

Lab Work CSS Box Models – Week 6

Important to do first

Before you start to complete the tasks below you should download the examples.zip file from loop, you will notice all the CSS styles have been removed, after you have completed the tasks below you should use your notes to enter the styles missing for each of the files from the lectures for the first set of CSS notes. **This should be completed after you do the tasks listed below**

Create a new folder on your local\lab machine called boxes.

At the end of these tasks you should be able to:

- Apply borders, margins and padding to elements in an HTML page
- Centre content in a web page using CSS
- Use box properties to format other HTML element

How to Use the CSS Box Model – Examples

To help you understand the theory of the box model in practical terms, I have created an example page with a block element in it, see below, copy the code and save the file as css-lab-work.html. It's basically an HTML document with some sample text inside a div box.

```
<!doctype html>

<html lang ="en">
<head>
<title>Box Model Tutorial</title>
<!-- use an external style sheet to style this page call it css_style1.css-->
<!--within the style sheet create a class called example-element/-->
</head>
<body>
<h1>Understand How to Use CSS and the Box Model</h1>
<h2>CSS Tutorial</h2>

<div class="example-element">
```

<p>How to use CSS? How to design a website?

If these questions often come to your head, you are in the right place. This CSS tutorial will help you understand what the best and easiest way to learn CSS is. Furthermore, after finishing this guide, you'll also know the main differences between CSS and HTML as well as why you usually need both languages to make a professional website. This CSS tutorial will reveal some brilliant tips that will help you master CSS in no time.

</p>

<p>

Creating and adding custom CSS to websites might seem like an intimidating task at first but don't worry! After discovering some tricks, it will be as simple as ABC.

</p>

<h3>CSS is not a difficult language to learn. </h3>

<p>You just need to know where to get started and occasionally try out different concepts in practice.

</p>

</div>

Let's get started.

<div>

CSS Tutorial: differences between CSS and HTML

 CSS Tutorial: HTML & CSS Syntaxes

 CSS Tips For Easier Learning Experience

</div>

<p>

CSS Tutorial: differences between CSS and HTML

Let's start with an example which will illustrate the difference between CSS and HTML.

Imagine that you're building a house. What's the first thing that comes to your mind? Is it the colour of your walls and how you would decorate them? Is it the shape of the dining table and the fabric of your sofa? Most people tend to think about the decorative aspects first when they're creating something. In the context of building a house, we call this interior design. However, there's another crucial element that we tend to take for granted. It's the structure that keeps everything from crumbling – the architecture of the house.

</p>

</body></html>

To start off, create a css style sheet called `css_style1.css`, also to help make it easier for you I have given you the first piece of code to start you off.

```
.example-element {  
    background-color: deepskyblue;  
}
```

Keep in mind that an external CSS style sheet must only contain CSS no html is permitted. This code simply adds a deepskyblue background to the DIV container, run your html file to see the output, if you like to change the colour to any colour you prefer.

Now let's make some changes to it.

Changing the Width

The first thing we will do is declare a width. At the moment the width is automatically set to 100%. If you want to change that, you need to use the width property.

We have several ways of setting it. You can use a fixed width like pixels or a proportional value like percentages. The latter is really important for things like responsive design. However, for simplicity's sake, you should go with pixels. Modify your CSS file to include a **width: 600px;** save the file and run your html file again to see the changes

You will notice how the div box is now limited in how far it stretches to the right. You can also see that it's height is adjusted automatically to make room for the text it contains. Let's increase the height of the DIV container further, include the following code in your example-element class in the CSS file

height: 400px; this value height should give the box a rectangle shape.

Introducing a Border to our box model

Next, I want you to change the border. The border is already there, you just can't see it. Time to change that via the **border** property.

What's important to know about this operator is that it takes three values: border width, style, and colour. Width is the thickness of the border (usually in pixels), style can be solid, dashed, dotted and a whole lot of other things we looked at in lectures and colour is either a declaration like "red" or the hex value of a colour.

Theoretically, it's enough to only define the border width. However, most of the time you achieve better results setting all three values. Here is what I want you to add to your CSS file, again within the example-element class:

border: 15px solid blue; save the CSS file and run your html file again to see the changes.

Be aware, that as mentioned in lectures you can also set different types of borders on different sides of an element via border-top, border-right, border-bottom and border-left.

Putting in Some Padding

When you looked at the example, one of the main things that stands out is that the text is bordering directly on the side of the div element. That makes it very hard to read and is far from pleasant to look at. Luckily, we can change that via the padding rule. Here's what you need to include to do:

padding: 16px; this will add 16 pixel between the content of the box model and the border.

just like for border, it's also possible to set different padding values on different sides of an element. For that, use padding-top, padding-right, padding-bottom and padding-left.

You can also go with a shorthand like padding: 10px 5px 15px 10px;. The numbers denote the padding at the top, right, bottom and left respectively.

Adding a Margin

Finally, you are getting to add a margin. This determines the space outside of an element, influencing its position on the page and in relation to other elements.

To make space clearer, you should add the following to your style sheet:

margin: 25px; again save the CSS file and run the html file to see the changes

And there you have it, the basics of the CSS box model. Learning the above will help you a lot with understanding how websites work. Doing so makes it much easier to manipulate your design to whatever you want and change the layout of your web pages.

Try creating another class in you CSS file and call it .nav-container include a background colour along with padding, margins, width and height properties and wrap the class using another div block element around all the links in your html document.

Finally copy and paste the .example-element class in your css file and rename the copied version and name it .example-element2, add the following code to the class

margin: 50px auto 50px auto; wrap the class using another div block element around the final Paragraph in your html document, again save the CSS file and run the html file to see the changes.

Once you have completed this try adding different style borders, different colour borders etc.

Now complete the tasks mentioned at the top of the page, download the files and using your lecture notes add the missing CSS styles.