TASK

HTML I

Visit our website

Introduction

Welcome to The First HTML Task!

This is the start of your journey with Hyperion! Welcome! In this task, you are going to jump right into creating a webpage using HTML. Once you've seen how simple it is to create a webpage you will take a step back, in the next task, to see how what you have learned in this task, fits in with the bigger picture. In the next task, you will learn how the web works.

For now though, let's get stuck into learning some basics of one of the core technologies that make the web work: HTML! After completing this task, you will understand the basics of HTML and be able to write a basic HTML page with titles, headers, and paragraphs. There are MANY different types of elements and tags in HTML, some of the most frequently used ones will be covered in this task.

Your mentor will be looking forward to reviewing your work and helping to clarify your understanding once you submit your work at the end of this task.

Get in touch

Connect for support

Remember that with our courses, you're not alone! You can contact your mentor to get support on any aspect of your course.

The best way to get help is to login to **www.hyperiondev.com/portal** to start a chat with your mentor. You can also schedule a call or get support via email.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

A note from the

Hyperion Team

Remember that with our courses, you're not alone! To become a competent software developer, it is important to know where to get help when you get stuck.

While you are enrolled with Hyperion, the best way to seek help is to drop your mentor a line in the comments.txt file in your Dropbox folder. You can also log into www.hyperiondev.com/support to start a chat with your mentor, or schedule a 1:1 call with them.

Hyperion knows what you need and this is why we have a Slack group for students of our courses to assist each other. Also, our blog is a great place to find detailed articles and tutorials on concepts into which you may want to dig deeper. Our blog is updated frequently, so be sure to check back from time to time for new articles or subscribe to updates through our Facebook page.

It is also good to be familiar with external resources that are useful. Stack Overflow is a great resource that software developers around the world use to get help. For this level of this Bootcamp, you are also likely to find resources on the MDN web docs site useful. Here you can find resources that provide extra information about HTML, CSS and JavaScript.

BEFORE YOU BEGIN

You are about to dive into learning the basics needed to create a webpage using HTML. Before you begin though, please make sure that you have installed and set up the text editor you will be using for web development. You could create a webpage using any text editor (like Notepad or TextEdit) and saving the file with the extension '.html'. Using a text editor that is designed to support web development (such as Sublime Text) will make your life much easier in the long run though. As you will see throughout this course, it is vital to get things just right when you are a developer. A text editor designed to support software development helps with this by assisting you to identify and correct errors. More on this later. For now, please make sure that you have followed the instructions in 'Setting up your text editor.pdf' before you proceed with the rest of this task.

INTRODUCTION TO HTML

HTML stands for Hypertext Markup Language. It is a language that we use to write files that tell the browser how to lay out the text and images on a page. There is nothing special about HTML pages. They are essentially text files with the special file extension: '.html'. Inside this simple text file, we use HTML *tags* to define how the page must be structured.

HTML TAGS

Every language has a set of symbols used to construct "sentences." These symbols in HTML are called ***tags***. They are placed on the left and the right of the element you want to markup, so as to wrap around the element.

For example:

```
<opening tag>Some text here.</closing tag>
```

This is the general pattern that we follow for all tags in HTML. There are a few exceptions, which we will discuss later. The words 'opening tag' and 'closing tag' are just placeholders we use to illustrate the pattern. Instead of those words, we are going to use special keywords, or elements, that actually modify the appearance of our webpage.

Note that the opening and closing tags are not the same. The opening tag consists of an opening angled bracket, <, the name of the element, and a closing angled bracket, >. The closing tag consists of an opening angled bracket, <, a forward slash, /, then the name of the tag, and finally the closing angled bracket, >. You must adhere to this structure at all times. Remember to use a forward slash, /, and not a backslash, \.

```
<!DOCTYPE html>

<html>
<head>
        <title>My first web page!</title>
</head>

<body>
        <p>I am learning to develop a dynamic web application.</p>
</body>
</html>
```

Example of HTML in a simple text file.

The HTML tags indicate to the browser what sort of structure the content is contained in. Note that HTML does not include the *style* of the content (e.g. font, colour, size, etc.), which is done using CSS (Cascading Style Scripts), but only the structure and content itself.

HTML ELEMENTS

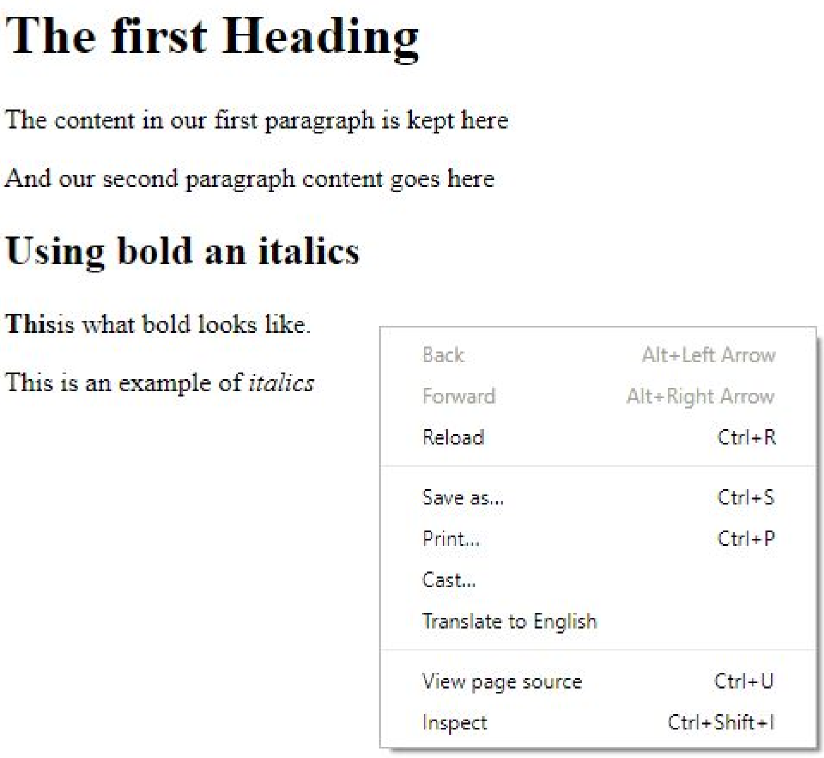
An element usually consists of an opening tag (<element_name>), a closing tag (</element_name>), which contain the element's name surrounded by angle brackets, and the content in between:
<element_name>...content...</element_name>

Example of HTML element:

```
<p>This element is going to result in this paragraph of text being displayed in the browser</p>
```

Try this:

- Double click on the file called 'example.html' (in the same Dropbox folder as this task) to open it in the browser.
- Examine how the HTML page renders in the browser.
- Now, right-click in the browser and select the option 'View page source.'



The first Heading

The content in our first paragraph is kept here

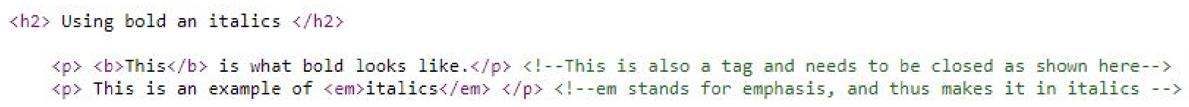
And our second paragraph content goes here

Using bold an italics

Thisis what bold looks like.

This is an example of *italics*

- You will see the HTML used to create this webpage that includes many HTML tags. For example, you will notice the tags shown below:



```
<h2> Using bold an italics </h2>

    <p> <b>This</b> is what bold looks like.</p> <!--This is also a tag and needs to be closed as shown here-->
    <p> This is an example of <em>italics</em> </p> <!--em stands for emphasis, and thus makes it in italics -->
```

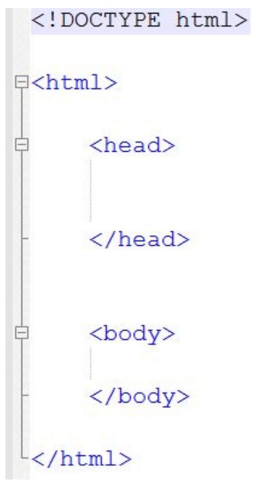
When the browser encounters the tag <h2> it knows to treat the information between the opening <h2> tag and the closing </h2> tag as a heading. Similarly, the browser will display the information between the tags <p> and </p> as a paragraph of text.

- You will learn more about specific HTML elements soon.

BASIC LAYOUT / TEMPLATE OF AN HTML PAGE

A typical HTML document consists of a **doctype** which indicates which version of HTML to load, a **head** which contains metadata about the page and a **body** which contains the actual content.

A general layout template that you can use before even starting to worry about what sort of content you want to display is set out below:



```
<!DOCTYPE html>

<html>

    <head>


    </head>


    <body>

    </body>

</html>
```

The Doctype is indicated at the top of the page and when typing 'html' it defaults it to HTML5. This is one of the only elements that does not need a closing tag. Note that throughout HTML, capitalisation is very important.

Next, we define what the content is to follow within the <html> tags (note the closing tag at the bottom). Within this <html> element, we introduce two other elements, namely <head> and <body>. Notice that although each of the tags is located on a separate line, we still have **opening tags** matching their **corresponding closing tags**. Notice how the <html> tag wraps around its contents. We use a nested order to structure tags on our web page; this means that tags are contained within other tags, which may themselves contain more tags. Above, the <html> tag *contains* the <head> and <body> tags. It is important to understand how elements are nested because one of the **most frequent mistakes that students make with HTML is getting the order all mixed up**. For example, it would be wrong to have a closing body tag (</body>) after a closing html tag (</html>) because the body element should be completely contained or nested within the <html> element. It should also be noted that white space is ignored by the browser, so you can lay out the physical spacing of the elements as you please.

ATTRIBUTES

Attributes are things that describe the objects created by HTML elements. For example, *<p>This element is going to result in this paragraph of text being displayed in the browser</p>*
would result in a paragraph that contains text. This paragraph can be described using various attributes including align, font-size etc.

Consider the following:

```
<title id="myTitle">My first web page</title>
```

In this case, the element is of type 'title'. Next, we have an 'id' which is an attribute of the element (title), and has a value of "myTitle". Attributes like this are used mainly for the purposes of CSS and JS (JavaScript) and will be covered later in the course. Then there is a closing '>' which indicates that you have finished defining the attributes of the element.

COMMON HTML ELEMENTS

We have already encountered some commonly used elements that are used to create most web pages. Some of these (and some new elements) are summarised below:

- A piece of metadata that should be included in all web page is the **<title>** element.
  The <title> element:
  - defines a title in the browser tab
  - provides a title for the page when it is added to favorites
  - displays a title for the page in search engine results

  As noted before, metadata should be contained in the <head> of the HTML document.
  E.g. of a title element:
  <head>
  <title>Portfolio</title>
  </head>

- As you would with a word document, use **headings** to show the structure of your web page. This is important because search engines use the headings to index the structure and content of your web pages. There are 6 heading elements you can use: <h1> to <h6> where <h1> element is used for the most important headings and <h6> for the least important.
  E.g. of a heading element:
  <h1>Online Portfolio of work</h1>
  <h2>About me</h2>

- Add paragraphs of text using the **<p>** element as follows:
  <p>This is an example of a paragraph. Paragraphs usually contain more text than headings and are not used by search engines to structure the content of your web page. </p>

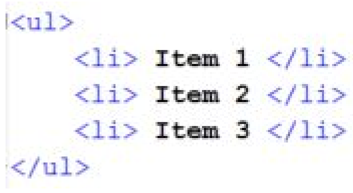
- **Line breaks.** To do the equivalent of pressing enter to get a line break between text, use the <br> element. This element does not have a matching closing tag. This should make sense because there is no content that you could put within a <br> element. Elements like this, with no content or matching closing tags, are known as void elements.

- **Horizontal rule.** This is another void element. By adding the HTML element <hr> to your web page you will create a horizontal rule.

- **Lists.** Lists can either be ordered lists <ol> or unordered lists <ul>. Ordered simply means that the list is numbered, i.e. 1, 2, 3, etc. and unordered is in the form of bullet points. In lists keeping track of how far you are with nesting of the various elements is VERY important. We highly recommend that you use indentations to keep track of what elements fall under what other elements. Remember that indentation and "whitespace" does not affect the layout of the elements on the web page.

  <u>Unordered Lists</u>

  In an unordered list, as with most elements, we have to open and close the tags. Within this element, we now want to display some content in our list. This content is inputted in the form of *list items* and thus have the tag <li>. So, to create an unordered list with three items in it, we would write it out as follows:



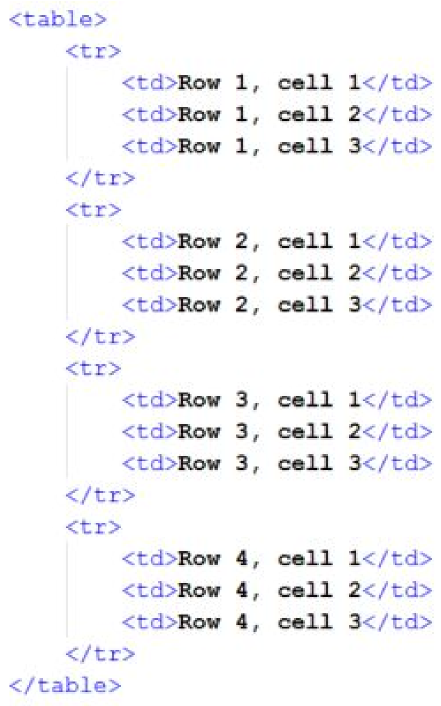
```
<ul>
    <li> Item 1 </li>
    <li> Item 2 </li>
    <li> Item 3 </li>
</ul>
```

Note how the indentation makes the entire structure a lot easier to read. The list items, as seen above, are also closed at the end of the content to indicate to the browser where the content of that specific item ends.

<u>Ordered Lists</u>

Ordered lists work almost the same as unordered lists, except that you use the tag, <ol>. You input list items in the same way as shown above. Instead of showing bullet points, these list items are numbered.

- **Tables.** Tables work in a similar fashion to lists in terms of nesting elements. First, define the fact that it's a table using the <table> tag, and then manually enter the data into the rows. Have a look at the example below:



```
<table>
    <tr>
        <td>Row 1, cell 1</td>
        <td>Row 1, cell 2</td>
        <td>Row 1, cell 3</td>
    </tr>
    <tr>
        <td>Row 2, cell 1</td>
        <td>Row 2, cell 2</td>
        <td>Row 2, cell 3</td>
    </tr>
    <tr>
        <td>Row 3, cell 1</td>
        <td>Row 3, cell 2</td>
        <td>Row 3, cell 3</td>
    </tr>
    <tr>
        <td>Row 4, cell 1</td>
        <td>Row 4, cell 2</td>
        <td>Row 4, cell 3</td>
    </tr>
</table>
```

  The table element is defined within the opening and closing tags. Immediately within these tags, there is a *table row* indicated by <tr> which also has a closing tag. Within that first table row, there is a <td> tag which indicates that there is *table data*. A table is shown in the "example2.html" file so that you can try and correlate what elements contribute to what visual appearance on the web page.

The most important elements for this course can be found in "example.html" and "example2.html".

A note from our coding mentor
Sabir

Code Comments

Comments are a means used by programmers to explain what they have done in their code where it might not be clear to someone reading the code. Even though they are written in a file containing code, comments are not executed by the computer. To add a comment to your HTML file, you need to use the following pattern:

```
<!-- Your comment goes here. -->
```

A comment comprises an opening sequence, <!--, followed by the text you want to include in your comment, then finally the ending sequence, -->. We will use comments in the example files to explain what the HTML tags we've included do.

HTML SYNTAX

As a web developer, you are going to learn many new languages. Each of these has their own rules which must be strictly followed in order for your instructions to be properly processed. The rules of a language are referred to as *syntax*. Examples of common HTML syntax errors include spelling the name of an element incorrectly or not closing tags properly or in the wrong order. You are bound to make mistakes that will violate these rules and that will cause problems when you try to view web pages in the browser! We all make syntax errors! Often! Being able to identify and correct these errors becomes easier with time and is an extremely important skill to develop.

To help you identify HTML syntax errors, copy and paste the HTML you want to check into this helpful tool.

A note from our coding mentor
Valerie

In the Dropbox folder for this task, you will notice additional reading material. All additional reading listed in this Bootcamp is optional. The additional reading for this task is an eBook entitled "HTML5 notes for professionals." It is written by the 'beautiful people of Stack Overflow'. The topics covered in this task correspond to chapters 1 - 5 and 7 - 9 of this book. Therefore, this book is a good resource if you would like further information about any of the HTML elements covered in this task.

Instructions

Open "example.html" and "example2.html" in Sublime Text and read through the comments before attempting these tasks.

Compulsory Task 1

Follow these steps:

- Use Sublime Text to create an HTML file called "firstPage.html" in this folder.
- Set out the basic document template.
- Make the title "My first web page with HyperionDev!"
- Make two headings ('h1's), with the content being "Hobbies" and "Goals". ["Goals" here refers to what you would like to get out of this Bootcamp/what you would like to be able to do by the end of this Bootcamp.]
- Each of these headings should have two subheadings of type h2 (total of four). List two items for your hobbies and goals respectively.

- Within these subheadings, create two paragraphs per h2 heading. In each of these paragraphs, mention something about the topic you're speaking about. You don't need to make this too long - a sentence is fine (unless you're feeling very creative!).
- Be sure to use italics and bold text in your page - you can choose where.
- Before submitting your code, check it with the HTML validator here.

Compulsory Task 2

Follow these steps:

- Create an HTML file called "tables.html" in this folder.
- Set out the basic document template, giving it a title and headings as you see fit.
- Create a table with 3 columns, with column names in bold. These names should be: "Topic", "Name of website" and "URL".
- Populate this table with the details of 3 resources that you have found on the web that provide useful advice for web developers. Feel free to use resources referred to in this task.
- Before submitting your code, check it with the HTML validator here.

Once you have completed the task in line with the instructions above, click the button below to request your mentor to review your work and provide feedback. If you have any questions while attempting the task, leave your mentor a note on the comments.txt file in your Student Dropbox folder.

Completed the task(s)?

Ask your mentor review your work!

Review work

Things to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Sublime Text** may not be installed correctly. Please make sure that you have followed the instructions in 'Setting up your text editor.pdf.'

Rate us

Share your thoughts

Hyperion strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think the content of this task, or this course as a whole, can be improved or think we've done a good job?

Click here to share your thoughts anonymously.