



TASK

The Box Model

Visit our website

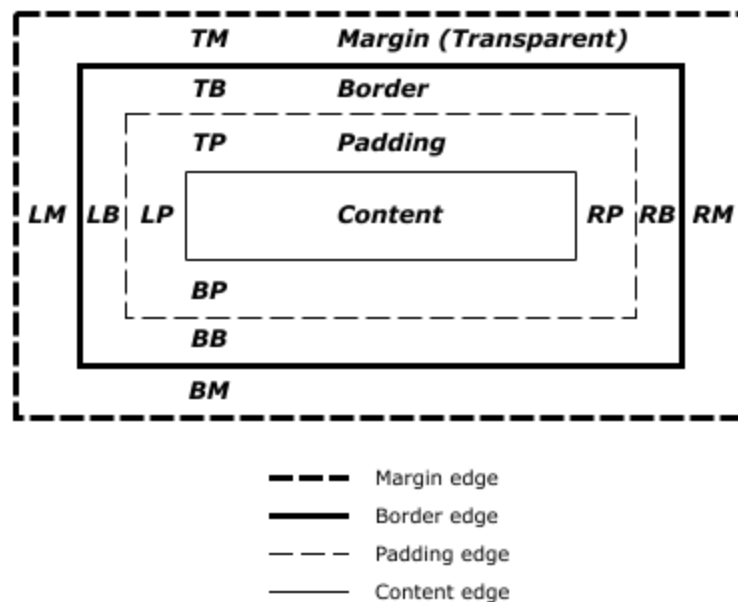
Introduction

WELCOME TO THE BOX MODEL TASK!

Okay, you're now a bit more familiar with the difference between CSS inline style and internal, external CSS, and the most common element selectors. In this task, we're going to explore the box model, which is a fundamental design and layout concept we use when deciding how to arrange the features on our web page. It's a useful concept that simplifies how we imagine the composition of page elements and their various boundaries. Most importantly, the box model prescribes the rules we must follow when altering the assets on our page.

THE BOX MODEL

An important fundamental concept to understand in CSS is what is called the *box model*. The browser creates a rectangle for each element in the HTML document. The Box Model describes how the padding, border, and margin are added to the content to create this rectangle. The following diagram is a depiction of the box model for an element.



Each box has a content area (e.g., text, an image, etc.) and optional surrounding padding, border, and margin areas; the size of each area is specified by properties defined below.

The margin, border, and padding can be broken down into top, right, bottom, and left segments (e.g., in the diagram, "LM" for left margin, "RP" for right padding, "TB" for the top border, etc.).

The perimeter of each of the four areas (content, padding, border, and margin) is called an "edge", so each box has four edges:

Content edge or inner edge

The content edge surrounds the rectangle given by the width and height of the box, which often depends on the element's rendered content. The four content edges define the box's content box.

Padding edge

The padding edge surrounds the box padding. If the padding has 0 width, the padding edge is the same as the content edge. The four padding edges define the box's padding box.

Border edge

The border edge surrounds the box's border. If the border has 0 width, the border edge is the same as the padding edge. The four border edges define the box's border box.

Margin edge or outer edge

The margin edge surrounds the box margin. If the margin has 0 width, the margin edge is the same as the border edge. The four margin edges define the box's margin box.

Each edge may be broken down into a top, right, bottom, and left edge.

Let's put this into practice! Create a folder named **"boxModel"**, and in this folder create an HTML file named **"index.html"**. Copy the following code into the index.html file.

```

<!DOCTYPE html>
<head>
  <title>Static Template</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <h2>Demonstrating the Box Model</h2>

  <p id="firstParagraph">
    The CSS box model is essentially a box that wraps around every HTML
    element. It consists of: borders, padding, margins, and the actual
    content.
  </p>

  <p class="secondParagraph">This is the second paragraph </p>

  <div class="div1">
    This text is the content of the box. We have added a 50px padding, 20px
    margin and a 15px green border.
  </div>
  <div class="div2">The total width of this element is 350px</div>
</body>
</html>

```

Now in the same folder create a file named “**style.css**”. We perform the link to this external stylesheet by using the following line of code in the **head section** of the “**index.html**” file.

```

<link rel="stylesheet" href="style.css" />

```

With no CSS code in the CSS file, when index.html is opened in the browser, it should look like this.

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.
The total width of this element is 350px

Let's change the CSS properties of all of the paragraph tags in the HTML file. We will use an element selector, "p", and then change the font to "Verdana" and the font colour to "red". You can copy and paste this into the CSS file.

```
p {  
  font-family: Verdana;  
  color: red;  
}
```

After saving and refreshing the browser, both paragraphs should change accordingly:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.
The total width of this element is 350px

Now let's style the first div element. (The <div> HTML element is the generic container for flow content. It has no effect on the content or layout until styled in some way using CSS; it is also known as a content division element.)

Let's set the background colour and width of the element first. We will use the class selector for the first div element with the class name div1, and change the background colour to **lightblue**, and the width of the element to 300px. Copy and paste the following into the CSS file, save, and look at the changes in the rendered web page after reloading the browser.

```
.div1 {  
  background-color: lightblue;  
  width: 300px;  
}
```

The result should be as follows:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.
The total width of this element is 350px

You will now notice that the background colour has changed and the container in which the text is situated now has a width of 300px. Let's add a border to the same div container. We will make the border width 15px, the border type solid and the colour lightcoral.

```
.div1 {  
  background-color: lightblue;  
  width: 300px;  
  border: 15px solid lightcoral;  
}
```

We should see the following change as this is applied:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.
The total width of this element is 350px

You will notice that there is no spacing between the border and the text (we need to add padding) and the border and the div element below it (with the content that reads "The total width of this element is 350px"). Let's add padding of 50 pixels around the text:

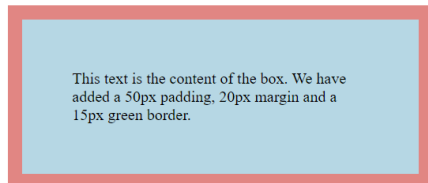
```
.div1 {  
  background-color: lightblue;  
  width: 300px;  
  padding: 50px;  
  border: 15px solid lightcoral;  
}
```

Have a look at the result:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph



The total width of this element is 350px

Now let's add a margin around the border of 20 pixels:

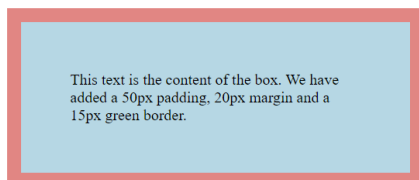
```
.div1 {  
  background-color: lightblue;  
  width: 300px;  
  padding: 50px;  
  border: 15px solid lightcoral;  
  margin: 20px;  
}
```

The result is spacing around the element:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph



The total width of this element is 350px

Now let's add styling to our second div element with the class name "div2":

```
.div2 {  
  width: 320px;  
  padding: 10px;  
  border: 5px solid gray;
```

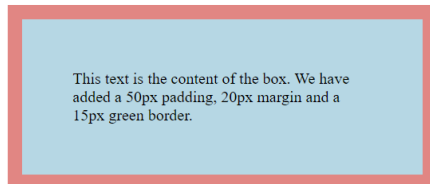
```
margin: 0;
}
```

The result is as follows:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph



The total width of this element is 350px

It is also possible to have different pixel sizes for the left, right, top ,and bottom padding and margins by using the following CSS properties:

- padding-top
- padding-right
- padding-bottom
- padding-left
- margin-top
- margin-right
- margin-bottom
- margin-left

You are encouraged to explore these using the files that you created so that you can see how the changes occur in the browser.

CASCADE

As we know, CSS stands for Cascading Style Sheets. You may have wondered why they are called *cascading* style sheets. ‘Cascading’ refers to how the 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.

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 on a page) whereas ID selectors are the most specific. Therefore, ID selectors will be applied over class selectors and element selectors.

Let's see this in action using the files from the box model demonstration which you have created.

Earlier we used the element selector “p” to style all paragraph elements red. We can override this by being a bit more specific and changing the colour of all paragraphs of a specific class. Let's do this for the “secondParagraph” class and change it to green:

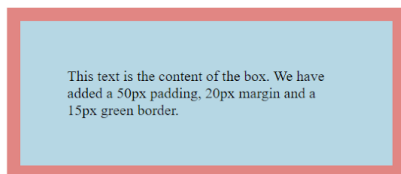
```
.secondParagraph {  
  color: green;  
}
```

We can now see that the second paragraph colour changed despite the red colour assigned to all paragraphs:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph



The total width of this element is 350px

Now let's give the first paragraph the same class name as the second paragraph:

```
<p id="firstParagraph" class="secondParagraph">  
  The CSS box model is essentially a box that wraps around every HTML  
  element. It consists of: borders, padding, margins, and the actual  
  content.
```

```
</p>
```

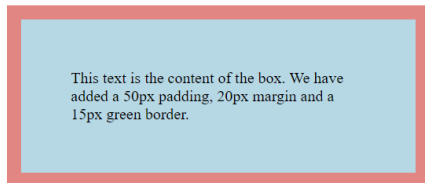
```
<p class="secondParagraph">This is the second paragraph </p>
```

Both paragraphs will now be green.

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph



The total width of this element is 350px

Lastly, we can see that the first paragraph has the id "firstParagraph". If we use an id selector and change the colour of this id to blue, we will see the class colour get overridden. Add the following to your CSS:

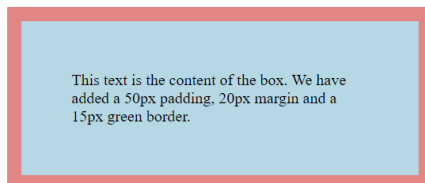
```
#firstParagraph {  
  color: blue;  
}
```

The result:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph



The total width of this element is 350px

Lastly, the most specific type of style is inline styling, which will override any other styles applied to an element. Let's add inline styling to the first paragraph by changing

the colour to magenta, and consider the result:

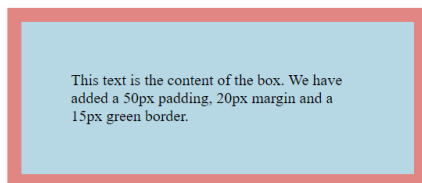
```
<p id="firstParagraph" class="secondParagraph" style="color: magenta;">
  The CSS box model is essentially a box that wraps around every HTML
  element. It consists of: borders, padding, margins, and the actual
  content.
</p>
```

The result:

Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This is the second paragraph



The total width of this element is 350px

CSS VALIDATOR

As a developer, you have no doubt come to realise that with HTML 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 precisely 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, “:”.

You will eventually make mistakes that will violate these rules and that will cause problems when you try to view web pages in the browser. Look, 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. The eBook entitled “HTML5 notes for professionals” by the ‘beautiful people of Stack Overflow’ provided with task 12 (HTML).
2. [Web Style Sheets CSS tips & tricks: Centering things](#) by W3C.

Instructions

Open all the examples in the directory called “Examples” (**from the CSS I task**) for this task and read through the comments before attempting these tasks. Also, please consult the [additional reading](#). This 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 folks that make the rules for how the web works!



A note from the HyperionDev Team

Please have a look at the following video playlist which explains CSS, [here](#).



Compulsory Task

Let's add to the tribute page you created in the previous task. Remember, you can find an example of a tribute page [here](#).

- You'll be working with your **tribute.html** and **myStyles.css** files from your previous task. Now in your **tribute.html** file:
- Apply styling to each of the following elements as described:
 - Ensure your h1 tags have padding of 20 pixels left and right and 10 pixels top and bottom.
 - Change your h1 tags to brown
 - Insert a blue border around all images you've used
 - Use an id selector (id="para1") to change the colour of your first paragraph to 'navy'.
 - Add two sets of 'ul' items to your document and highlight them with CadetBlue.
 - Change your caption(s) for any images to bold, centred, and 20 pixels in size.
 - Give all div elements rounded corners or 5 pixels.
 - Ensure that all h2 tags are italic.
 - All h3 tags should have a width of 25%.
 - Ensure that all aside tags are indented by 20 pixels.
 - Select or create a paragraph and add a 'spring green', mixed border of 5 pixels to it.
 - Add a class selector with these properties and values:
 - Font size of 25 pixels
 - Line height of 20 pixels
 - Colour should be 'plum'
 - Font type should be monospace
 - Padding of 4 pixels
 - Left margin of 6 pixels



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.



REFERENCES

CSS reference - CSS: Cascading Style Sheets | MDN. Mozilla.org. Published June 16, 2022. Accessed June 28, 2022. <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

Powell K. HTML & CSS for beginners. YouTube. Published online 2022. Accessed June 28, 2022. <https://www.youtube.com/playlist?list=PL4-IK0AVhVjM0xE0K2uZRvsM7LklhsPT->

Pinimg.com. Published 2022. Accessed June 29, 2022. <https://i.pinimg.com/originals/ab/6b/61/ab6b61d2f4197fb41a07fa04038836df.png>