



TASK

HTML II

Visit our website

Introduction

WELCOME TO THE SECOND HTML TASK!

In this task, some more HTML elements will be introduced. We will be looking at using links. This includes linking to other places within your website, linking to other websites, and importing images. All of these links can be used in the HTML elements that we have already covered. You will also learn to create HTML forms. You will also learn to start using coding best practice to make sure that your code not only works, but is written in a professional manner that is easy to read and maintain.



Get in touch
Connect for support

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

The best way to get help is to login to Discord at <https://discord.com/invite/hyperdev> where our specialist team is ready to support you.

Our team 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!



LINKS

You can add links to your web page as follows:

```
<a href="url" target="_blank">link text</a>
```

The `<a>` element is used to add all links on a web page (the `a` stands for anchor, and the `href` for hypertext reference, so the element is used to anchor a link to the link text you provide). Using this element you can link to other pages within your own website and to external web pages, and enable users to send an email, using the format:

```
<a href="mailto:exampleperson@example.com">link text</a>
```

Linking to other places on your web page

Often on your web page, you will want your users to be able to click on a link which will then take them to another part of the same page. Think about the “back to the top” button — you click on this and you are suddenly viewing the top of the page again!

In order to do this, we need to use *ID* attributes. An ID is used to uniquely identify one of your HTML elements, such as a paragraph, heading or table. Then we can use the link tag to make the text or image a link that the user clicks on to take them to whichever position we choose.

An ID can be assigned to any of your elements, and is done as follows:

```
<h1 id="theHeading">My first web page</h1>
```

Notice how the attribute `id` is within the opening tag.

Now that we have this heading, we can look at how to reference it within our text. We use the `<a>` tag which shows which address we are using. To reference a structure with an ID, we need to precede the value assigned to the `id` attribute with a “#”, otherwise the browser will think you are looking for a website.

```
<h1 id="theHeading">My first web page</h1>  
<a href="#theHeading">Back to top</a>
```

Consider the **example.html** file that contains the elements shown above. If you open it you will see that it will make the text “Back to top” look like a hyperlink (this defaults to blue and underlined). When the link text is clicked, it will take you to the Heading with the **id** “theHeading”.

Linking to other web pages

Similarly, we can link to another page. This is done as follows:

```
<a href="http://www.hyperiondev.com">This is a cool place!</a>
```

The **http://** in front of the address lets the browser know that you are linking to an external website rather than a file on your system.

That’s not to say, however, that you can only link through text! All the content that is between the **<a>** tags is what is to be clicked on in order to get to the destination address.

With the link specified above, if you click on the link it will change the window you’re currently on. What if you wanted to open the destination address of a link in a new tab? Well, in that case you can add an attribute to the link tag called *target* which specifies how the link should be opened, e.g. in the same window, new browser instance, or new tab. To modifying the link as follows to open in a new tab:

```
<a target="_blank" href="http://www.hyperiondev.com">This is a cool place!</a>
```

IMAGES

We can add images to our website using the **** element as shown below:

```
  

```

There are a few things to note about the **** element.

- Unlike most of the other elements we have explored so far, **the element doesn’t have a closing tag.**

- The `` element has a number of attributes that can describe it. These include:
 - **src=** The `src` attribute gives the path to where the image can be found or the *source* of the image.
 - **alt=** The `alt` attribute defines the *alternate text* that will be displayed if the image won't display.
 - Intuitively, the **height** and **width** attributes define the height and width of the image.
- The **src** attribute can point to a URL or a file location. In the example above, the first image uses a URL as the source of an image. The second image shows how the `src` attribute is defined to display an image named **image1.jpg** that is stored in a folder named *images* that resides in the same folder as your web page.



Take note:

When adding images to your web page, it is important to remember that this page may be viewed on many different devices with widely differing screen sizes, resolutions etc. You want the images to look good independent of the device that is used to

view the page. Thus responsive images (images that work well on devices with widely differing screen sizes and resolutions) are important. To see how to create responsive images, look [here](#) and read Chapter 15 of “HTML5 notes for professionals” (additional reading in the Dropbox folder of the first task).

SPOT CHECK 1

Let's see what you can remember from this section.

1. What is wrong with the following lines of code:
 - a. `This is a cool place!`
 - b. `<h1 id="theHeading">My first web page
Back to top`
2. What is the `alt` attribute used for?

HTML FORMS

A dynamic website is driven by user interaction. In order for users to be able to interact with your website, you need to provide them with the means to enter the information that is to be used and displayed on the pages. Forms are the instruments that we use to allow users to enter data in HTML. Forms can be

structured in various ways; in fact, web designers often try to make them as cool as possible to encourage users to interact with the site. Here are some examples of different kinds of forms on the Web:

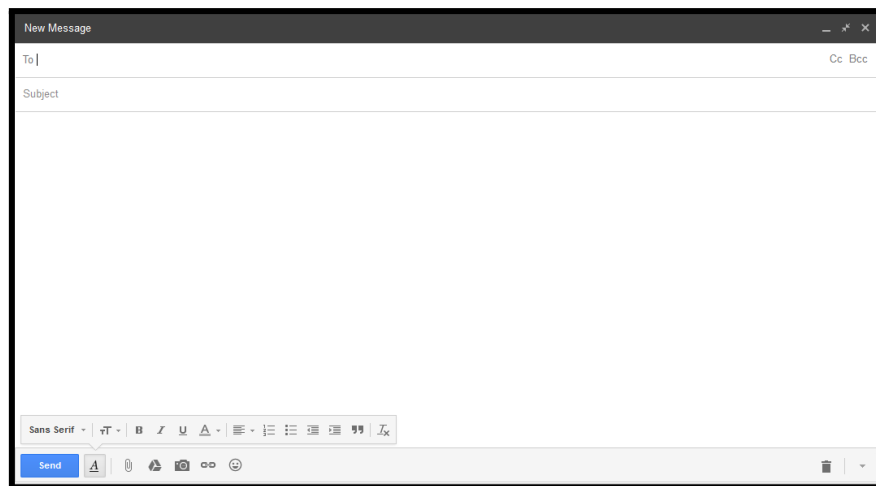
A screenshot of a Gmail 'New Message' form. The form has a dark header bar with the title 'New Message' and window controls. Below the header, there are fields for 'To', 'Cc', 'Bcc', and 'Subject'. The main body of the form is a large, empty text area. At the bottom, there is a rich text editor toolbar with options for font face (Sans Serif), font size, bold, italic, underline, link, unlink, bulleted list, numbered list, indent, outdent, and text color. A 'Send' button is located at the bottom left of the form.

Figure 1: Here's a sophisticated form from gmail (mail.google.com) - this is the popup text editor used to draft an email.

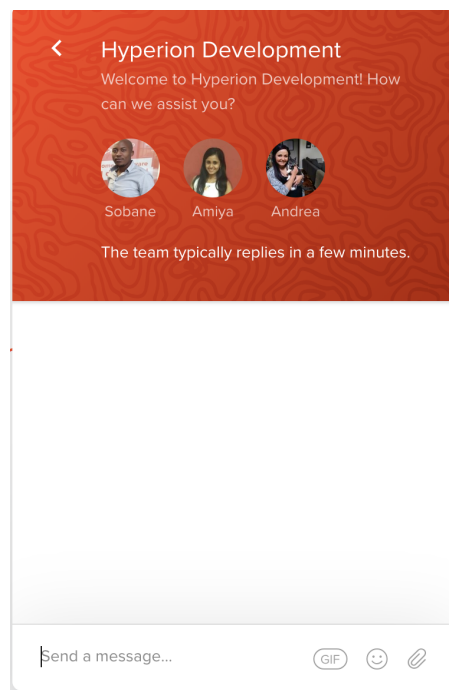
A screenshot of HyperionDev's chat box. The chat box has a red header bar with a back arrow, the title 'Hyperion Development', and a welcome message: 'Welcome to Hyperion Development! How can we assist you?'. Below the header, there are three circular profile pictures of team members: Sobane, Amiya, and Andrea. Below the profiles, it says 'The team typically replies in a few minutes.' The main body of the chat box is a large, empty text area. At the bottom, there is a text input field with the placeholder 'Send a message...' and three icons: a GIF icon, a smiley face icon, and a paper plane icon.

Figure 2: HyperionDev's (www.hyperiondev.com) chat box is actually a very sophisticated form, but a form nonetheless

CREATING A FORM

We won't begin with complex forms like the ones you see above. First, we're going to build a simple form and focus on investigating some of the components of a

form. At this stage, our forms won't be functional. You will create forms that function correctly later when you learn JavaScript.

```
<form>
  <label>First Name:</label>
  <input type="text"><br>
  <label>Surname:</label>
  <input type="text"><br>
  <label>Gender:</label>
  <select>
    <option value="male">Male</option>
    <option value="female">Female</option>
    <option value="other">Other</option>
  </select>
  <label>Age:</label>
  <input type="text">
</form>
```

In the example above we create a form to capture our user's biographical information. It captures the following information:

- First Name
- Surname
- Gender
- Age

We expect the user to enter text for their name and surname. We, therefore, use the **input** element. This element has a **type** attribute with the **text** property assigned to it. This displays text boxes in the browser into which users can type input. We add labels to tell our visitors what information we want them to enter into the boxes.

The **select** element is used to create a drop-down menu that users can select from instead of typing out their gender.

To see a list of other HTML input types, look [here](#) and Chapter 17 of “HTML5 notes for professionals” (additional reading in the Dropbox folder of the first task). Look at **example2.html** to see an example of an HTML form.

READABILITY

As you start to create HTML pages with more elements, it becomes increasingly important to make sure that your HTML is easy to read. In software development,

readability is an important principle! Code and markup that is easy to read are easier to debug and maintain than code or markup that is not easy to read.

Indenting your HTML is an important way of improving the readability of your code. For example, consider the HTML below:

```
<!DOCTYPE html><html><head>
<title>My first web page</title>
</head>
<body>
<form><label>First Name:</label><input type="text"><br>
<label>Surname:</label><input type="text"><br>
<label>Gender:</label>
<select><option value="male">Male</option>
<option value="female">Female</option>
<option value="other">Other</option></select>
<label>Age:</label><input type="text">
</form></body></html>
```

The above is perfectly correct HTML that will render properly in the browser but it is certainly not as easy to read and understand as the code below which is properly indented:

```
<!DOCTYPE html>
<html>

<head>
  <title>My first web page</title>
  <!--This is a comment by the way -->
</head>

<body>
  <form>
    <label>First Name:</label>
    <input type="text"><br>
    <label>Surname:</label>
    <input type="text"><br>
    <label>Gender:</label>
    <select>
      <option value="male">Male</option>
      <option value="female">Female</option>
      <option value="other">Other</option>
    </select>
    <label>Age:</label>
    <input type="text">
```



```
</form>
</body>

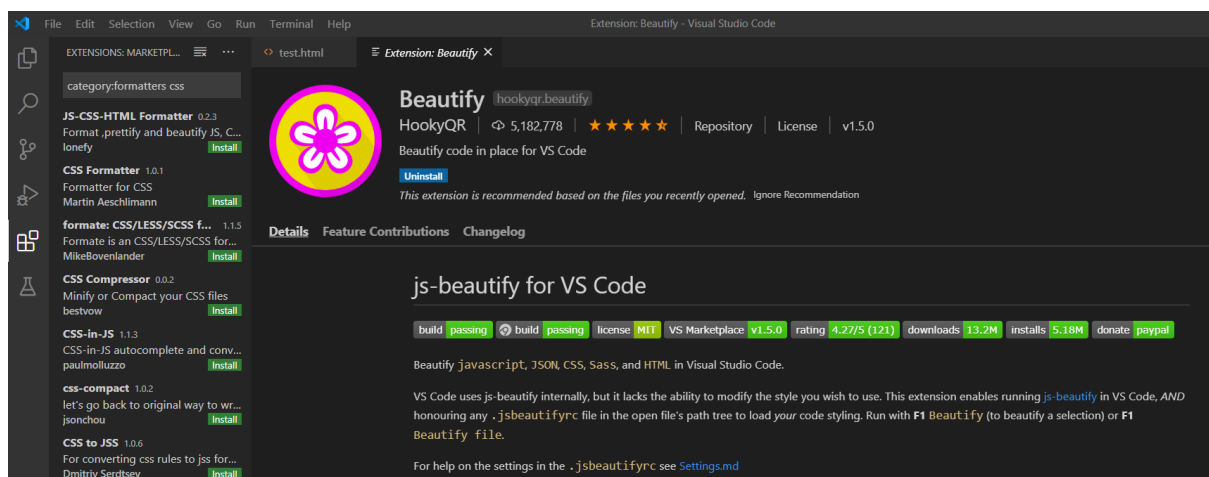
</html>
```

As you can see above, indentation should be used to show which HTML elements are nested within which other HTML elements. As shown above, all the other elements are nested within the `<html>` element.

There are several tools that can help you improve the readability of your HTML. Since you will soon be working with CSS and JavaScript too, we highly recommend that you use **Beautify**.

You can add this package to VSCode by doing the following:

1. In VSCode, go to the extensions icon on the left hand side and type in 'Beautify'.
2. Click 'Install'.



3. Now, when you need to format your code, go to your code and right-click. Select 'Command Palette...' → 'Beautify file' and select the file type of your file (HTML, CSS or JavaScript).
4. That's it! Your code is beautifully formatted!

SPOT CHECK 2

Let's see what you can remember from this section.

1. What HTML element do you need to use to create a form?
2. What is the mandatory attribute that an input element needs?
3. What does the select element do?
4. Why do good programmers strive to create readable code?

Instructions

Open **example.html** and **example2.html** in VSCode and read through the comments before attempting these tasks.

Compulsory Task

Follow these steps:

- Open the **WebFundamentals.html** file which you created in the previous task.
- If you have not yet done so, make sure that your webpage contains appropriate headings and paragraphs.
- Add an appropriate heading and paragraph to your webpage in which you explain in your own words what you should think about before adding images to your website. Note the information [here](#) in the **warning** box.
- Add at least 3 relevant pictures (either from your PC or online) to your webpage. Don't worry about the sizes and position for now - we will deal with that in the CSS section.
- Add a heading called "Acknowledgements" to your webpage. Under this heading list all the websites that you have used to access images or information for this task. Add links to each of the websites you have referenced.
- Add a "Back to top" link at the bottom of your webpage that will return the user to the top of the webpage when clicked.
- Place the HyperionDev logo (URL given in the example.html file) after your last paragraph. Link this picture so that, when you click on it, it takes you to the HyperionDev website. Hint: Think about what the content in the link that users should click on might be. (Normally it would be text, but what is it now?).

- Before submitting your code, check it with the HTML validator [here](#).

If you are having any difficulties, please feel free to contact our specialist team [on Discord](#) for support.

Completed the task(s)?

Ask an expert to review your work!

[Review work](#)

Things to look out for:

1. Make sure that you have installed and set up all programs correctly. You have set up **Dropbox** correctly if you are reading this, but **Visual Studio Code** may not be installed correctly.



Rate us

Share your thoughts

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

Think that 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.



SPOT CHECK 1 ANSWERS:

1.
 - a. You need to use the **href** attribute to create a link, not the **src** attribute.
 - b. The **h1** (heading 1) and **a** (link) elements are both missing their closing tags: **</h1>** and ****
2. The **alt** attribute is used to show alternate text in place of an image if it does not show.

SPOT CHECK 2 ANSWERS:

1. The **form** element.
2. The **input** attribute.
3. The **select** element creates a dropdown menu where the user needs to select one option.
4. Readable code makes the code more understandable, more scalable, and more maintainable, especially as it gets more complicated and multiple programmers end up working on the same code.