

MICHAEL GABRIEL

BEGINNER'S GUIDE TO HTML

Beginner's Guide to HTML

by Michael Gabriel

1. Getting Started: What You Need to Do to Get Going and Make Your First HTML Page

- 1.1 What is HTML?
- 1.2 Why learn HTML?
- 1.3 What you need to know about HTML to get started

2. Tags, Attributes and Elements

- 2.1 What's the difference?
- 2.2 HTML elements
- 2.3 Tips
- 2.4 Attributes

3. Titles. Headings. Paragraphs.

- 3.1 The title tag
- 3.2 Headings
- 3.3 Paragraphs

4. Unnumbered, Numbered, Definition and Nested Lists

- 4.1 Ordered lists
- 4.2 Unordered lists
- 4.3 Description lists
- 4.4 Nested lists

5. How to Make Links to Other Pages and Elsewhere

- 5.1 Regular or text links
- 5.2 Image links
- 5.3 Links to e-mail addresses
- 5.4 Anchor links
- 5.5 Other considerations when creating links

6. How to Work with Images

- 6.1 The alt attribute
- 6.2 The width and height attributes
- 6.3 Align your images
- 6.4 Adding links to your images

7. How to Use Tabular Data

- 7.1 When to use tables and when not to
- 7.2 Creating tables
- 7.3 More on the border attribute
- 7.4 Captioning a table
- 7.5 Working with cells that take up two rows or two columns
- 7.6 Making data easier to read
- 7.7 HTML tables are very flexible

8. Text Boxes and Other User-Input "Thingamajigs"

- 8.1 Form fields
- 8.2 Text field
- 8.3 Password
- 8.4 Checkboxes
- 8.5 Radio buttons
- 8.6 Submit
- 8.7 Reset button
- 8.8 Drop-down menus
- 8.9 Text area

9. HTML for Beginners: CSS

- 9.1 Benefits of using CSS
- 9.2 A brief history of CSS
- 9.3 How to use CSS with your HTML document
- 9.4 What you should know about cascading stylesheets?

10. Putting It All Together

- 10.1 Start by making simple web pages
- 10.2 Why HTML should be easy
- 10.3 Simplifying HTML and CSS
- 10.4 Other things you need to know when learning HTML
- 10.5 Learning resources

1. Getting Started: What You Need to Do to Get Going and Make Your First HTML Page

If you have always wanted to make your own Web page, but you were under the impression that it would be very difficult, then there should really be no problem. To create your own Web page, all you need to have is a thorough understanding of HTML. And HTML is not all that difficult to learn.



1.1 What is HTML?

HTML is actually shorthand for Hypertext Markup Language. It is the language of Web pages that tells a browser how to display certain elements, such as text and images through the use of codes and symbols.

HTML is the standard when it comes to creating Web pages. The World Wide Web Consortium, or W3C recommends it. Being such, most browsers implement HTML to help display Web pages more or less uniformly.



HTML is the brainchild of Tim Berners-Lee. In 1990, Berners-Lee needed something that would help scientists coming from different colleges and universities access documents and research from other scientists. That problem led to Berners-Lee inventing the World Wide Web, the hypertext transfer protocol or HTTP, and HTML.

1.2 Why learn HTML?

These days, the proliferation of content management systems such as WordPress as well as HTML editors can help you create a Web page without knowing HTML.

But even with these tools, knowing how HTML works plus knowing even just the basic tags can go a long way.

For example, it would make your life easier if you knew how to work with HTML to correct design flaws. Like if your WordPress installation is using a CSS rule that displays images wrong, you can view the source and correct how your images are

being displayed if you know HTML.

Also, HTML is very simple and easy to learn. It literally makes no sense why you should not take time to learn it!

1.3 What you need to know about HTML to get started

This is another good news: when you are learning HTML, you do not need much.

You can start with using Notepad, a text editor that is included in your Windows installation. If you are not using Windows, or if you prefer other text editors, there is a lot of free software that you can download from the Internet.

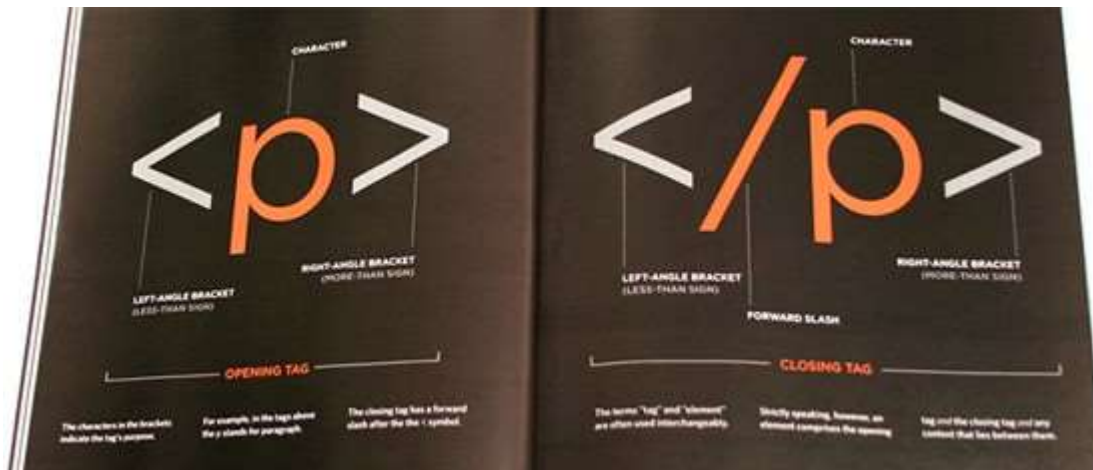
Or you could get one of those HTML or WYSIWYG editors. WYSIWYG stands for "what you see is what you get." Think of it as a layout tool that allows you to easily create an HTML page and format it without bothering too much about tags and elements.

So we do recommend hard-coding if you are still learning. Hard-coding means to write HTML using a text editor. There are two things that you need to know to get started with HTML:

- 1. The basic tags and elements.*
- 2. The structure of an HTML page.*

1. Basic tags and elements

HTML is a markup language. As such, you will need to know the various tags and elements that it uses.



Tags usually come in pairs. An opening tag will signify that the browser should treat the succeeding text using that tag's properties and a closing tag would end it.

For example, the opening `` tag and its corresponding closing `` tag will render all the text in between in bold. i.e.,

"My name is Clark Kent" will be displayed as:

My name is **Clark Kent**.

And then there are what we call the "empty elements" or those that work without a closing or opening tag. Empty elements are often described as self-closing tags.

There are quite a few empty elements such as:

- area
- base
- basefont
- br
- col
- frame
- hr
- img

- `input`
- `isindex`
- `link`
- `meta`
- `param`

All of these can stand alone. For example, line breaks in HTML documents are often denoted by `
`.

2. Structure of an HTML page

HTML is very simple and it is logical. A browser would start reading an HTML page from the top going down, from left going right.

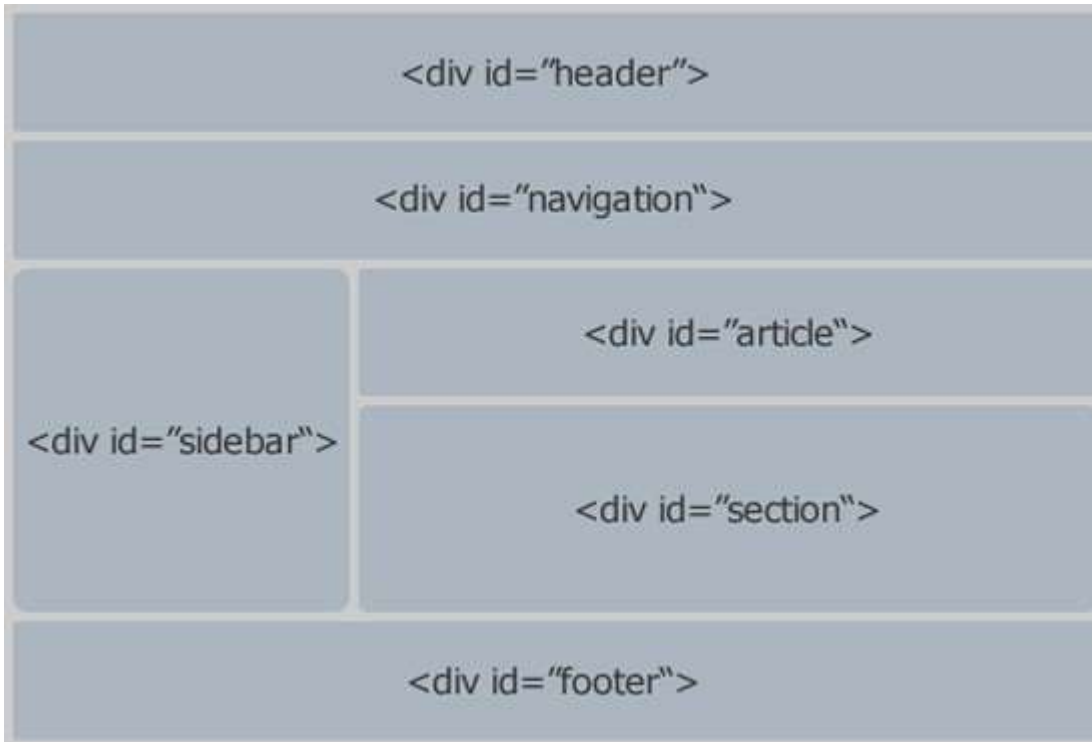
It does follow a basic structure. First, you have to declare that the document is an HTML document. You can do this by using the `<html>` tag.

Then there are the two other sections called the `<head>` and `<body>` of your document.

The `<head>` is where you put in all the information you want to include about your document.

The `<body>` is basically the content of your HTML document.

Put in another way, the `<body>` section is what people see when they view your HTML document.



Remember how you need to close a tag in HTML? You would also need to close `<html>`, `<head>` and `<body>`, so that the basic HTML document would look like this:

```
<html>  
  <head>  
  </head>  
  <body>  
  </body>  
</html>
```

Now you have the foundations of creating your first HTML page!

2. Tags, Attributes and Elements

When learning about HTML, you will come across different resources that may use HTML elements and tags interchangeably. Chances are, that's because they wanted to

simplify everything and make it easier for you to understand. However, there is a difference when you refer to HTML elements and HTML tags.

2.1 What's the difference?

Technically, HTML documents contain only tags. When they are accessed on a browser, HTML documents are then parsed so that these could be displayed using the Document Object Model (DOM). The tags are there in the HTML document, but HTML elements only appear after the document has been parsed.

But let's not get into the technical stuff and instead simplify our lives by using what is commonly understood. To put it simply, HTML tags are the markup language that you use in HTML. These are generally the start or opening tag and the closing or ending tag. Tags are enclosed in angle brackets `< >`.

HTML elements, on the other hand, include the content.

So, given this: `<p>This is the difference between HTML elements and tags. </p>`. The tags here are `<p>` and `</p>` , while the whole thing is called an HTML element.

2.2 HTML Elements

Now that we have that out of the way, let's take a look at HTML elements.

An HTML element has the following syntax:

1. An HTML element begins with an opening or start tag, i.e., `<p>`.
2. An HTML element ends with a closing or end tag, i.e. `</p>`.
3. The property of an HTML content is enclosed within the start and end tags, i.e., `<p>This is the element content. </p>`.
4. It may also contain an attribute.

1. Empty Elements

Some HTML elements do have content and are simply called empty elements. These

elements are ended or closed with the opening tag, i.e, `
`.

2. Nestled Elements

Then there is also what is called the "nestled elements." Nestled elements are HTML elements that are found within other HTML elements. For example, this is a nestled element:

`<p>I do not know what it was.</p>`

Lastly, an HTML element may contain attributes, which we will discuss later on.

3. Some Commonly Used HTML Tags Include:

`<a>` - which shows a hyperlink

``- which displays boldfaced text

`
` - which enters a line break into paragraphs

`<div>`- which introduces a section of the document

``- which shows emphasis

`<h1>`, `<h2>`, `<h3>`, through `<h6>`- display HTML headings

`<hr>`- which inserts a horizontal rule

`<p>`- which introduces a paragraph

`<table>`- which introduces a table

4. Uses of HTML elements

HTML elements serve a variety of purposes:

1. HTML elements can describe the purpose of the text, as well as give it structure. A good example of this type of HTML elements is `<h1>`, `<h2>` and other HTML headers. For instance, `<h1>Introduction</h1>` communicates that "Introduction" is a first level heading. Take note that structural HTML elements only provide structure and do not have specific standards for display. A browser, however, can have default display for this type of elements, hence you often see header tags being bigger than normal text on your browser.

2. HTML elements can dictate how certain content would appear. For instance, using `` would render the text in boldface, while using `<i>` will display the text in italics.

3. HTML elements that link your HTML document to another documents. For instance, you can use `<a>` to link your document to another document.

2.3 Tips

1. Always put in the closing tag.

There are HTML elements that are not empty elements but may be written without a closing tag. Because browsers are able to recognize them, they present no problem. For example, the `<p>` element, which starts a paragraph, is often not closed.

`<p>`*This is what we are talking about.*

However, it is still best practice to close HTML elements.

Most Web browsers will consider the closing tag for `<p>` optional and the paragraph will display properly. Yet, this would not be the case for all Web browsers. To be sure, always put the closing tag at the end.

How is this case different from empty elements? An empty element is when no closing tags are required. Additionally, no closing tags exist or are allowed in empty elements. For example, `
` does not have a corresponding closing tag: `</br>` does not exist in HTML.

Whereas, technically, you would need to close the HTML element above by using `</p>`.

2. Use lowercase when writing elements.

HTML tags are not case sensitive. This means that you can write `<p>`, `<P>` or even `</P>` or `</p>`.

However, standards and recommendations from the World Wide Web Consortium or W3C recommend using lowercase for HTML elements in HTML4 and requires it in XHTML. So you might as well start training yourself in writing HTML elements in lowercase.

2.4 Attributes

Another term you would need to know when it comes to HTML elements are attributes. Attributes modify the tags where they appear. Attributes are name-value pairs, which are separated by the equal sign.

The HTML element `Next of kin` contains the attribute `href="nextofkin.html"`. Attributes usually use the following syntax:

`<tag attribute="value of attribute">content</tag>`

A lot of people confuse the alt attribute as a tag. It is not. The alt attribute modifies the ``, `<area>`, `<input>` and `<applet>` tags, which makes it an attribute.

Some of the most commonly used attributes, aside from href and alt include:

- align
- bgcolor
- src
- height
- width
- value
- href
- hspace
- id
- class

- style
- title
- dir
- lang

These are the basics you need to know about tags, attributes and elements.

3. Titles. Headings. Paragraphs

One of the things that you should realize when working with HTML is that it is not just a markup language that you need to use in order to create a Web page. HTML is highly readable and understandable, even though you are just looking at its code. For the most part, you would see how everything is structured just by looking at its code.

So looking at the source code, you would be able to look at an HTML document and understand how a page would look like.

With that in mind, you should be able to work pretty well with titles, paragraphs and headings.

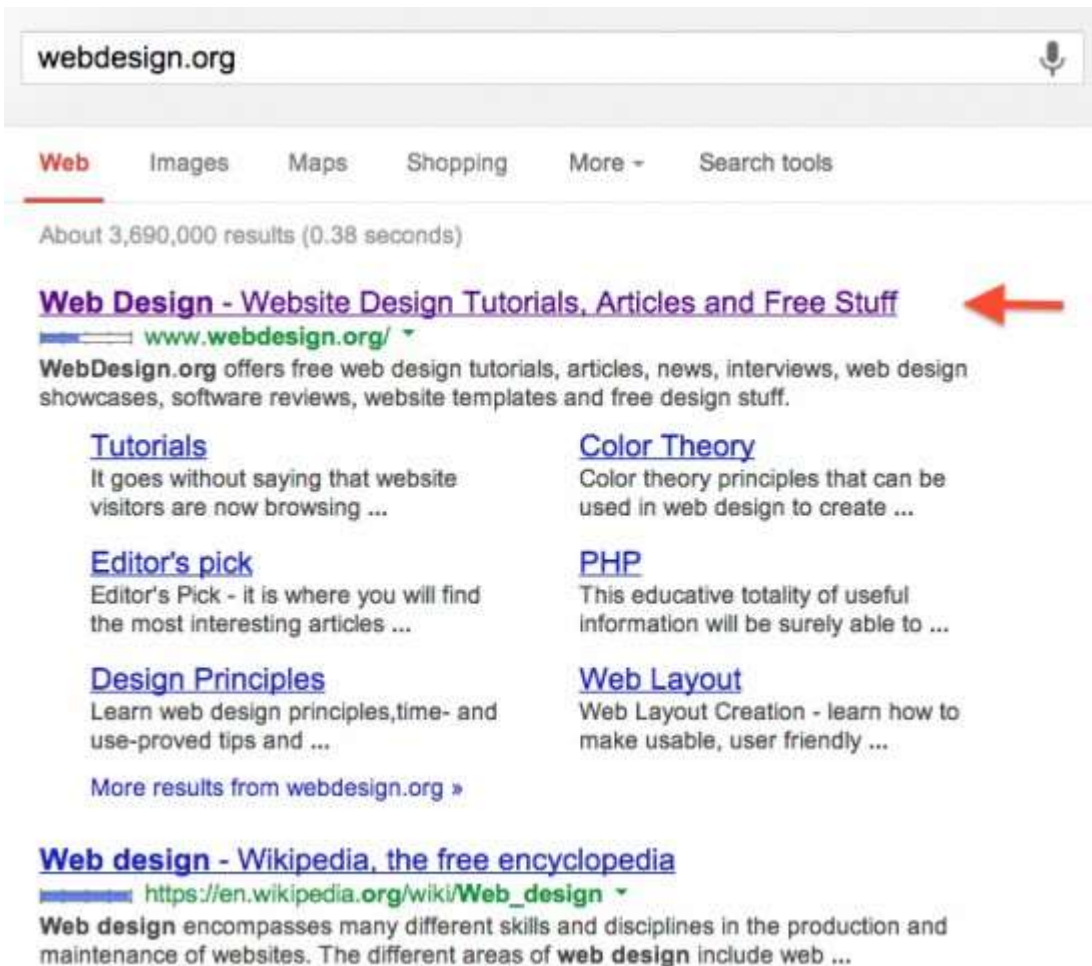


3.1 The Title Tag

As with any book, article or any printed works, your HTML document should have a title.

The title is required for all HTML documents and it comes with other benefits:

- It will identify what your document is all about.
- Titles also show up on the browser window or tab, making it easy for people to identify which tab to click on to access your Web page if they have a lot of tabs open at the same time.
- It is one of the things that search engines look at in order to determine the content of your page. Search engine result pages (SERPs) will display the title of your document as the result's main heading.



Ultimately, it will help you rank higher on search engines if you write your titles right.

- It can be used by other systems such as WAIS searches and document archiving systems to identify your document.
- Title tags are also used for bookmarks. So if you are in the habit of bookmarking pages, title tags would provide a good way to identify the pages in the future.

The correct way to write your title is to use the following syntax:

`<title> Write Your Document's Title Here. </title>`

You should place your title element within the the `<head>` section of your HTML document.

When you have the `<title>` in place, you can now start writing the content that is visible

to humans. It's the content that you put within the `<body>` section of your HTML document.

Let us start with headings and paragraphs.

3.2 Headings

Headings are important to use in your document. They give it some structure. HTML allows you to have six different headings: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` and `<h6>`.

Headings follow the following syntax:

```
<h1>The texts for Heading 1 go here</h1>  
<h2>The texts for Heading 2 go here</h2>  
<h3>The texts for Heading 3 go here</h3>  
<h4> The texts for Heading 4 go here</h4>  
<h5> The texts for Heading 5 go here</h5>  
<h6>The texts for Heading 6 go here</h6>
```

You can use headings according to importance. For instance, you can use `<h1>` for your most important headings and `<h2>` to `<h6>` for your less important ones.

You can also use headings to group and organize your content so that `<h1>` becomes your main section headings, while `<h2>` will introduce a subsection within your main section and `<h3>` will introduce further subsections within your `<h2>` subsection. If your page includes a title, you might want to use `<h1>` for the title, and `<h2>` for the main sections, and through `<h3>` for subsections.

As such, it is very important to use headings to:

- Make your document more readable.
- Make your document easy to skim. By looking at the headings, readers would be able to know the content of your page, the main sections and subsections.

- Headings can also help your page become more search engine-friendly (SEO).

When you have headings in your document, make use of the headings tag. There are other tags that you could use to achieve the same formatting as headings, but refrain from using them.

For example:

`<big>This Is a Heading</big>`
may be displayed the same way as
`<h1>This Is a Heading </h1>`

However, if you are looking at the source code, it would be much easier to know that `</p><h1>` is the main heading than if you use `<big>` and other similar tags.

Take note that headings are not formatting tags and that they do not tell the browser how to display the content in any particular manner. It just so happens that most browsers recognize headings and they have their own way of displaying these. The different formatting you see for `<h1>`, `<h2>` through `<h6>` will depend on what browser you use.

You can, however, format your headings to be displayed by using cascading style-sheets.

3.3 Paragraphs

When working with HTML documents, you would need to use the paragraph tag `<p>` in order to insert white space between two paragraphs.

You can use the following syntax when writing a paragraph for your HTML document:

`<p>Paragraph 1 starts here</p>`.

`<p>The second paragraph would follow here.</p>`

It is important to remember that you would need to add the closing tag `</p>` after you write every paragraph, because paragraphs are not empty elements.

You may skip on adding `</p>` at the end, and most browsers will ignore it and display your paragraph right. But some browsers may return errors when you do not put `</p>`. And that's everything you need to know first about titles, headings and paragraphs in HTML.

4. Unnumbered, Numbered, Definition and Nested Lists (Part 4)



There are two most common lists used in HTML and these are:

- Ordered list.
- Unordered list.

4.1 Ordered Lists

You use ordered lists if the sequence of your items is important. Ordered lists enumerate the various items on your list on separate lines. It will have less space in between items, unlike when you use `or` paragraph tags. Lastly, ordered lists are typically indented. The usual syntax for ordered lists is as follows:

```
<ol>  
<li>This is the first item on the list. </li>  
<li>This is the second item on the list. </li>  
<li>This is the third item on the list. </li>  
</ol>
```

This will be displayed as:

1. This is the first item on the list.
2. This is the second item on the list.
3. This is the third item on the list.

Attributes that you can use with ordered lists

As with other tags, you can change how ordered lists look by adding attributes. Two of the most used attributes are `type` and `start`.

1. Type

Instead of numbers, you can also have letters and Roman numerals. Types and their corresponding syntax are as follows:

```
<ol type="a"> will display a, b, c  
<ol type="A"> will display A, B, C  
<ol type="i"> will display i, ii, iii  
<ol type="I"> will display I, II, III
```

2. Start

You can have your ordered lists start with any number you want. For example:

```
<ol start="3">
```

The resulting list will start with 3 accompanying the first item.

```
<ol start="3">
```

```
<li>This is the first item on the list.</li>
```

```
<li>This is the second item on the list.</li>
```

```
<li>This is the third item on the list.</li>
```

```
</ol>
```

This will be displayed as:

1. This is the first item on the list.
2. This is the second item on the list.
3. This is the third item on the list.
- 4.

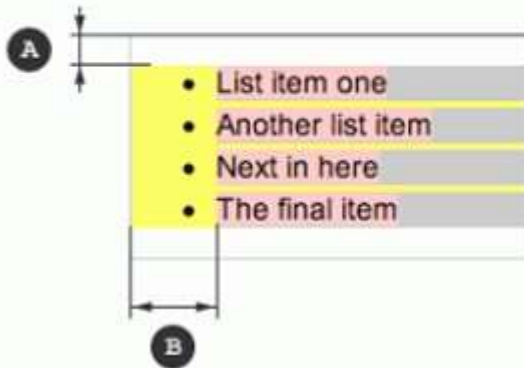
This is particularly useful if you have to interrupt your list to explain something. You can start off where the previous list ends.

4.2 Unordered Lists

Unordered lists are also called bulleted lists and, unlike ordered lists, the sequence of items is not important in this type of list.

Anatomy of HTML lists

```
<ul>
<li>List item one</li>
<li>Another list item</li>
</ul>
```



- A** margin applied to `` or ``
- B** left padding of `` or `` in standards based browsers, like Gecko, Webkit, Opera.
- or
- left margin of `` or `` in Internet Explorer.

Similar to ordered lists, each item in your list is presented on separate lines, introduced by a bullet. Unordered lists are also indented.

You can create an unordered list by using the following syntax:

```
<ul>
<li>This is item #1.</li>
<li>This is item #2.</li>
<li>This is item #3.</li>
</ul>
```

This will display as:

- This is item #1.
- This is item #2.
- This is item #3.
-

Further, if you are using HTML versions earlier than HTML 4, you can change the bullet by specifying the type.

For example, using:

`<ul type="square">` will display a square filled-in bullet.

`<ul type="circle">` will display a circle bullet that is not filled in.

While ordered and unordered lists are basic forms of lists, you might need to make use of lists that have more information other than just list items. HTML can handle that as well. Let us take a look at description lists and nested lists.

4.3 Description lists

Description lists allow you to have a description for each item on your list. Unlike `` and ``, you will not use `` for list items. Instead, `<dl>` tags need `<dt>` for the list items and `<dd>` for the description. Description lists are also known as definition lists.

The correct syntax would be:

```
<dl>
```

```
<dt>This is item #1.</dt>
```

```
<dd>- This is a description of item #1</dd>
```

```
<dt>This is item #2.</dt>
```

```
<dd>- This is a description of item #2</dd>
```

```
</dl>
```

The browser will display this as:

This is item #1.

- This is a description of item #1

This is item #2.

- This is a description of item #2

As you can see, only the description is indented.

4.4 Nested lists

A nested list may be an unordered list or an ordered list where you can have sub-lists for any item you want. This is good if you have complicated lists wherein you might need to enumerate another list for an item in your original list. Perhaps the simplest way to describe nested lists is that it is a list within a list.

The correct syntax for nested lists would be:

```
<ul>
<li>This is item #1.</li>
<li>This is item #2.
<ul>
<li>This is item #2 (a).</li>
<li>This is item #2 (b).</li>
</ul>
</li>
<li>This is item #3.</li>
</ul>
```

This would be displayed as:

- This is item #1.
- This is item #2.
 - This is item #2 (a).
 - This is item #2 (b).
- This is item #3.

If you need to use an ordered list, you can just replace `` with ``. For example:

```
<ol>
<li>Jeans</li>
<li>Shirts
</ul>
```


Navy blue.

Plain white.

This is item #3.

Will be displayed as:

1. Jeans

2. Shirts

- Navy blue.
- Plain white.

3. This is item #3.

And that's how you come up with lists for your HTML codes. Screenshot showing the use of various HTML list styles:

Duis ultrices eleifend ipsum. Pellentesque eget odio. Donec lobortis. Aenean viverra. arcu sed interdum rhoncus.

- pede erat faucibus quam, eu porta lorem justo in odio
- donec rutrum nisi ut nibh. In felis diam.
- ultrices nec, mollis at, euismod non, nunc. Sed at mi.

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae. In hac habitasse platea dictumst. Morbi nunc augue, mattis quis, lacinia in, blandit id, neque. Curabitur cursus magna et massa. Nullam quis sem. Mauris aliquam elementum libero. Maecenas laoreet. Nulla pellentesque tincidunt massa. Mauris at augue ut tellus tincidunt porta.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In gravida. Maecenas libero ligula, aliquet a, consequat at, dignissim vitae, eros. Vestibulum posuere urna sed turpis. Cras vulputate tortor sed libero. Nam nulla. Ut et purus id erat mattis nonummy.

- Donec blandit ante at metus. Aliquam erat volutpat. Suspendisse nisi magna, venenatis eget, lobortis sit amet, fringilla ut, eros. Morbi pretium, leo eget vulputate blandit, pede sem ornare quam, sed rutrum odio eros eget elit.
- Nam nulla nisi, laoreet quis, sagittis quis, volutpat sed, diam. Cras diam magna, faucibus sit amet, ornare eget, consequat sit amet, ligula.
- Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aliquam pede. In hac habitasse platea dictumst. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae.

Duis ultrices eleifend ipsum. Pellentesque eget odio. Donec lobortis. Aenean viverra. arcu sed interdum rhoncus.

Maecenas nunc ipsum, tincidunt a, luctus id, volutpat sed, orci. Maecenas aliquam. Donec fringilla aliquet risus. Nullam a elit non tellus molestie ullamcorper. Proin porta rhoncus dolor. Morbi rutrum. Aliquam id metus sed purus blandit sollicitudin.

- ④ In hac habitasse platea dictumst.
- ④ Integer malesuada faucibus mauris
 - ④ Sed semper, turpis vitae
 - ④ commodo ullamcorper, augue arcu
 - ④ sollicitudin lectus, ac ultrices diam ante ac orci.

Vestibulum molestie, tellus ac pretium accumsan, iacus tellus aliquet nulla, vel commodo ligula ligula sit amet mi. Cras enim lacus, rhoncus vel, molestie tincidunt, ornare at, pede. Sed sapien erat, adipiscing sed, facilisis vel, sollicitudin ut, velit. Nullam quis nulla.

Sed sed quam in magna bibendum ullamcorper. Vestibulum posuere pede non felis. Quisque gravida iaculis sapien.

1. Vivamus odio ipsum, luctus ac, gravida eget, lacinia sit amet, magna. Praesent convallis augue non lectus.
2. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sollicitudin quam.
3. Sed sodales mi et pede. Phasellus ultricies urna. Pellentesque lobortis. Fusce pellentesque porta lorem.

Maecenas nunc ipsum, tincidunt a, luctus id, volutpat sed, orci. Maecenas aliquam. Donec fringilla aliquet risus. Nullam a elit non tellus molestie ullamcorper. Proin porta rhoncus dolor. Morbi rutrum. Aliquam id metus sed purus blandit sollicitudin.

5. HTML for Beginners: How to Makes Links to Other Pages and Elsewhere

As you already know, links are a vital part of Web pages. Links allow you to move smoothly from one page to another. It allows you to navigate from page to page. It also enables you to get more information about a certain topic.



As such, there are a lot of different types of links that you should know when learning HTML. Let us first concentrate on the body of your HTML document. What are these link types that you should know about?

- Regular or text links
- Image links
- Links to e-mail addresses
- Placeholder or anchor Links

5.1 Regular or Text Links

The most common type of link is the text link. These are text that are usually rendered underlined and in blue, and allows you to visit a linked resource.

To create a simple text link, use the `<a>` tag by following this syntax:

```
<a href="URL of page you want to link to">link text</a>
```

What does this syntax mean?

- The `<a>` tag opens the link element.
- The `href` attribute will tell you the destination of the link.
- The "link text" could be anything from a simple description of the linked page to something like "click here" or "visit the page."
- The `` tag closes the element.

For example:

```
<a href="http://www.google.com">Google It!</a>
```

Would show as:

[Google It!](http://www.google.com)

Clicking on that link will take you to <http://www.google.com>. Take note that the destination URL does not have to be an HTML page or a Web page. Links could take you to an image file, a Word document, a PDF file, or even audio, video and other multimedia files.

5.2 Image Links

Sometimes, you would want to use an image instead of a text link. That is possible too.

To use an image as your link, use the following syntax:

``

``

On your page, the picture called image.jpg will be linked to any URL you have.

5.3 Links to e-Mail Addresses

Instead of an URL, image or video and other similar files, you could create an e-mail link. An e-mail link would open the user's default e-mail client (Gmail, Outlook, Apple Mail, etc) and start a new message with the specified e-mail addresses already filled into the TO: field. If you have also specified a subject, the subject line will also be filled in automatically.

How do you do this? Use the following syntax:

` Send me an e-mail!`

On your page, this will show up as

"Send me an e-mail!"

When you click on that link, it will open your e-mail client to compose a new message window. On the TO field, you will see you@yourdomain.com already entered and the subject line will read as "Creating an e-mail link!".

5.4 Anchor Links

There are times when you need to create links to different parts of your document. For example, FAQ pages usually have all the questions at the top of the page and then the answers to these questions follow below.

A good way to create a link that would take your users to a specific spot of the same

page is by creating anchor links. To do this, you must first create a bookmark using the "id" attribute.

For example, the answer to question number 10 is *"10. Yes, we do handle that."* Look at your source code, find that corresponding line and just add the bookmark like this:

```
<a id="AnswerTen">10. Yes, we do handle that.</a>
```

At the top of your page, create a link to that bookmark, i.e.:

```
<a href="#AnswerTen">Question 10: Do you handle HTML tutorials?</a>
```

On your page, Question 10 will be highlighted and when you click on it, it will take you directly to the bookmarked answer below. Do not forget to add the # sign before the bookmark name when linking.

5.5 Other considerations when creating links:

1. Take note that you can specify where a linked resource would be opened. It could be

- opened in the same page (default).
- opened in a new window, in which case you would need to add the `target="_blank"` attribute to your link element.

If you are working with frames, there are also attributes that would allow you to specify on which frame the link would open.

- `target="_self"` would open the link in the same frame where you clicked it.
- `target="_parent"` would open the linked page in the parent frame.
- `target="_xyz"` would open the linked page in the frame named "xyz".

2. Using absolute or relative URLs.

Absolute URLs specify the full URL, i.e. `http://www.you.com/me.html`. This is usually used when you are linking to another Web site on another domain.

Relative URLs, on the other hand, will not have the domain part, but instead it will be written as `me.html`. This is usually used when you are linking to pages within your own domain.

6. HTML for Beginners: Images

Images are a big part of any Web page. It helps to drive your point across and helps to get your site visitor's attention. How do you insert images into your page?



You need to use this code:

```

```

The `` tag introduces an image, the attribute `src` or `source` allows you to point to your image file. The `alt` attribute stands for alternative text. This is what shows up when the image cannot be shown for some reason (slow Internet connection or server is not responding). The `width` and `height` attributes determine the dimensions of the image.

Take a note that you should use the full URL of the image if it is not hosted on the same directory as your HTML document.

For example:

```
`

## **6.1 The alt attribute**

It is very important that you should fill in the alt attribute. At the very least, it would give people an idea what the image is all about, even if they are not able to see it.

However, the alt attribute, which is most often mistakenly called the alt tag, also serves other purposes.

For example:

1. A blind person accessing your Web site via a screen reader will hear the alt attribute, allowing him or her to "see" the image you have included in your page.
2. Text-based browsers and search engine spiders will be able to read the alt attribute.

Speaking of search engines, the alt attribute can also help to make your page or at least your image rank higher in search engine results pages. There is even speculation among SEO professionals that putting your keywords in the alt text actually has more weight than putting your keywords in the title or header tags. Also, image search services, such as Google Images, Yahoo Image Search and Bing Images, use the alt attribute.

## **6.2 The width and height attributes**

When inserting an image, it would be a good practice to specify the width and height of the image. This way, the browser will know just how much space to leave for your image. This means that even though your image does not load right away, the page will still be properly laid out as if the image has already completely loaded.

It is also important to use the right proportions so that your image does not look

distorted. For example, if you have an image that is 200 pixels wide and 200 pixels high, specifying a width and height of 200 pixels and 100 pixels respectively will result in the page showing a deformed image.

### 6.3 Align your images

It is also important to align your images. For example, you can have an image displayed at the center of the paragraph, or you could have it flushed to the right.

You could do this using the align attribute.

- To align your images to the left, use `<img align="left">`
- To align your images to the right, use `<img align="right">`
- To align your images to the middle, use `<img align="middle">`
- To align your images to the top, use `<img align="top">`
- To align your images to the bottom, use `<img align="bottom">`

The align attribute for the `<img>` tag, however, has already been deprecated in HTML 4.01, and if you are using HTML 5, it is not supported anymore.

You could still, however, use CSS to align your images by using the following syntax: `<img style="float:right">` in order to align your image to the right.

### 6.4 Adding links to your images

Sometimes, you will want to use an image to link to another content. It could be another page or perhaps you would want to give people a larger view of the image you have used for your page.

You can add a link to your image by using this syntax:

```

```



That sounds simple enough. But what if you need to add two or more links to a single image? How do you do that?

Though it's a bit outdated, you want to at least know about how you can create such things. An image map is what you would call a single image that has many hyperlinked areas. For example, if you have an image of a face and you would want to link the eyes to a page you call eye.html and the lips to lip.html, you would need to create an image map.

To create an image map, you would need to get the coordinates of your hot spots. Hot spots are those areas that are clickable within your image. To get the coordinates, you can use an image processing software or an online tool such as the Poor Person's Image Mapper.

Once you know the coordinates, you can now insert an image map into your page. You can use the following syntax:

```

<map name=face>
<area shape=rect coords=0,0,129,129 href="eye.html">
<area shape=rect coords=130,130,159,159 href="lip.html">
</map>
```

In this image map, the rectangular area within the coordinates (0, 0) and (129, 129) is clickable and linked to the eye.html page. Similarly, the rectangular area within the coordinates (130, 130) and (159, 159) is also clickable, but this time it is linked to the lip.html page. There are other shapes that you could use:

```
<area shape=circle coords=0,0,129,129 href="url">
```

*and*

```
<area shape=polygon coords= 0,0,129,129 ..., xn,yn Href="url">
```

So that's how you add images to your site.

## **7. HTML for Beginners: How to Use Tabular Data**

Tables enable you to easily present data and information in such a way that it is also easy to read and understand.

Tables are just a collection of cells. Or to be more accurate, it is just columns and rows of cells that contain information such as text, numbers, links, other tables, images and other content.

We are not going to make you think, however, that tables are a breeze to create in HTML. As you have more cells, columns and rows, you will find that the code you have to write becomes a little bit more unwieldy too.

Nevertheless, knowing the basics and practicing would allow you to get tables right.

### **7.1 When to Use Tables and When Not to**

First, let us be clear that tables are not to be used for layouting, formatting and positioning. For example, if you need to have a block quote, or spaces around images, do not use a table for it, you can get similar results using other HTML tags.

You should only use tables if you have to present data. If your content makes sense when you use a spreadsheet to present it, then you should use a table to put it on an HTML document.

## 7.2 Creating Tables

The `<table>` tag defines all tables in an HTML document. To make it easier for you to hard-code your table, you should always remember that HTML tables work first with rows (designated by the `<tr>` tag), then columns (defined by the `<td>` tag).

Let us start with just one cell. To do this, use the following code:

```
<table border="1">
<tr>
<td>one cell</td>
</tr>
</table>
```

This will show up as:

one cell
-------------

The border attribute shows the grid lines for your table. Next, try to create a table with just one row, but three different columns.

```
<h4>One row, multi column:</h4>
<table border="1">
<tr>
<td>column 1</td>
<td>column 2</td>
```

```
<td>column 3</td>
</tr>
</table>
```

If you need to have four columns, just add another `<td>`. The number of `<td>` lines would correspond to the number of columns you have in your table.

This will show up as:

column 1	column 2	column 3
----------	----------	----------

Now for something a little more challenging, why not try to have a table with two rows and three columns?

```
<table border="1">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
<td>Row 1, Column 3</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
<td>Row 2, Column 3</td>
</tr>
</table>
```

As you can see, there are two different sets for `<tr>`, which would mean that the table would have two rows. Three `<td>` tags for each `<tr>` set would mean that each row has

three columns.

This code will show as:

Row 1, Column 1	Row 1, Column 2	Row 1, Column 3
Row 2, Column 1	Row 2, Column 2	Row 2, Column 3

### 7.3 More on the Border Attribute

We have already explained that the border attribute would display the grid lines or border of your table. If you do not want borders for your table, you can just omit the border attribute or set its value to zero. Conversely, if you want a thicker border on your table, you can put in a higher value for your border attribute, i.e.,

```
<table border="10">
```

HTML Table Headers

If you need to include headers in your table, you would need to use the <th> tag instead of the <td> tag in your first <tr> set.

For example, to create the following table:

Header 1	Header 2	Header 3
Row 1, Column 1	Row 1, Column 2	Row 1, Column 3

You would need to write the following HTML code:

```
<table border="1">
```

```
<tr>
```

```
<th>Header 1</th>
```

```
<th>Header 2</th>
```

```

<th>Header 3</th>
</tr>
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
<td>Row 1, Column 3</td>
</tr>
</table>

```

## 7.4 Captioning a Table

A caption could be your table's title or short description. You can add this at the top or the bottom of your table, but within your `<table>` tags.

To add a caption, use this syntax:

```
<caption>Title of Your Table</caption>
```

## 7.5 Working with cells that take up two rows or two columns

There are times when you need to have headers or cells that spans two or more rows or columns. You can accomplish this using `colspan` or `rowspan`.

For example, to create this type of table:

Category	Examples		
Fruits	Apples	Oranges	Grapes

You would write this HTML code:

```
<table border="1">
<tr>
<th>Category</th>
<th colspan="3">Examples</th>
</tr>
<tr>
<td>Fruits</td>
<td>Apples</td>
<td>Oranges</td>
<td>Grapes</td>
</tr>
</table>
```

Take note that your second header, has the attribute colspan. Conversely, writing this code:

```
<table border="1">
<tr>
<th>Department:</th>
<td>Accounting</td>
</tr>
<tr>
<th rowspan="2">Contact Person:</th>
<td>Freya Olsen</td>
</tr>
<tr>
<td>Maita Cot</td>
```

```
</tr>
</table>
```

would allow you to have a cell that spans two rows. Notice that the header for Contact Person has the attribute `rowspan`.

Department:	Accounting
Contact Person:	Freya Olsen
	Maita Cot

## 7.6 Making Data Easier to Read

Sometimes, when you work with blocks of data, it is difficult to read it because it is displayed too close to the border. This is where the attribute “`cellpadding`” comes in handy.

```
<table border="1"
cellpadding="10">
<tr>
<td>This cell is padded. </td>
<td>This cell is padded too. </td>
</tr>
<tr>
<td>The world is round.</td>
<td>The sky is grey.</td>
</tr>
</table>
```



The resulting table will have your text with sufficient white space around it.

A	B
C	D

Instead of looking like this:

A	B
C	D

## 7.7 HTML Tables Are Very Flexible

As you can see, HTML tables are very flexible. Aside from allowing you to have exactly the cells, rows and columns that you need, you can also put in any data you want. For example, you could have words, paragraphs, bulleted lists and even another table within a table.

## 8. Text Boxes and Other User-Input "Thingamajigs"

HTML forms provide a great way to capture data from your web site. A form in an HTML page can be used to submit data to a server for further processing.

Each HTML form is introduced by the <form> tag, and is made up of different input elements. The <form> tag has several attributes that would tell the browser what to do with the information that is entered in the different fields. These attributes are:

- action
- method

The action attribute tells your browser the location of the cgi script that you are going to use to process the form. On the other hand, the method attribute has two possible values, *method=get* or *method=post*, both of which specify a method of submitting data to the script you have specified in the action attribute.

## 8.1 Form Fields

Any HTML form can contain the following fields. First you have the input elements:

*Text*

*Password*

*Checkbox*

*Radio*

*Submit*

*Reset*

*File*

*Hidden*

*Image*

*Button*

There are also other input methods that may be used in an HTML form, including drop-down lists, radio buttons, check boxes, text areas and others.

Let us see how each of these text elements is used in an HTML form.

## 8.2 Text Field

A text field is just that, a field in your form where the user can enter a line of text. This is ideal for names, states, zip codes and other similar information that does not need too much space.

The syntax for a text field is as follows:

```
<input type="text" name="shortnameforfield">
```

For example:

School: `<input type="text" name="schoolname">`

### 8.3 Password

Very much like a text field, but this one is especially earmarked for passwords and other sensitive data. Whatever you enter into the password field is masked.

The syntax for a password field is as follows:

```
<input type="password" name="passwd">
```

### 8.4 Checkboxes

A checkbox is used if you have a list of choices, a list from which you want the visitor to select as many options as possible or let them select none of the choices you provided.

```
<input type="checkbox" name="color" value="yes">
```

*Yes, I am interested!*  
<br>

```
<input type="checkbox" name="color" value="no">No, do not contact me.
```

If you want to have one of the options in your checkbox already checked (checked by default), you can use the checked attribute for that option. For example:

```
<input type="checkbox" name="color" value="yes" selected>
```

*Yes, I am interested!*  
<br>

```
<input type="checkbox" name="color" value="no">No, do not contact me.
```

## 8.5 Radio Buttons

Radio buttons are used when you have a set number of options and you only want to let your site's visitor to choose only one of these choices.

```
<input type="radio" name="reply" value="yay">
```

*Yup*<br>

```
<input type="radio" name="reply" value="nay">Nope
```

## 8.6 Submit

The submit button allows your visitors to send their input to the server.

The correct syntax for the submit button is:

```
<input type="submit" value="Submit">
```

*You can use an image for your submit button, or simply an image button.*

```
<input type="image" src="image.gif" name="image">
```

## 8.7 Reset Button

If the submit button sends all the inputs to a server, the reset button allows your users to clear the form or revert it back to its original state.

The correct syntax for the reset button is

```
<input type="reset" value="Reset">
```

## 8.8 Drop-down Menus

Drop-down menus allow you to give your visitors a set number of options to choose from. Drop-down menus can function like a checkbox where you can choose more than one option. It can also work like a radio button wherein you are forced to choose only one option.

What's more, drop-down menus tend to take up less space than both checkboxes and radio buttons because you do not have to lay out everything. For example, with radio

buttons and checkboxes, you would need to allot five lines for five options, but with a drop-down box you only have to allot one line.

So if you have a field wherein you do not need people to see all the choices all at once, drop-down menus are a great idea. The typical code for a drop-down box would be:

```
<select name="alphabets">
<option value="a">A</option>
<option value="b">B</option>
<option value="c" selected>C</option>
<option value="d">D</option>
</select>
```

Having the attribute "selected" would mean that when the drop-down box is first displayed on the page, it will show option "C" as the selected value (selected by default).

This HTML code will force your visitors to choose only one option. If you need to allow your visitors to choose more than one option, you just add the multiple attribute to the `<select>` tag, like so:

```
<select multiple name="alphabets">
```

## 8.9 Text Area

A text area acts like a text field, only that you have more space than just one line. It is not defined by an `<input>` tag, but by a tag such as:

```
<textarea>
Input text here
</textarea>
```

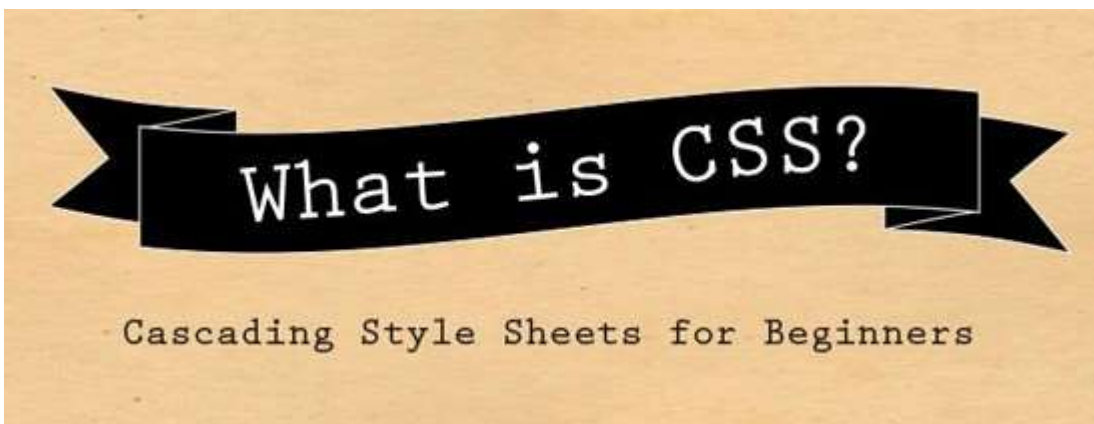
This will show a text box with the words "Input text here" filled in. You can also specify the number of rows and columns to control the size of your text box, i.e., `<textarea rows="15" cols="20">`

These are the different form fields that you can use in an HTML form.

## 9. HTML for Beginners: CSS

Cascading style sheets or CSS controls how a Web site would appear. There are many ways of styling a particular Web page, but CSS is the one recommended by the World Wide Web Consortium.

Using CSS can help you separate the content from the styling. What this means is that you spend less time in maintaining the overall look of your site. If you need to change how a certain element looks, you do not have go through each and every page to do so, you only need to tweak your CSS.



### 9.1 Benefits of Using CSS

CSS helps you save on bandwidth. Your style sheets are cached after you load them

for the first time. Putting your entire presentational markup in one CSS file can significantly help cut your bandwidth requirements.

Because you get to save on bandwidth, your page loads faster. It benefits visitors in such a way that they get to access the information they need quickly and easily.

Without presentation markup in your pages, you can easily port the content you have to RSS feeds and other similar ways to repurpose your content.

To be clear, CSS is a different language than HTML, but the two are complementary. In most cases, however, CSS is used with XML, HTML and XHTML.

## **9.2 A Brief History of CSS**

CSS is not really new. The first set of standards for CSS was first recommended by the W3C in 1996. Back then, it was mostly bare bones allowing you to control typography (spacing, margins, lists, alignment and fonts), boxes, borders and not much else.

In 1998, CSS2 came out with tons of new features introduced. However, designers had some difficulties in implementing CSS2 that it had to be revised. CSS2.1 became the second CSS standard to be implemented by browsers.

At the moment, CSS3 is implemented.

## **9.3 How to Use CSS with Your HTML Document**

There are three distinct ways to link cascading stylesheets to your HTML documents:

1. Use an external style sheet and refer to it in the HTML document by using the `<link>`

element or *@import* rule.

You can use the following code to do this:

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

Place this code in the `<head>` section of your page. Additionally, you can use the *@import* rule for your stylesheet:

```
<style type="text/css">
```

```
@import url(/style.css);
```

```
</style>
```

2. Use an inline style sheet that is introduced by the style attribute.

3. Use a separate but internal style sheet, introduced by the style tag. This must be placed in the `<head>` section of your Web page. Use the following syntax:

```
<style type="text/css">
```

```
(list of CSS rules here)
```

```
</style>
```

## 9.4 What you should know about cascading style sheets?





### ***1. Get the terms right.***

CSS is a totally different language, so better put a finger on the different terms you are most likely to encounter when studying CSS. What are these?

#### ***Comment.***

Comments are not markup. It does, however, provide information about CSS that makes it easier to maintain. This is especially helpful when two or more developers are coding the same CSS file.

Create a CSS comment by using the following syntax:

```
/* This is a sample comment. You can put whatever you want here. */
```

### *At-rules.*

At-rules, as the name suggests, begin with the @ sign. It typically signifies instructions to the CSS parser to do a variety of functions. Common at-rules are:

- @charset: specifies character encoding of the external CSS
- @import: allows you to use or import one CSS into another CSS
- @media: allows you to assign CSS rules to certain types of media
- @page: allows you to indicate margins for anything that has pages in it
- @font-face: allows you to specify customized fonts

### *Rule sets.*

Rule sets are what you use to assign presentation markups to a certain tag. For example:

```
h1 {color: #333;}
```

Rule sets have a selector (in this case, the selector is h1) and declaration blocks (color: #333).

### *Statements.*

CSS is just a collection of statements. At-rules and rule sets are examples of statements.

## ***2. How does a browser know what declarations to use in case the same elements have two or more contrasting declarations?***

Simple, the browser would assign a weight to each declaration of an element. To do this, it takes into consideration the following factors:

- importance
- origin
- specificity
- source order

## ***3. Hacks and Workarounds***

CSS is by no means perfect. One of the biggest problems when using CSS is that some parts of it might not be compatible with certain browsers. How your CSS is used to display your Web page largely depends on the browser. There are times when the page is displayed incorrectly, and there are times when CSS has been known to cause browsers to hang.

There are solutions to these. First, there is the hack that allows you to exploit buggy features in CSS so that you could specify a different styling for certain problematic browsers.

The second is called filters. Filters usually make use of advanced CSS features so that if you are using newer features that are not compatible with older browsers, the filter

can exclude these browsers.

What's the difference between hacks and filters? Hacks work on the browser side so that your page remains readable even when there are compatibility issues with your CSS, while filters are CSS-oriented.

These are the basic things you should know about CSS.

## **10. HTML for Beginners: Putting It All Together**

Learning HTML may seem daunting at first. This is especially true if you do not have any experience working with HTML at all.

But that does not mean that you should just forget about it. On the contrary, you should go ahead and take the plunge. The best way to learn HTML is to just do it.

### **10.1 Start by Making Simple Web Pages**

When learning HTML, always start with simple, easy-to-do and basic Web pages.

In this series, we have given you a list of things that you need to know in order to get started, as well as knowing the terms (tags, elements, attributes, and others).

We also gave you a rundown of the most basic elements that you are going to use when creating a Web page. This includes lists, paragraphs, headings, images, tables and forms.

Not only that, you were also given the proper syntax for each of these tags, as well as

the widely used attributes for each.

Start with a simple Web page first. Make sure that you have the primary tags in place, then work at adding elements.

Learn the tags one by one. For example, you could learn about `<u/>` today and then move on to `<o/>` tomorrow.

If you hit a roadblock, try to see what others are doing. What tags are people using and how do you write these tags?

Next, check out what attributes are valid for these tags to see how you could customize your Web page.

Or, better yet, check out some online tutorials to help you out.



## 10.2 Why HTML Should Be Easy

The best thing about HTML is that it is easily readable by humans. You can take a look at the source code of an HTML document and know which ones are paragraphs, which ones are headings, and which ones are table cells and table headers.

It is also very easy to remember tags. Most of the time, HTML tags take the first or the first two letters of what it is supposed to do. For example, `<p>` are for paragraphs, `<em>` is for emphasis, `<th>` is for table heading, and so on.

## 10.3 Simplifying HTML and CSS

You should always remember that Web pages have two very distinct parts: content and presentation.

Content is the information that you put on your Web page. This could be the block of text or the numbers and statistics that you have tabulated and computed.

Presentation, on the other hand, would include every element that would dictate how you present your content to your reader. This would include the layout, colors, fonts and even the formatting you use for each and every element in your Web page.

HTML takes care of how you structure content on your site, while CSS takes care of how you present your content.

Think of it as using a word processor to type a document. You first write your title, your headings, your paragraphs. You can use lists and tables to make your information even easier to read and understand. This is what HTML does.

After typing everything, you can then prettify or improve your document by assigning different properties to your headings, changing the font color for your title, changing your paragraph's font from Times New Roman to Verdana. This is what CSS does.



## 10.4 Other Things You Need to Know When Learning HTML

Aside from learning about HTML itself, along with tags, attributes and the proper syntax, you should also know the following:

- You could easily write an HTML document as long as you have a text editor like Notepad. You can also use something like Microsoft Word just as long as you remember to save in plain text. Always use the .html file type when saving.
- You could check your HTML document using any browser you might have.
- How can you check the source code of HTML pages? When learning HTML it is important for you to know how to see a Web page's source code. This way, you can determine and follow how HTML tags and attributes function. Different browsers have different ways to do this. In Firefox and Chrome, you can right click on the page you



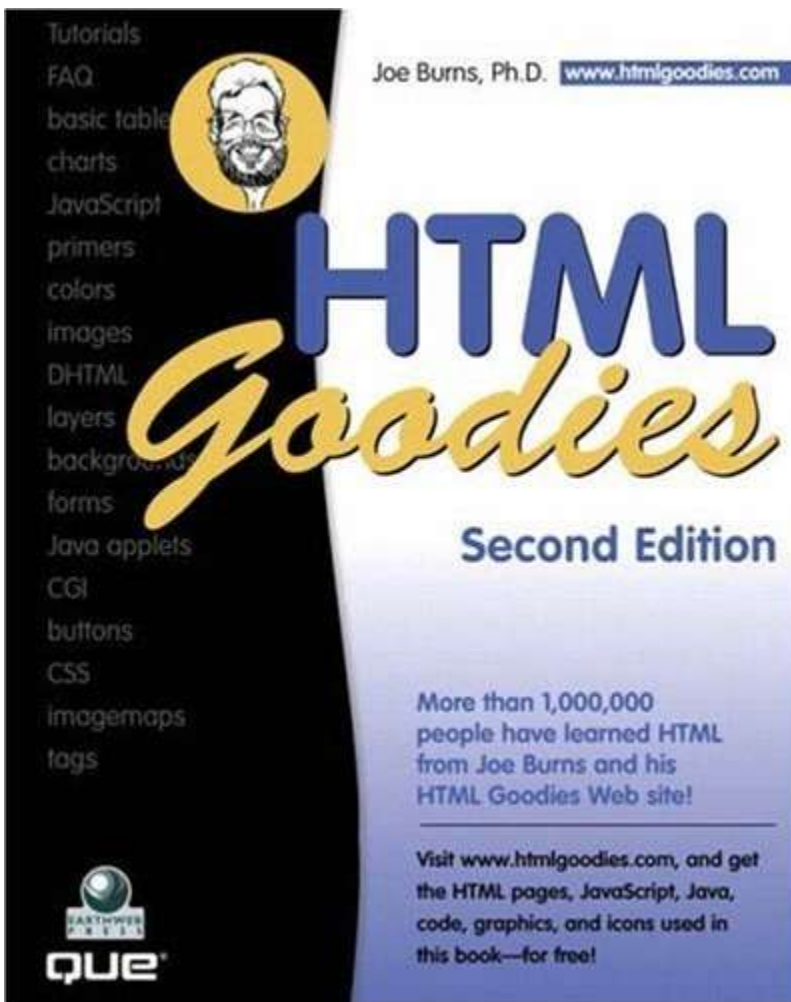
want to see the source code for and select View Page Source in the menu that comes out.

## **10.5 Learning Resources**

There are also some resources that could help you with learning both HTML and CSS.

Online classes. There are different HTML classes online that allow you to learn at your own pace and can teach you about anything related to HTML. There are also online classes that send you daily or weekly lessons via e-mail.

Online tutorials. Unlike online classes wherein you get a sustained and continuous lesson, online tutorials are there to help you learn about HTML quickly. For example, if you need to create an HTML table, you can look for online tutorials that teach you just how to do that.



HTML how-to books. If you would like to learn HTML even without an Internet connection, Web design books are a great idea. There are several great books on HTML and Web design out there.

To start, check out [HTML for the World Wide Web](#) by Elizabeth Castro, Jennifer Niederst's [Learning Web Design](#) 4th Edition or [HTML Goodies](#) by Joe Burns. HTML Goodies is also an excellent online resource for HTML newbies. You can visit their website [here](#). If funds are tight, you can probably borrow a Web design book from a local library.

## **Learn from others.**

If you come across a site that you like, try to check out their source code and see what tags they are using. You can then research these tags and learn how they are used. Practice makes perfect!