Hyperiondev

TASK

CSS II

Visit our website

Introduction

Welcome to The Second CSS Task!

This task will look at combining the styles considered previously in order to create a slightly more complicated web page which will set you up for the Capstone Project where we will be building a fully functional website!

Get in touch

Connect for support

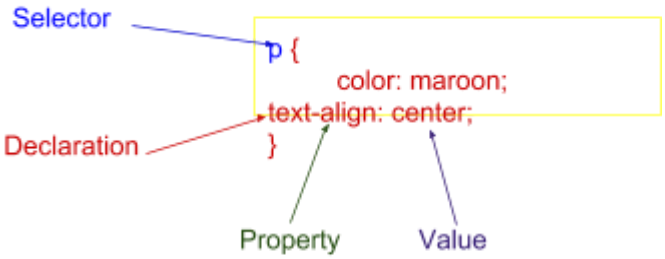
Remember that with our courses, you're not alone! You can contact your mentor to get support on any aspect of your course.

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!

CSS SELECTORS

As you learned in the previous task, CSS follows the following syntax:
A style sheet consists of a selector and a declaration.



- The **selector** indicates which HTML element you want to style.
- The **declaration** block contains one or more declarations separated by semicolons. A declaration always ends with a semicolon and is surrounded by curly braces.
- Each declaration includes a **property** and a **value**, separated by a colon.

We have thus far only worked with one type of selector. The CSS below should look familiar. Here the selector is always an *element*.

```
p {
    color: red;
    font-family: Arial, Helvetica;
    background-color: blue;
    }

body{
    text-align: center;
}
```

However, you can also use *class* and *ID selectors*. A class selector is used when the selector describes the rule for all elements that have a *class attribute with the same name defined*. An ID selector describes the style of an element with a specific id attribute defined.

Open *example.html*. Notice that in this file we have several elements for which either the class or id attribute has been defined. For example, notice how there are several <a> elements that have the class attribute set to “moreButton” as shown below.

```
<a href="" class="moreButton">More</a>
```

However, there are also <a> elements that do not have a class attribute specified.

```
<li><a href="Contact.html">Contact</a></li>
```

Therefore, instead of creating a CSS with an element selector (‘a’) we could create a rule that is specific to a class selector. See an example of this below. When you use

a class selector, the name of the class will always be preceded by a dot, ' . '. The style rule below will cause all elements where the class attribute has been set to 'class="moreButton"' to have bold text.

```
.moreButton {
    font-weight: bold;
}
```

In a similar manner, you could create a stylesheet that uses ID selectors. ID selectors start with a hash, '#', as is shown in the example below.

```
#mainNav {
    font-family: cursive;
}
```

The rule above would cause the text of the element where the id attribute equals "mainNav" to be cursive.

Although you can have many elements that have a class attribute with the same value, an ID name must be unique in the document!

A note from our coding mentor

Masood

Readability

As a software developer, it is obviously important to be able to write code/markup that is correct and runs to produce the desired outcome. As a professional web developer, more than this is needed though. You also want to be sure that your code is written in such a way that it is easy to debug, update and maintain by yourself or by teams of developers. Readability is therefore very important. Your code/markup should be easy to read.

In this regard, note the following about the values you assign to class and id attributes: You can name a class whatever you like but it is best practice to use meaningful names. For example, "moreButton" is a good example of a name for a class because it is descriptive. "a1" would not be a good name for a class because it is not descriptive.

Also remember to use tools like HTML-CSS-JS Prettify to improve the readability of your CSS.

CASCADE: SPECIFICITY

Remember that the word “Cascade” in the term “Cascading Style Sheets” basically has to do with the order in which rules are applied.

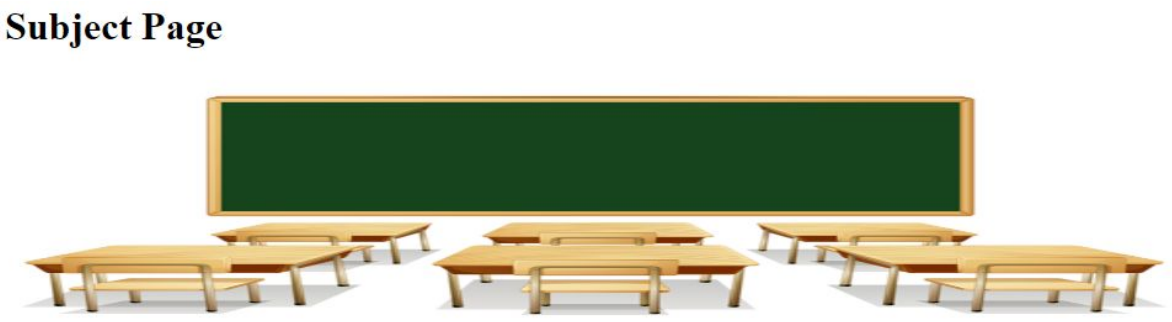
Another important rule to remember is that *the more specific a rule is, the higher its precedence*. For example, in a stylesheet that uses element selectors, class selectors and ID selectors, *element selectors are the least specific* (because they could match the most elements in a page) whereas ID selectors are the most specific. Therefore, ID selectors will be applied over class selectors and element selectors. See *example2.html* to clarify this concept more.

WHAT WILL WE BE DOING?

In this task, we are going to style a page which advertises different subjects to take at a high school level. We recommend using external CSS wherever possible, but you may use inline and internal as needed. All of the required HTML elements have been provided for you, but feel free to add more if you so wish!

REQUIREMENTS

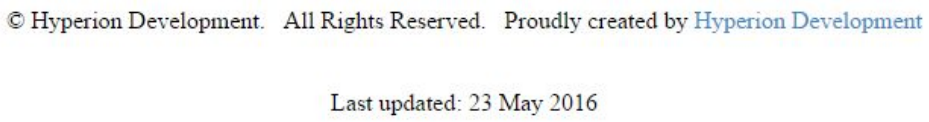
There will be a main heading and an image at the top of the page, before the navigation bar, as shown below:



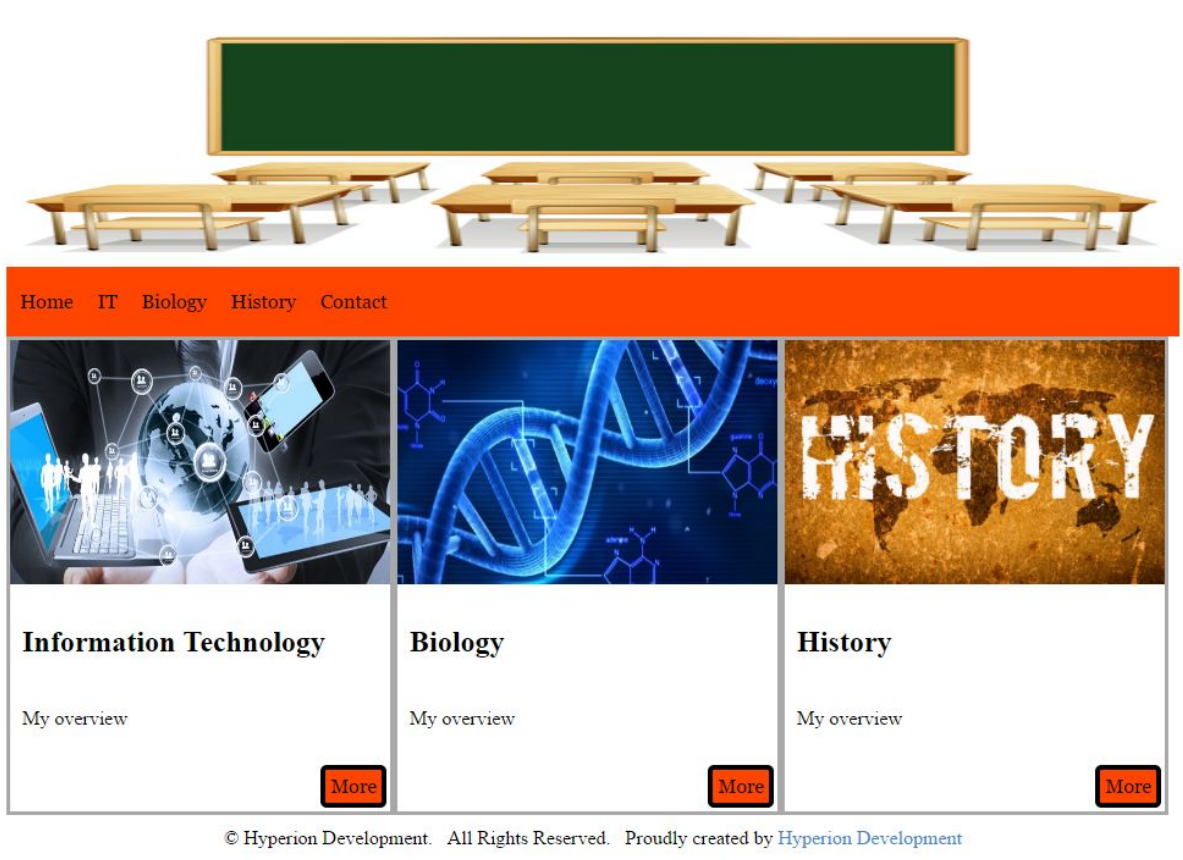
The webpage will have a navigation bar which will have links to the homepage, one per subject and a contact page. These do not need to be functional, but the supposed links should still exist.

We will offer three subjects to choose from, namely *Information Technology*, *Biology* and *History*. These subjects will be encapsulated in *article* elements. Each article will consist of a heading (the name of the subject), a picture relating to the subject, and a small paragraph which gives a brief overview of the subject. There will be a details button which will react when a mouse hovers over it, but will not provide extra functionality.

In the footer element, there will be all the extra information which is usually contained in web pages.



You have been provided with all of the necessary HTML and images (already linked). After applying the various CSS rules outlined in the instructions, your webpage should look something like the one below:



Open the HTML file to view what the page looks like without CSS having been applied!

A note from our coding mentor

Ridhaa

Here are two more cool CSS tricks:

1. *If you would like to apply a certain style rule to an element when it is in a certain state (e.g. if you hover over it) you can do this as shown below:*

```
/* Any button over which the user's pointer is hovering */
button:hover {
  color: blue;
}
```

In this example, 'hover' is a pseudo-class (a keyword that describes the state of the selector). To see a list of pseudo-classes, see here.

2. *If you would like to apply a certain rule to an element that is a descendant (child) of another element, you can do so as shown below:*

```
li li {
  list-style-type: circle;
}
```

The rule above will make all list items that are descendants of other list items have a circle as a bullet point. To see more about using a descendant combinator, see here.

A note from our coding mentor
Jared

Before you attempt this task, take a quick look at the 'example.html' file. Notice that this file uses <article> tags. <article> elements are new sectioning elements used with HTML5. HTML5 is the latest evolution of the standard that defines HTML. Learn more about HTML5 here. Find out more about the <article> element and how it compares to elements here.

Getting the layout of elements correct is one of the trickiest aspects of CSS. You will be required to do some research in this regard to complete this task successfully. Feel free to use whatever resources you like but this Introduction to CSS layout and this CSS layout cookbook are excellent places to find help.

Instructions

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

Compulsory Task

Follow these steps:

- Create a CSS file and link it in your HTML file (*task.html*).

The Heading and Navigation

- Centre your whole page, using 50% as your width. (Hint: look at margin types).
- Set your main picture (of the classroom) to span across the screen, and make it align with the rest of your page.
- Make your unordered list have no bullets, be aligned in the middle, and have a different font (of your choice). (Hint: look at list-style-type). Give them no padding.
- Set the background colour of your navigation to “orangered”, and give it 3px padding. (See additional reading ‘CSS notes for professionals’ Chapter 9: Padding)
- When links on the page are hovered over, the text should change to white. (Hint: use a pseudo-class)
- All links should have no text-decoration and be black.

The Articles

- Each article should be floated left, have a width of 32.4%, height of auto and a border which is dark grey and solid. (See additional reading ‘CSS notes for professionals’ chapter 10 and 14)
- The articles should have an additional property of display:inline-block.

- The paragraphs in the articles should be justified and have a padding of 10 px. (Hint: look at html children).

- The pictures should take up the full width of the articles and have a height of 200px. This will distort the images slightly, but don't worry about this.

- When the pictures are hovered over, the opacity should be reduced to half. (See additional reading 'CSS notes for professionals' Chapter 19: Opacity)

- The "more" buttons should look the same as the pics above (in terms of colours). This is left to you to work out.

- These buttons should be floated right, given padding and a border, as well as a margin of 3px.

- When these buttons are hovered over, the background colour should change to black and the text to orangered.

The Footer

- The text should be centered.

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.