font>` determines the size of the font (or text) placed with the tag; try it!

```
<p>
This is some normal text while <font size="3">this is some
larger text</font>
</p>
```

Attributes are added to tags in this format:

```
attributename="value"
```

Note: Spaces are NOT allowed in attribute names! e.g. size="3"

For attribute values, it is okay to use either single quotes or double quotes, so we can have cases like:

```
<input name="name" value="what's your name?" type="text">
```

So using single quotes (') to specify attribute values allows you to used double (") quotes within attribute values, while using double quotes (") to specify attribute values allows you to use single quotes (') within attribute values.

Default Attribute Values

When a tag is used by itself (e.g. <p>), all its attributes are set to their default values. When you set any value manually, the default values ignored. For example, the *dir* attribute of <p> by default is set to "ltr" (left-to-right), so the (default) text direction is ltr. If we set it by ourselves like below,

```
<p dir="ltr">some text</p>
```

the text direction still won't change because we've set it to its default value and in this case, adding the *dir* attribute is as good as not adding it at all! If we want something different, then we can change it like this:

```
<p dir="rtl">some text</p>
```

Now the text direction is set to rtl (right-to-left) which is no longer the default value. This works for all the attributes of all other tags: default attribute values are used if you didn't set any!

Basic HTML Template

The basic HTML webpage temple is something that I recommend each time you create a new webpage with HTML; it should contain at least the following tags at ALL times:

```
<html>
<head>
<title>You page title here</title>
</head>
<body>
Other stuff you want to add goes here
</body>
</html>
```

Since I’ve explained the tags in this template and their functions, I will now go further to explain some other things about writing HTML.

Note: For practice sake, all new tags I teach in this tutorial should be nested within the body tag in the HTML template suggested above, unless I state otherwise.

HTML Tag Nesting

Each time you code in HTML, it is important that you nest you HTML tags to ensure that your webpage is properly displayed. Please refer to my HTML template above; you'll notice that I used <html> as the first and last tags on the page, then <title> </title> within <head> </head>. This arrangement is called 'Tag Nesting'. Now, to properly nest your HTML tags, follow this simple rule:

The tag you open first MUST be closed last, then the tag you open second must be closed before closing the first opened tag… get it? So we can have something like:

```
<html>
<head>
<title>You page title here</title>
</head>
<body>
<p>
This is my first paragraph with some <font size="3"><b>BIG
BOLD</b></font> text and some <u>underlined text in
<i>italics</i></u>
</p>
</body>
</html>
```

So, following the rule above, the tags above should be easy for you to understand. Improperly nested tags could look something like this:

```
<html>
<head>
<title>You page title here
</head>
</title>
<body>
<p>
This is my first paragraph with some <b><font size="3">BIG
BOLD</b></font> text and some <u>underlined text in
<i>italics</u></i>
</body>
</html>
```

These tags are terribly nested and you should make it like second-nature to ALWAYS nest your tags properly!

Another rule for nesting your tags is to create the start and end tags at once, and then do all the other tags within:

```
<html>

</html>
```

So I start with <html> and end it immediately so I'll never have to forget. Then the next tags can follow in the same manner:

```
<html>
<head>

</head>
</html>
```

And so on until…

```
<html>
<head>
<title>Welcome to HTML 101 with Kheme</title>
</head>
<body>

</body>
</html>

</html>
```

Follow these rules when nesting your tags and you would have learned to keep you code tidy!

I prepared a short video to explain this rule; click here to download it.

HTML and White Spaces

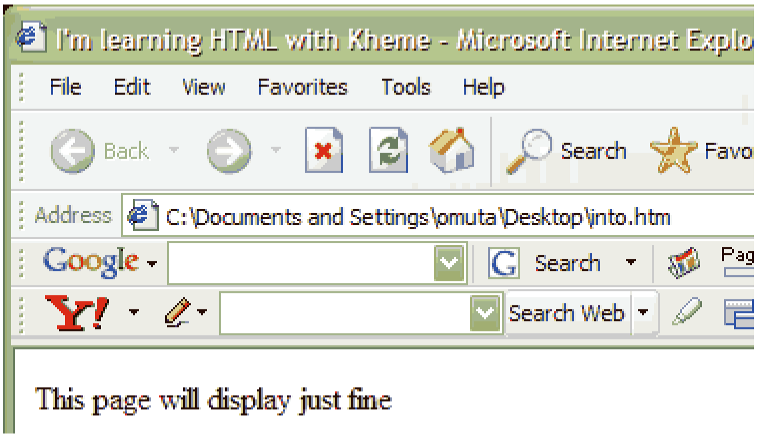
A white space is your normal space when you press the spacebar on your keyboard, but the deal with HTML and white spaces is that HTML only sees one (1) white space even when more than one (1) white spaces are used consecutively! HTML also sees carriage returns (↵ Enter) or new lines in your HTML file as a single space. Using out HTML template try the lines below and observe the results:

```
<p>This   page will



            display                          just fine</p>
```

What did you get? Here's what I got:


I'm learning HTML with Kheme - Microsoft Internet Explo
File
Edit
View
Favorites
Tools
Help
Back
Search
Favo
Address
C:\Documents and Settings\omuta\Desktop\into.htm
Google
Search
Pag
Y!
Search Web

This page will display just fine

So, as you're probably wondering "how on earth am I going to use multiple spaces?" Well, it’s pretty simple; we use Character Entities. These are HTML forms of special characters like the angle brackets (< and >) or the white space or the copyright sign (©) or the hash sign (#), etc. Without these entities it would be impossible for you to write the HTML codes on webpages as I displayed here because the web browser sees every <p> as the 'start' of a paragraph tag and doesn't know if you want it to display or ‘interpret’ it! I will give you a list of these entities for you to play with.

Name	Character	Entity
white space		&nbsp;
angle bracket	>	&lt;
angle bracket	<	&gt;
copyright	©	&copy;
registered	®	&reg;
trademark	™	&trade;
multiplication	×	&times;
Division	÷	&divide;
ampersand	&	&amp;

You can also use the ASCII format of these entities too. Here are the same entities in the ASCII form; they work just the same way, try them!

Name	Character	Entity (ASCII format)
white space		&#160;
angle bracket	>	&#60;
angle bracket	<	&#62;
copyright	©	&#169;
registered	®	&#174;
trademark	™	&#153;
multiplication	×	&#215;
Division	÷	&#247;
ampersand	&	&#38;

Note: I have created a page that might be fun for you to use; it generates as many HTML entities as you want and gives their corresponding ASCII form. You can access it at http://www.htmlonline.tk/tools/ascii.php

Basic HTML Tags

In this lesson, I'll be teaching you a few HTML tags that we may call basic HTML tags.

Note: from this point forward, I will indicate HTML tag names in ***bold italic words*** (e.g. table), while HTML tag attributes will be indicated in normal italic words (e.g. *align*).

The Paragraph Tag

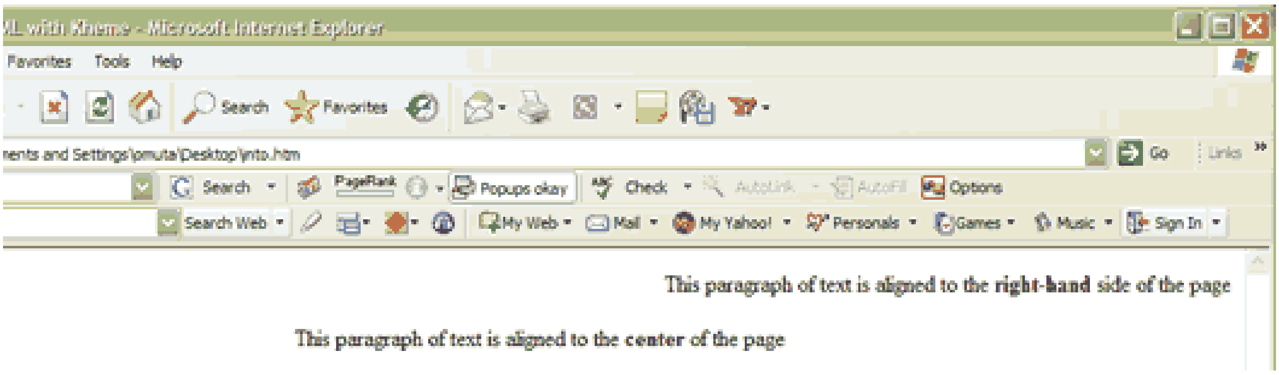
The ***paragraph*** tag (<p> </p>) is used to indicate paragraphs of text on webpages. For every paragraph you want, use <p> </p>; try this:

```
<p>This is my first paragraph of text</p>
<p>This is the second paragraph</p>
<p>This of course is my third paragraph. To add more
paragraphs, use more &lt;p&gt; &lt;p/&gt;</p>
```

The *align* attribute sets the text alignment of each paragraph. Try this:

```
<p align="right">
This paragraph of text is aligned to the <b>right-hand</b>
side of the page</p>
<p align="center">
This paragraph of text is aligned to the <b>center</b> of the
page</p>
```

Here's what I got:



ML with Xheme - Microsoft Internet Explorer
Favorites Tools Help
Search Favorites
nents and Settings\pmuta\Desktop\nto.htm
Go Links
Search PageRank Popups okay Check AutoLink AutoFill Options
Search Web My Web Mail My Yahoo! Personals Games Music Sign In

This paragraph of text is aligned to the right-hand side of the page

This paragraph of text is aligned to the center of the page

HTML Comment Tags

HTML ***comment*** tags allow you to add comments to your HTML codes. Although these comments are visible in your HTML codes, it will not be displayed when viewed in a web browser. There are two ways of writing comments in HTML and I'll treat them together; try these lines below:

```
<comment>
This is a comment and it will not show on your webpage
</comment>

<!--
This is another comment; comments can be used to provide extra
information like explanation for your HTML code to aid you in
editing it later.
-->
```

Everything within `<comment> </comment>` or `<!-- -->` are treated as HTML comments and will not be displayed in the browser.

Line Brakes

To insert new lines in your webpages, should use the ***line brake*** tag `<br>`; try these lines:

```
<p>
This is a regular paragraph.
<br>This will be a new line on the same paragraph
</p>
```

Note: In HTML, not all tags have ending tags and <br> is one of them.

Horizontal Lines

To insert horizontal lines in your webpages, you should use the ***horizontal rule*** tag <hr>; try these lines:

```
<p>
This is a regular paragraph,
<br>and this will be a new line on the same paragraph
<hr>
This text will appear just under a grayish horizontal line
</p>
```

Also observe that <hr> is another tag that does not have an ending tag. Some other tags without ending tags are: <meta>, <img>, <link> and <bgsound>. You can make use of the following attributes of <hr>:

Attribute	**Function**	**Examples**
color	sets the color of the line	<hr color="black">
size	sets the size (thickness) of the line	<hr size="2">
align	sets the alignment of the line	<hr align="right">

Heading Tags

You can insert headings into your webpages with the ***heading*** tag; try these lines below:

```
<h1>This is a heading</h1>
<h2>This is a smaller heading</h2>
<h3>This is a much smaller heading</h3>
<h4>This is another much smaller heading</h4>
<h5>This is yet another much much smaller heading</h5>
<h6>This is the smallest heading possible</h6>
```

You can make use of the *align* attribute with the ***heading*** tags as you wish. Try something new by yourself!

Body Tag

I know we've done some stuff with <body> </body> since we've been using it in our basic HTML template, but I will now show you some useful attributes of the ***body*** tag.

Below is a table of attributes, their function and examples; I want you to try them out in your own way, using your own different combinations!

Attribute	Function	Example
bgcolor	sets page background color	`<body bgcolor="navy">` or `<body bgcolor="lightgrey">`
bgsound	sets page background sound	`<body bgsound="loop.mp3">`
text	sets color of text	`<body text="teal">`
link	sets the color of hyperlinks	`<body link="black">`
scroll	adds/removes page scrollbars	`<body scroll="yes">` or `<body scroll="no">`

Now we've covered these basic tags, we can get ready for the next lesson: Formatting with HTML

Formatting with HTML

In this lesson, I will teach you how to format text on webpages using HTML; I will teach you how to underline text, make them **bold**, *italics* or even ***bold italics and underlined*** altogether! I will also teach you how to color your text, set line spacing, text width, etc.

Bold

To make a text bold, we use the ***bold*** (<b> </b>) tag; try these lines:

```
This is normal text while <b>these ones are bold</b>
```

I will now give you a list of tags for you to play with; try all of them in different combinations, experiment with them, okay :) ?

Tag Name	**Function**	**Examples**
Bold	makes text bold	<b>bold text</b>
Italics	italicizes text	<i>italics</i>
Underline	underlines text	<u>underlined</u>
Delete	strikes text horizontally, indicating it has been deleted!	<del>struck through</del>
Big	makes text bigger in size	<big>bigger text</big>
Small	makes text smaller in size	<small>smaller text</small>
Code	makes text appear like computer *code*	<code>some code</code>
Reformat	reformats text	<pre>some text</pre>
Keyboard	indicates text to be typed (with keyboard)	<kbd>CTRL + V</kbd>

Note: `<u> </u>` is rather depreciated now and *styles* or *CSS* are recommended instead.

`<code> </code>` is applicable when you want to write out some codes (such as HTML codes or other program codes) on your webpage

`<pre> </pre>` is very good when you want to write some code on your web page also, but you can try it against `<code>` and `<kbd>`, and decided whichever works best for you. You will also notice that `<code>`, `<kbd>`, and `<pre>` all produce same or similar effects.

Font Tag

I will now teach you how to format text using the ***font*** tag and some of its attributes. The font tag (`<font> </font>`) in itself, does nothing to text until you add some attributes. We will not go through a few of these ***font*** tag attributes one at a time. Please note that `<font>` is now depreciated and will not longer be in use with the new HTML (XHTML); but you can still perform all I'm about to teach you using styles or CSS.

Size: The *size* attribute of the font tag sets the size of the text. Values for this attribute are the numbers 1 through 7 with one being the smallest size and 7 being the largest size.

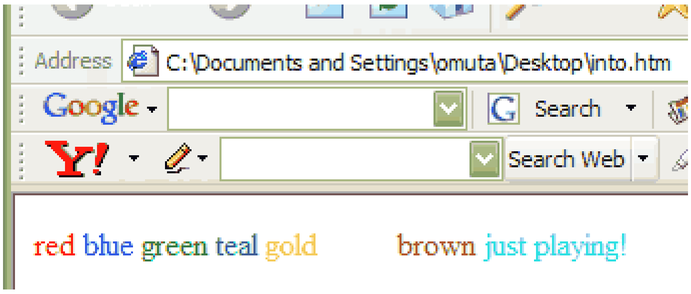
```
<font size="1">size 1</font>
<font size="2">size 2 </font>
<font size="3">size 3 </font>
<font size="4">size 4 </font>
<font size="5">size 5 </font>
<font size="6">size 6 </font>
<font size="7">size 7 </font>
```

Color: The *color* attribute sets the color of the text. Please note that the spelling of the attribute is "color" (American) and not "colour" (British), and using "colour" instead of "color" doesn't work!

In another lesson (Specifying Colors), I will teach you the different ways you can specify colors in HTML but for now, we'll use the known names of colors, so try these lines:

```
<font color="red">red</font>
<font color="blue">blue</font>
<font color="green">green</font>
<font color="teal">teal</font>
<font color="gold">gold</font>
<font color="white">white</font>
<font color="brown">brown</font>
<font color="kheme">just playing!</font>
```

Note: I discovered a little trick, that using regular words as color names also works! Try my surname (omuta) as a color name and it gives you navy blue, while "kheme" gives you something like aqua! Try your name, and see what it gives!



C:\Documents and Settings\omuta\Desktop\into.htm

red blue green teal gold brown just playing!

Face: The *face* attribute sets the font face type of the text; try these lines:

```
<font face="comic sans ms">
famous font for e-greeting cards online!
</font>
```

Note: It is recommended that you use fonts that are present on most computer systems by default! If you use a font that's on your system but absent on someone else's system, your page won't display well on that other system! One way to solve this is to set alternate fonts to be used in the absence of some font; look at the line below:

```
<font face="arial, verdana, tahoma">some text</font>
```

What this tag and it's attributes tell the browser is this: "use arial to display the text, and if no arial is absent on this computer system, use verdana, and if no verdana, then use tahoma, and if none of these fonts are present, then go ahead and use which ever font you like, to display the text, okay?" and that's exactly what the browser does!

If the browser is asked to display text using a font that's not present on the user's system, it will display the text using the browsers default set font which is usually "Times New Roman" (Internet Explorer's default) or whatever you have set in your browser. Always try to avoid using foreign fonts so your page won't appear "ugly" on someone else's browser!

Lists

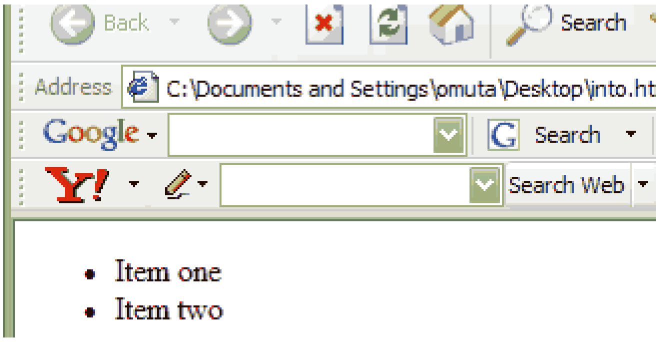
I'll now show you how to do this:

- this is a list item
- this is another one
- and I can go on listing items
- etc.

What I have above is a "list" called an "unordered list", while there's another called an "ordered list" which is numbered; let’s look at the ***unordered list*** tag:

```
<ul>
<li>Item one</li>
<li>Item two</li>
</ul>
```

I've used two tags here: the first is <ul> (**u**nordered **l**ist) and the other is <li> (**l**ist **i**tem). <ul> tells the browser to expect a "list of items" and that it should NOT order them, so they'll be listed as bullets. Try it, here's what I got:



- Item one
- Item two

Now, to tell the browser to number the list or *order* the list, you simply change <ul> to <ol> (**o**rdered **l**ist); try it now!

There's also another kind of list called ***definition list***. The definition list is not actually a list of items but a list of terms or words and their definitions. To use this ***definition list***, you're going to work with three (3) tags: <dl>, <dt>, <dd>.

```
<dl>
<dt><b>HTML</b></dt>
<dd>HyperText Markup Language</dd>
<dt><b>CSS</b></dt>
<dd>Cascading Style Sheets</dd>
</dl>
```

<dl> means **d**efinition **l**ist and tells the browser to expect a list of definitions. <dt> which means **d**efinition **t**erm indicates a **d**efinition **t**erm while <dd> indicates the definition for the term in the previous <dt>.

Specifying Colors

In this lesson, I will teach you the various ways in which you can specify colors on webpages with HTML.

Color Names

One of the easiest ways to specify colors in HTML is to use its name, if you know it! We've done this before and I believe I don't have to spend extra time on this.

Hexadecimal Values

```
<font color="#ffff00">yellow colored text</font>
<font color="#800000">maroon colored text</font>
```

In the above example, the hash (#) and the numbers/alphabets that follow are together known as "hexadecimal" numbers. Hexadecimal (hex for short) is another type of number system like the decimal number system, but the hex number system uses 16 digits (0 to F); that is, 0, 1, 2, ..., 8, 9, a, b, c, d, e, f (decimal uses 10 digits, 0 through 9).

The hex values define the mixture ratio of the colors red, green and blue. So the first two characters define the degree of red to add to the color mixture, then the next two characters define the degree of green to add to the mixture while the last two defines the degree of blue to add to the mixture.

You can try out different combinations, but always add the hash sign (#) before the characters. My favorite hex color is the MSN blue (#336699), which is what I used in gradient with black for ProTx$^{\text{TM}}$.

Note: I have prepared a HTML color table listing some colors, their names and their hexadecimal values in the appendix section.

RGB Values

Using RGB (Red Green Blue) values is very similar to hex only that the hex values are converted back to decimal and used in a different way; so instead of:

```
<font color="#ffff00">yellow colored text</font>
```

you should use:

```
<font color="rgb(255,255,0)">yellow colored text</font>
```

To get the RGB equivalent of the hex values, you have to convert the first two digits from hex to decimal to give the first set of numbers before the first comma, then the next two digits from hex to decimal to give the next set of numbers after the first comma and lastly convert the last two digits from hex to decimal to give the last set of numbers.

FF in Hexadecimal = 255 in Decimal, while 00 in Hexadecimal = 0 in Decimal. You can do this conversion with a scientific calculator if you have one or you can use the small Hex to Decimal (H2D$^{\text{TM}}$) converter I made for this purpose: http://www.htmlonline.tk/tools/h2d.php

So converting `#ffff00` from Hexadecimal to RGB gives `rgb(255,255,0);` Try out some numbers by yourself!

Now, you should be able to specify colors at any point while writing your HTML codes, using up to four different methods. You can use these values to color any tag.

See you at the next lesson!

Working with Images

In this lesson, I'm going to teach you how to insert images (pictures) into your webpages using HTML, and a little image formatting.

Image Tag

To insert images into webpages, we make use of the ***image*** tag `<img>`. Note now that the ***image*** tag, `<img>`, has NO ending tag; in fact, the ending tag for `<img>` is forbidden! Please follow my example below and my explanations:

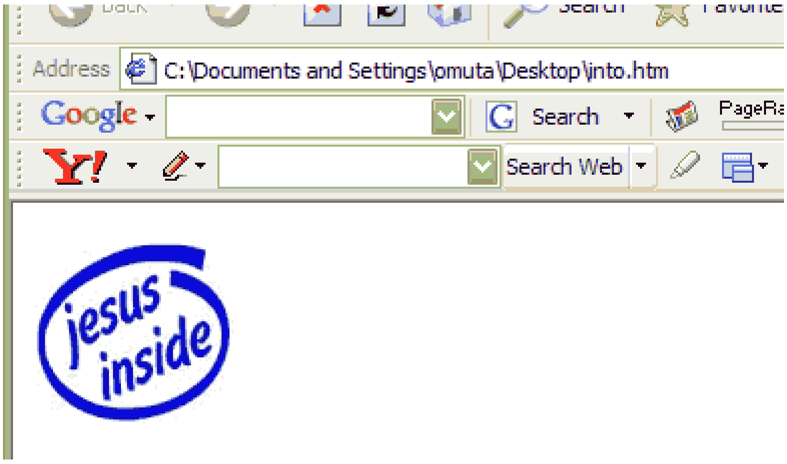
```
<img src="mypic.jpg">
```

What I did here was to use the *source* attribute (written as "src") to specify the location of the image I want to insert into my webpage. I have used the image name and its extension.

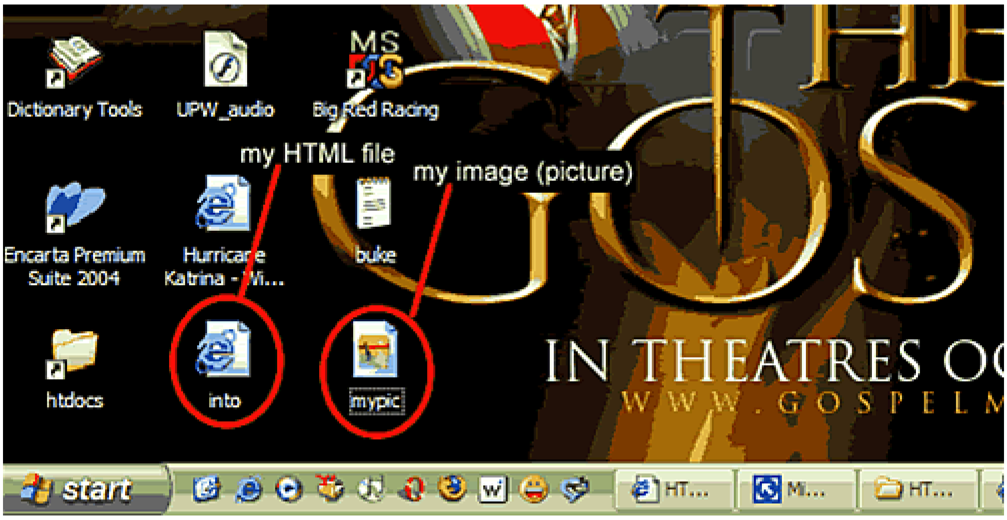
For first-timers, the extension of the image is the last three letters in the name (.jpg in my example) while the file name itself is "mypic". You must use the name and extension to specify an image file name when dealing with images.

Try This:

Make a copy of the image you want to use and place it on your desktop now. In your HTML file on your desktop, copy the line from my last example and enter the image’s name and its extension like in my example above, save it and preview it. What did you get?



One thing to point out here is this: to insert images by using only its name (e.g. mypic.jpg), then the HTML file and the image to be inserted MUST be in the same folder! So, if your HTML file is still on your desktop, like I suggested earlier, make a copy of the image you wish to use and paste it on your desktop now. Look at how I have it on my desktop right:



Now, if you want to insert an image that is in a different location, you have to know the image's absolute path location. This means that you need to know its location in relation to the drive in which it is saved, or its complete internet address, including the http://www. etc. (if the image is located online). The location I used here is 'relative' to the containing folder (which is my desktop).

If you want to insert an image located, for example, in your WINDOWS folder on drive C: then your tag should look something like this:

```
<img src="c:/windows/Coffee Bean.bmp">
```

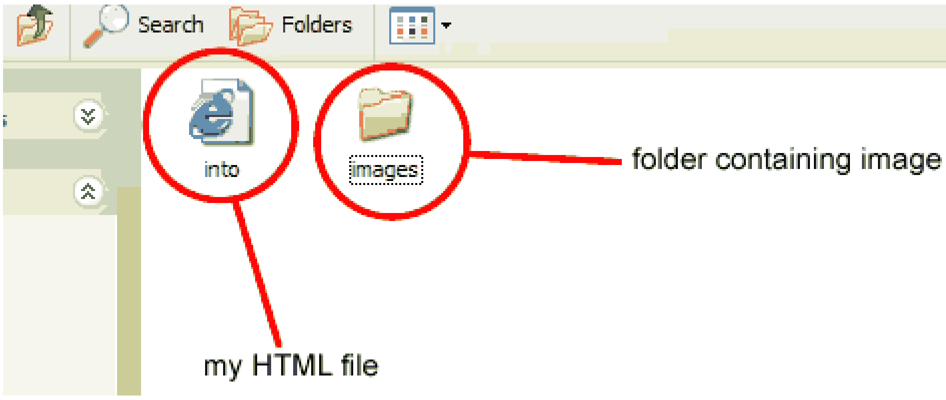
Or if it were in some other folder, it might look something like this:

```
<img src="C:/My Games/Bejeweled 2 Deluxe/images/action.gif">
```

If you want to insert an image located on the internet, you'll have to use the full URL of that image:

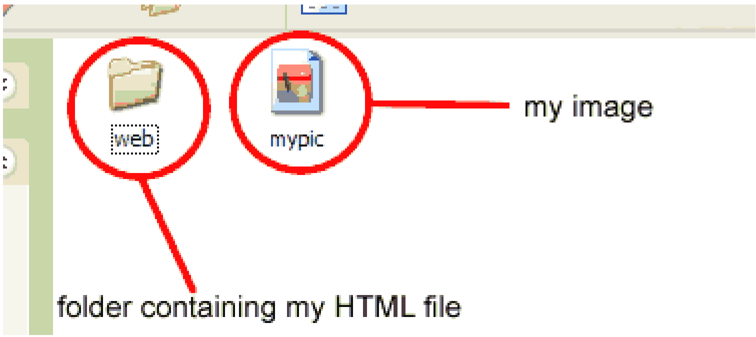
```
<img src="http://www.htmlonline.tk/images/htmlonline.png">
```

If you want to insert an image that's located in a folder within the folder containing your HTML file (i.e. the folder containing the image and the HTML file are in the same folder) you simply enter the name of the folder, followed by a slash (/) and then the name of the image.



```
<img src="images/mypic.jpg">
```

Now, if you want to insert an image that's located in a folder just outside the folder containing your HTML file (i.e. the image and the folder containing your HTML file are in the same folder), you simply enter two (2) dots followed by a slash (/) and then the image name.



```
<img src="../mypic.jpg">
```

Note: Avoid using spaces in your file names (HTML files, images, CSS files, etc.); instead, use underscores (_) or dots (.) instead of spaces. This would be proper: "star_logo.gif", while this would be improper: "star logo.gif". Using spaces may cause some problems displaying them on some other browsers, so beware!

Alternate Text

With the *alt* attribute of `<img>`, you can set an "alternate" text for the image. This text will be displayed on the image if for one reason or another, the image couldn't load.

```
<img src="http://www.htmlonline.tk/images/htmlonline.png"
alt="HTML Online">
```

Borders

To insert a border around your image, use the *border* attribute:

```
<img src="mypic.jpg" border="1">
```

Values for this attribute are in numbers: the higher the number, the thicker the border. Try it!

HTML Head

I'm going to teach you some of the tags that are placed within `<head> </head>`, one of which we've already used (`<title> </title>`).

Usually, anything placed within `<head>` is not visible on the webpage, but some browsers still display things placed here! In this lesson, I'll treat only five (5) tags briefly: `<meta>`, `<script>`, `<style>`, `<link>` and `<base>`; little has to be said about `<title> </title>`.

Meta Tag

The ***meta*** tag is used to carry information used by the web browser and some search engines. Using the appropriate attributes and values, you can use `<meta>` to do the following:

- redirect the browser to another page after some time (in seconds)
- set the default editor (for Internet Explorer) when you attempt to edit the web page you're viewing
- describe a webpage content for a search engine
- list keywords for search engines to use in indexing your webpage

Note: Please note that the ending tag for `<meta>` is forbidden!

Browser Redirection: Below is a typical web browser redirection page; you can change the things you want if you like to use it.

```
<html>
<head>
<title>You page title here</title>
<!--
you can change the time (in seconds) when you want the browser
to be redirected. You can also replace "mypage.htm" with the
location of the page you’re redirecting the browser to;
remember to include the full location of the page, unless it
is located within the same folder as webpage from which your
redirecting.
-->
<meta http-equiv="refresh" content="2;url=mypage.htm">
</head>
<body>
<p align="center">
You will be redirected to a new page shortly!
</p>
</body>
</html>
```

For Search Engines: In the example below, I have written "description" and "keywords" for search engines to use when indexing or crawling my webpage on the internet.

```
<html>
<head>
<title>You page title here</title>
<meta name="description" content="HTML Online - Free HTML
online tutorials!">
<meta name="keywords"
content="HTML,free,lessons,tutorials,Kheme,BackandFront">
```

```
</head>
<body>
<p>Some text</p>
</body>
</html>
```

Other Uses: You can use `<meta>` to carry some other information. Such information may include the author of the webpage, contact e-mail, etc. Look at what I have here:

```
<html>
<head>
<title>You page title here</title>
<meta name="Author" content="Kheme">
<meta name="Email" content="kheme@htmlonline.tk">
</head>
<body>
<p>Some text</p>
</body>
</html>
```

Script Tag

The ***script*** tag is used to insert client-side scripts such as JScript or JavaScript and VBScript. I will say a little more about this tag later in the Introduction to JavaScript lesson. Anyway, try these lines:

```
<html>
<head>
<title>You page title here</title>
<script>
a=new Date()
alert("The time according to your computer is
"+a.getHours()+":"+a.getMinutes())
</script>
```

```
</head>
<body>
<p>put some content here if you want!</p>
</body>
</html>
```

Style Tag

The ***style*** tag is used for defining your Cascading Style Sheet or CSS properties. With CSS, you can do your entire formatting; positioning, colors, widths & heights, etc. I will say a little more about CSS later in the Introduction to CSS lesson. Meanwhile, try this:

```
<html>
<head>
<title>You page title here</title>
<style>
body {direction:rtl;background-color:black;color:white;cursor:help}
</style>
</head>
<body>
<p>
Put some content here to try out with the style I've defined.
</p>
</body>
</html>
```

Link Tag

The ***link*** tag (with no ending tag) is used when you have your CSS in an external file and you want to use it on a particular webpage. The tag below creates a link to the CSS file so that the present page can use the CSS defined in that CSS file.

```
<html>
<head>
<title>You page title here</title>
<!–
the "href" attribute defines the location of the CSS file
-->
<link href="styles.css" rel="StyleSheet">
<!–
the "rel" defines the relationship between the HTML file & CSS
file
-->
</head>
<body>
<p>
Nothing will affect this text because you do not have a
&quot;styles.css&quot; file that defines any styles
</p>
</body>
</html>
```

Base Tag

The ***base*** tag defines the "base" reference point for all locations on that page (e.g. for images, external CSS, external JavaScript, etc.)

```
<html>
<head>
<title>You page title here</title>
<base href="http://www.khemeonline.tk/blog/">
</head>
<body>
<p><a href=”30080601.php”>An interesting blog of mine</a></p>
</body>
</html>
```

The *href* attribute defines the reference point for all locations.

```
<html>
<head>
<title>You page title here</title>
<base href="http://www.htmlonline.tk">
</head>
<body>
<img src="mypic.gif">
</body>
</html>
```

From what I have here, it means that the actual location of the image is http://www.htmlonline.tk + mypic.gif which is http://www.htmlonlinetk/mypic.gif

And from the previous example above, the link to my blog entry would actually be http://www.khemeonline.tk/blog/30080601.php and any other link I place on that page would start with http://www.khemeonline.tk/blog/

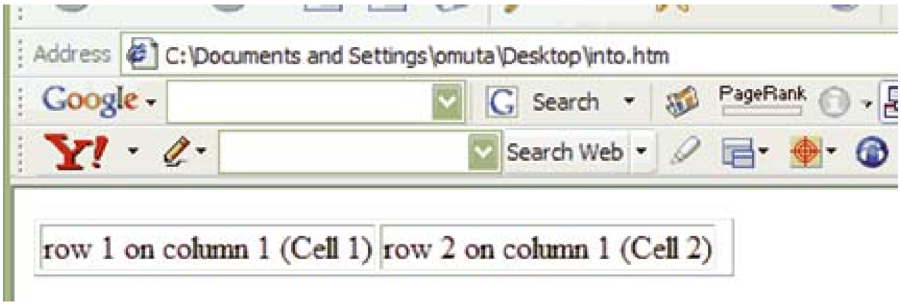
Using HTML Tables

Creating tables with HTML can be quite confusing for starters, and yes it was for me too! But if you to follow this lesson closely, you would do just fine ;)!

Table Tag

We use the ***table*** tag to tell the browser to expect a HTML table; but `<table> </table>` alone doesn't give us a table, because a table cannot exist without a cell! So we'll need two (2) extra tags nested within `<table> </table>` to give us a table.

```
<table border="1">
<tr>
<td>row 1 on column 1 (Cell 1)</td>
<td>row 2 on column 1 (Cell 2)</td>
</tr>
</table>
```

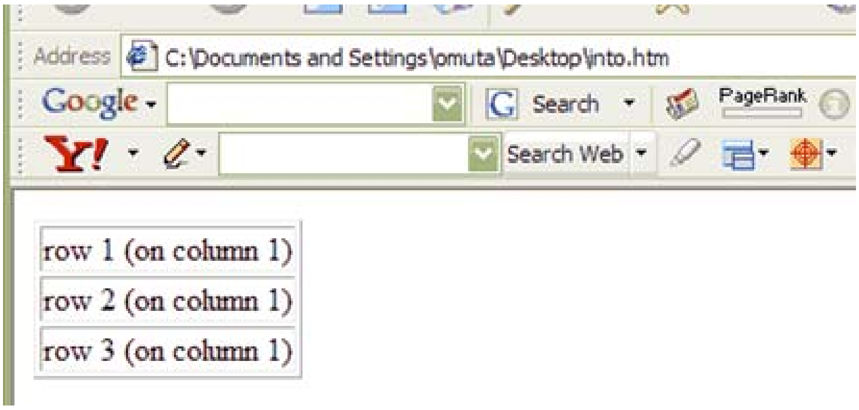


row 1 on column 1 (Cell 1)	row 2 on column 1 (Cell 2)

So like is said, `<table> </table>` tells the browser to expect a table while `<tr> </tr>` denotes a **t**able **r**ow and `<td> </td>` denotes at **t**able **d**ata or a (table) cell.

Table Rows: To insert rows into a ***table***, merely duplicate `<tr> </tr>` for as many rows as you want, but also remember that within each `<tr> </tr>` you MUST nest at least one (1) `<td> </td>` because no table can exist without a cell (which is an intersecting row and column).

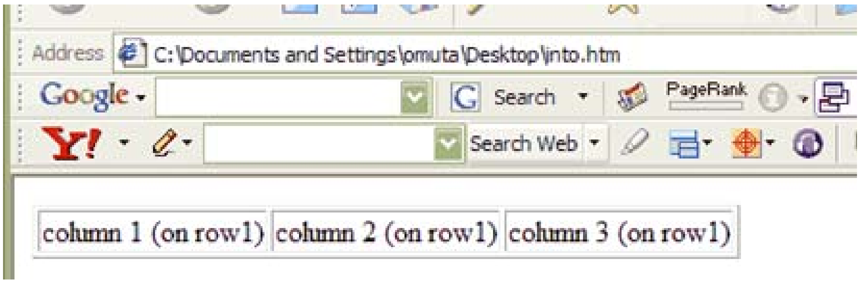
```
<table border="1">
<tr><td>row 1 (on column 1)</td></tr>
<tr><td>row 2 (on column 1)</td></tr>
<tr><td>row 3 (on column 1)</td></tr>
</table>
```



row 1 (on column 1)
row 2 (on column 1)
row 3 (on column 1)

Table Columns: To insert columns into tables, merely duplicate `<td> </td>` nested within a `<tr> </tr>` and that will give you a new cell, or rather, a column on the same row.

```
<table border="1">
<tr>
<td>column 1 (on row1)</td>
<td>column 2 (on row1)</td>
<td>column 3 (on row1)</td>
</tr>
</table>
```

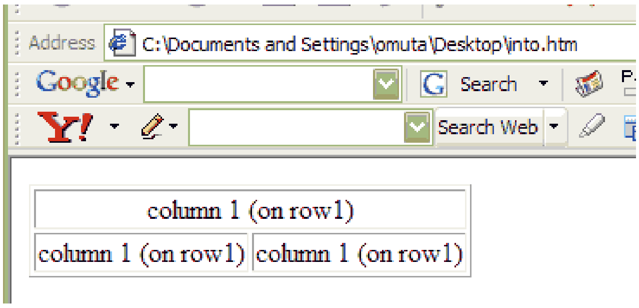


Address C:\Documents and Settings\pmuta\Desktop\into.htm

column 1 (on row1)	column 2 (on row1)	column 3 (on row1)

So, with what I've shown you, you can combine `<tr>` and `<td>` to give you the rows and columns you want.

Column Spanning: What if you want to do something like:



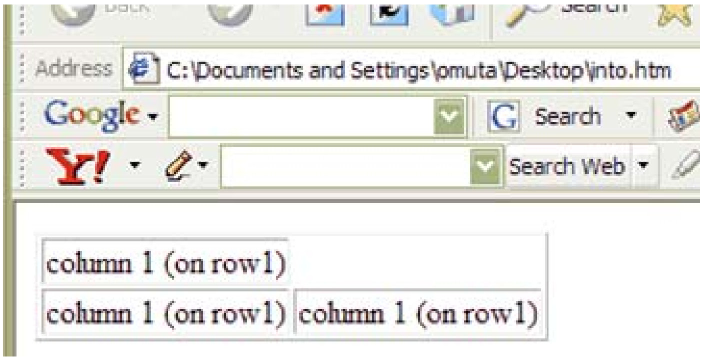
Address C:\Documents and Settings\pmuta\Desktop\into.htm

column 1 (on row1)	
column 1 (on row1)	column 1 (on row1)

Any ideas how to do it? Do you want to try it on your own? Go ahead and try! So here's what we'll do: First of all, note that we have one (1) column on the first row and two (2) columns on the second row, right? Okay. But if you go ahead and try this:

```
<table border="1">
<tr>
<td align="center">column 1 (on row1)</td>
</tr>
<tr>
<td>column 1 (on row1)</td>
<td>column 1 (on row1)</td>
</tr>
</table>
```

you will get this:



C:\Documents and Settings\omuta\Desktop\into.htm

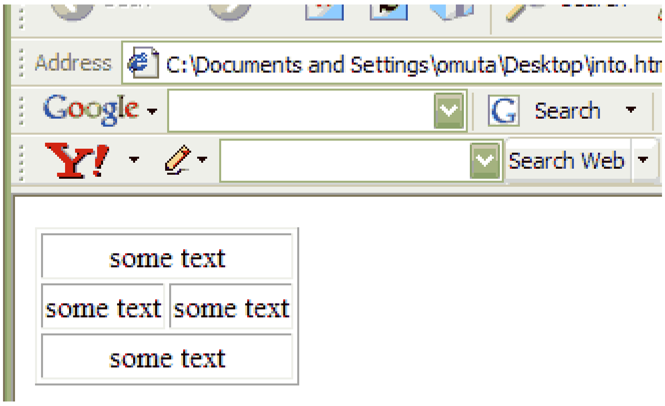
column 1 (on row1)	
column 1 (on row1)	column 1 (on row1)

What we need to do here is to add to the first row's tag, an appropriate value for the *colspan* attribute. The *colspan* attribute tells the browser to "span" the cell over a certain number of columns.

```
<table border="1">
<tr>
<td colspan="2" align="center">column 1 (on row1)</td>
</tr>
<tr>
<td>column 1 (on row1)</td>
<td>column 1 (on row1)</td>
</tr>
</table>
```

Of course you should know by now that the *align* attribute controls the text alignment within the cell.

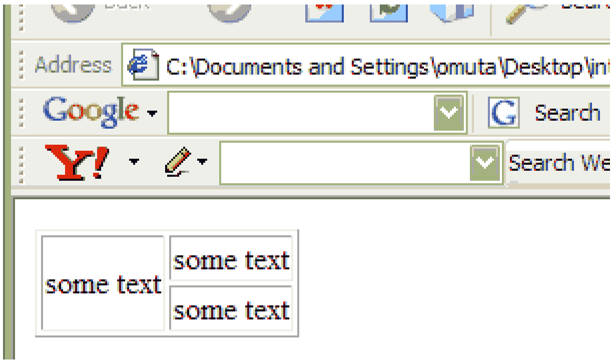
So get this right: when you want a column to *span* over x number of columns, add the *colspan* (**col**umn **span**) attribute with x as the value. Get it? Now that you know how, try to do this:



some text	
some text	some text
some text	

Click on this sticky note to see the codes I used.

Row Spanning: Now, what if we want to do this:



some text	some text
	some text

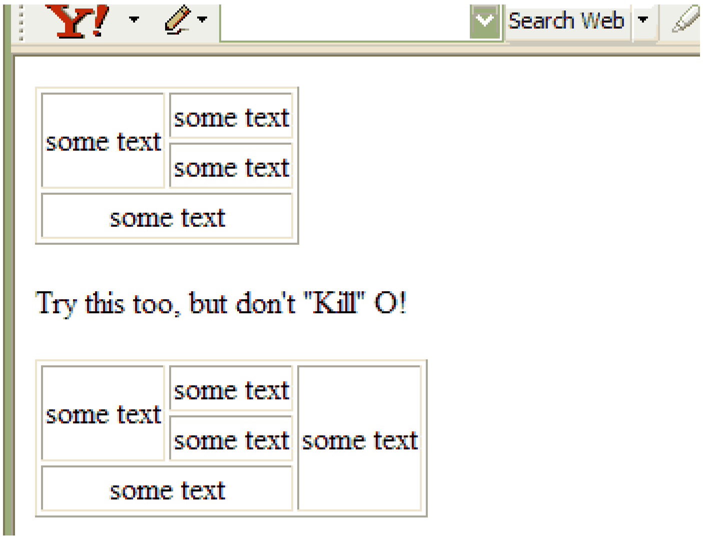
Can you guess? Okay, let's go through this together: we have two (2) rows and two (2) columns with the first column stretching (spanning) down over two (2) rows. So what we have to do first of all is create the table as I explained it; look at my tags:

```
<table border="1">
<tr>
<td>some text</td><td>some text</td>
</tr>
<tr>
<td>some text</td>
</tr>
</table>
```

Now, what we have to do is add a *rowspan* attribute with an appropriate value. The *rowspan* attribute tells the browser to stretch (span) a row over the next certain number of rows.

```
<table border="1">
<tr>
<td rowspan="2">some text</td><td>some text</td>
</tr>
<tr>
<td>some text</td>
</tr>
</table>
```

Try out some other ways you can span rows and columns, also try doing what I have below:



some text	some text
	some text
some text	

Try this too, but don't "Kill" O!

some text	some text	some text
	some text	
some text		

Table Width & Height

Now, I'll teach you how to manipulate the *width* and *height* of a table and its cells.

Table Width: Using the *width* attribute, we can control the width of a table, simply enter a number to set the value. Try this:

```
<table border="1" width="300">
<tr><td>some text</td></tr>
</table>
```

We can also set the width (as well as the height) in percentages:

```
<table border=”1” width=”50%">
<tr><td>some text</td></tr>
</table>
```

What I have above tells the browser to set the width of the table to 50% of the width of the tag containing the table (in the case, the ***body***), that's how percentage values work!

Table Height: With the *height* attribute, we can set the height of the table. Just enter a number (e.g. 100) or use percentage (e.g. 25%).

```
<table border="1" width="50%" height="100">
<tr><td>some text</td></tr>
</table>
```

Now try different widths and heights for your table, use numbers and percentages too.

Note: The effect of using percentage is that when the browser's page is resized, the table automatically re-adjusts its height/width to fit the new size of the page. I made a short video clip to demonstrate this. Click here to download the short clip and observe the black border as it resizes.

Cell Width: We can also manipulate the width of a cell but using the *width* attribute on the `<td>`.

```
<table border="1" width="50%" height="100">
<tr>
<td width="20%">some text</td>
<td width="80%">some text</td>
</tr>
</table>
```

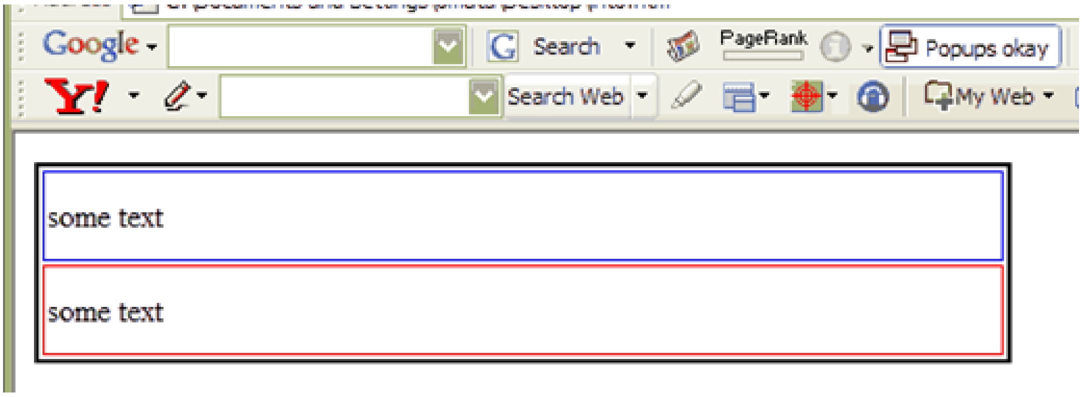
Now, in the HTML I have above, I used percentage to set the width of the second row. Please note that in this case, the percentage here is of the table's width because it's the table that contains the cell. This means that the width of the `<td>` (80%) is actually 80% of the 50% of the width of the table, which is actually 40% of the total table width! You can also use numbers to set the width of the cell.

```
<table border="1" width="50%" height="100">
<tr>
<td width="200">some text</td>
<td width="400">some text</td>
</tr>
</table>
```

I think I really don't have to give you a lesson on cell height, right? You can do that on your own... I trust you ;) ! Now, you can also adjust with thickness of the table by using higher numbers (e.g. 2 or 3) as *border* for the table. You can also use the *align* attribute to set the table's text alignment to left, center or right.

Border Color

You can change the color of your table borders using the *bordercolor* attribute on your ***table*** tag. You should know how to specify colors by now, so I don't have to dwell on that! You can also use the same attribute (*bordercolor*) to set the color of the borders of individual cells too. Now try doing this (click on the sticky note to see how I did it):



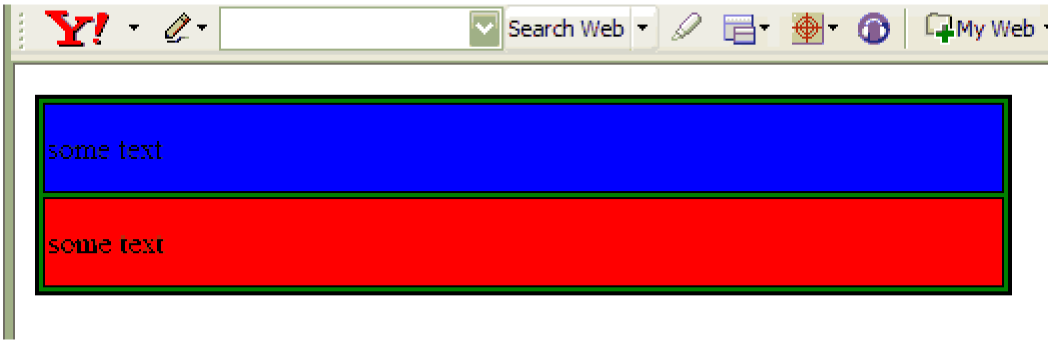
some text

some text

Note: If you don't know how to specify colors in HTML, please read the Specifying Colors lesson.

Background Color

Using the *bgcolor* (**back**g**round color**) attribute on `<table>`, `<tr>` and `<td>`, you can set their background colors. Now try to do this (click on the sticky note to see how I did it):



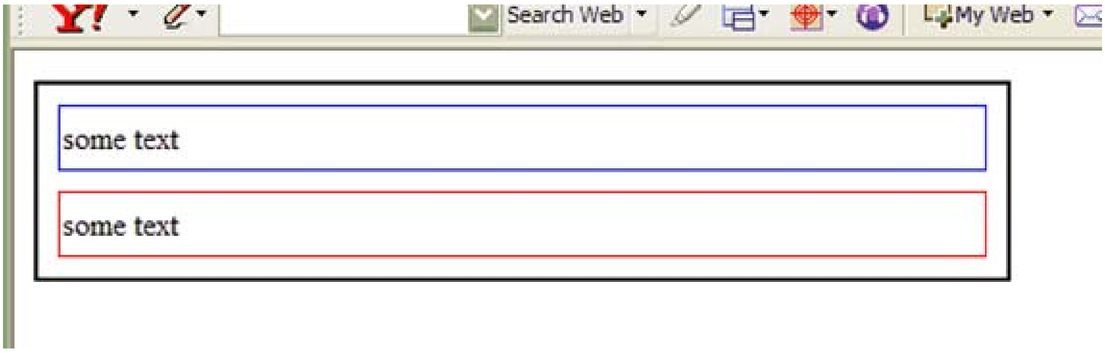
some text

some text

Cell Spacing & Padding

You can set the space between each cell in a table and the space between the cells & the table's border using the *cellspacing* attribute. In the example I have below, I used a large value for my table's *cellspacing* so you can easily see it; notice the space between both cells and the table's border.

```
<table border="1" width="50%" height="100" cellspacing="5">
<tr>
<td width="200">some text</td>
<td width="400">some text</td>
</tr>
</table>
```

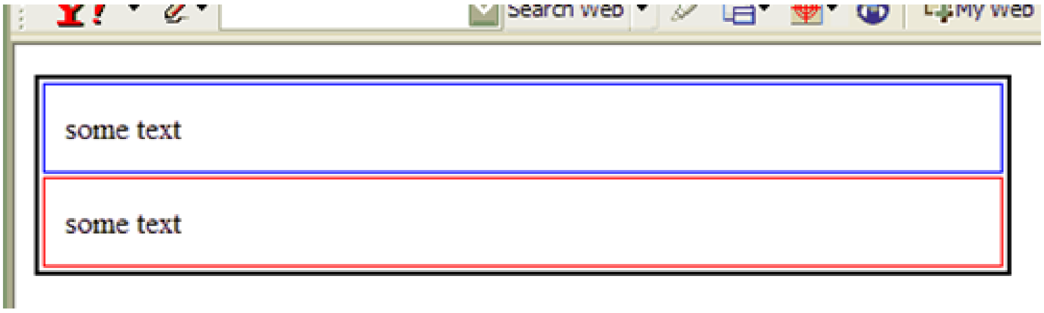


some text

some text

Cell Padding: You can also set the padding of each cell; by padding I mean, the space between the cell's border and the cell's content. Look at this:

```
<table border="1" width="50%" height="100" cellpadding="5">
<tr>
<td width="200">some text</td>
<td width="400">some text</td>
</tr>
</table>
```

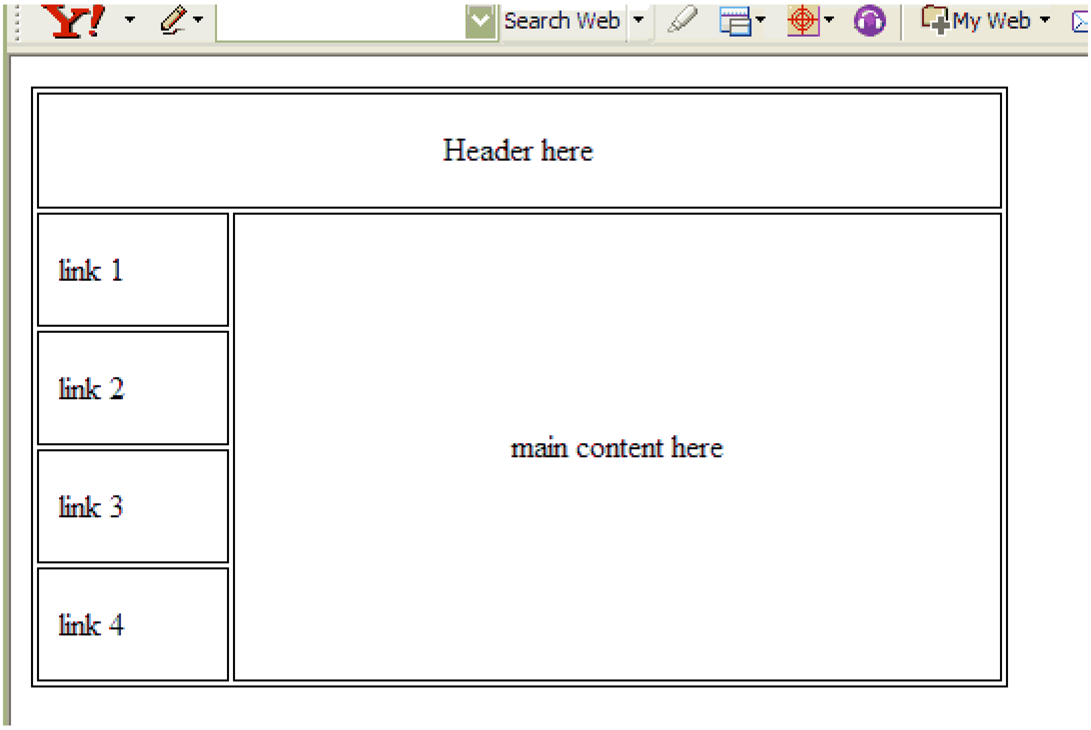


some text

some text

Notice the space between the "some text" and the borders of each cell.

Now try to do this (click the sticky note to see how I did it):



Header here	
link 1	main content here
link 2	
link 3	
link 4	

This will be all about HTML tables. Let's move over to the next lesson: Working with Forms.

Working with Forms

In this lesson, I'm only going to teach you several HTML ***form*** elements for you to manipulate HTML forms on webpages. I am not going to teach you how to process forms in this lesson, but after this lesson, you can download a form mailer from the internet, and manipulate the HTML form elements without problems :)

Form Tag

The ***form*** tag is similar to the ***table*** tag in the sense that `<form> </form>` it tells the browser to expect a form, but the ***form*** tag alone does nothing just like `<table> </table>`. The following tags are needed when working with HTML forms:

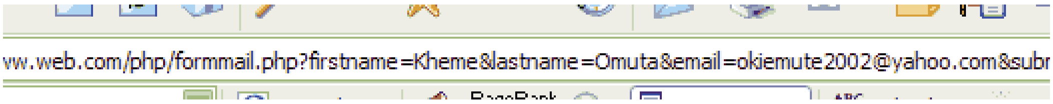
Form Input: Most form fields are placed using the ***input*** tag, but with different values for the *type* attribute. I will explain this over the next few paragraphs. Please note now that the ending tag is forbidden for `<input>`.

Text Input

To place a text form input field just like the one in which you enter your e-mail address on websites, you should set the *type* attribute of the ***input*** tag to 'text'. Try what I have below:

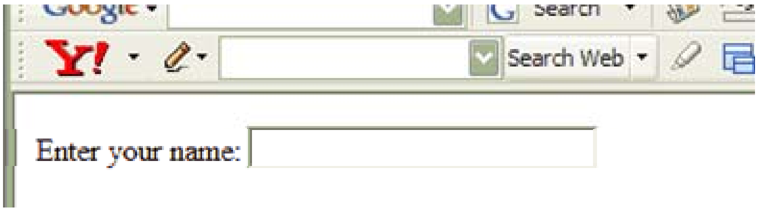
```
<form method="post" action="formmailer.php">
Enter your name:
<input type="text" name="first_name" size="25">
</form>
```

In the HTML code above, I used `<form>` and `<input>`. In `<form>`, the *action* attribute is the location of the form handler to which the form is submitted. A form handler is a script (usually serve-side) that handles the data sent from the form. The *method* attribute sets the "method" used in sending the information from the form to the form handler: using "post" sends the information through the browser in a way that it is not visible to the user, while using "get" sends the information to the form handler as variables in the address bar of the web browser. When "get" is used, after the form is submitted, the address bar may have something like this:



ww.web.com/php/formmail.php?firstname=Kheme&lastname=Omuta&email=okiemute2002@yahoo.com&subr

I've also set the input *type* to "text" and this places a "text" input field. The *name* attribute here is the "name" that the form handler will use to identify the field and its value. The *size* attribute sets the width of the text field.



Enter your name:

Now, to get other form input fields, try changing the *type* attribute in the previous example from “text” to any of the values in the table below:

Attribute Values	**Function**	**Examples**
hidden	places an invisible form field	`<input type="hidden" name="IP">`
button	places a normal clickable button	`<input type="button" name="Submit">`
reset	places a button to reset the form when clicked (test it!)	`<input type="reset" name="Clear Form">`

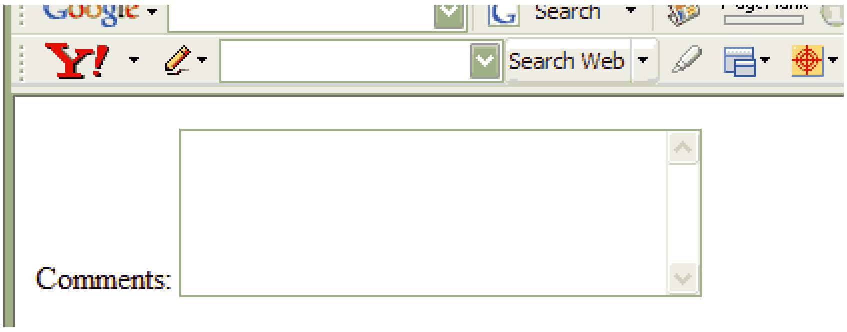
submit	places a button used to submit the form to the form handler	`<input type="submit" name="Submit Form">`
password	places a password field	`<input type="password" name="password">`
file	places a filed for uploading files	`<input type="file" name="your_picture">`
images	places an image as a submit button	`<input type="images" name="Submit" src="images/images_button.gif">`
check box	places a check box	`<input type="checkbox" name="likes">`
radio button	places a radio button	`<input type="radio" name="gender">`

Text Area

To insert a text area like the one you into which you type e-mail messages, you should use the ***textarea*** tag. Use the *cols* attribute of the ***textarea*** tag to set the width of the text area while the *rows* attribute sets the height of the ***textarea***.

```
<form method="post" action="formmailer.php">
Comments: <textarea cols="30" rows="5"></textarea>
</form>
```

Here's what I got:

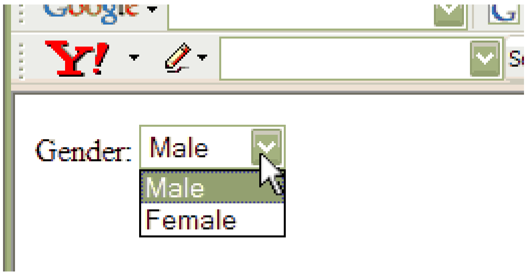


Drop-down Menu

To insert a drop-down menu, we use the ***select*** tag; but `<select> </select>` alone will give us an empty drop-down menu with no options. Now, to add options to this ***select*** (or drop-down menu), you should nest `<option> </options>` within `<select> </select>`. See below:

```
<form method="post" action="formmailer.php">
Gender:
<select name="gender">
<option value="Male">Male</option>
<option value="Female">Female</option>
</select>
</form>
```

In my example, I used the *value* attribute in `<option>` to set the value of that option. This is the value the form handler gets from the selected option. The text in between `<option> </option>` is what the user sees for that option. Now try it; here's what I got:



Hyperlinks

In this lesson, I will show you how to link webpages together to form a website using ***hyperlinks***. I will also show you how to use ***hyperlinks*** to link to other parts of the same page.

Anchor Tag

With the ***anchor*** tag, you can link to other webpages or even a different part of the same page! First, I will show you how to link to other webpages with the ***anchor*** tag using the *href* (**h**ypertext **ref**erence) attribute.

```
<a href="page2.htm">Page 2</a>
```

The *href* attribute sets the location of the page you're linking to. The text placed between <a> </a> is what the user sees and clicks on, to access the page.

Target: The *target* attribute when used with <a> </a> is used to tell the browser where to open the page; whether in the window or a new window, etc. Below is a table with various *target* values and their functions.

Value	**Function**	**Examples**
_blank	opens page in a new blank page	`<a href="page1.htm" target="_blank">Page 1</a>`
_search	opens page in the search window of the browser	`<a href="page1.htm" target="_search">Page 1</a>`

_self	opens page in the same window
	(this is the default value)
		`<a href="page1.htm" target="_self">Page 1</a>`
		or simply
		`<a href="page1.htm">Page 1</a>`

_top	opens page in the topmost window	<a href="page1.htm" target="_top">Page 1</a>

Link to a Different Part of the Same Page

We can also create a hyperlink that links to another part of the same page. I have used this in all of my tutorial web pages; if you've noticed, I always have a link at the bottom of every lesson that takes you back to the top of the page; that's what I'm about to teach you!

To do this, you should use <a> </a> and, the *name* & *href* attributes in different combinations. To eventually achieve this, you should first indicate what part of the page you want the link to pointed to, and you do so by using <a> </a> and the *name* attribute.

```
<a name="top"></a>
```

What I have here is called a "bookmark". Put this at the part of the page where you want the link to point to. Anything can go in between the tag, such as text or image, etc. Next, you create the hyperlink itself, but in your *href*, you shoulder enter a hash sign (#) followed by the "name" of the bookmark. So, from my example above, if I want the link to go to the book mark I just created, then my hyperlink will look like this:

```
<a href="#top">Back to top</a>
```

Email Link

You can also use the anchor tag to create a link so that when it's clicked, it opens up your email page to compose an email to a particular email address.

```
<a href="mailto:okiemute1984@yahoo.com">Email Kheme</a>
```

You can also predefine the subject of the email message.

```
<a href="mailto:kheme@htmlonline.tk?subject=Hi!">email me</a>
```

This will be all for these lessons, and I wish you all the best as you learn HTML and website design and development. But I still have 2 more introductory lessons for you!

Introduction to CSS

CSS or Cascading Style Sheets are used to define styles on webpages, in CSS files or in HTML tags. First of all, I will talk about the various ways you can use CSS on your webpage.

External CSS

You can have a CSS file on its own, outside your HTML file and still use it on your webpage. All you need to do is add the link tag like what I have below:

```
<link href="styles.css" rel="StyleSheet">
```

You should change the href to contain the location of your CSS file and that's all (CSS files usually have the extension .css).

Note: Please remember that `<link>` must be placed within `<head> </head>`.

Internal CSS

You can also have your CSS defined within the webpage, but you'll do that with `<style> </style>`. Below is a typical example of internal CSS. Please observe that `<style> </style>` must always be placed in `<head> </head>`.

```
<style>
body {color:black;font-family:verdana,arial,tahoma}
a:link {text-decoration:none}
a:hover {text-decoration:underline}
</style>
```

Inline CSS

You can also place your CSS inline with HTML tages; all you need to do is add the *style* attribute to the tag like I have below, and define your styles within the *style* attribute.

```
<html>
<head>
<title>You page title here</title>
</head>
<body style="font-family:verdana,arial,tahoma;color:black">
<p align="center" style="text-decoration:underline;font-
weight:bold">
Welcome to my home page!</p>
</body>
</html>
```

Defining Styles

Now, to define styles in your CSS file or within a webpage using `<style> </style>`, you'll follow this format:

```
tagname {style-attribute:attribute value;}
```

What I have here will define the style for all tags named *tagename*. Please note that attributes are separated from their values with a colon (:), while attribute:value pairs are separated by semicolons (;). Below is a typical example (notice the colons and semicolons):

```
body {color:black; font-family:verdana, Arial, tahoma}
```

In the line above, I have defined for the ***body*** tag (<body> </body>) that its text color be black, and that it's font face (or font family) be verdana, and if verdana is not available on the system, then arial should be used, and if arial is also not available on the system, the tahoma should be used.

We can also define styles for only certain tags. In this case, we'll have to give it a name called ‘class’. You can use any name but the only difference it that you have to put a dot (.) before the name when defining it. For example,

```
.name {color:black; font-family:verdana, arial, tahoma}
```

Now, to use this on any tag, you should add the *class* attribute to the tag and set its value as the name of the style class, but without the dot. Doing so applies the style defined for "name" to the tag. Look at my example below:

```
<p class="name">
This text will be black and displayed in "Verdana"
</p>
```

Inline Styles: Defining styles within you HTML tags is similar; enter the attribute:value pairs like I had them in the other examples, as values for the *style* attribute for the tag you want to affect. Look at my example below:

```
<p style="color:black; font-family:verdana, arial, tahoma">
This text will be black and displayed in "Verdana"
</p>
```

On the next page is a table of several style attributes, their functions and some examples.

Attributes Values	**Function**	**Examples**
font-family	sets font face type or family	style="font-family:comic sans ms"
font-size	sets font size	style="font-size:11pt"
font-weight	sets font weight	style="font-weight:bold"
line-height	sets line height or spacing	style="font-height:150%" (1.5 line spacing)
letter-spacing	sets space between letters	style="letter-spacing:10px"
text-align	sets text alignment	style="text-align:center"
color	sets font color	style="color:gold"
background-color	sets background color	style="background-color:#336699"
background-image	sets background images	style="background-image:url(mypic.jpg)"
border	sets all border thickness	style="border:2px" (0px for no border)
border-bottom	sets bottom border thickness	style="border-bottom:1px"
border-top	sets top border thickness	style="border-top:2px"
height	sets object height	style="height:100%"
visibility	sets the visibility of an object	style="visibility:hidden" or style="visibility:visible"
width	sets object width	style="width:250px"

cursor	sets the cursor to display when the mouse is held over object	style="cursor:help" or
		style="cursor:hand"
		or style="cursor:help"

Introduction to JavaScript

JavaScript is a client-side scripting language used to make webpages dynamic and interactive. By "client-side" I mean, it runs only after the webpage has been downloaded to the user's computer via the browser. If a script is not client-side, then it is "server-side", which means the script runs on the server before the webpage is downloaded to the user’s computer via the browser. And just like CSS, there are various ways of using JavaScript on webpages.

External JavaScript

You can have your JavaScript coded on a separate file and still use it on a webpage by using the ***script*** tag. Please note now that `<script> </script>` should be nested within `<head> </head>` in most cases. In other cases, it can be placed anywhere else within the ***body*** tag.

```
<html>
<head>
<title>You page title here</title>
<script src="myscript.js"></script>
</head>
<body>Other stuff you want to add goes here</body>
</html>
```

The *src* (**s**ou**rc**e) attribute contains the location of the JavaScript file (JavaScript files usually have the extension .js).

Internal JavaScript

We can also use `<script> </script>` to insert JavaScript directly into your webpages. Simply do our coding within `<script> </script>` but without the *src* attribute.

```
<html>
<head>
<title>You page title here</title>
<script>
alert("Good Day!");
</script>
</head>
<body>
Other stuff you want to add goes here
</body>
</html>
```

Note: When the *src* attribute is specified, whatever codes you place between `<script> </script>` will be ignored completely!

Inline JavaScript

You can also insert JavaScript directly into HTML tags, but it's usually not advisable. Best way to use JavaScript and CSS on your webpages if they are much, is to write them in an external file, and then link them to the webpage that needs them. This reduces the total size of your webpage, and saves time when your page is being view over a slow internet connection.

```
<a href="#" onclick="alert('Okay!')">Click me!</a>
```

Appendix

HTML Color Table

- Named Colors

aliceblue (#F0F8FF)	antiquewhite (#FAEBD7)	aqua (#00FFFF)	aquamarine (#7FFFD4)	azure (#F0FFFF)
beige (#F5F5DC)	bisque (#FFE4C4)	black (#000000)	blanchedalmond (#FFEBCD)	blue (#0000FF)
blueviolet (#8A2BE2)	brown (#A52A2A)	burlywood (#DEB887)	cadetblue (#5F9EA0)	chartreuse (#7FFF00)
chocolate (#D2691E)	coral (#FF7F50)	cornflower (#6495ED)	cornsilk (#FFF8DC)	crimson (#DC143C)
cyan (#00FFFF)	darkblue (#00008B)	darkcyan (#008B8B)	darkgoldenrod (#B8860B)	darkgray (#A9A9A9)
darkgreen (#006400)	darkkhaki (#BDB76B)	darkmagenta (#8B008B)	darkolivegreen (#556B2F)	darkorange (#FF8C00)
darkorchid (#9932CC)	darkred (#8B0000)	darksalmon (#E9967A)	darkseagreen (#8FBC8B)	darkslateblue (#483D8B)
darkslategray (#2F4F4F)	darkturquoise (#00CED1)	darkviolet (#9400D3)	deeppink (#FF1493)	deepskyblue (#00BFFF)
dimgray (#696969)	dodgerblue (#1E90FF)	firebrick (#B22222)	floralwhite (#FFFAF0)	forestgreen (#228B22)
fuchsia (#FF00FF)	gainsboro (#DCDCDC)	ghostwhite (#F8F8FF)	gold (#FFD700)	goldenrod (#DAA520)
gray (#808080)	green (#008000)	greenyellow (#ADFF2F)	honeydew (#F0FFF0)	hotpink (#FF69B4)
indianred (#CD5C5C)	indigo (#4B0082)	ivory (#FFFFF0)	khaki (#F0E68C)	lavender (#E6E6FA)
lavenderblush (#FFF0F5)	lawngreen (#7CFC00)	lemonchiffon (#FFFACD)	lightblue (#ADD8E6)	lightcoral (#F08080)
lightcyan (#E0FFFF)	lightgoldenrodyellow (#FAFAD2)	lightgreen (#90EE90)	lightgray (#D3D3D3)	lightpink (#FFB6C1)
lightsalmon (#FFA07A)	lightseagreen (#20B2AA)	lightskyblue (#87CEFA)	lightslategray (#778899)	lightsteelblue (#B0C4DE)

lightyellow (#FFFFE0)	lime (#00FF00)	limegreen (#32CD32)	linen (#FAF0E6)	magenta (#FF00FF)
maroon (#800000)	mediumaquamarine (#66CDAA)	mediumblue (#0000CD)	mediumorchid (#BA55D3)	mediumpurple (#9370DB)
mediumseagreen (#3CB371)	mediumslateblue (#7B68EE)	mediumspringgreen (#00FA9A)	mediumturquoise (#48D1CC)	mediumvioletred (#C71585)
midnightblue (#191970)	mintcream (#F5FFFA)	mistyrose (#FFE4E1)	moccasin (#FFE4B5)	navajowhite (#FFDEAD)
navy (#000080)	oldlace (#FDF5E6)	olive (#808000)	olivedrab (#6B8E23)	orange (#FFA500)
orangered (#FF4500)	orchid (#DA70D6)	palegoldenrod (#EEE8AA)	palegreen (#98FB98)	paleturquoise (#AFEEEE)
palevioletred (#DB7093)	papayawhip (#FFEFD5)	peachpuff (#FFDAB9)	peru (#CD853F)	pink (#FFC0CB)
plum (#DDA0DD)	powderblue (#B0E0E6)	purple (#800080)	red (#FF0000)	rosybrown (#BC8F8F)
royalblue (#4169E1)	saddlebrown (#8B4513)	salmon (#FA8072)	sandybrown (#F4A460)	seagreen (#2E8B57)
seashell (#FFF5EE)	sienna (#A0522D)	silver (#C0C0C0)	skyblue (#87CEEB)	slateblue (#6A5ACD)
slategray (#708090)	snow (#FFFAFA)	springgreen (#00FF7F)	steelblue (#4682B4)	tan (#D2B48C)
teal (#008080)	thistle (#D8BFD8)	tomato (#FF6347)	turquoise (#40E0D0)	violet (#EE82EE)
wheat (#F5DEB3)	white (#FFFFFF)	whitesmoke (#F5F5F5)	yellow (#FFFF00)	yellowgreen (#9ACD32)

- Web Safe Colors

#000000	#000033	#000066	#000099	#0000CC	#0000FF
#003300	#003333	#003366	#003399	#0033CC	#0033FF
#006600	#006633	#006666	#006699	#0066CC	#0066FF
#009900	#009933	#009966	#009999	#0099CC	#0099FF
#00CC00	#00CC33	#00CC66	#00CC99	#00CCCC	#00CCFF
#00FF00	#00FF33	#00FF66	#00FF99	#00FFCC	#00FFFF
#330000	#330033	#330066	#330099	#3300CC	#3300FF
#333300	#333333	#333366	#333399	#3333CC	#3333FF
#336600	#336633	#336666	#336699	#3366CC	#3366FF
#339900	#339933	#339966	#339999	#3399CC	#3399FF
#33CC00	#33CC33	#33CC66	#33CC99	#33CCCC	#33CCFF
#33FF00	#33FF33	#33FF66	#33FF99	#33FFCC	#33FFFF
#660000	#660033	#660066	#660099	#6600CC	#6600FF
#663300	#663333	#663366	#663399	#6633CC	#6633FF
#666600	#666633	#666666	#666699	#6666CC	#6666FF
#669900	#669933	#669966	#669999	#6699CC	#6699FF
#66CC00	#66CC33	#66CC66	#66CC99	#66CCCC	#66CCFF
#66FF00	#66FF33	#66FF66	#66FF99	#66FFCC	#66FFFF
#990000	#990033	#990066	#990099	#9900CC	#9900FF
#993300	#993333	#993366	#993399	#9933CC	#9933FF
#996600	#996633	#996666	#996699	#9966CC	#9966FF
#999900	#999933	#999966	#999999	#9999CC	#9999FF
#99CC00	#99CC33	#99CC66	#99CC99	#99CCCC	#99CCFF
#99FF00	#99FF33	#99FF66	#99FF99	#99FFCC	#99FFFF
#CC0000	#CC0033	#CC0066	#CC0099	#CC00CC	#CC00FF
#CC3300	#CC3333	#CC3366	#CC3399	#CC33CC	#CC33FF

#CC6600	#CC6633	#CC6666	#CC6699	#CC66CC	#CC66FF
#CC9900	#CC9933	#CC9966	#CC9999	#CC99CC	#CC99FF
#CCCC00	#CCCC33	#CCCC66	#CCCC99	#CCCCCC	#CCCCFF
#CCFF00	#CCFF33	#CCFF66	#CCFF99	#CCFFCC	#CCFFFF
#FF0000	#FF0033	#FF0066	#FF0099	#FF00CC	#FF00FF
#FF3300	#FF3333	#FF3366	#FF3399	#FF33CC	#FF33FF
#FF6600	#FF6633	#FF6666	#FF6699	#FF66CC	#FF66FF
#FF9900	#FF9933	#FF9966	#FF9999	#FF99CC	#FF99FF
#FFCC00	#FFCC33	#FFCC66	#FFCC99	#FFCCCC	#FFCCFF
#FFFF00	#FFFF33	#FFFF66	#FFFF99	#FFFFCC	#FFFFFF

ASCII Character List

The 1st 1000 ASCII character list generated from:
www.htmlonline.tk/tools/ascii.php

Character	ASCII Form
	&#1
	&#2
	&#3
	&#4
	&#5
	&#6
	&#7
	&#8
	&#9
	&#10
	&#11
	&#12
	&#13
	&#14
□	&#15
□	&#16
□	&#17
□	&#18
□	&#19
□	&#20
□	&#21
□	&#22
□	&#23
□	&#24
□	&#25
□	&#26
□	&#27
□	&#28
□	&#29
-	&#30

Character	ASCII Form
ŏ	&#335
Ő	&#336
ő	&#337
Œ	&#338
œ	&#339
Ŕ	&#340
ŕ	&#341
Ŗ	&#342
ŗ	&#343
Ř	&#344
ř	&#345
Ś	&#346
ś	&#347
Ŝ	&#348
ŝ	&#349
Ş	&#350
ş	&#351
Š	&#352
š	&#353
Ţ	&#354
ţ	&#355
Ť	&#356
ť	&#357
Ŧ	&#358
ŧ	&#359
Ũ	&#360
ũ	&#361
Ū	&#362
ū	&#363
Ŭ	&#364

Character	ASCII Form
□	&#669
□	&#670
□	&#671
□	&#672
□	&#673
□	&#674
□	&#675
□	&#676
□	&#677
□	&#678
□	&#679
□	&#680
□	&#681
□	&#682
□	&#683
□	&#684
□	&#685
□	&#686
□	&#687
□	&#688
□	&#689
□	&#690
□	&#691
□	&#692
□	&#693
□	&#694
□	&#695
□	&#696
□	&#697
□	&#698

	&#31
	&#32
!	&#33
"	&#34
#	&#35
$	&#36
%	&#37
&	&#38
'	&#39
(	&#40
)	&#41
*	&#42
+	&#43
,	&#44
-	&#45
.	&#46
/	&#47
0	&#48
1	&#49
2	&#50
3	&#51
4	&#52
5	&#53
6	&#54
7	&#55
8	&#56
9	&#57
:	&#58
;	&#59
<	&#60
=	&#61
>	&#62
?	&#63
@	&#64
A	&#65

ŭ	&#365
Ů	&#366
ů	&#367
Ű	&#368
ű	&#369
Ų	&#370
ų	&#371
Ŵ	&#372
ŵ	&#373
Ŷ	&#374
ŷ	&#375
Ÿ	&#376
Ź	&#377
ź	&#378
Ż	&#379
ż	&#380
Ž	&#381
ž	&#382
ſ	&#383
□	&#384
□	&#385
□	&#386
□	&#387
□	&#388
□	&#389
□	&#390
□	&#391
□	&#392
□	&#393
□	&#394
□	&#395
□	&#396
□	&#397
□	&#398
□	&#399

□	&#699
□	&#700
□	&#701
□	&#702
□	&#703
□	&#704
□	&#705
□	&#706
□	&#707
□	&#708
□	&#709
ˆ	&#710
ˇ	&#711
□	&#712
ˉ	&#713
□	&#714
□	&#715
□	&#716
□	&#717
□	&#718
□	&#719
□	&#720
□	&#721
□	&#722
□	&#723
□	&#724
□	&#725
□	&#726
□	&#727
˘	&#728
˙	&#729
˚	&#730
˛	&#731
˜	&#732
˝	&#733

B	&#66
C	&#67
D	&#68
E	&#69
F	&#70
G	&#71
H	&#72
I	&#73
J	&#74
K	&#75
L	&#76
M	&#77
N	&#78
O	&#79
P	&#80
Q	&#81
R	&#82
S	&#83
T	&#84
U	&#85
V	&#86
W	&#87
X	&#88
Y	&#89
Z	&#90
[	&#91
\	&#92
]	&#93
^	&#94
_	&#95
`	&#96
a	&#97
b	&#98
c	&#99
d	&#100

□	&#400
□	&#401
ƒ	&#402
□	&#403
□	&#404
□	&#405
□	&#406
□	&#407
□	&#408
□	&#409
□	&#410
□	&#411
□	&#412
□	&#413
□	&#414
□	&#415
Ơ	&#416
ơ	&#417
□	&#418
□	&#419
□	&#420
□	&#421
□	&#422
□	&#423
□	&#424
□	&#425
□	&#426
□	&#427
□	&#428
□	&#429
□	&#430
Ư	&#431
ư	&#432
□	&#433
□	&#434

□	&#734
□	&#735
□	&#736
□	&#737
□	&#738
□	&#739
□	&#740
□	&#741
□	&#742
□	&#743
□	&#744
□	&#745
□	&#746
□	&#747
□	&#748
□	&#749
□	&#750
□	&#751
□	&#752
□	&#753
□	&#754
□	&#755
□	&#756
□	&#757
□	&#758
□	&#759
□	&#760
□	&#761
□	&#762
□	&#763
□	&#764
□	&#765
□	&#766
□	&#767
`	&#768

e	&#101
f	&#102
g	&#103
h	&#104
i	&#105
j	&#106
k	&#107
l	&#108
m	&#109
n	&#110
o	&#111
p	&#112
q	&#113
r	&#114
s	&#115
t	&#116
u	&#117
v	&#118
w	&#119
x	&#120
y	&#121
z	&#122
{	&#123
|	&#124
}	&#125
~	&#126
□	&#127
€	&#128
�	&#129
‚	&#130
ƒ	&#131
„	&#132
…	&#133
†	&#134
‡	&#135

□	&#435
□	&#436
□	&#437
□	&#438
□	&#439
□	&#440
□	&#441
□	&#442
□	&#443
□	&#444
□	&#445
□	&#446
□	&#447
□	&#448
□	&#449
□	&#450
□	&#451
□	&#452
□	&#453
□	&#454
□	&#455
□	&#456
□	&#457
□	&#458
□	&#459
□	&#460
□	&#461
□	&#462
□	&#463
□	&#464
□	&#465
□	&#466
□	&#467
□	&#468
□	&#469

́	&#769
□	&#770
̃	&#771
□	&#772
□	&#773
□	&#774
□	&#775
□	&#776
̉	&#777
□	&#778
□	&#779
□	&#780
□	&#781
□	&#782
□	&#783
□	&#784
□	&#785
□	&#786
□	&#787
□	&#788
□	&#789
□	&#790
□	&#791
□	&#792
□	&#793
□	&#794
□	&#795
□	&#796
□	&#797
□	&#798
□	&#799
□	&#800
□	&#801
□	&#802
̣	&#803

ˆ	&#136
‰	&#137
Š	&#138
‹	&#139
Œ	&#140
□	&#141
Ž	&#142
□	&#143
□	&#144
‘	&#145
’	&#146
“	&#147
”	&#148
•	&#149
–	&#150
—	&#151
˜	&#152
™	&#153
š	&#154
›	&#155
œ	&#156
□	&#157
ž	&#158
Ÿ	&#159
	&#160
¡	&#161
¢	&#162
£	&#163
¤	&#164
¥	&#165
¦	&#166
§	&#167
¨	&#168
©	&#169
ª	&#170

□	&#470
□	&#471
□	&#472
□	&#473
□	&#474
□	&#475
□	&#476
□	&#477
□	&#478
□	&#479
□	&#480
□	&#481
□	&#482
□	&#483
□	&#484
□	&#485
□	&#486
□	&#487
□	&#488
□	&#489
□	&#490
□	&#491
□	&#492
□	&#493
□	&#494
□	&#495
□	&#496
□	&#497
□	&#498
□	&#499
□	&#500
□	&#501
□	&#502
□	&#503
□	&#504

□	&#804
□	&#805
□	&#806
□	&#807
□	&#808
□	&#809
□	&#810
□	&#811
□	&#812
□	&#813
□	&#814
□	&#815
□	&#816
□	&#817
□	&#818
□	&#819
□	&#820
□	&#821
□	&#822
□	&#823
□	&#824
□	&#825
□	&#826
□	&#827
□	&#828
□	&#829
□	&#830
□	&#831
□	&#832
□	&#833
□	&#834
□	&#835
□	&#836
□	&#837
□	&#838

«	&#171
¬	&#172
	&#173
®	&#174
¯	&#175
°	&#176
±	&#177
²	&#178
³	&#179
´	&#180
µ	&#181
¶	&#182
·	&#183
¸	&#184
¹	&#185
º	&#186
»	&#187
¼	&#188
½	&#189
¾	&#190
¿	&#191
À	&#192
Á	&#193
Â	&#194
Ã	&#195
Ä	&#196
Å	&#197
Æ	&#198
Ç	&#199
È	&#200
É	&#201
Ê	&#202
Ë	&#203
Ì	&#204
Í	&#205

□	&#505
Ǻ	&#506
ǻ	&#507
Ǽ	&#508
ǽ	&#509
Ǿ	&#510
ǿ	&#511
□	&#512
□	&#513
□	&#514
□	&#515
□	&#516
□	&#517
□	&#518
□	&#519
□	&#520
□	&#521
□	&#522
□	&#523
□	&#524
□	&#525
□	&#526
□	&#527
□	&#528
□	&#529
□	&#530
□	&#531
□	&#532
□	&#533
□	&#534
□	&#535
□	&#536
□	&#537
□	&#538
□	&#539

□	&#839
□	&#840
□	&#841
□	&#842
□	&#843
□	&#844
□	&#845
□	&#846
□	&#847
□	&#848
□	&#849
□	&#850
□	&#851
□	&#852
□	&#853
□	&#854
□	&#855
□	&#856
□	&#857
□	&#858
□	&#859
□	&#860
□	&#861
□	&#862
□	&#863
□	&#864
□	&#865
□	&#866
□	&#867
□	&#868
□	&#869
□	&#870
□	&#871
□	&#872
□	&#873

Î	&#206
Ï	&#207
Ð	&#208
Ñ	&#209
Ò	&#210
Ó	&#211
Ô	&#212
Õ	&#213
Ö	&#214
×	&#215
Ø	&#216
Ù	&#217
Ú	&#218
Û	&#219
Ü	&#220
Ý	&#221
Þ	&#222
ß	&#223
à	&#224
á	&#225
â	&#226
ã	&#227
ä	&#228
å	&#229
æ	&#230
ç	&#231
è	&#232
é	&#233
ê	&#234
ë	&#235
ì	&#236
í	&#237
î	&#238
ï	&#239
ð	&#240

□	&#540
□	&#541
□	&#542
□	&#543
□	&#544
□	&#545
□	&#546
□	&#547
□	&#548
□	&#549
□	&#550
□	&#551
□	&#552
□	&#553
□	&#554
□	&#555
□	&#556
□	&#557
□	&#558
□	&#559
□	&#560
□	&#561
□	&#562
□	&#563
□	&#564
□	&#565
□	&#566
□	&#567
□	&#568
□	&#569
□	&#570
□	&#571
□	&#572
□	&#573
□	&#574

□	&#874
□	&#875
□	&#876
□	&#877
□	&#878
□	&#879
□	&#880
□	&#881
□	&#882
□	&#883
□	&#884
□	&#885
□	&#886
□	&#887
□	&#888
□	&#889
□	&#890
□	&#891
□	&#892
□	&#893
;	&#894
□	&#895
□	&#896
□	&#897
□	&#898
□	&#899
΄	&#900
΅	&#901
Ά	&#902
·	&#903
Έ	&#904
Ή	&#905
Ί	&#906
□	&#907
Ό	&#908

ñ	&#241
ò	&#242
ó	&#243
ô	&#244
õ	&#245
ö	&#246
÷	&#247
ø	&#248
ù	&#249
ú	&#250
û	&#251
ü	&#252
ý	&#253
þ	&#254
ÿ	&#255
Ā	&#256
ā	&#257
Ă	&#258
ă	&#259
Ą	&#260
ą	&#261
Ć	&#262
ć	&#263
Ĉ	&#264
ĉ	&#265
Ċ	&#266
ċ	&#267
Č	&#268
č	&#269
Ď	&#270
ď	&#271
Đ	&#272
đ	&#273
Ē	&#274
ē	&#275

□	&#575
□	&#576
□	&#577
□	&#578
□	&#579
□	&#580
□	&#581
□	&#582
□	&#583
□	&#584
□	&#585
□	&#586
□	&#587
□	&#588
□	&#589
□	&#590
□	&#591
□	&#592
□	&#593
□	&#594
□	&#595
□	&#596
□	&#597
□	&#598
□	&#599
□	&#600
□	&#601
□	&#602
□	&#603
□	&#604
□	&#605
□	&#606
□	&#607
□	&#608
□	&#609

□	&#909
Ύ	&#910
Ώ	&#911
ΐ	&#912
Α	&#913
Β	&#914
Γ	&#915
Δ	&#916
Ε	&#917
Ζ	&#918
Η	&#919
Θ	&#920
Ι	&#921
Κ	&#922
Λ	&#923
Μ	&#924
Ν	&#925
Ξ	&#926
Ο	&#927
Π	&#928
Ρ	&#929
□	&#930
Σ	&#931
Τ	&#932
Υ	&#933
Φ	&#934
Χ	&#935
Ψ	&#936
Ω	&#937
Ϊ	&#938
Ϋ	&#939
ά	&#940
έ	&#941
ή	&#942
ί	&#943

Ĕ	&#276
ĕ	&#277
Ė	&#278
ė	&#279
Ę	&#280
ę	&#281
Ě	&#282
ě	&#283
Ĝ	&#284
ĝ	&#285
Ğ	&#286
ğ	&#287
Ġ	&#288
ġ	&#289
Ģ	&#290
ģ	&#291
Ĥ	&#292
ĥ	&#293
Ħ	&#294
ħ	&#295
Ĩ	&#296
ĩ	&#297
Ī	&#298
ī	&#299
Ĭ	&#300
ĭ	&#301
Į	&#302
į	&#303
İ	&#304
ı	&#305
Ĳ	&#306
ĳ	&#307
Ĵ	&#308
ĵ	&#309
Ķ	&#310

□	&#610
□	&#611
□	&#612
□	&#613
□	&#614
□	&#615
□	&#616
□	&#617
□	&#618
□	&#619
□	&#620
□	&#621
□	&#622
□	&#623
□	&#624
□	&#625
□	&#626
□	&#627
□	&#628
□	&#629
□	&#630
□	&#631
□	&#632
□	&#633
□	&#634
□	&#635
□	&#636
□	&#637
□	&#638
□	&#639
□	&#640
□	&#641
□	&#642
□	&#643
□	&#644

ΰ	&#944
α	&#945
β	&#946
γ	&#947
δ	&#948
ε	&#949
ζ	&#950
η	&#951
θ	&#952
ι	&#953
κ	&#954
λ	&#955
μ	&#956
ν	&#957
ξ	&#958
ο	&#959
π	&#960
ρ	&#961
ς	&#962
σ	&#963
τ	&#964
υ	&#965
φ	&#966
χ	&#967
ψ	&#968
ω	&#969
ϊ	&#970
ϋ	&#971
ό	&#972
ύ	&#973
ώ	&#974
□	&#975
□	&#976
□	&#977
□	&#978

ķ	&#311
ĸ	&#312
Ĺ	&#313
ĺ	&#314
Ļ	&#315
ļ	&#316
Ľ	&#317
ľ	&#318
Ŀ	&#319
ŀ	&#320
Ł	&#321
ł	&#322
Ń	&#323
ń	&#324
Ņ	&#325
ņ	&#326
Ň	&#327
ň	&#328
ŉ	&#329
Ŋ	&#330
ŋ	&#331
Ō	&#332
ō	&#333
Ŏ	&#334

□	&#645
□	&#646
□	&#647
□	&#648
□	&#649
□	&#650
□	&#651
□	&#652
□	&#653
□	&#654
□	&#655
□	&#656
□	&#657
□	&#658
□	&#659
□	&#660
□	&#661
□	&#662
□	&#663
□	&#664
□	&#665
□	&#666
□	&#667
□	&#668

□	&#979
□	&#980
□	&#981
□	&#982
□	&#983
□	&#984
□	&#985
□	&#986
□	&#987
□	&#988
□	&#989
□	&#990
□	&#991
□	&#992
□	&#993
□	&#994
□	&#995
□	&#996
□	&#997
□	&#998
□	&#999
□	&#1000
□	&#1001
□	&#1002