

Chapter 5: The Box Model, Links, Lists and Tables

Learning Outcomes:

- ✓ Be able to manipulate the appearance of HTML hyperlinks, lists and tables using CSS
- ✓ Identify the features of the CSS box model and how the properties can manipulate HTML tags
- ✓ Apply the box model properties to HTML tags



Prerequisite Knowledge:

- ✓ Complete Chapter 4
- ✓ Be able to apply CSS using all three methods: inline, internal and external
- ✓ Be able to manipulate the appearance of simple HTML elements using basic CSS



State Marker Padding Margin Border Inline Block

Keywords

5.1 Practical: CSS and Hyperlinks

This chapter will focus on other key CSS techniques and properties that are necessary in the modern-day web; specifically, it will demonstrate examples of how CSS can be applied to HTML tables, lists and hyperlinks. This chapter will also examine the CSS box model, the crucial theory behind creating web page layouts.

Styling Hyperlinks with CSS

Like every HTML element, hyperlinks can be styled (including their colour, font family and decoration) using CSS. Previously, it was learnt that the anchor tag (`<a>`) is used to create hyperlinks and the `href` attribute is used to specify the source of a requested page. Therefore, to manipulate the appearance of a hyperlink, the `<a>`



HTML Tips!

When applying CSS to hyperlink states, there are some important ordering rules to be aware of. They are as follows: `a:hover` must proceed `a:link` and `a:visited`. `a:active` must come after `a:hover`.

Activity 5.1

Create a web page that has three hyperlinks to other external websites. Use CSS to manipulate the appearance of the hyperlinks, including their various states.



element is targeted using CSS selectors. However, be aware that there is a key difference when styling hyperlinks and that is that their state can be targeted too. For example, a different set of CSS properties can be applied to a hyperlink if it is hovered over using the mouse cursor. The four available states are listed below:

- `a:link` – a standard, unvisited, hyperlink
- `a:visited` – a link that has been visited by the user previously
- `a:hover` – a link that is hovered over with the mouse cursor
- `a:active` – a link that is clicked by the mouse cursor

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS and Hyperlinks</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <a href="http://www.bbc.co.uk">The BBC</a>
  <a href="http://www.itv.co.uk">ITV</a>
</body>
```

The BBC ITV



```
a:link{color:yellow;background-color:black;text-decoration:none;}
a:visited{color:purple;background-color:black;text-decoration:none;}
a:hover{color:red;background-color:black;text-decoration:underline;}
a:active{color:pink;background-color:black;text-decoration:overline;}
```

5.2 Practical: CSS and Lists

Styling Lists with CSS

As explained in Chapter 2, HTML provides two main types of lists: ordered (numbered/lettered) and unordered (bulleted). Like other HTML elements, CSS can be used to manipulate the appearance of lists, including the markers that are used by the lists. For example, an unordered list can be set with a square, circle or disc marker. An ordered list can be set to numbers, letters and roman numerals. Below are some examples of general list styles:

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS and Lists</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <ul class="one">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ul>
  <ul class="two">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ul>
  <ol class="three">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ol>
  <ol class="four">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ol>
</body>
</html>
```

- Item 1
- Item 2
- Item 3
- Item 4
- Item 1
- Item 2
- Item 3
- Item 4
- i. Item 1
- ii. Item 2
- iii. Item 3
- iv. Item 4
- A. Item 1
- B. Item 2
- C. Item 3
- D. Item 4

```
ul.one{list-style-type:circle;color:red;}
ul.two{list-style-type:square;color:blue;}
ol.three{list-style-type:lower-roman;color:green;}
ol.four{list-style-type:upper-alpha;color:orange;}
```

HTML Tips!

The default marker for an unordered list is 'disc'. An ordered list can be set to alpha (both upper and lower), roman (both upper and lower) and decimal (for numbers). There is also another marker type called 'decimal-leading-zero'; this is numeric, but the number is preceded by a zero.

Activity 5.2

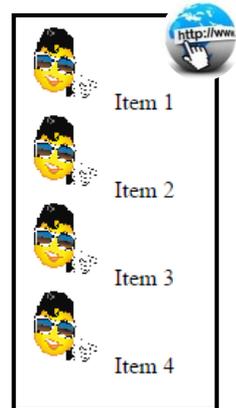
Create a single web page that consists of two ordered and two unordered lists. The lists can be based on any topic; however, use CCS to manipulate their appearance, including their marker.

5.3 Theory: List Image Markers and List Position

CSS and List Image Markers

Markers do not have to necessarily be the preset values available in CSS; it is also possible to use a custom image marker. This is achieved by using the `list-style-image` property and setting the value of the property to the URL path of the chosen image. A syntax example of this is shown below:

```
ul{list-style-image:url('elvis.png');}
```

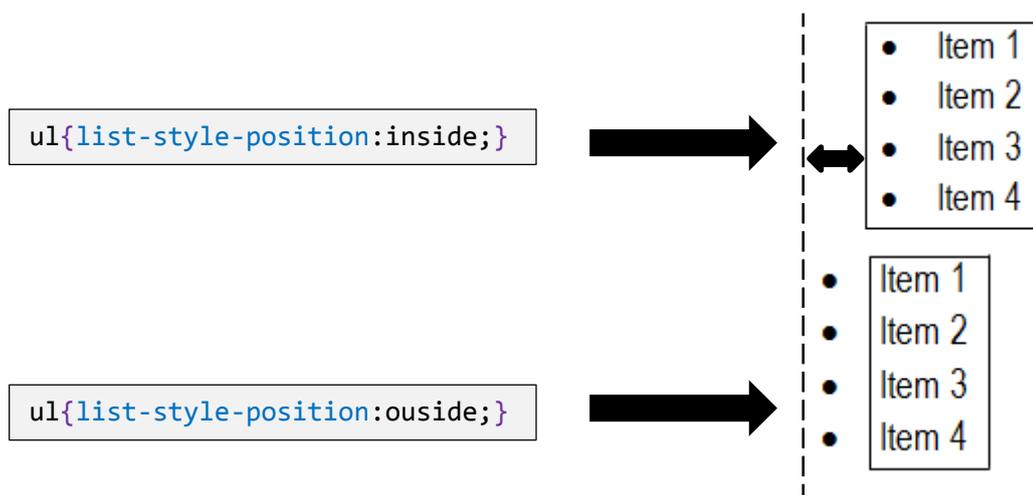


Unfortunately, there is a known issue when using this CSS property and that is that not all browsers display the content equally. In other words, each displays them slightly differently, which can be a problem if seeking a consistent style across all browsers. Specifically, Internet Explorer and Opera will display the image marker higher than Firefox, Chrome and Safari. Therefore, if seeking a consistent approach a cross-browser solution is required; fortunately there is one and this is shown below:

```
ul{
  list-style-type:none; /* removes the list marker */
  padding:0px; /*sets the padding to 0 (for browser compatibly) */
  margin:0px; /*sets the margin to 0 (for browser compatibly) */
}
ul li{
  background-image:url('elvis.png'); /* sets the background image */
  background-repeat:no-repeat; /* shows the image only once, with no repeat*/
  background-position:0px 0px; /* positions image - left 0px and down 5px */
  /*-->Adjust the padding sets below dependent on the image marker size */
  padding-left:44px; /* positions the text in the list - indents from the left */
  padding-bottom:30px; /*positions the text in the list from the bottom */
  padding-top:12px; /*positions the text in the list from the top */
}
```

CSS and List Style Position

The `list-style-position` property can be used to specify whether or not list markers should appear inside or outside the document's content flow; this affects the positioning of the list on a web page. This property can be set using two values: `inside` or `outside`. An example of this is shown below:



5.4 Theory: CSS and Tables

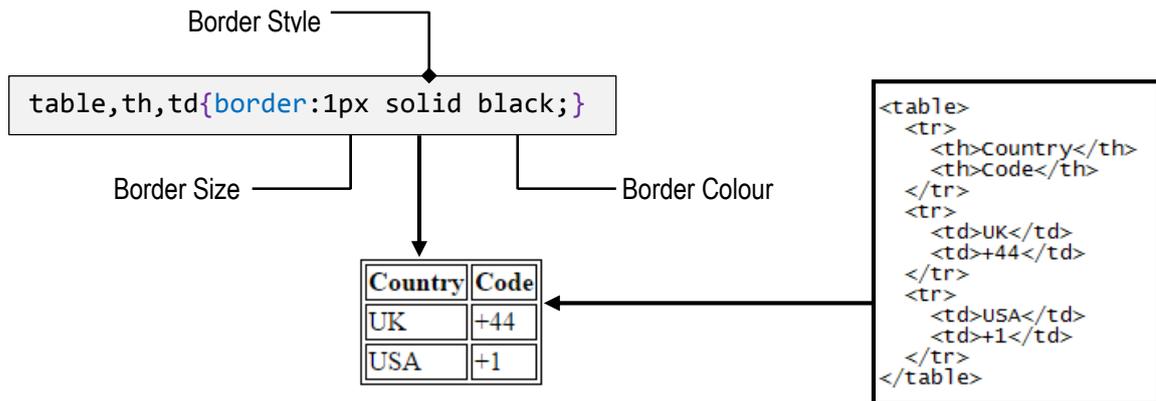
CSS and Table Borders

Standard tables in HTML are not visually appealing and, therefore, CSS can be used to improve their appearance immensely, specifically by adding colours and borders that complement the theme of a web page. By default, a HTML table does not have any borders when printed to a browser and looks as though the data is just tabbed into columns. However, a border can easily be applied to a table by using the `border` property. The following example applies a border to the table, table header and table data elements;



HTML Tips!

The border style property can be set with several values, including dotted, dashed, double and grove!



Collapsing Table Borders

The CSS syntax used to create the example above, has resulted in the table having a double border; this is the result of each element (table, table header and table data) having its own individual border. However, it is possible to display a table with a single border, by using the `border-collapse` property and settings its value to 'collapse'. The example below demonstrates how this can be applied:

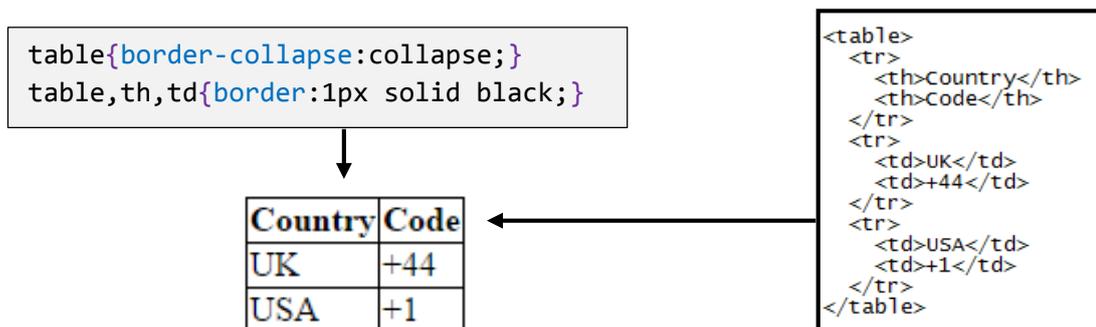


Table Width and Height

The width and height of a table can be manipulated using the CSS `width` and `height` properties; this is little different to manipulating the height and width properties of other HTML elements. The `height` property is set using pixels, whereas the `width` property can be set with either pixels or a percentage (of the width of the containing HTML element). The width and height properties can be generally applied to the `<table>` element; however, for more control they can also be applied to specific `<tr>`, `<td>` and `<th>` elements. For example, a heading row could have a larger height than a standard row. An example of this is demonstrated in the following pages.

5.5 Practical: CSS and Tables

Table Alignment, Height and Padding

When working with HTML tables, there are some tricks (in CSS) that can be used to greatly improve their presentation (and also to display data clearer); one such technique is 'stripping' every other table row. Below is a complete example, that demonstrates how CSS can be used to improve the look of a HTML table:

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS and Tables</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <h1>The Beatles - My Favourite Tracks!</h1>
  <table id="tracks">
    <tr>
      <th>My Ranking</th>
      <th>Track</th>
      <th>Album</th>
    </tr>
    <tr class="odd">
      <td>1st</td>
      <td>She Came In Through The Bathroom Window</td>
      <td>Abbey Road</td>
    </tr>
    <tr>
      <td>2nd</td>
      <td>Love Me Do</td>
      <td>Please Please Me</td>
    </tr>
    <tr class="odd">
      <td>3rd</td>
      <td>Sgt. Pepper's Lonely Hearts Club Band</td>
      <td>Sgt. Pepper's Lonely Hearts Club Band</td>
    </tr>
    <tr>
      <td>4th</td>
      <td>Michelle</td>
      <td>Rubber Soul</td>
    </tr>
    <tr class="odd">
      <td>5th</td>
      <td>Here Comes the Sun</td>
      <td>Abbey Road</td>
    </tr>
  </table>
</body>
</html>
```

This id is used to identify the table so that CSS properties can be applied. As this is an id, it can only be applied to a unique table.

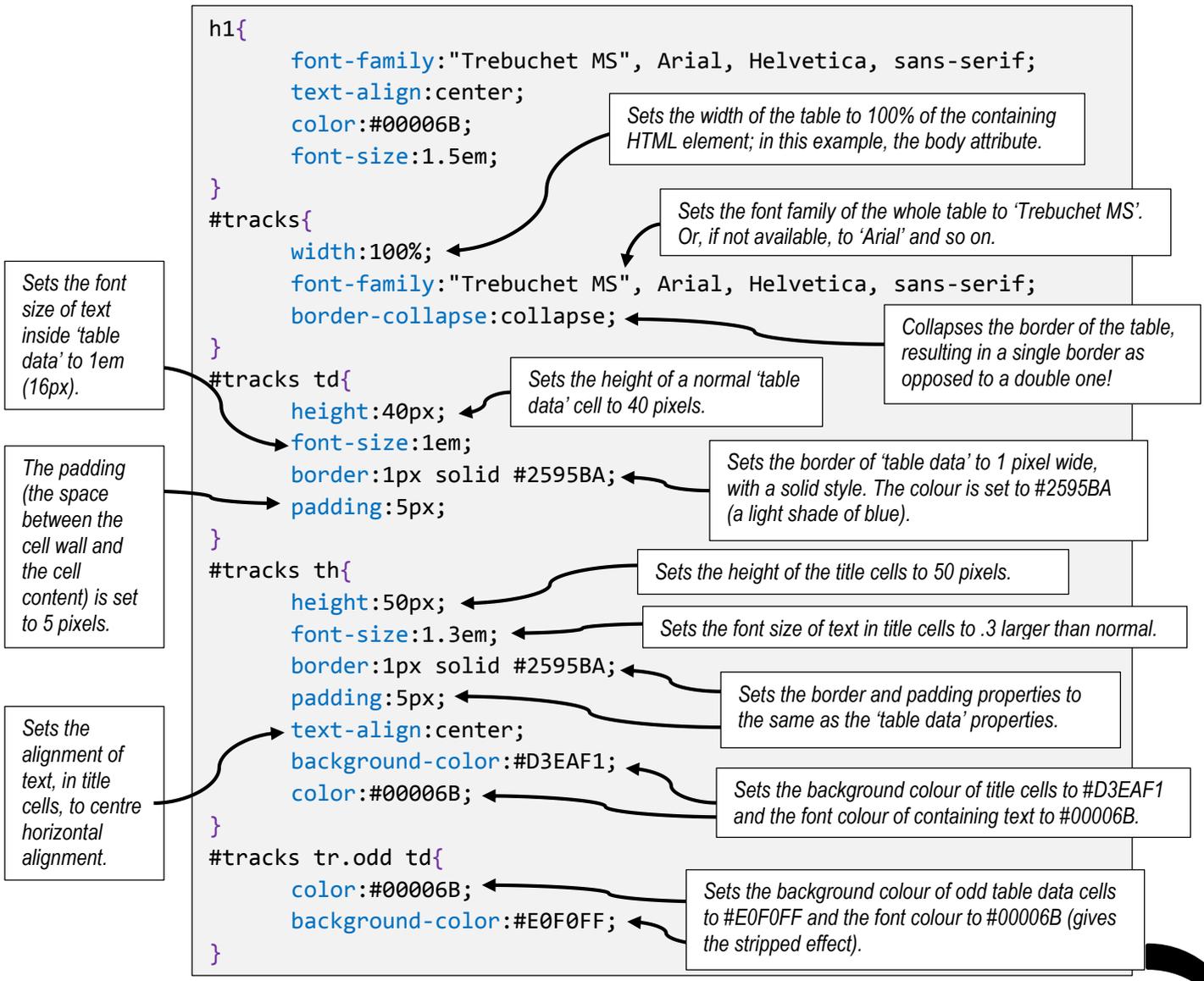
This class is used to identify every other row (all odd rows).

HTML Tips!

The `text-align` property can be used to set the horizontal alignment of text in a table; to set the vertical alignment of text, use the `vertical-align` property, the value of which can be set to the likes of baseline, top, text-top, middle, bottom, text-bottom and initial.

Activity 5.3

Create a web page that has a table featuring three columns and five rows, not including the title row. The data in the table can be on any subject; however, CSS should be used to style the table (including stripped rows). The web page should also feature a suitable title; this should also be styled.



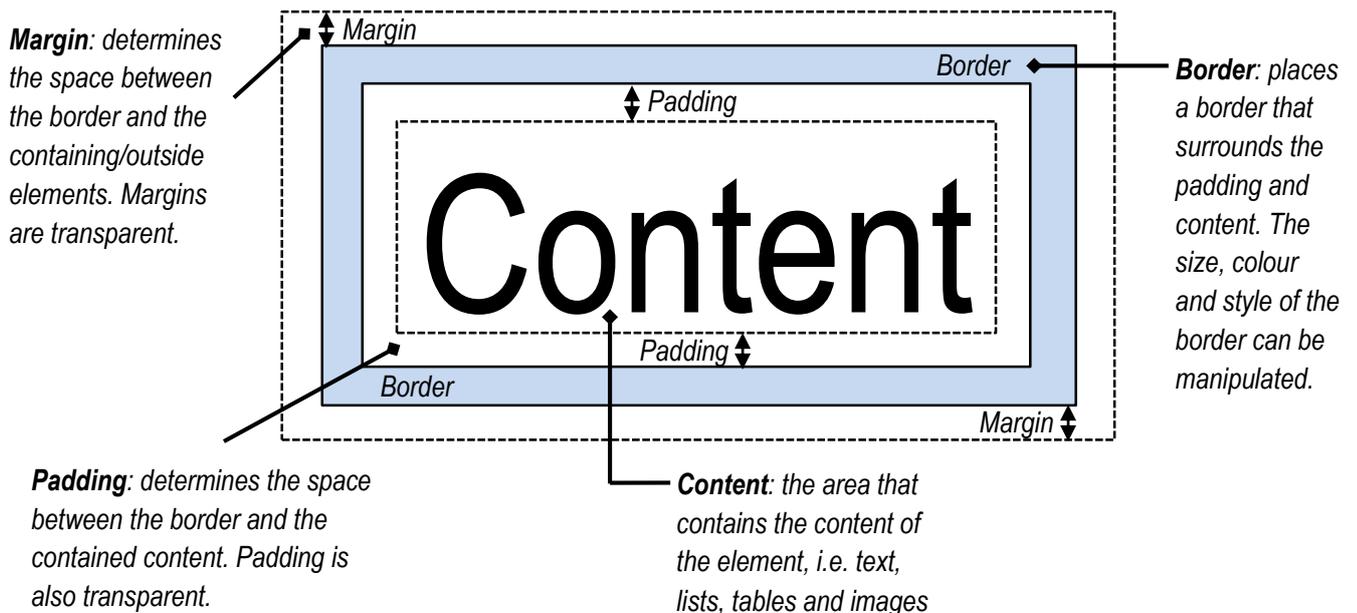
The Beatles - My Favourite Tracks!

My Ranking	Track	Album
1st	She Came In Through The Bathroom Window	Abbey Road
2nd	Love Me Do	Please Please Me
3rd	Sgt. Pepper's Lonely Hearts Club Band	Sgt. Pepper's Lonely Hearts Club Band
4th	Michelle	Rubber Soul
5th	Here Comes the Sun	Abbey Road

The #tracks selector is used to target the HTML table labelled with the 'tracks' id (`id="tracks"`). The #tracks td selector identifies table data elements (`<td></td>`) that are found in the table labelled with the 'tracks' id. The #tracks th selector identifies all table heading elements (`<th></th>`) that are found in the table labelled with the 'tracks' id. The #tracks tr.odd td selector states that the CSS properties will be applied to all table data cells that are contained inside rows labelled with the 'odd' class (`class="odd"`).

5.6 Theory: The CSS Box Model

Perhaps one of the most important lessons (when learning CSS) is understanding the 'box model' and its effects on the layout of HTML elements; without understanding this theory, it is difficult to create web pages to a desired (and professional) effect. In HTML, all elements can be considered as 'boxes'; and the box model describes the layout of these boxes. Every HTML element is essentially wrapped in a box, and all contain the same properties that enable the appearance of that box to be manipulated, including the padding, margin, border and content. Effectively, the box model enables borders to be applied to HTML elements and the space between elements to be determined. The illustration below demonstrates the available properties of the box model:

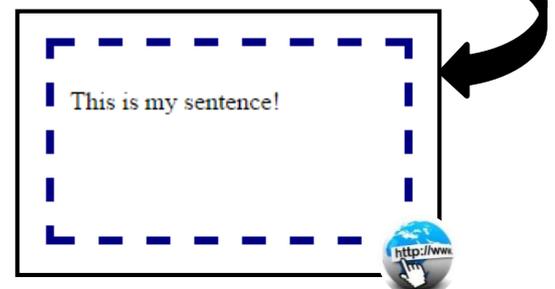


Below is an example of the box model properties being applied to a HTML division element:

```
<div>
  <p>This is my sentence!</p>
</div>
```

```
div{width:200px;
  height:100px;
  margin:20px;
  border:5px dashed navy;
  padding:10px;
}
```

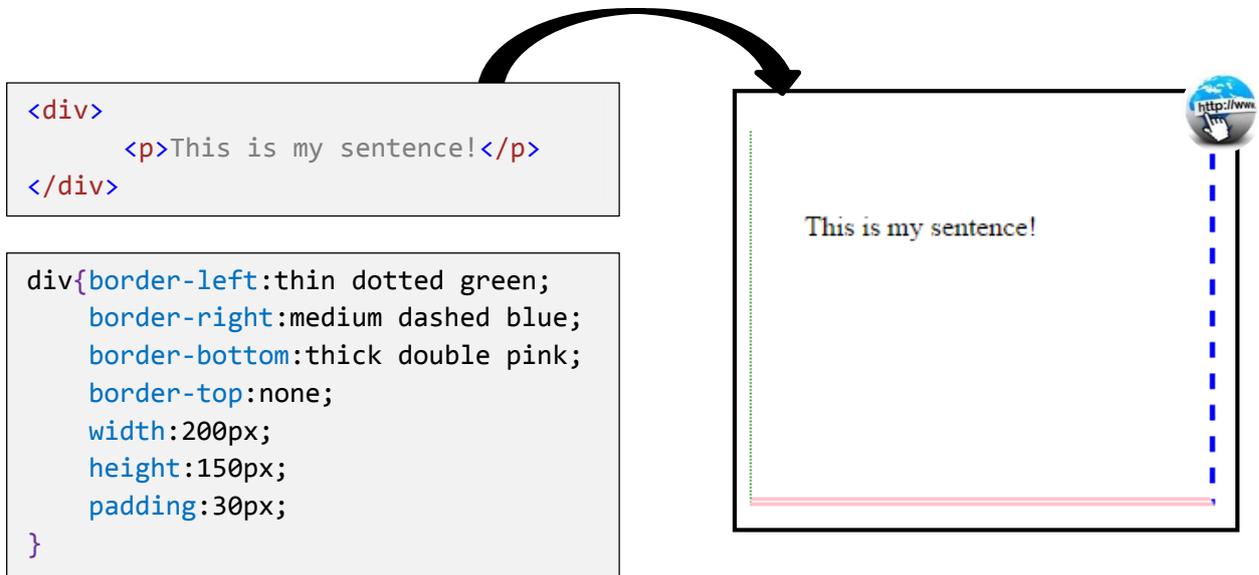
When setting the width/height of an element, this property addresses the content area only. To calculate the full width/height of a property, then the size of the margins, padding and border must be included too. For example, the total width of the element in this example is 270 pixels; 200px for the width + 20px for the margin (x2 for the left and right margin) + 5px for the border (x2) + 10px for the padding (x2). Using the same algorithm, the height of this element is 170 pixels. Earlier versions of Internet Explorer included the padding and border in the width property; however, this issue can be resolved by adding the HTML5 declaration: `<!DOCTYPE html>`.



5.7 Practical: CSS Custom Borders and Outlines

CSS Custom Borders

CSS borders have been demonstrated several times previously, including how to change their colour, style and size. However, borders do not have to maintain the same style on each side; they can, in fact, have unique values. For example, the bottom and top borders can be set with completely different values. An example of this is illustrated below:



The diagram illustrates the process of applying CSS to HTML. On the left, two boxes contain the code. The top box shows the HTML structure:

```
<div>
  <p>This is my sentence!</p>
</div>
```

 The bottom box shows the CSS rules:

```
div{border-left:thin dotted green;
border-right:medium dashed blue;
border-bottom:thick double pink;
border-top:none;
width:200px;
height:150px;
padding:30px;
}
```

 A curved arrow points from the code to a visual representation of the rendered box on the right. The box contains the text "This is my sentence!". The border is custom: the top border is none, the left border is thin dotted green, the right border is medium dashed blue, and the bottom border is thick double pink. A small globe icon with "http://www" is in the top right corner of the box.

The `border-width` property can be set with either three predefined values (thin, medium or thick) or can be set specifically by specifying a value in pixels (px). The `border-style` property can be set with the following values: none, solid, dotted, dashed, double, groove, ridge, inset or outset. It is worth experimenting with these styles to become familiar with all of them. The `border-color` property can be set using a colour name (for example, blue, yellow, and so on) or by specifying a hexadecimal value (for example, #0000BB). The border property can also be set to 'transparent' (to allow the background to show through). If the `border-color` property is not set by default, it will inherit the `color` property of the HTML element itself. Note that the `border-color` property also does not work, unless the `border-style` property is set too!



HTML Tips!

Writing `border-style:solid dotted double dashed` is actually shorthand syntax for the following properties;

```
border-top-style:solid;
border-right-style:dotted;
border-bottom-style:double;
border-left-style:dashed;
```



HTML Tips!

Writing `border:thin dotted green` is actually shorthand syntax for the following properties;

```
border-width:thin;
border-style:dotted;
border-color:green;
```

Activity 5.4

Create a web page that features four paragraphs; the topic of each paragraph can be based on any appropriate subject. Each paragraph should have a unique border style, including a custom border (for example, a different bottom and top border style). Ensure that the groove and ridge values (for the border style) are used at least once.



CSS Outlines

In the world of CSS, an outline is an additional line that is drawn outside a HTML element's border, enabling that element to 'stand out' in a web page. The properties that are available to manipulate an outline are similar to that of a border; the outline's style, colour and width. The key difference between an outline and a border is that an outline is not part of an element's dimensions. In other words, the width/height of an element is not impacted by the width of the outline. Below is an example of how the outline property can be used:

The diagram illustrates the process of applying a CSS outline to an HTML element. On the left, two boxes contain the code. The top box shows the HTML structure: `<div>`, `<p>This is my sentence!</p>`, and `</div>`. The bottom box shows the CSS rules for the `div` element: `border:thick dashed blue;`, `width:200px;`, `height:150px;`, `padding:30px;`, and `outline:10px solid green;`. A large black arrow points from the code boxes to a visual representation on the right. This representation shows a rectangular box with the text "This is my sentence!" inside. The box has a thick, dashed blue border and a solid green outline. A small globe icon with "http://www" is in the top right corner of the visual box.



HTML Tips!

Writing `outline:10px solid green;` is actually shorthand syntax for the following properties:

```
outline-style:solid;
outline-color:green;
outline-width:10px;
```



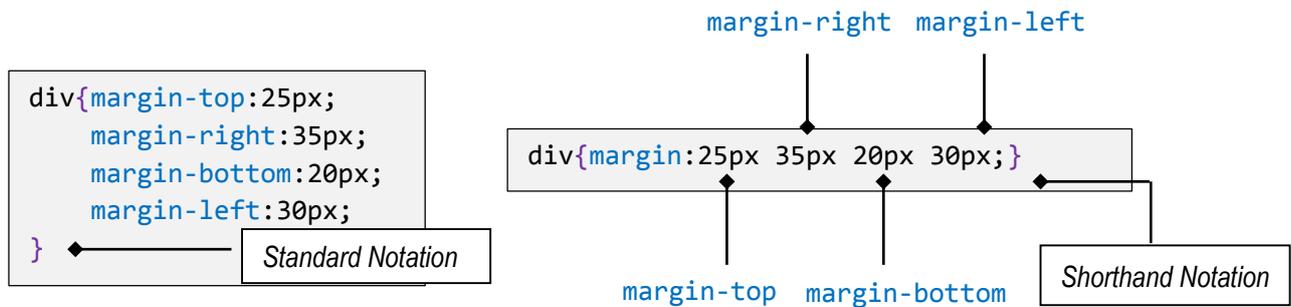
Activity 5.5

Edit the web page created for Activity 5.3, so that the first and second paragraphs have an outline applied to them; manipulate the appearance of both outlines so that they appear differently.

5.8 Theory: CSS Margins and Padding

CSS Margins

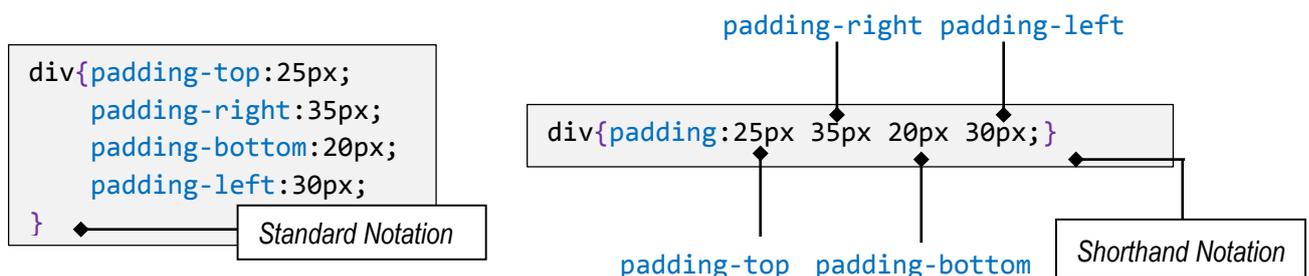
The box model diagram (previously illustrated in this chapter) shows that a margin property specifies the space around an element and its containing element. Margins are transparent and thus do not have a background colour; however, each respective side can be set independently. For example, the top margin can be set independent to the left, right and bottom margins (examples of this were also demonstrated earlier). However, this can also be done using a shorthand notation, which is demonstrated below and to the right:



The value of margins does not have to be set in pixels (px); margins can also be set using a percentage (of the width of the containing element), with the keyword 'inherit' (to inherit the margin values specified in the parent element) or with the keyword 'auto' (which allows the browser to 'automatically' calculate the margin). Margins can also be set with a negative value, which is used to overlap content.

CSS Padding

Like the margin property, the padding property was also illustrated in the box model diagram. Padding determines the area between content and a HTML element's border; unlike margins, however, padding is affected by the background colour property. However, similar to margins, the top, left, right and bottom margins can also be set independently. An example of this is illustrated below:



When working with padding, like margins, they can only be set with a 'length unit' (for example pixels) or a percentage of the containing HTML element.

5.9 Practical: CSS Dimensions and Display Types

CSS Dimensions

CSS has several dimension properties that can be used to manipulate the height and width of HTML elements. The most common of those properties is the standard `height` and `width` property, which is used to set the basic dimensions of an element. Both of these properties can be set with either a value (in pixels), or alternatively, a percentage of the outside container.

Other useful dimension properties include the `max` and `min` properties (which can be applied to both the height and width of a HTML element). These are particularly useful when working with containers that have `width/height` properties set with percentage values; this is because the containers will grow/shrink as the browser is resized, sometimes resulting in undesirable affects.

The `max` property can be used to set the maximum width/height that an element cannot exceed, including when the browser is resized. For example, if a division is set with a `width` of 90%, and a `max-width` of 1,000 pixels, this will result in the division being 90% of the container, as long as this does not exceed 1,000 pixels. This is because the `max-width` property overrides the `width` property. An example of this is demonstrated below:

```
div{margin-left:auto;
margin-right:auto;
width:90%;
background-color:red;
height:100px;
max-width:1000px;
}
```



HTML Tips!

The `max-height` property applies the same principles as the `max-width` property, except to the height of a HTML element. However, if the content inside the element is too large, the additional content can be hidden and accessed with a scroll bar by using the following CSS property: `overflow:auto`;

This division has a width less than 1000 pixels and therefore the division fills 90% of the outside container.



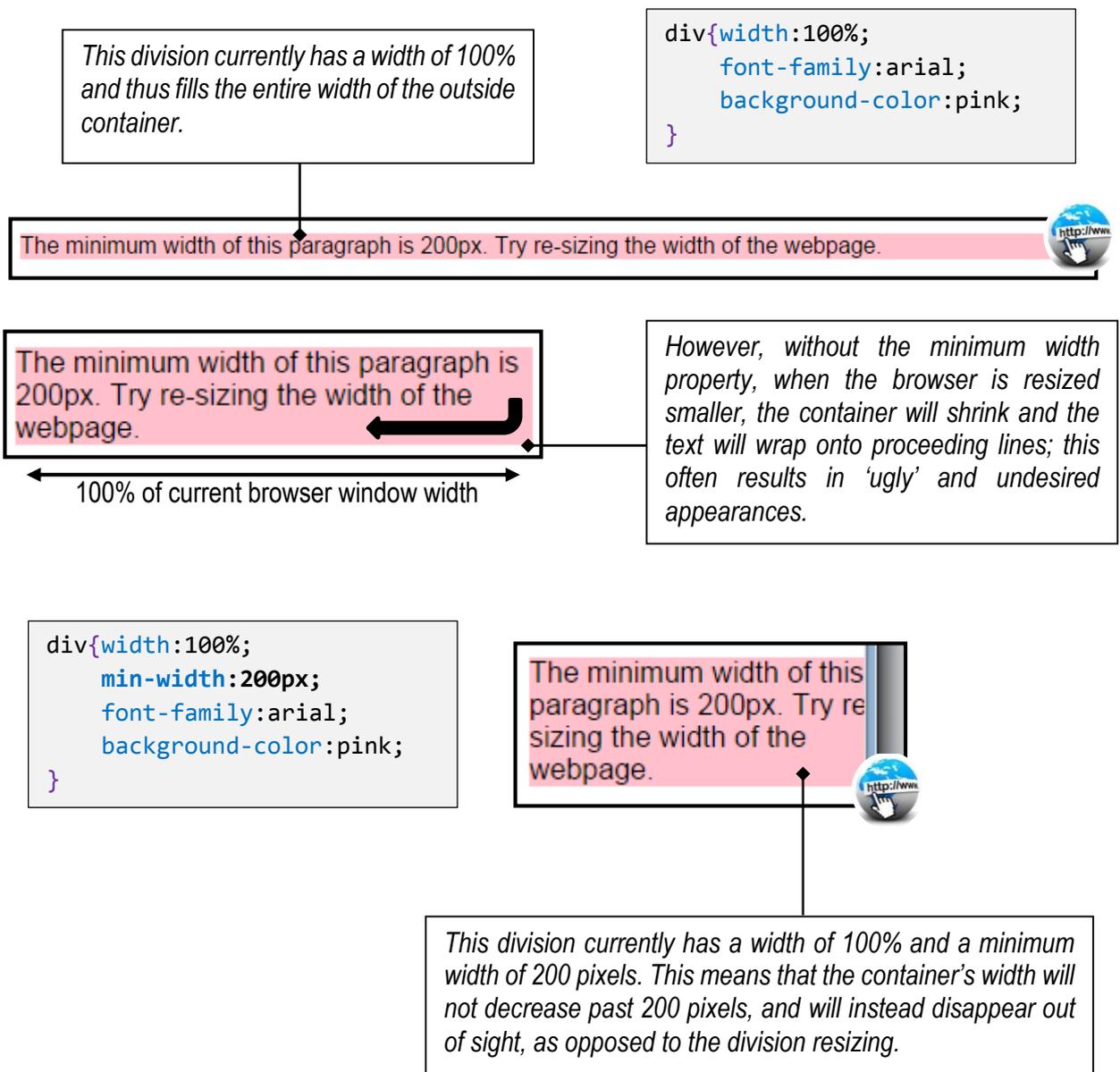
Activity 5.6

Create a web page that has two divisions, each containing paragraphs. The first division should demonstrate the `max-width` property and the second division should demonstrate the `max-height` property. Add a suitable title to the web page.



This division has a width of 1,000 pixels, because a width of 90% (of the outside container) would exceed the `max-width` property of 1,000 pixels.

Similar to the max property, the min property can be used to set the minimum dimensions that a HTML element can be, including when the browser is resized. Again, this is particularly useful when working with divisions that have dimensions set with a percentage value. This results in divisions being 'elastic', but without them resizing below a specific width size. The `min-width` property is demonstrated below:



HTML Tips!

The `min-height` property applies the same principles as the `min-width` property, except to the height of a HTML element!



Activity 5.7

Amend the web page from Activity 5.6 so that one division demonstrates the minimum width property. Resize the browser to ensure that the properties work as expected.

CSS Display Types

HTML elements have a default 'display type' that informs the browser how to display the element on a web page; however, this can be manipulated using CSS. For example, a list does not have to be displayed vertically; it can, for example, be displayed horizontally by changing the display type. There are two display types in CSS, 'block' (vertical) and 'inline' (horizontal), and both of these values can be used to change the default display type of a HTML element.

Activity 5.8

Create a web page that displays a list of items horizontally (using the `` element and the `display` property) and displays five `` elements vertically.

A block element refers to HTML elements that typically fill the entire width of the available space in a container, and normally have a line break before and after them. This includes elements such as `<p>`, ``, `<div>` and `<h1>`. Consider the example below:

This is my paragraph.

Although the paragraph only has a small amount of text, the entire width of the available space is filled by the paragraph tag; this is because it is a block element.

Therefore, 'normal' block elements can be changed to 'inline', resulting in them only taking up the space needed and not enforcing any line breaks. An example of this is shown below:

```
<!DOCTYPE html>
<html>
<head>
  <title>Inline and Block</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <p>I am paragraph one. </p>
  <p>I am paragraph two. </p>
</body>
</html>
```



I am paragraph one. I am paragraph two.

```
p{display:inline;}
```

By default, HTML elements such as `` and `<a>` are inline; however, these too can be manipulated and changed to block elements, displaying them vertically and with line breaks. An example of this is shown below:

```
<!DOCTYPE html>
<html>
<head>
  <title>Inline and Block</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <a href="http://www.bbc.co.uk">The BBC</a>
  <a href="http://www.itv.co.uk">ITV</a>
  <a href="http://www.channel5.com">Channel 5</a>
</body>
</html>
```

The BBC
ITV
Channel 5

```
a{display:block;}
```

Chapter Summary

- ✓ CSS can manipulate the appearance of hyperlinks dependent on their various states, such as visited, mouse hover and active.
- ✓ The `list-style-type` property can be used to manipulate the appearance of both ordered and unordered lists. The value of this property can include square, disc, lower-roman and upper-alpha.
- ✓ Lists do not have to use the default list markers; the list marker can be replaced with a custom image (although there are some associated compatibility issues) using the following syntax: `list-style-image:url('elvis.png');`
- ✓ The appearance of tables can be greatly improved by using CSS. To remove the double border from a table use the `border-collapse` property. The alignment of cell data can be manipulated both horizontally and vertically using the following properties, respectfully, `text-align` and `vertical-align`. CSS can stripe tables too, by creating a class with a different style (mostly background colour) and applying it to all odd rows.
- ✓ The `margin` property determines the space between the border and the containing/outside elements. Margins are transparent.
- ✓ The `padding` property determines the space between the border and the contained content. Padding is also transparent.
- ✓ The `border` property places a border that surrounds the padding and content. The size, colour and style of the border can be manipulated.
- ✓ Custom borders can be applied to HTML elements by manipulating the border sides individually. For example, `border-right` would be used to manipulate the appearance of the right border only.
- ✓ Individual sides of the `margin` and `padding` properties can be set using a shorthand notation; for example, `margin:25px, 35px, 20px, 30px;` which would set the top, right, bottom and left margin properties (in that order).
- ✓ The `max-width/height` properties can be used to set the maximum dimensions of a HTML element.
- ✓ The `min-width/height` properties can be used to set the minimum dimensions of a HTML element.
- ✓ A HTML element's display type can be changed from default, by setting the `display` property to either inline (horizontal) or block (vertical).