- HTML Color codes/Color Names – List of 140 HTML color names including HEX code, RGB value and HSL value.

HTML Files Structure

A HTML file is a text file with the `.html` extension.
The code in the file needs to start with `<!DOCTYPE html>`.

An HTML element is defined by a start tag and an end tag, with the content inserted between:

`<element-type>My element value</element-type>`

Notice how a tag is closed with the /
Some tags don't need to be closed, they're called **orphan tags**

- All the HTML code is inside an HTML element (`<html>`).
- The HTML code is separated into two big parts, the `<head>` and the `<body>`.
  The body is the visible part of the HTML code, while the head is here to add some config to the file.

Before content is added, most HTML files basically look like this:

```
<!DOCTYPE html>
<html>
<head>
    <title>Title of the document</title>
</head>

<body>
    The content of the document......
</body>

</html>
```

- Every HTML file begins with the opening tag `<html>` and ends with the closing tag `</html>`.
- The HTML file's first line should declare the type of document so that the browser knows what HTML version you use. You see that before the HTML code begins, it starts with `<!DOCTYPE html>` .
- The section in the `<head>` tag usually contains information about the document such as its title, meta tags, and where to locate its CSS file – content that is not directly visible on the browser page.
  - Title - Example : `<title>This is our page title</title>`
  - `This is our page title` will be displayed in the browser tab. It's also indexed as the title for the page when the search engine bots crawl your website.
  - Meta element: often used to specify the information, search engines can describe the content in their listings. This includes the description, keywords, author information, etc. The `<meta>` element also specifies the character set the HTML document uses.
- The section between the tag `<body></body>` is where the HTML file's main content is located. It's what the viewer will see in the browser. This includes everything from the first paragraph to hyperlinks, formatting, multimedia, and everything else.

Basic Building Blocks Of HTML

When coding in HTML, it is essential to understand these building blocks. They include tags, attributes, and elements.

Introduction To Tags

In essence, **tags** separate normal text from HTML code.
Tags are practically the building block of HTML – you can't do HTML without tags! If you're stuck on which tag to use, be sure to check out the Useful Resources above.

Opening And Closing Tags

Almost every open tag must be closed. Keep in mind that there are exceptions.

An example of a tag that does not have to be closed is an empty tag, such as the line break: `<br>`.

Tags are contained in less than (“<”) and greater than (“>”) angle bracket. Closing tags have a trailing slash that before the name of the tag being closed.

Example of an open tag: `<b>`. Example of a closed tag `</b>`.

When it comes to HTML, tags make the difference between whether your document is displayed as ordinary text or ‘transformed text’ . This ‘transformed text’ is a code that can display a series of things (hyperlinks, images, media, or other methods of formatting).

Bold & Italic Text

Let’s take a look at the word “He is a boy” as an example: * In ordinary text format you get: He is a boy. * In bold text format you get: **He is a boy**

To achieve what we have in the bold format, you have to use the `<b></b>` tag : `<b>He is a boy</b>.`

This sentence could also come out italicized.
This is achieved using the `<i></i>` tag : `<i>He is a boy</i>`.

Hyperlink

The `<href>` tag is used to make hyperlinked in HTML.

Example: In raw HTML we have:

```
<a href="https://developers.institute">Developers Institute</a>
```

Introduction To Elements

In HTML, an “element” consists of the opening and closing tag and the content between the tags.

Example : `<b>He is a boy</b>`.
The text “He is a boy” is surrounded by an open and closed tag. Everything, including the opening tag, the content, and the close tag is an **element**.

An element could be in a basic form or a complex form. Why? Because anything in between an opening tag and a closing tag is an element. It means that we can have elements within an element.

In our current example, “He is a boy” (`<b>He is a boy</b>`) is an element.

Introduction To Attributes

While HTML documents use tags for everything, we sometimes want to communicate additional information inside an element. In this case, we use an **attribute**. The attribute is used to define an element’s characteristics and it’s used inside the opening tag of the element. Attributes are made up of a name and a value.

Note that the value of an attribute is placed inside a quotation mark using the format

```
<tag attribute="value">Your Text</tag>
```

Example

```
<p align="center">He is a boy</p>
```
In this example, we instruct that “He is a boy” is aligned in the center of the document.

Creating Headings In HTML

Headings play a significant role in the success of a website. Firstly, they make it easy for readers to scan the content of your pages. Secondly, and perhaps more importantly, they communicate your web pages' structure to search engines and, therefore, often impact how your content is ranked in search engine results.

That said, it is essential to avoid using heading tags to make your text big or bold. Other tags can be used to make your text big (which we'll get to later in this section). Instead, heading tags should be used solely for structure.

There are six heading tags in HTML: `<h1>` to `<h6>`, with the `<h1>` tag indicating the most important heading and the `<h6>` tag indicating the least important heading.

- HTML Headings Example:

  Headings are used for titles and big texts

  # H1 Heading

  ## H2 Heading

  ### H3 Heading

  #### H4 Heading

  ##### H5 Heading

  ###### H6 Heading

Creating Paragraphs

Paragraphs can be created with the `<p>` tag.

Example

```
<p>This is your first paragraph.</p>
<p>This is your second paragraph, and you will be creating many more paragraphs.</p>
```

Keep in mind that writing in HTML is very different from writing in pure text. Therefore, if you break up text inside HTML without starting a new paragraph, it won't matter when the browser displays the text. Instead, you want to use a line break, represented by the `<br>` tag.

Example

```
<p>This is a new paragraph.
And I want to use a number of new lines.
So I'm breaking it up.</p>
```

This **will not** come out as the following:

This is a new paragraph.
And I want to use a number of new lines.
So I'm breaking it up.

Instead, it will come out as this:

This is a new paragraph. And I want to use a number of new lines. So I'm breaking it up.

So, how do you break texts into new lines so that it would show like this:

This is a new paragraph.
And I want to use some new lines.
So I'm breaking it up.

By using line breaks.

```
<p>This is a new paragraph.<br>And I want to use several new lines.<br>So I’m breaking it up.</p>
```

If you want to create a horizontal line, you can use the `<hr>` tag.

```
<p>This is a new paragraph.<hr>And I want to create a horizontal line</p>
```

It will display like this:

This is a new paragraph.

And I want to create a horizontal line

Formatting Text In HTML

The next step is to format your text in HTML. This is where you can indicate whether you want your text to come out bold, italicized, underlined, subscripted, superscripted, etc. This is a basic guide, so this section won’t be the be-all-end-all for formatting. Instead, we will only include some basic formatting tags. The process for using other forms of formatting is simple – just find the tag you want and go ahead:

Using bold: `<b>He is a boy</b>` comes out as **He is a boy**

Using italics: `<i>He is a boy</i>` comes out as *He is a boy*

Underlining text: `<u>He is a boy</u>` comes out as He is a boy.

It’s worth noting that the `<u>` tag was deprecated in HTML 4.01 and was redefined to represent stylistically different text in HTML5. As a result, it is recommended to use CSS to indicate text that should be underlined. Since this article is a basic guide, we’re keeping it simple.

Using subscript: `<sub>He is a boy</sub>` comes out as $_{\text{He is a boy}}$

Using superscript: `<sup>He is a boy</sup>` comes out as $^{\text{He is a boy}}$

For other tags that can be used to format, take a look at **Useful Resources** at the beginning of the page.

Exercise 1

Create a structured HTML file as explained above.

Write about yourselves : your name, age, birthdate and why you decided to learn to code. Style this small text with paragraph, headings, italic and bold characters.

Ordered And Unordered Lists

Sooner or later, you will have to create lists. Lists could be ordered (i.e., numbered) or unordered (i.e., unnumbered).

Here is an example of an ordered list:

1. Item 1
2. Item 2
3. Item 3

Here is how to create it:

```
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ol>
```

Here is an example of an unordered list:

- Item 1
- Item 2
- Item 3

Here is how to create it:

```
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```

If it is not already obvious. Here's a breakdown:

The `<li>` tag is used to indicate each item on the list.

When making a list, you can use the `<ol>` tag to indicate an ordered list ("o" = ordered and "l" = list) or the `<ul>` tag to indicate an unordered list ("u" = unordered and "l" = list).

Nested Lists

We can also have nested lists or a list within a list.

Example

- Item 1
  - Item 1 nested
  - Item 2 nested
  - Item 3 nested
- Item 2
- Item 3

This can be created with:

```
<ul>
    <li>Item 1
        <ul>
            <li>Item 1 nested</li>
            <li>Item 2 nested</li>
            <li>Item 3 nested</li>
        </ul>
    </li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```

As you can see, you open another listing tag – ordered (`<ol>`) or unordered (`<ul>`) – depending on what you want before closing the item you want to be nested.

Hyperlinking

The web is one massively interconnected network of pages. If you create a website – whether internally or externally, or both – you will have to link to other pages. A link to an internal page on your website or an external page on the web is called a hyperlink. While people usually link a text, any HTML element – an image, for example – can be linked.

The `<a>` tag is used to define links in HTML, while the “href” attribute is used to specify a link’s destination. The link is then put into a quote after the “equal to” sign before the tag is closed.

Example

To create a hyperlink,

this:

```
<a href="https://websitesetup.org">YOUR LINK TEXT HERE</a>
```

will come out as: YOUR LINK TEXT HERE.

Now, assume you are linking to a local file in which you have all your HTML pages in the same folder. In this case, you don’t have to include a website URL. Just add the file path.

For example, if linking to a file titled `about-page.html`, your hyperlink becomes

```
<a href="about-page.html">YOUR LINK TEXT HERE</a>
```

which links to the `about-page.html` file.

If the HTML file you want to link to is local but different from the main folder, you specify the file path.
For example, if the file is in the “files” folder under the main document where your current document is, your hyperlink goes something like this

```
<a href="files/about-page.html">YOUR LINK TEXT HERE</a>
```

You might want to specify how you want the link to be opened e.g., in a new window/tab. You will use the target attribute for that.

Example

To specify that the link should be opened in a new tab.

```
<a href="about-page.html" target="_blank">YOUR LINK TEXT HERE</a>
```

To specify that the link should be opened in the same tab.

```
<a href="about-page.html" target="_self">YOUR LINK TEXT HERE</a>
```

Other attributes can be used to communicate different ways a link should be opened:

- `_blank` – Open in a new tab.
- `_self` – Open in the same tab.
- `_parent` – Open in the parent frame
- `_top` – Open in the full body of the window
- `framename` – Open in a named frame

You can also have an image point to a link.

Example

```
    <a href="https://websitesetup.org">
        <img src="heisaboy.jpg" alt="He is a boy">
    </a>
```

This tells the browser to display an image pulled from the image file “heisaboy.jpg” and have it linked to `https://websitesetup.org`.

The “alt” attribute represents what should be shown if the image cannot be displayed (due to browser settings or other stuff that prevents images from showing).

Using Images

Images are added to an HTML document with the `<img>` tag. The `<img>` tag is empty, so it doesn't need to be closed.

Example

```
<img src="heisaboy.jpg" alt="He is a boy">
```

This is a basic example of telling the browser to display an image titled "heisaboy.jpg" pulled from the same directory as the HTML document.

The "alt" attribute tells the browser to show a text (known as an "alternative text") called "He is a boy" if, for some reason the browser or internet settings prevents the browser from displaying images. If you want to pull an image in another directory or on an external site, the full address/path needs to be specified.

Going beyond the basics, we could also use other attributes to customize the image better. For example, the "style" attribute can be used to specify the width, the height, or both.

Example

```
<img src="IMAGE PATH" alt="YOUR ALTERNATIVE TEXT" style="width:Xpx;height:Ypx;">
```

(the values "Xpx" and "Ypx" should be replaced with the actual values in "px" (e.g. "20px") you want for your image.)

Exercise 2

Work on the HTML file you previoulsy created

Make ordered and unordered lists of the things you love to do, to eat, etc ... Don't forget to style it a bit

Create also links to your social media profile

Bonus: Use image and icons to make these links more userfriendly

Creating Tables

As you get deeper into HTML, you will want to learn various ways to present information in a more organized way. One such way is through the use of tables.

Tables are created with the `<table>` tag.

- Each header in a table is specified with the `<th>` ("table header") tag
- while each row is specified with the `<tr>` ("table row") tag.
- The table data is then indicated with the `<td>` tag.

Example

```
<table>
    <tr>
        <th>Table Header 1</th>
        <th>Table Header 2</th>
        <th>Table Header 3</th>
    </tr>
    <tr>
        <td>Input 1 under header 1</td>
        <td>Input 1 under header 2</td>
        <td>Input 1 under header 3</td>
    </tr>
    <tr>
        <td>Input 2 under header 1</td>
        <td>Input 2 under header 2</td>
        <td>Input 2 under header 3</td>
    </tr>
    <tr>
        <td>Input 3 under header 1</td>
        <td>Input 3 under header 2</td>
        <td>Input 3 under header 3</td>
    </tr>
</table>
```

This gives us something like this:

Table Header 1	**Table Header 2**	**Table Header 3**
Input 1 under header 1	Input 1 under header 2	Input 1 under header 3
Input 2 under header 1	Input 2 under header 2	Input 2 under header 3
Input 3 under header 1	Input 2 under header 2	Input 3 under header 3

Here, the table is styled with borders, set horizontal or vertical alignment for the table's content, introduce dividers, padding, etc. However, you will need to learn some CSS to do stuff like this. We're keeping this tutorial basic and won't get into that in this article.

Quotations In HTML

There are different types of quotations, and other elements represent these quotations.

For a basic quotation, here's an example:

```
<p>This is a sample. And <q>Here is our quote</q></p>
```

This comes out as:

This is a sample. And *"Here is our quote"*

When you use the `<q>` tag, quotation marks are automatically added to what is enclosed within the tag.

You can also use blockquotes, which is done with the `<blockquote>` tag.

Example

```
<p>
    <blockquote>
        This is a sample.
        And Here is our blockquote. In this example, we try to demonstrate how to format text to in
        This is different from ordinary quotes, in that the actual "quote" symbol may or may not be
        CSS styling, but the text is highlighted.
    </blockquote>
</p>
```

It will look like this:

> *This is a sample. And Here is our blockquote. In this example, we try to demonstrate how to format text to indicate a blockquote in HTML. This is different from ordinary quotes, in that the actual "quote" symbol may or may not be added depending on CSS styling, but the text is highlighted*

Using Comments In HTML

When doing any form of coding, it is vital to learn how to include comments in your code. Comments are used to make your code more organized. You can include a reminder to yourself or a note to others to make things easier.

Initially, it might not seem important, but when you start writing hundreds or thousands of lines of code, and when things seem to start to get complicated, comments will come in handy.

Comments won't be shown to the viewer by the browser. Remember that they can be seen in the source code.

Comments can also be used anywhere in the code. They won't change the function of your code in any way.

You can include a comment by opening a bracket, including a double hyphen, adding your comment, including another double hyphen, and then closing your bracket.

Example

```
<!-- This is a comment used to indicate information I want to remember or that I want others to tak
```

It is also worth noting that, especially when debugging, you can comment out lines of code. This way, the code remains in the document but does not function since you have commented it.

Example

```
<!-- <b>He is a boy</b> -->
```

While "He is a boy" is supposed to come out as bold, we have commented it, and the browser would ignore it. In this scenario, as far as the browser is concerned, we have nothing.

This could also be used this way:

```
<b>He <!-- is a --> boy</b>
```

which shows **He boy**.

This is because we commented out "is a". We're just trying to show you that practically anything, within or outside an element, can be commented out. You can include a note in your comment for reference – it doesn't change the nature of what is displayed.

Exercise 3

Work on the HTML file you previously created

Create a table with 3 columns and 4 rows. The first row should be the headers (ie. the title of each column), The first column should display 3 projects that you would like to create here at Developers Institute, the second column should display your 3 best skills in life, and the third column should display your 3 hobbies.

Bonus: Try to style the table with CSS (Hint: Check out "Inline CSS Table Borders" on Google)

Display a quote that represents you the best

Using IFrames In HTML

In an HTML document, an iFrame ("inline frame") can be used to embed a file in the current document. In essence, you can use it to display another website or content from another web page inside the current one.

Example

```
<iframe src="https://websitesetup.org"></iframe>
```

With the above code, I tell the browser to display the content of `https://websitesetup.org` in a frame directly inside this page.

Iframes can be used to display a website, video, images, or any other form of content.

The code for this is simple:

```
<iframe src="URL"></iframe>
```

(simply replace "URL" with the link to the website/content you want to be displayed.)

You can further customize to specify specific values, e.g., your iframe's height and width.

Example

```
<iframe src="URL" height="350px" width="400px"></iframe>
```

We tell the browser to display the iframe using a height of 350px and a width of 400px.

HTML Forms

HTML Forms are required to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.

You can see HTML forms on (almost) every website, as the "sign-in" section.

Here is a simple form example:

```
<form>
    <label>Username:</label><input type="text">
    <label>Password:</label><input type="password">
    <input type="submit" value="Submit">
</form>
```

Defining A Form

To define a form, use the `<form></form>` element.

The Input Element

This is the most commonly used element within HTML forms.

The `<input>` tag allows you to specify various user input fields, depending on the `type` attribute. An input element can be of type:

- text field
- password field
- checkbox
- radio button
- submit button
- reset button
- file select box

And a lot more!

To change the type of a field, pass the `type="<field_type>"` as attribute in the `<input>` element.

You can specify the `size` of the field by passing it in the attributes.

The Label Element

This element is created to define the *label* of a `<input>` element.

Conclusion

In conclusion, HTML is more complicated than what you have in this guide. Do not give up hope – if you can grasp the basic concepts in this guide, you are up to a good start. It's like knowing the ABC's of a language, and everything else becomes much easier.

If you went through this tutorial intending to create a stunning website, we're sorry to disappoint – it just doesn't work that way. This is a basic HTML tutorial. It's here to help you develop a foundation you can build upon. You can be sure that it will take significantly more effort – and a lot of development time, often using a combination of languages – to create an amazing website.

Back to Top

Feedback

Tell us what you thought about the chapter: HTML5