

Forum:[Hack Forums](#)

Thread: [\[The Empire\] Beginner HTML guide \[The Empire\]](#)

Rap!st - February 18, 2014, 6:55 am

In this tutorial I'm explaining the basics of HTML in order to get started with your webdeveloping adventure.

This tutorial is an official tutorial of The Empire.

Requirements:Editor

Patience

Internet

Difficulty:1/10, Read and you'll succeed.

Table of contents:Choosing your text editor.

Getting started.

Page Title.

Elements, Attributes and Tags.

Paragraphs.

Ordered and unordered lists.

Divisions.

Tables.

Choosing your text editor.

Obviously you need an editor to modify and create your code. Below are a few examples of common used Editors including the downloadlink.

Windows Editors: [Notepad++](#)

[Brackets](#)

[Dreamweaver](#)

[Sublime text](#)

[NVU](#)

[Eclipse](#)

Mac OSX Editors: [Brackets](#)

[Sublime text](#)

[TextWrangler](#)

[Aptana Studio](#)

You're free to choose the one you'd like to however I would recommend Notepad, Eclipse or Brackets for Windows and Brackets or Aptana Studio for Mac.

Getting started.

After you've downloaded one of the editors above you're going to create your first HTML page. Open a new file in the editor and save it as .html

(Hypertext Markup Language)

Every basic HTML file has a default code in order to declare it's an HTML file and in order to properly create a structure.

Here's how the code looks like:

With Comments:

Spoiler [\(Click to View\)](#)

PHP Code

```
<?php
<!DOCTYPE html>
<html> [b]<!--[/b] Opening HTML tag -->
<head> [b]<!--[/b] Opening Head tag -->

</head> [b]<!--[/b] Closing Head tag -->
<body> [b]<!--[/b] Opening Body tag -->

</body> [b]<!--[/b] Closing Body tag -->
</html> [b]<!--[/b] Closing HTML tag -->
```

Without Comments:

Spoiler [\(Click to View\)](#)

PHP Code

```
<?php
```

```
<!DOCTYPE html>
<html>
<head>

</head>
<body>

</body>
</html>
```

The following code can be added within the `<html>` `</html>` tag in order to add a comment which is mainly used to remember something once you've got a bigger code. In this tutorial I'm mostly using this to explain what something is or does.

PHP Code

```
<?php
[b]<!--[/b] Comment here -->
<!DOCTYPE html>
```

The line on the top; `<!DOCTYPE html>`, is a document type declaration and it lets the browser know which flavor of HTML you're using (HTML5, in this case). I would highly recommend to use this. If you don't, browsers will assume you don't really know what you're doing and may act in a peculiar way depending on what you're trying to do.

`<html>` / `</html>` The `<html>` tag is the opening tag that tells the browser that everything between that and the `</html>` closing tag is an HTML document.

`<head>` / `</head>` The `<head>` tag is basically everything you want preloaded before the things in the body begin to load as example things like things like meta tags, titles, scripts, stylesheets, scripting functions, etc.

`<body>` / `</body>` The stuff between `<body>` and `</body>` is the main content of the document that will appear in the browser window.

Page Title.

Every webpage has their own title which will be shown somewhere at the top, this depends on which browser you're using. The title can be declared by using the `<title>` tag. Note that this tag must be placed within the `<head>`

Here's an example:

PHP Code

```
<?php
<!DOCTYPE html>
<html>
```

```
<head>
<title> Rapist Tutorial </title>
</head>
<body>

</body>
</html>
```

The result in Google Chrome:
Spoiler ([Click to View](#))

The result in Mozilla Firefox:
Spoiler ([Click to View](#))

Elements, Attributes and Tags.

HTML(5) has several Elements, Attributes and Tags. Below I'm going to explain what they are and how they're used.

Tags.

You may didn't know but you've already used some tags. Every tag has an "Opening" and "Closing" tag. within the tag you can add Attributes and sometimes you can fill it with Elements.

The opening tag is almost the same as the closing tag, We're using the body and paragraph tags as example.

The body tag

PHP Code

```
<?php
<body> [b]<!--[/b] Opening tag of Body -->
</body> [b]<!--[/b] Closing tag of Body -->
```

The P (Paragraph) tag

PHP Code

```
<?php
<p>[b]<!--[/b] Opening tag of P ( Paragraph ) -->
Text within the Paragraph tag. All the text placed here will be declared by the paragraph settings
</p> [b]<!--[/b] Closing tag of P ( Paragraph ) -->
```

I'll list some of the most used tags below so you've got a better view on them.

Spoiler ([Click to View](#))

PHP Code

```
<?php
<html> -- </html>
<head> -- </head>
<title> -- </title>
<body> -- </body>
<p> -- </p> ( Paragraph )
```

 -- (Bold)
<u> -- </u> (Underline)
<i> -- </i> (Italic)

 -- </br> (Break rule)
<div> -- </div> (Division)
<marquee> -- </marquee>
 --
<strike> -- </strike> (strikethrough text)
<a> -- (Anchor Tag -- Hyperlinks)
 -- (Images tag)
<table> -- </table>
<tr> -- </tr> (Table Row, Requires to be within the <table> tag)
<td> -- </td> (Defines a cell in a table, Also requires to be within the <table> tag)
 -- (ordered list)
 -- (defines an unordered (bulleted) list.)
 -- (defines a Listitem, Requires to be used within an or tag.)

[Click Here](#) to see a full list of HTML tags.

Self-closing Tags

You may use Self-closing tags someday which is the reason I'll give a quick explanation about them. Self-closing tags are tags which aren't required to use the same "Closing" tag as normally. The easiest example I could give you is the
 tag. You may close this with </br> but since this is one of the Self-closing tags you may also close it like this:
 You may only need to think about this whenever you're using XHTML as example. XHTML is almost the same as HTML but doesn't allow any mistakes like HTML may does. Feel free to use google for information about this.

Some of the examples above require an attribute or a parent tag in order to function properly. I'll explain those later.

Attributes.

Now we've had most of the HTML tags, It's time to explain what an Attribute is and how it could / should be used. Every attribute must be within the "Opening" tag and their value must be between quotation marks.

PHP Code

```
<?php  
<tag attribute="valuehere"> Something Here </tag>
```

Let's use the Anchor and Image tag as examples.

Anchor Tag Example:

PHP Code

```
<?php  
<a href="https://hackforums.net" title="Rapist Tutorial"> Link Text Here </a>
```

The following Attributes have been used in the example above:Attribute HrefThe href attribute is to define a link for the Anchor tag. The value is the Hackforums website which will be the link obviously.

Attribute TitleThe title href is to define a title when you're hovering the text "Link Text Here " on a webpage. this attribute is filled with the value "Rapist Tutorial" obviously the value will show up when you're hovering the link text.

Result:
Spoiler ([Click to View](#))
The link:

The title when Hovering:

Image Tag Example:

PHP Code

```
<?php
```

```
<img src"images/hackforums.png" title="Hackforums Image" alt="Hackforums Image"></img>
```

The following Attributes have been used in the example above:Attribute srcThe src attribute is to define the directory where the Image is saved. In this example the image is saved in the images folder and is named hackforums.png (PNG is the extension ofcourse)

Note that the images folder must be in the same folder as the page you're using this code at since it's simply browsing from there to folder "images" and picks the hackforums.png out of that folder.

Attribute TitleThe title href is to define a title when you're hovering the specified image. The value of the title attribute is what's showing up when you're hovering the image.

Attribute altThe alt attribute stands for alternative if I'm correct. It's used to show a text instead of the image when you've specified an invalid directory in the value of the src attribute.

If you'd like to know the attributes for other tags, feel free to reply here or put some effort in it and use Google.

Elements.

Elements is the easiest of the 3 things to explain. It's simply the text within some tags. Below are some examples.

As I've stated above, the <title> code is a tag. Whenever you're filling it with text, the text is called the Element. Since it's not within quotation marks and it has nothing to do with an attribute, it's called

an Element.

Title element:

PHP Code

```
<?php
```

```
<title> This is the Title Element </title>
```

The text "This is the Title Element" will be the title of the webpage.

Anchor element:

PHP Code

```
<?php
```

```
<a href="" title=""> This is the Anchor Element </a>
```

The text "This is the Anchor Element " will be the clickable text for the link.

Paragraphs.

Paragraphs are used alot in websites. They're also really easy to use when you'd like to show alot of text.

Here's an Example on how to properly create a paragraph.

PHP Code

```
<?php
```

```
<p>
```

This is some text you can type which will be defined by the paragraph settings.

```
</p>
```

```
<p>
```

Simply close the paragraph tag and open a new one in order to create a new "Paragraph".

```
</p>
```

Result:

Spoiler [\(Click to View\)](#)

As you can see, it automatically added some space between the paragraphs.

Ordered and unordered lists.

Ordered and unordered lists are mostly used to create menu's, or lists.

Ordered list

The ordered list element, ol, works just like the unordered list element, including how each individual list item is created. The main difference between an ordered list and an unordered list is that with an ordered list the order of which items are represented is important. Instead of showing a dot as the default list item element, an ordered list uses numbers. You're able to change these numbers to letters as example by editing this in CSS

Unordered list

Unordered lists are purely a list of related items, in which their order does not matter nor do they have a numbered or alphabetical list element. Creating an unordered list in HTML is accomplished using the unordered list, ul, block level element. Each list item within an unordered list is individually marked up using the list item, li, block level element.

By default most browsers represent each list item with a solid dot. This solid dot is referred to as the list item element and can be changed using CSS.

Here's an example on how to use an ordered list:

PHP Code

```
<?php
<!DOCTYPE html>
<html>
<head>
<title> Rapist Tutorial </title>
</head>
<body>
<ol>
<li> List item 1</li>
<li> List item 2</li>
<li> List item 3</li>
</ol>
</body>
</html>
```

Ordered list Result:

Spoiler [\(Click to View\)](#)

Here's an example on how to use an unordered list:

PHP Code

```
<?php
<!DOCTYPE html>
<html>
<head>
<title> Rapist Tutorial </title>
</head>
<body>
<ul>
<li> List item 1</li>
<li> List item 2</li>
<li> List item 3</li>
</ul>
</body>
</html>
```

Unordered list Result:

Spoiler [\(Click to View\)](#)

You're able to remove these dots by using the following code in your CSS file.

PHP Code

```
<?php
list-style-type: none;
```

More about CSS will be explained in another tutorial.

Divisions.

Divisions are mostly used to add different "Blocks" in a website. As example you could make different divisions for the Header, Content and footer of the website. You're able to add an id or class to a division in order to separate them from each other and add different settings to each.

It's allowed and highly suggested to use divisions within other divisions in order to create a properly working webpage.

Below is an example of divisions used to create a header, content and footer block. These blocks will be in divisions which will be the main divisions.

PHP Code

```
<?php
<!DOCTYPE html>
<html>
<head>
<title> Rapist Tutorial </title>
</head>
<body>
<div id="main"> [b]<!--[/b] Closing the div with the id "Main" -->
<div id="header"> [b]<!--[/b] Opening the div with the id "Header" -->
</div>[b]<!--[/b] Closing the div with the id "Header" -->
<div id="content"> [b]<!--[/b] Opening the div with the id "Content" -->
</div> [b]<!--[/b] Closing the div with the id "Content" -->
<div id="footer">[b]<!--[/b] Opening the div with the id "Footer" -->
</div>[b]<!--[/b] Closing the div with the id "Footer" -->
</div> [b]<!--[/b] Closing the div with the id "Main" -->
</body>
</html>
```

You can fill these divisions with tags you'd like to use. As example you could use the unordered list and list-item tag in the header to create a menu. I'm going to give you an example to create a paragraph in the content division.

PHP Code

```
<?php
<!DOCTYPE html>
<html>
<head>
<title> Rapist Tutorial </title>
</head>
<body>
<div id="main">
<div id="header">
</div>
<div id="content">
<p> Paragraph text which will be in the content division/block </p>
</div>
<div id="footer">
</div>
</div>
```

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

Each division could have another style now because we gave all of them their own "id". You can do this in your CSS file with the following code:

PHP Code

```
<?php
#main {
Background-color: ValueHere;
Height: valuehere;
Width: valuehere;
}

#header {
Background-color: ValueHere;
Height: valuehere;
Width: valuehere;
}

#content {
Background-color: ValueHere;
Height: valuehere;
Width: valuehere;
}

#footer {
Background-color: ValueHere;
Height: valuehere;
Width: valuehere;
}
```

As stated before, More about CSS will be explained in another tutorial.

Tables.

The table tag speaks for itself, you're able to create tables with this tag or you could even create your whole website as a table. Let's take a look at a quick example which would explain enough about this subject.

PHP Code

```
<?php
<table>
<tr>
<td>Firstname1</td>
<td>Lastname1</td>
<td>Age1</td>
</tr>
<tr>
<td>Firstname2</td>
<td>Lastname2</td>
<td>Age2</td>
</tr>
</table>
```

Result:
Spoiler ([Click to View](#))

You're able to add cellspacing, borders, colors etc trough your CSS file.

Some detailed information about tables:
Spoiler ([Click to View](#)) Tables are defined with the <table> tag.

A table is divided into rows with the <tr> tag. (tr stands for table row)

A row is divided into data cells with the <td> tag. (td stands for table data)

A row can also be divided into headings with the <th> tag. (th stands for table heading)

The <td> elements are the data containers in the table.

The <td> elements can contain all sorts of HTML elements like text, images, lists, other tables, etc.

The width of a table can be defined using CSS.

Dafoo - February 18, 2014, 7:01 am

Thanks for taking the time to write this for us!
In depth and well written! Bookmarked!

Mr. New Vegas - February 18, 2014, 7:12 am

Well written basic tutorial on HTML, excelent work mate!

tast3blood - February 18, 2014, 7:18 am

Nice guide OP big Vouch, if i was trying to learn html this would be a perfect guide. I recommend all users to check out the guide

Brooklyn - February 18, 2014, 7:19 am

Nice tutorial OP , Eclipse is compatible for Mac.

Rap!st - February 18, 2014, 7:19 am

Thanks for the kind words. I'm also using this tutorial and give live support with it for the customers I'm mentoring atm.

(Feb 18, 2014 08:19 AM)Brooklyn Wrote: ([CITE](#)) Nice tutorial OP , Eclipse is compatible for Mac. I know that, I've just added a few of them and since I didn't like eclipse on a Mac while I use it on my windows I didn't add it though.

You could always check the application you'd like to use if it works on your OS.

Rapist?

M4RTIJN - February 18, 2014, 2:55 pm

Thank you for this very useful and informative tutorial!

Good job, Rap!st!

M4RTIJN ?

Tibit - February 18, 2014, 3:09 pm

Nice tutorial you made noob :P

Good work.

Rap!st - February 19, 2014, 4:06 am

Thanks for the feedback guys. Glad you like it.

Rapist?

Haxalot - February 19, 2014, 9:09 am

Not a bad tutorial, though I can see a few of improvements.

The first improvement would be to specify the character encoding of the HTML file. This is done to ensure proper interpretation of the characters within the file - UTF-8 is recommended.

Also, since you're using HTML5, why not use its (newly introduced) semantic markup tags? You are using div tags for sectioning parts of your page when you could be (and should be) using the new tags introduced in HTML5 (such as header, footer, section, article, etc).

It may also be worth mentioning that tables should only be used for organising data, and not for layout or design of the web page.

I'd also indent your code for better legibility.

Rap!st - February 19, 2014, 9:54 am

(Feb 19, 2014 10:09 AM)Haxalot Wrote: [CITE](#) Not a bad tutorial, though I can see a few of improvements.

The first improvement would be to specify the character encoding of the HTML file. This is done to ensure proper interpretation of the characters within the file - UTF-8 is recommended.

Also, since you're using HTML5, why not use its (newly introduced) semantic markup tags? You are using div tags for sectioning parts of your page when you could be (and should be) using the new tags introduced in HTML5 (such as header, footer, section, article, etc).

It may also be worth mentioning that tables should only be used for organising data, and not for layout or design of the web page.

I'd also indent your code for better legibility.

As you may could read in the title and tutorial itself. It's just a beginner guide. If I'm going to explain every single thing about HTML I need way to much time to get this done. You're right about the Character encoding, I'll edit that asap. The reason I've used divisions in this tutorial is because I always liked to use this and never even tried the new tags since I'm fine without it.

I've never said you should use tables to create your website. I'm just saying if you'd like to work with tables you're able to create a whole website with it. The reason I mentioned you're able to create a full website in tables is simply to let them imagine what you're able to do with a simple table.

Rapist?

Haxalot - February 19, 2014, 10:01 am

(Feb 19, 2014 10:54 AM)Rap!st Wrote: [\[CITE\]](#) As you may could read in the title and tutorial itself. It's just a beginner guide. If I'm going to explain every single thing about HTML I need way to much time to get this done. You're right about the Character encoding, I'll edit that asap. The reason I've used divisions in this tutorial is because I always liked to use this and never even tried the new tags since I'm fine without it.

I've never said you should use tables to create your website. I'm just saying if you'd like to work with tables you're able to create a whole website with it. The reason I mentioned you're able to create a full website in tables is simply to let them imagine what you're able to do with a simple table.

Rapist?

I know it's a beginners tutorial, but what's advanced about using (at least) the header and footer tags? Introducing them into good habits early on is better than ignoring them. As for the tables, I just thought it would be worth a mention since beginners do tend to over use tables since they are easy to structure web pages with (especially if they do not know CSS).

What about indentation?

Rap!st - February 19, 2014, 10:06 am

(Feb 19, 2014 11:01 AM)Haxalot Wrote: [\[CITE\]](#) I know it's a beginners tutorial, but what's advanced about using (at least) the header and footer tags? Introducing them into good habits early on is better than ignoring them. As for the tables, I just thought it would be worth a mention since beginners do tend to over use tables since they are easy to structure web pages with (especially if they do not know CSS).

What about indentation?

You're right that could/should be done indeed. I also agree on the over usage of the table option when creating a website however I really hate to work with that myself. I'm indenting it atm but it seems that I'm not able to update my post because of the character limit.

Rapist?

shqipenet - February 13, 2015, 8:07 am

Thank you very much for this great guide mate !!