Hyperiondev

TASK

CSS I

Visit our website

Introduction

WELCOME TO THE FIRST CSS TASK!

You are now able to create basic web pages using HTML. You have likely by now wanted to be able to change the colour/font of text or you may have wanted to change the size of an image or center it. This has not been possible so far but in this task, you will learn how to style your HTML elements. This task will introduce you to the basics of cascading style sheets (CSS) and how to format various HTML elements using “in-line” and “internal” as well as “external” approaches. There are many different ways to style an element and you are encouraged to do some research of your own. After all, it would be impossible to put a course together that covered every aspect of styling an element in CSS, since there are new ones being made almost every day. By the end of this task, you should be able to change the font, size, colour and position of titles, headers and paragraphs. You should also be comfortable with CSS in-line, internally, and externally.

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!

INTRODUCTION TO CSS

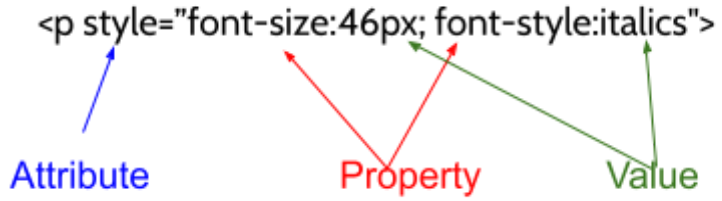
Cascading Style Sheets (CSS) is a language which is used to change the presentation and look of a particular document which has been written in a markup language, such as HTML. CSS is usually applied to web pages, but can also be used in other areas, such as XML documents.

INLINE STYLE

HTML **elements** are described using **attributes** and **properties**. You can style a web page by changing the properties of the elements that make up that webpage. For example, any text that you add to a web page has several properties that you can change. For example: font-family (Arial, Times New Roman etc), font-style (normal, italics etc) and font-size. An example of using the style attribute to change the font of an element is shown below:

```
<p style="font-size:46px;color:white;">
    This is the paragraph where I describe myself.
  </p>
```

Like all other attributes, the style attribute goes inside the element's beginning tag, right after the tag name. After specifying that you are changing the style attribute, you type =, and then, within double quotes, list the properties you want to change and after a colon specify the value for that property.



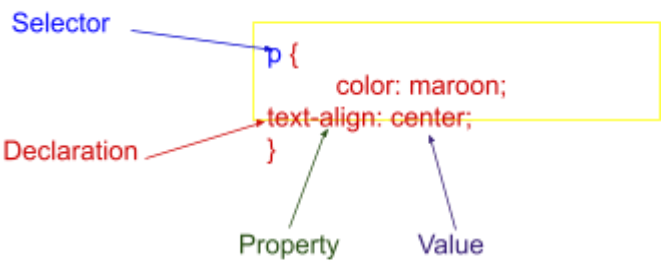
When you style an element individually by changing that element's properties, it is known as **inline styling**. Inline styling allows you to specify the style of an individual element in the line where that element is declared. What if you wanted to apply similar styles to all elements of a certain type. For example, what if you wanted to change the font of all paragraphs on your web page. You can do this by creating a CSS rule.

INTERNAL CSS

The example below shows how you can define a CSS rule in the *head* part of your HTML template. This is called internal CSS. The example below shows a CSS rule that will cause all paragraphs to be in the colour red and be of the font-family Arial. If the browser can't find Arial, then it will look for Helvetica. Paragraphs will also have a background colour of blue!

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

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.

See what other properties can be modified using CSS **here**.

You could use a combination of internal CSS (declared in the head of your HTML document) and inline style. How would the style rules apply? Essentially, the closer to the element the style is, the higher the precedence. For example, if you had the internal CSS rule shown in the image above in your page but you wanted one paragraph to be styled differently from the rest, you would simply use inline style for that one paragraph and that would overwrite the rule specified by the internal CSS.

EXTERNAL CSS

If your website consists of many HTML files, you are likely to want to be able to apply the same style rules to all the web pages. To accomplish this use external CSS instead of internal CSS. To do this, create a separate file with the extension .css. Within this file write all the style rules that you would like to specify. You can then

link this external CSS file to all the HTML files in which you would like the style rules applied. To link an external CSS file to a specific HTML file, do the following:

```
<link href = "examplesCSS.css" rel = "stylesteet" type = "text/css">
```

In the <head> part of your HTML create a reference to your CSS file so that the styles can be used in your web page. “href” refers to the name and path of your CSS file. In the example above, the file *exampleCSS.css* is in the same folder as the HTML page. “rel” says what sort of relation the file is to the HTML - i.e. the stylesheet. “type” tells the browser what sort of file it is and how to interpret it.

INTERNAL, EXTERNAL OR INLINE: WHICH APPROACH IS THE BEST?

If we were to include the CSS shown in the image below in our CSS file, the result would be the same as if it were in the <style> tags in the HTML page. So is it better to use internal or external CSS? Generally, it is ***better to use external CSS wherever possible***. Why? *Readability* is an important factor. Imagine trying to read through a whole bunch of different languages at once (CSS, HTML, JavaScript) all in one file. Rather separate them - it’s much easier to follow what’s happening, especially when you are building fancy websites where plenty of different styles are being used.

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

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

Another important reason to separate CSS from HTML files is to improve the *maintainability* of your website. If you wanted to update the look and feel of a website, this could easily be done by simply replacing the external CSS file if only external CSS is used for the website. Using external CSS also makes it easier to *debug* errors since all the CSS is in one place.
You may find though that it is necessary to use a combination of external, internal and inline style. In this case, it is important to understand the concept of cascade.

CASCADE

As we know, CSS stands for Cascading style sheets. You may have wondered why they are called *cascading* style sheets. Cascade basically has to do with the way in which rules are applied.

If your website contains external, internal and inline CSS, inline CSS overrides internal CSS rules and external CSS files. Internal CSS overrides external CSS rules. If there are conflicting rules regarding properties, properties override other properties, but entire rules don't override other rules. When several CSS rules match the same element, they are all applied to that element. Only after that are any conflicting properties evaluated to see which individual styles will win over others. See "Example 3" for more about this.

CSS VALIDATOR

As you have no doubt come to realise with HTML, as a developer it is very important to follow the syntax rules when developing websites. The same is true of CSS. You need to follow the rules for formatting your CSS rules exactly or unexpected errors will occur when you try to view your web page in the browser. Examples of common errors include spelling the name of an element incorrectly, not having matching opening and closing braces { } or leaving out semicolons, ";", or colons, ":", where they should be. 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 errors in your CSS, use this helpful **tool**.

Extra resource

The additional reading for this task includes two excellent resources:

1. An eBook entitled "HTML5 notes for professionals" by the 'beautiful people of Stack Overflow'.
2. **Web Style Sheets CSS tips & tricks: Centering things** by W3C.

SPOT CHECK 1

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

1. What does CSS stand for?
2. What is the difference between inline styling, internal CSS and external CSS? When would you use each one?
3. What is the hierarchy of the three methods of CSS styling? If you have all three, in what order do they take precedence?
4. If you wanted to make a paragraph red with the font size 18 and aligned to the right, show how you would write this with inline styling, internal CSS and external CSS.

Instructions

Open all the examples in the directory called "Examples" for this task and read through the comments before attempting these tasks. Also, please consult the **additional reading**. The additional reading is a resource provided by the World Wide Web Consortium (W3C). The W3C is the international community that develops the standards that govern the web. In other words, these are the guys that make the rules for how the web works!!

Compulsory Task 1

Follow these steps:

- In the folder 'Compulsory Task' open the file **task1.html**.
- Create an external stylesheet called **myStyle.css** which you will use to style **task1.html**. Use the external stylesheet whenever possible but use internal style sheets or in-line style if necessary. The following style rules should be applied to **task1.html**:
    - All headings should have a background colour of 'orangered'. (See additional reading 'CSS notes for professionals' Chapter 18: Colors)
    - All headings should be centered.

    - The text in all paragraphs should be maroon and Times New Roman. (See additional reading 'CSS notes for professionals' Chapter 15: Typography)
    - All the `<h3>` elements should be chocolate in color except for the heading that contains the text "The info below is NOT mine…." which should be in red.
    - The image should be centered. (See additional reading 'CSS centering things')
    - There should be a margin of 2% around the entire page. (See additional reading 'CSS notes for professionals' Chapters 7 and 8)
- At the bottom of the page add a 3x2 table with the headings "Web 1.0", "Web 2.0" and "Web 3.0". Add a sentence that describes the key characteristics of each version of the web. Make the column headings of this table centered and red, and the rest of the content purple.
- Centre the table on the page.
- Before submitting your code, check it with the HTML and CSS validators **here** and **here**.

Compulsory Task 2

Follow these steps:

- Create files called **about.html** and **myStyles.css** in this folder. **about.html** should contain information about you: your educational history, your work history, who you are and what you're passionate about. Eventually, you will be building a **developer portfolio** to showcase your skills. "About.html" could form part of this portfolio.
- Set out the basic HTML document template and give it a title as you see fit.
- Use the following six elements at least once:
    - h1

    - p
    - img
    - h2
    - h3
    - nav (find help **here** and **here**.)
- Depending on the content you have chosen, align some of them on the left, some on the right, and some in the middle.
- The nav bar should be centered and the elements be placed next to each other (horizontally).
- Change the background colour of the entire page (hint: think about where all of the content is kept).
- Make all the headings in italics, and change their font family to Times New Roman.
- The following should be added to your elements (1 per element)
    - Text-colour
    - Image-size
    - Border
    - Padding
- Before submitting your code, check it with the HTML and CSS validators **here** and **here**.

Completed the task(s)?

Ask your mentor 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. Cascading Style Sheets
2.
    a. Inline styling is used to style a specific line of code and is written in the HTML file as an attribute of the line it is intended to style. You would use this when you only want to style one specific piece of code, rather than a particular element.
    b. Internal CSS is where the styling is written in the HTML file, but it is within the style element inside the `head` element. This is used if you want specific elements to have the same styling, e.g. you want all instances of heading 1 to be blue, but you want the styling to be within the HTML file.
    c. External CSS is where styling takes a similar format to internal CSS, except it is within a CSS file that is *linked* in the `head` element of the HTML file. This is used if you want particular elements to have consistent formatting and you want to keep your code neat and readable by having the styling in a separate file.
3. Inline styling takes this highest priority, then internal CSS then external CSS. So if you have all three, you will see the inline styling execute over internal and external CSS, and internal CSS will execute over external CSS.
4.
    a. Inline styling:

```
<p style="font-size: 18px;color: red;text-align: right;">This paragraph
is size 18 font, red and aligned to the right.</p>
```

    b. Internal CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    font-size: 18px;
    color: red;
    text-align: right;
    }
</style>
</head>
```

    c. External CSS

```
/*Within the style.css file:*/
p {
    font-size: 18px;
    color: red;
```

```
    text-align: right;
    }
```

Then in your HTMLfile:

```
<!DOCTYPE html>
<html>
<head>
    <link href = "style.css" rel = "stylesteet" type = "text/css">
</head>
```