CSS

SUBJECT: BASICS OF WEB DESIGN (09CE2102)

DEPTRTMENT: CE-DIPLO

BY: ZEEL MEHTA

CONTENT

- CSS Introduction
- CSS Syntax
- CSS box model
- Types of Style Sheet
- CSS selectors
- CSS Comments
- CSS Colors
- CSS Backgrounds

- CSS Borders
- CSS Margins
- CSS Padding
- CSS Outlines
- CSS Fonts
- CSS Text
- CSS Positions

CSS Introduction

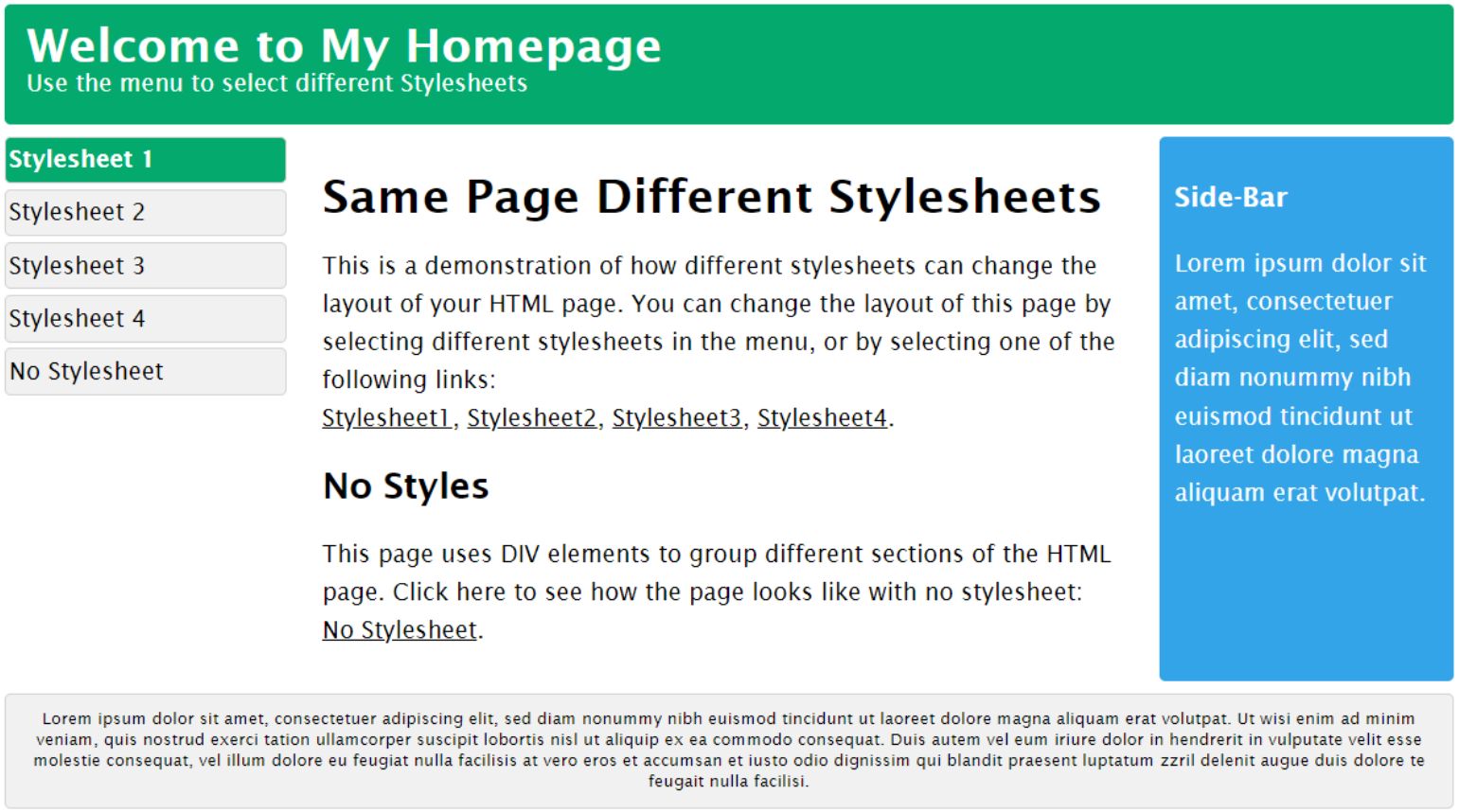
What is CSS?

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- External stylesheets are stored in CSS files.

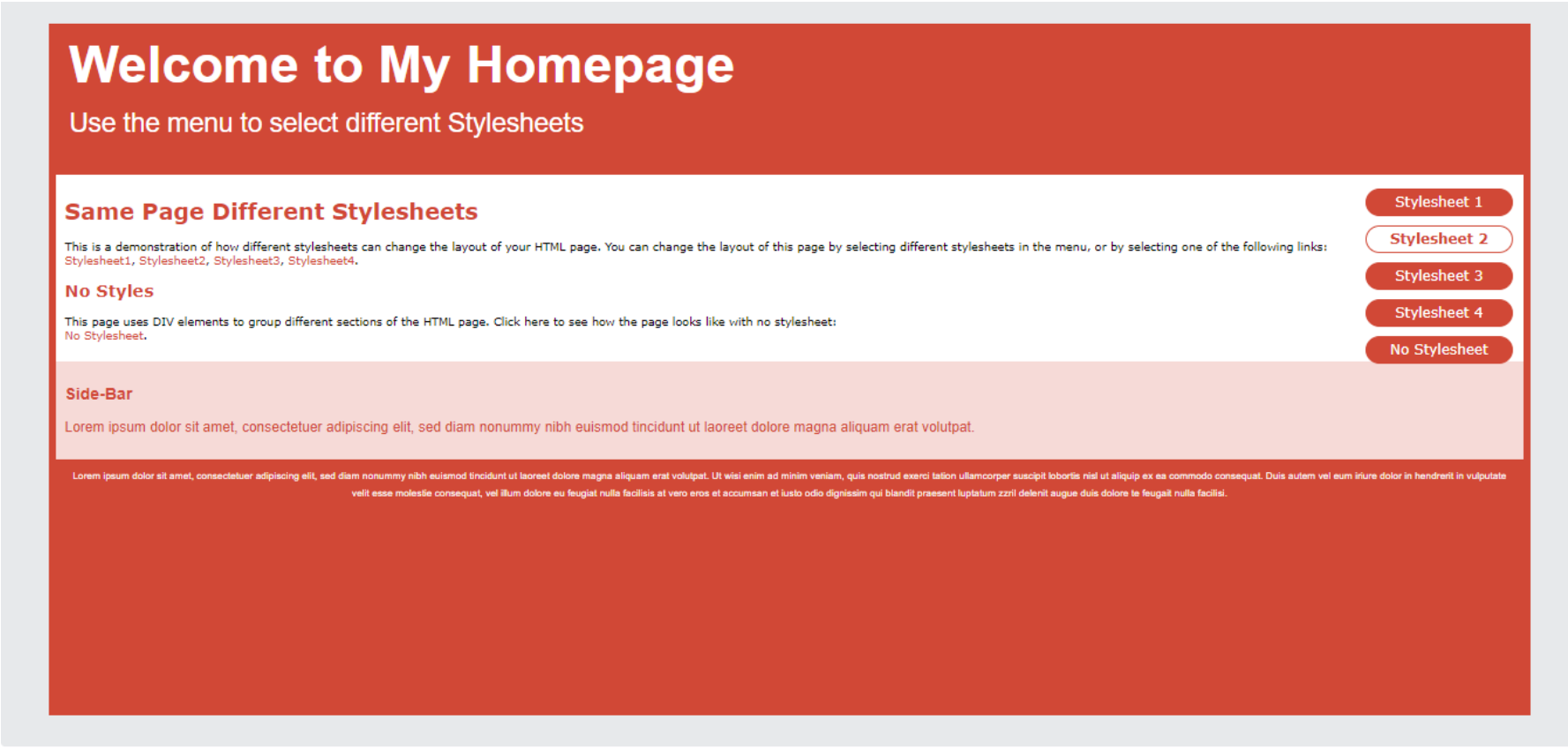
Why Use CSS?

- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

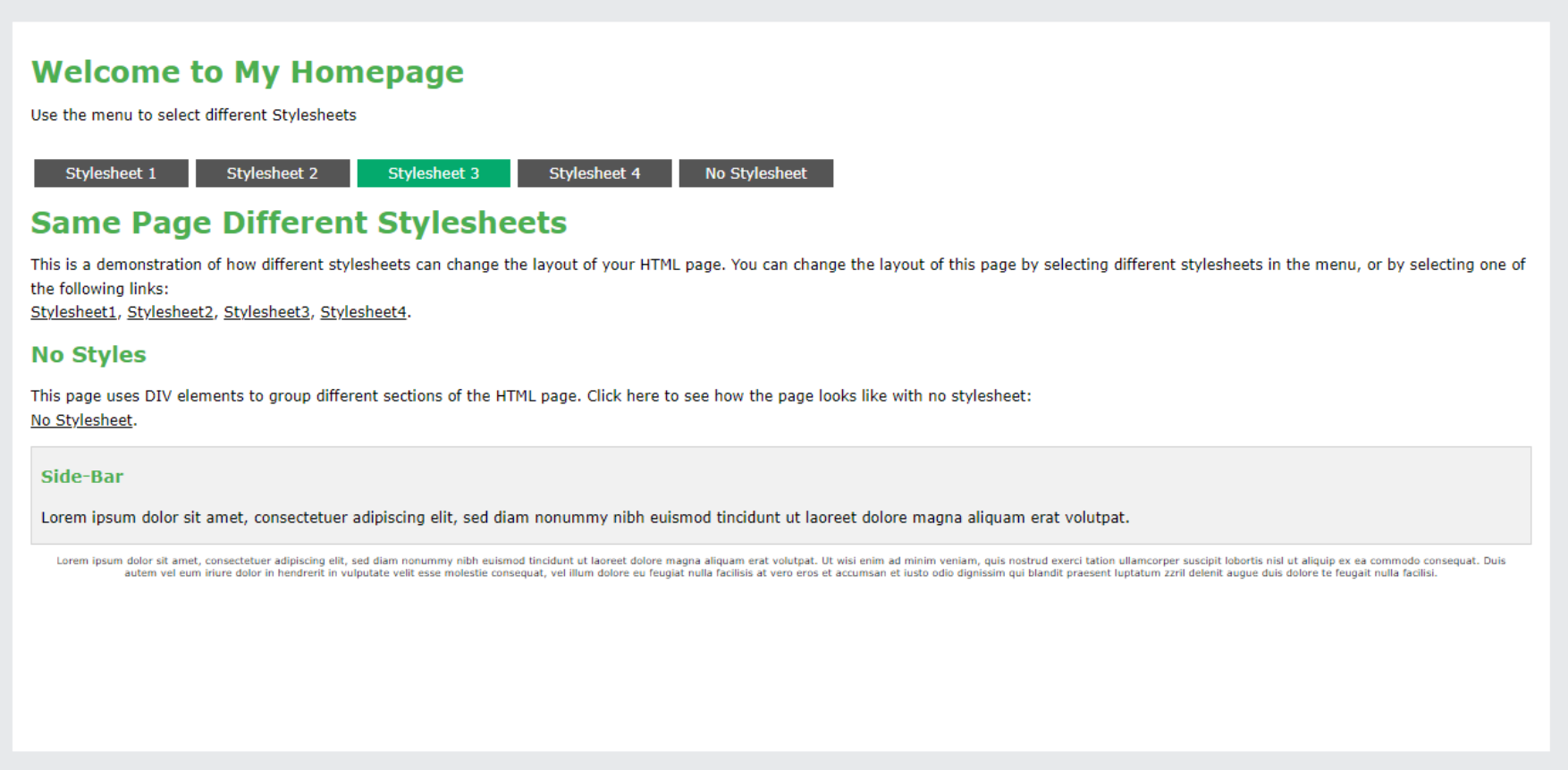
Style 1


Welcome to My Homepage
Use the menu to select different Stylesheets
Stylesheet 1
Stylesheet 2
Stylesheet 3
Stylesheet 4
No Stylesheet
Same Page Different Stylesheets
This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:
Stylesheet1, Stylesheet2, Stylesheet3, Stylesheet4.
No Styles
This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:
No Stylesheet.
Side-Bar
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Style2


Welcome to My Homepage
Use the menu to select different Stylesheets
Same Page Different Stylesheets
This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links: Stylesheet1, Stylesheet2, Stylesheet3, Stylesheet4.
No Styles
This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet: No Stylesheet.
Stylesheet 1
Stylesheet 2
Stylesheet 3
Stylesheet 4
No Stylesheet
Side-Bar
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

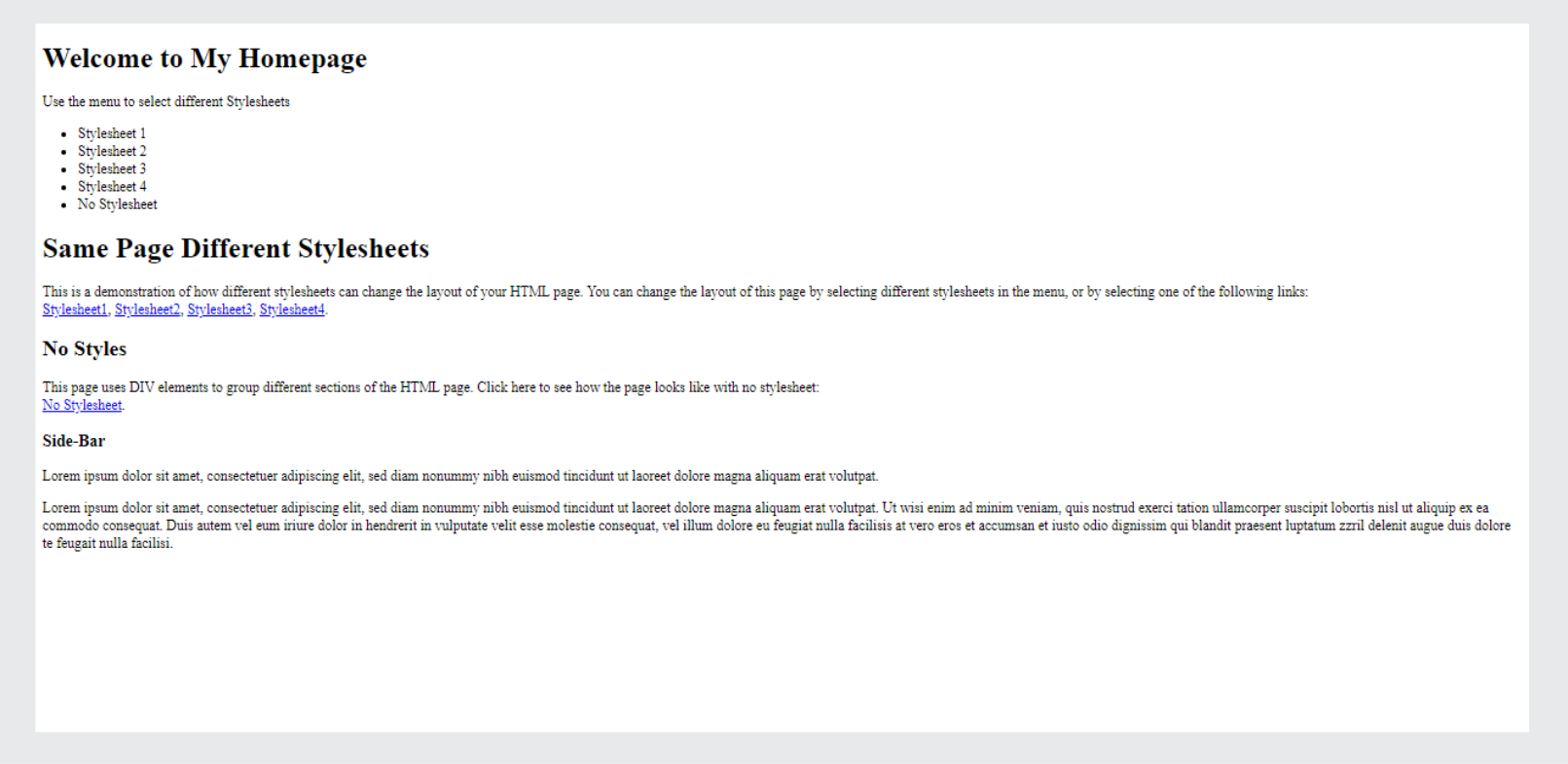
Style3


Welcome to My Homepage
Use the menu to select different Stylesheets
Stylesheet 1
Stylesheet 2
Stylesheet 3
Stylesheet 4
No Stylesheet
Same Page Different Stylesheets
This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:
Stylesheet1, Stylesheet2, Stylesheet3, Stylesheet4.
No Styles
This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:
No Stylesheet.
Side-Bar
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Style4


Welcome to My Homepage
Use the menu to select different Stylesheets
Stylesheet 1
Stylesheet 2
Stylesheet 3
Stylesheet 4
No Stylesheet
Side-Bar
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Same Page Different Stylesheets
This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:
Stylesheet1, Stylesheet2, Stylesheet3, Stylesheet4.
No Styles
This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:
No Stylesheet.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

No Style



Welcome to My Homepage

Use the menu to select different Stylesheets

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:
Stylesheet1, Stylesheet2, Stylesheet3, Stylesheet4.

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:
No Stylesheet.

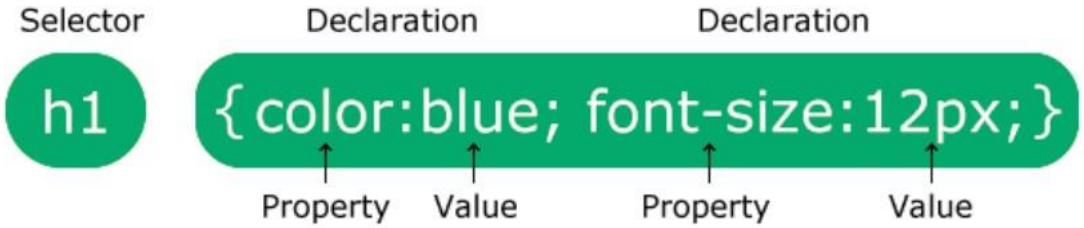
Side-Bar

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

CSS Syntax

- A CSS rule consists of a selector and a declaration block.



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example:

In this example all <p> elements will be center-aligned, with a red text color:

```
p {
  color: red;
  text-align: center;
}
```

Example Explained:

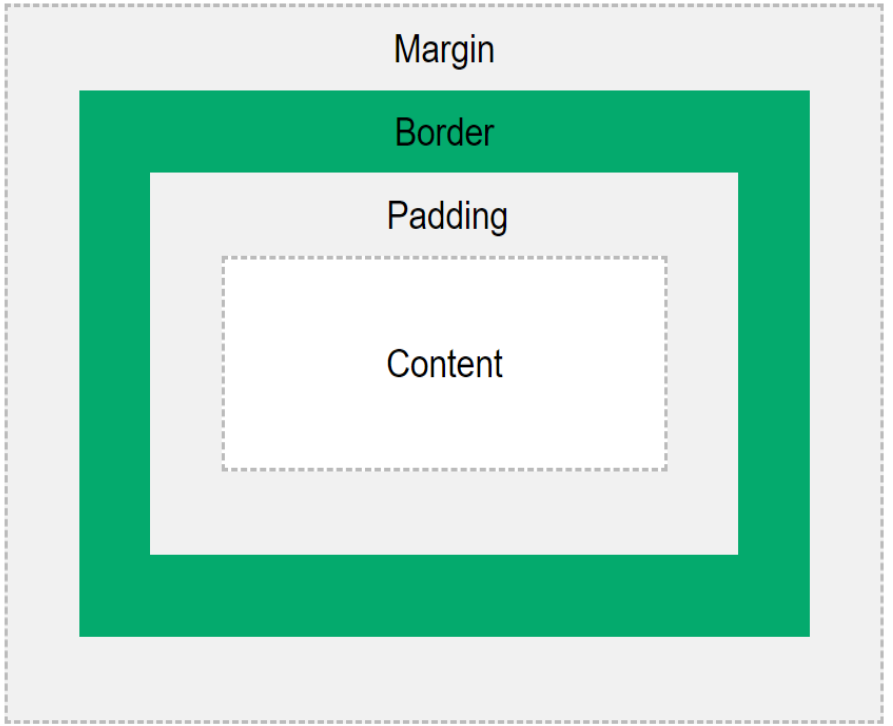
- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
p {  color: red; text-align:
center;}
</style>
</head>
<body>
<p>Hello World!</p>
<p>These paragraphs are styled
with CSS.</p>
</body>
</html>
```

CSS box model

- All HTML elements can be considered as boxes.
- In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element.
- It consists of: margins, borders, padding, and the actual content.
- The image show the box model:



Explanation of the different parts:

Content - The content of the box, where text and images appear

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

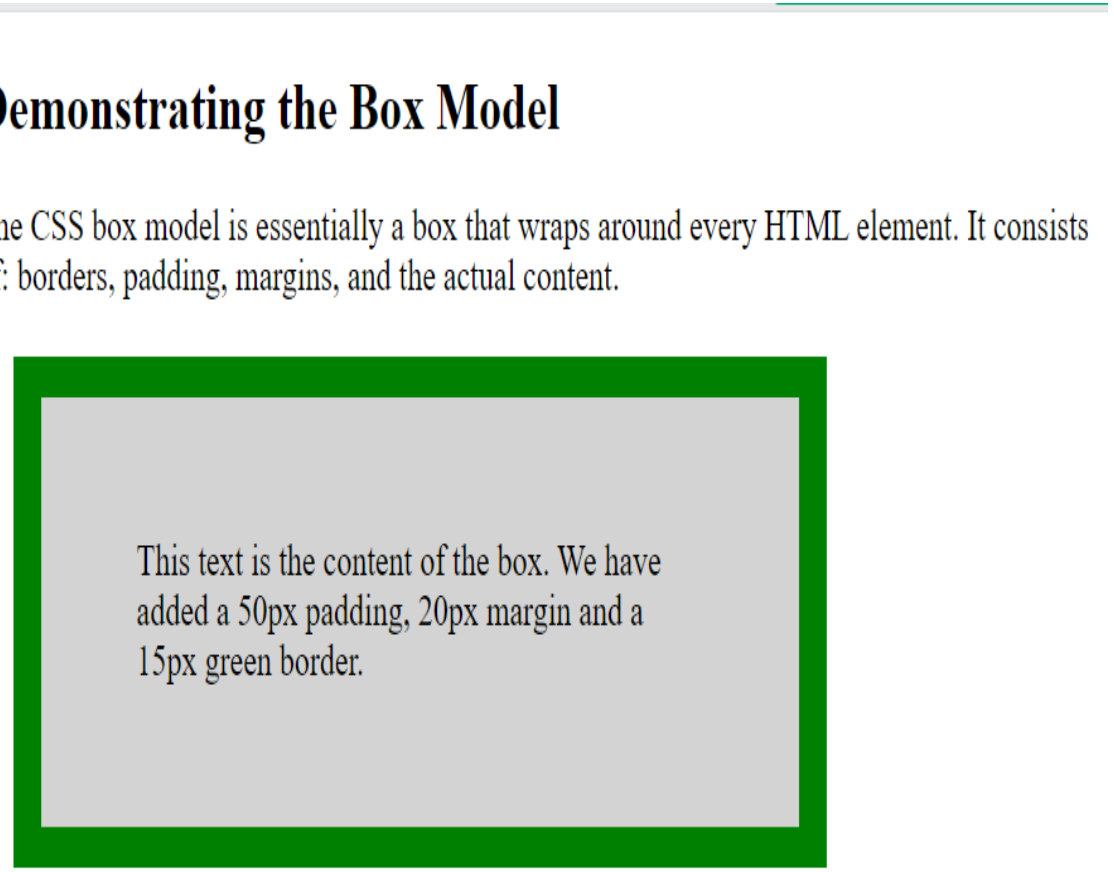
Example:

Demonstration of the box model:

```
div {
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>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>
<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. </div>
</body>
</html>
```



Width and Height of an Element:

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

Example:

This element will have a total width of 350px:

```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
```

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
</style>
</head>
<body>
<h2>Calculate the total width:</h2>
<img src="klematis4_big.jpg" width="350" height="263" alt="Klematis">
<div>The picture above is 350px wide. The total width of this element is
also 350px.</div>
</body>
</html>
```

Calculate the total width:

The picture above is 350px wide. The total width of this element is also 350px.

Here is the calculation:

320px (width) + 20px (left + right padding) + 10px (left + right border) + 0px (left + right margin) = 350px

- The total width of an element should be calculated like this:
- Total element width = width + left padding + right padding + left border + right border + left margin + right margin
- The total height of an element should be calculated like this:
- Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

Important: **When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins.**

Types of Style Sheet

- When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.
- There are three ways of inserting a style sheet:

External CSS
Internal CSS
Inline CSS

External CSS:

- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

Example:

External styles are defined within the <link> element, inside the <head> section of an HTML page:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

- An external style sheet can be written in any text editor, and must be saved with a .css extension.
- The external .css file should not contain any HTML tags.

Here is how the "mystyle.css" file looks:

"mystyle.css"

```
body {  background-color: lightblue; }
h1 {  color: navy;  margin-left: 20px; }
```

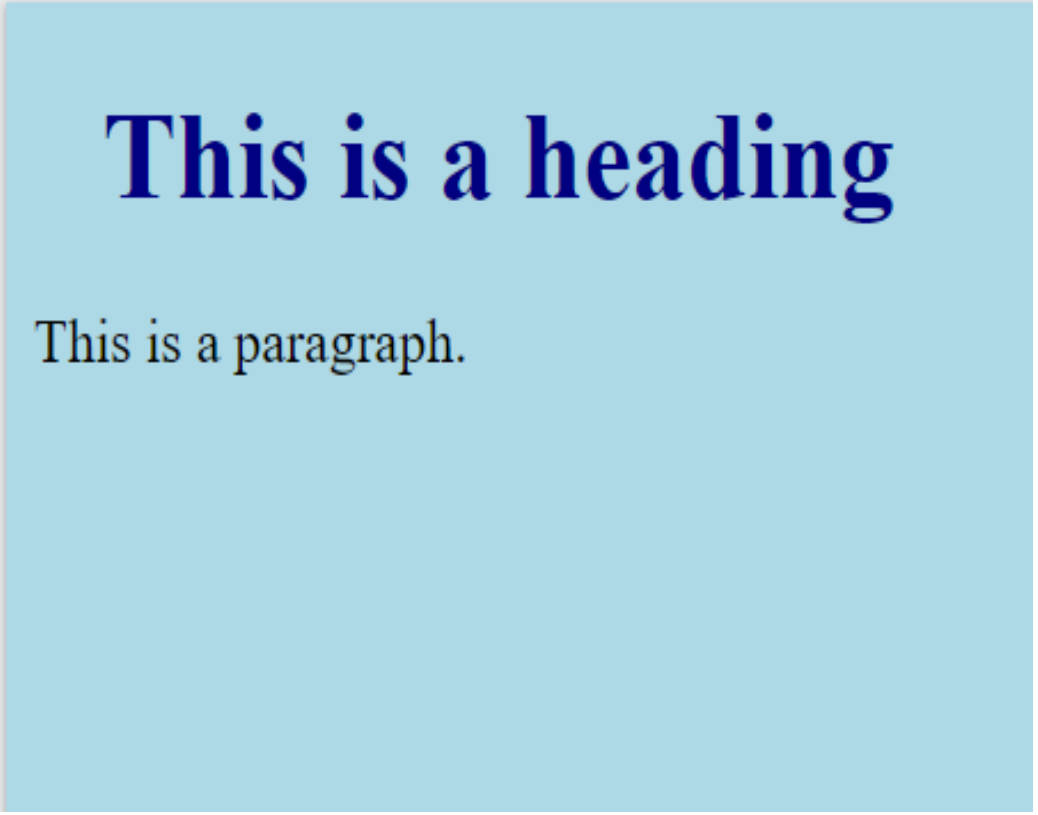
Note: Do not add a space between the property value and the unit:

Incorrect (space): margin-left: 20 px;

Correct (no space): margin-left: 20px;

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet"
href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```



Internal CSS:

- An internal style sheet may be used if one single HTML page has a unique style.
- The internal style is defined inside the <style> element, inside the head section.

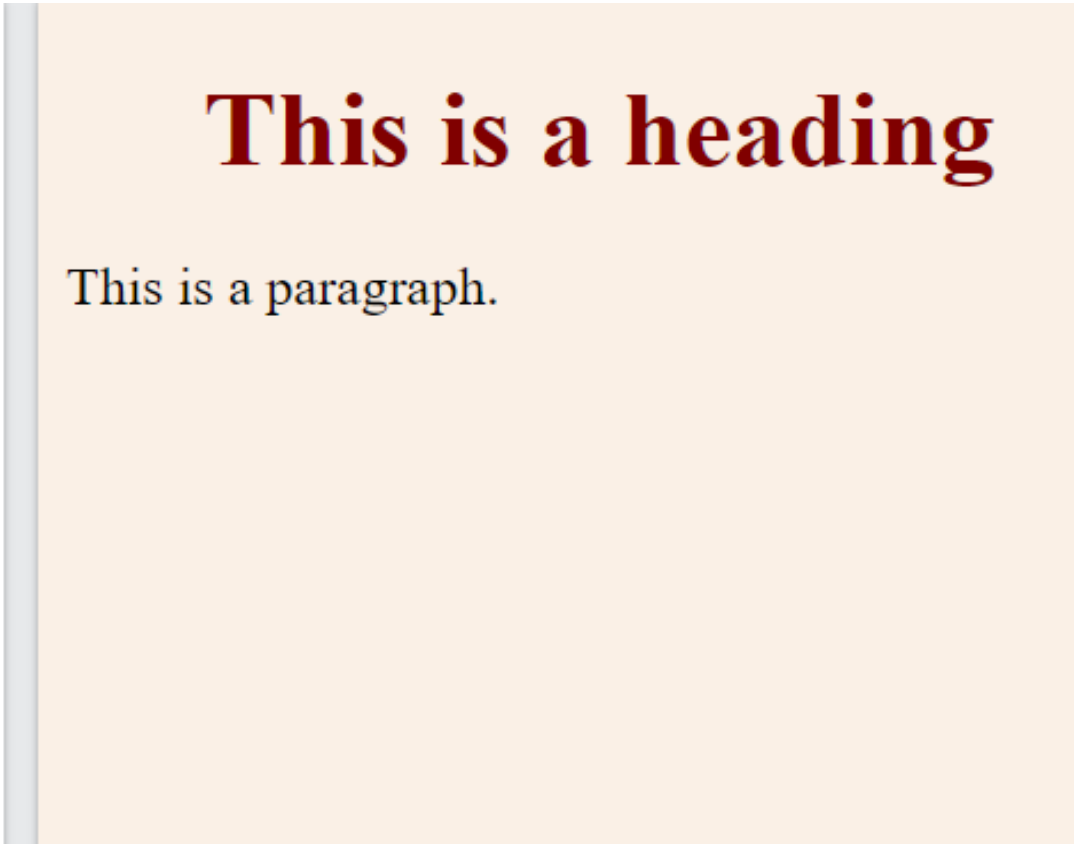
Example:

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<head>
<style>
body {  background-color: linen; }
h1 {  color: maroon;  margin-left: 40px; }
</style>
</head>
```

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
body { background-color: linen;}
h1 { color: maroon; margin-left:
40px;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```



Inline CSS:

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Example:

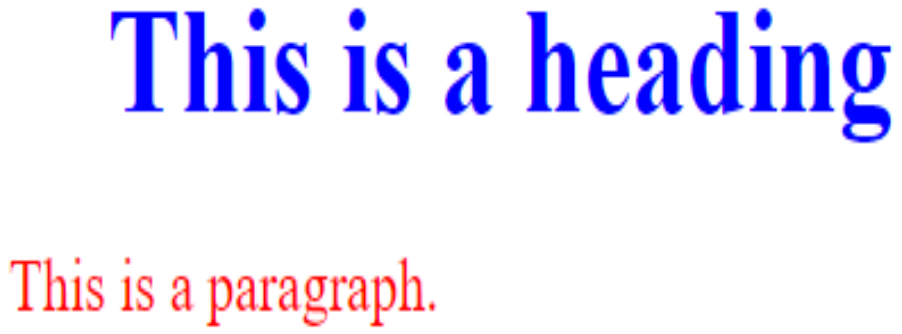
Inline styles are defined within the "style" attribute of the relevant element:

```
<body>
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
</body>
```

Tip: An inline style loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly.

Example with Output

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;text-
align:center;">This is a
heading</h1>
<p style="color:red;">This is a
paragraph.</p>
</body>
</html>
```



CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

The CSS element Selector :

The element selector selects HTML elements based on the element name.

Example:

Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p {
  text-align: center;
  color: red;
}
```

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
p {text-align: center;color: red;}
</style>
</head>
<body>
<p>Every paragraph will be
affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

Every paragraph will be affected by the style.

Me too!

And me!

The CSS id Selector:

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example:

The CSS rule below will be applied to the HTML element with id="para1":

```
#para1 {
  text-align: center;
  color: red;
}
```

Note: An id name cannot start with a number!

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {text-align: center; color:
red;}
</style>
</head>
<body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected
by the style.</p>
</body>
</html>
```

Hello World!

This paragraph is not affected by the style.

The CSS class Selector:

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.

Note: A class name cannot start with a number!

Example:

In this example all HTML elements with class="center" will be red and center-aligned:

```
.center {
  text-align: center;
  color: red;
}
```

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {text-align: center;color:
red; }
</style>
</head>
<body>
<h1 class="center">Red and center-
aligned heading</h1>
<p class="center">Red and center-
aligned paragraph.</p>
</body>
</html>
```

You can also specify that only specific HTML elements should be affected by a class.

Example:

In this example only <p> elements with class="center" will be red and center-aligned:

```
p.center {
  text-align: center;
  color: red;
}
```

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {text-align: center;
color: red;}
</style>
</head>
<body>
<h1 class="center">This heading
will not be affected</h1>
<p class="center">This paragraph
will be red and center-aligned.
</p>
</body>
</html>
```

HTML elements can also refer to more than one class.

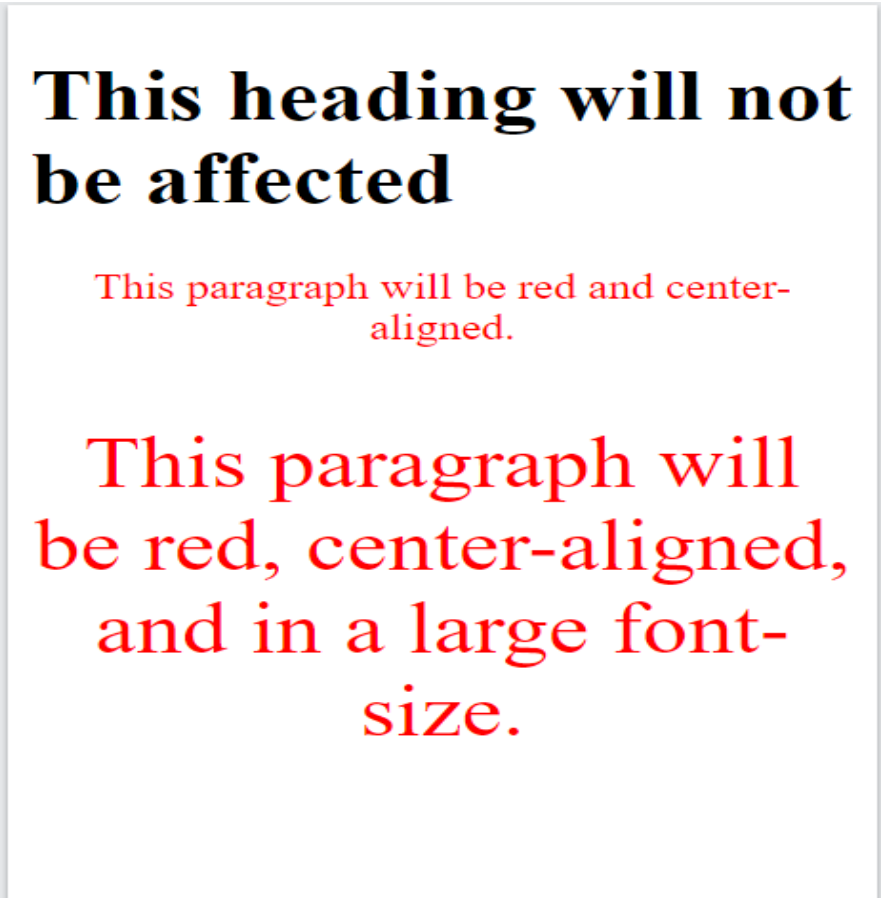
Example:

In this example the element will be styled according to class="center" and to class="large":

This paragraph refers to two classes.

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {text-align: center;
color: red;}
p.large {font-size: 200%;}
</style>
</head>
<body>
<h1 class="center">This heading
will not be affected</h1>
<p class="center">This paragraph
will be red and center-aligned.
</p>
<p class="center large">This
paragraph will be red, center-
aligned, and in a large font-size.
</p>
</body>
</html>
```



The CSS Universal Selector:

The universal selector (*) selects all HTML elements on the page.

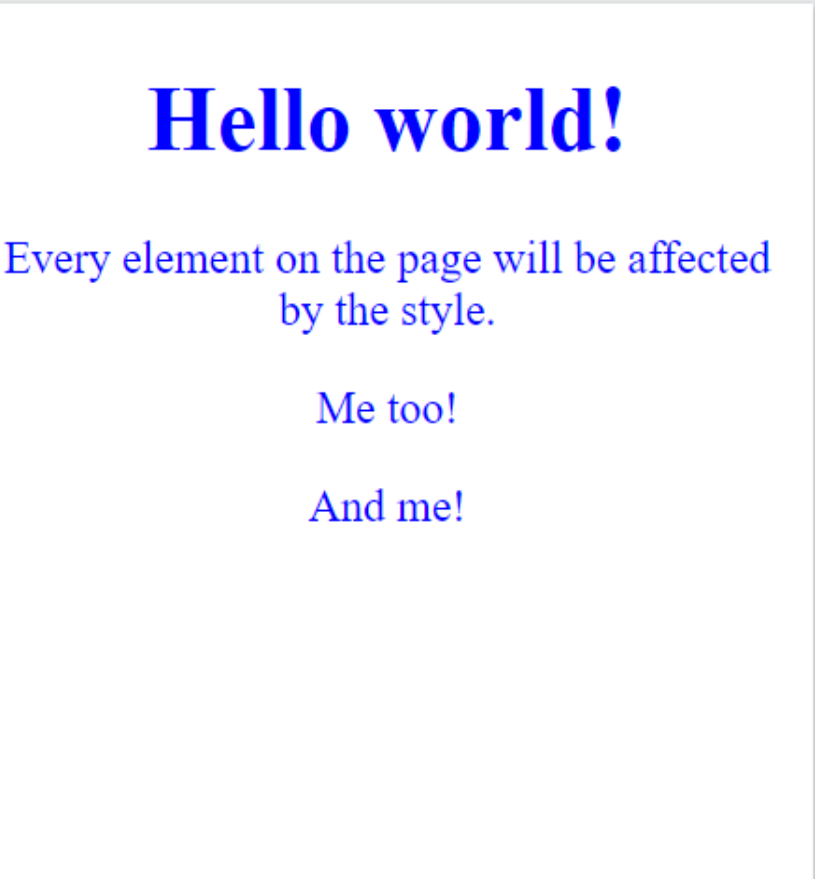
Example

The CSS rule below will affect every HTML element on the page:

```
* {
  text-align: center;
  color: blue;
}
```

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
* {text-align: center; color:
blue; }
</style>
</head>
<body>
<h1>Hello world!</h1>
<p>Every element on the page will
be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```



The CSS Grouping Selector:

The grouping selector selects all the HTML elements with the same style definitions.
Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1 { text-align: center; color: red; }
h2 { text-align: center; color: red; }
p { text-align: center; color: red; }
```

It will be better to group the selectors, to minimize the code.
To group selectors, separate each selector with a comma.

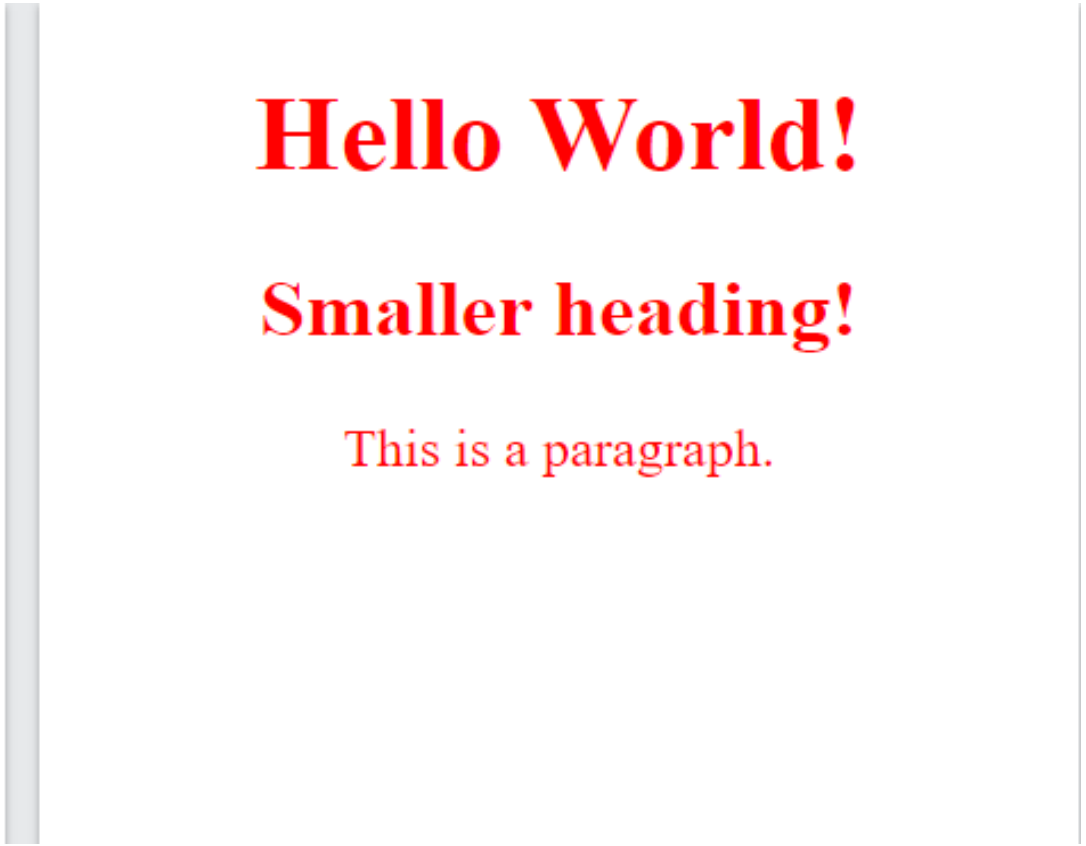
Example:

In this example we have grouped the selectors from the code above:

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

Example with Output

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {text-align: center;
color: red; }
</style>
</head>
<body>
<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>
</body>
</html>
```



All CSS Simple Selectors

Selector	Example	Example description
#id	#firstname	Selects the element with id="firstname"
.class	.intro	Selects all elements with class="intro"
element.class	p.intro	Selects only <p> elements with class="intro"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element,..	div, p	Selects all elements and all <p> elements

CSS Comments

- CSS comments are not displayed in the browser, but they can help document your source code.
- Comments are used to explain the code, and may help when you edit the source code at a later date.
- Comments are ignored by browsers.
- A CSS comment is placed inside the <style> element, and starts with /* and ends with */.

Example:

```
/* This is a single-line comment */
p { color: red; }
```

You can add comments wherever you want in the code:

Example:

```
p { color: red;  /* Set text color to red */ }
```

Comments can also span multiple lines:

Example:

```
/* This is
a multi-line
comment */
p { color: red; }
```

CSS Colors

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

CSS Background Color

You can set the background color for HTML elements:

Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

CSS Text Color:

You can set the color of text:

Hello World

You can set any color for your text like blue....

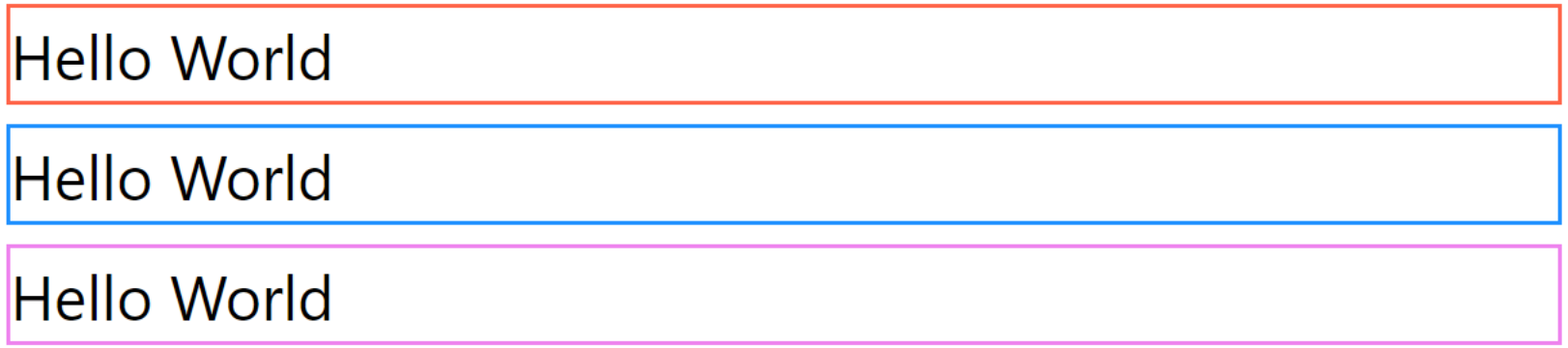
You can set any color for your text like green.....

Example

```
<h1 style="color:Tomato;">Hello World....</h1>
<p style="color:Blue;"> Hello World...</p>
<p style="color:MediumSeaGreen;"> Hello World....</p>
```

CSS Border Color:

You can set the color of borders:



Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

```
<!DOCTYPE html>
<html>
<body>
<h3 style="color:Tomato;">Hello
World</h3>
<p style="color:DodgerBlue;">
Colors are specified using
predefined color names, or RGB,
HEX, HSL, RGBA, HSLA values.</p>
</body>
</html>
```

CSS Backgrounds

- The CSS background properties are used to add background effects for elements.
- In these chapters, you will learn about the following CSS background properties:
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position
  - background (shorthand property)

CSS background-color:

The background-color property specifies the background color of an element.

Example
The background color of a page is set like this:

```
body {
  background-color: lightblue;
}
```

With CSS, a color is most often specified by:

a valid color name - like "red"
a HEX value - like "#ff0000"
an RGB value - like "rgb(255,0,0)"

Other Elements:

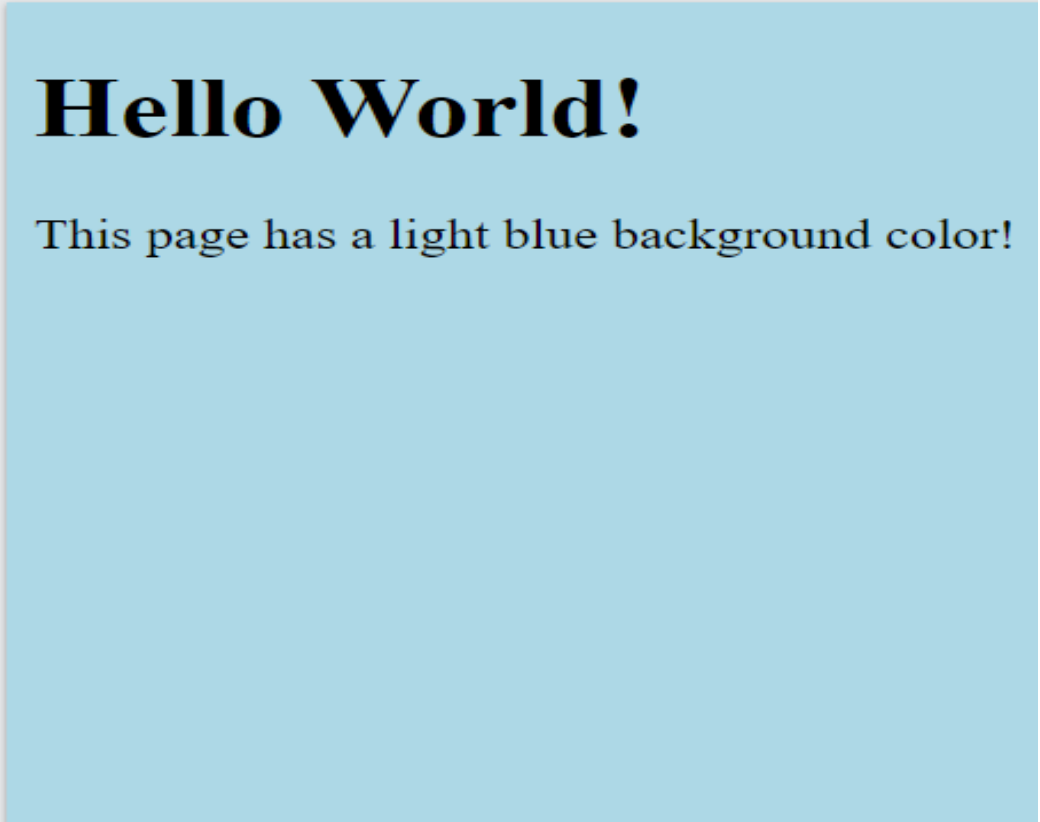
You can set the background color for any HTML elements:

Example:

Here, the <h1>, <p>, and elements will have different background colors:

```
h1 { background-color: green; }
div { background-color: lightblue; }
p { background-color: yellow; }
```

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>This page has a light blue
background color!</p>
</body>
</html>
```



CSS background-image:

The background-image property specifies an image to use as the background of an element.
By default, the image is repeated so it covers the entire element.

Example:
Set the background image for a page:

```
body {
  background-image: url("paper.jpg");
}
```

Note: When using a background image, use an image that does not disturb the text.

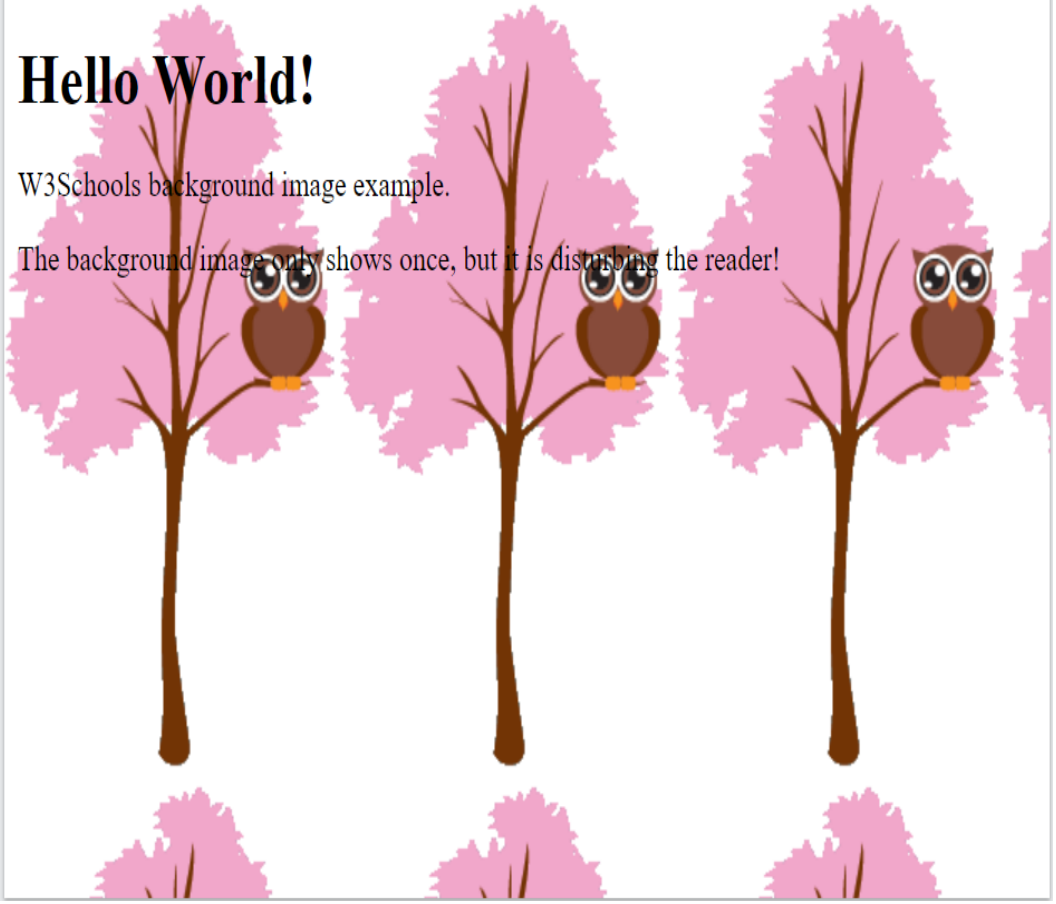
CSS background-repeat:

By default, the background-image property repeats an image both horizontally and vertically.
Some images should be repeated only horizontally or vertically, or they will look strange, like this:

Example

```
body {
  background-image: url("bg.png");
        }
```

```html
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>W3Schools background image example.</p>
<p>The background image only shows once, but it is disturbing the reader!
</p>
</body>
</html>
```



If the image above is repeated only horizontally (background-repeat: repeat-x;), the background will look better:

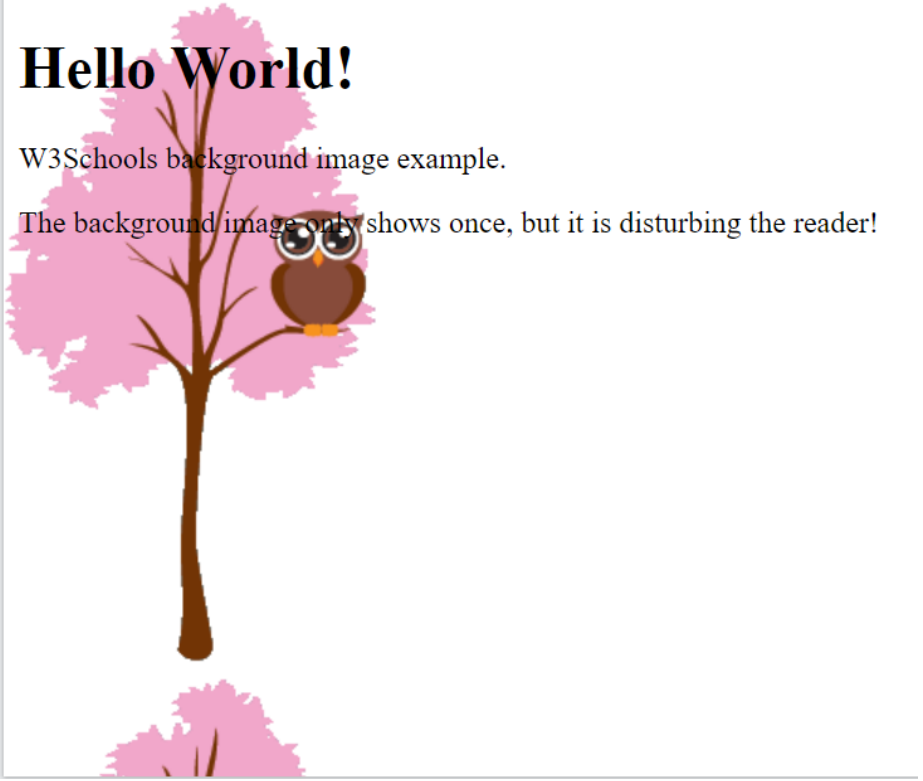
Example:

```
body {
  background-image: url("bg.png");
  background-repeat: repeat-x;
        }
```

To repeat an image vertically, set background-repeat: repeat-y;

```html
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat:repeat-x;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>W3Schools background image example.</p>
<p>The background image only shows once, but it is disturbing the reader!
</p>
</body>
</html>
```

```html
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat:repeat-y;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>W3Schools background image example.</p>
<p>The background image only shows once, but it is disturbing the reader!
</p>
</body>
</html>
```



CSS background-repeat: no-repeat

Showing the background image only once is also specified by the background-repeat property:

Example:

Show the background image only once:

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
}
```

```html
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image:
url("img_tree.png");
  background-repeat: no-repeat;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>W3Schools background image
example.</p>
<p>The background image only shows
once, but it is disturbing the
reader!</p>
</body>
</html>
```

CSS background-position:

The background-position property is used to specify the position of the background image.

Example:

Position the background image in the top-right corner:

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-image:
url("img_tree.png"); background-
repeat: no-repeat; background-
position: right top;margin-
right: 200px; }
</style>
</head>
<body>
<p>Here, the background image is
only shown once. In addition it
is positioned away from the
text.</p>
<p>In this example we have also
added a margin on the right
side, so that the background
image will not disturb the text.
</p>
</body>
</html>
```

Here, the background image is only shown once. In addition it is positioned away from the text.

In this example we have also added a margin on the right side, so that the background image will not disturb the text.

The CSS Background Repeat and Position Properties.

Property	Description
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated

CSS background - Shorthand property:

To shorten the code, it is also possible to specify all the background properties in one single property. This is called a shorthand property.
Instead of writing:

```
body {
  background-color: #ffffff;
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

You can use the shorthand property background.

Example:

Use the shorthand property to set the background properties in one declaration:

```
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
}
```

```html
<!DOCTYPE html>
<html>
<head>
<style>
body {background: #ffffff url("img_tree.png") no-repeat right top;
margin-right: 200px;
}
</style>
</head>
<body>
<h1>The background Property</h1>
<p>The background property is a shorthand property for specifying all the background properties in one declaration.</p>
<p>Here, the background image is only shown once, and it is also positioned in the top-right corner.</p>
<p>We have also added a right margin, so that the text will not write over the background image.</p>
</body>
</html>
```

The background Property

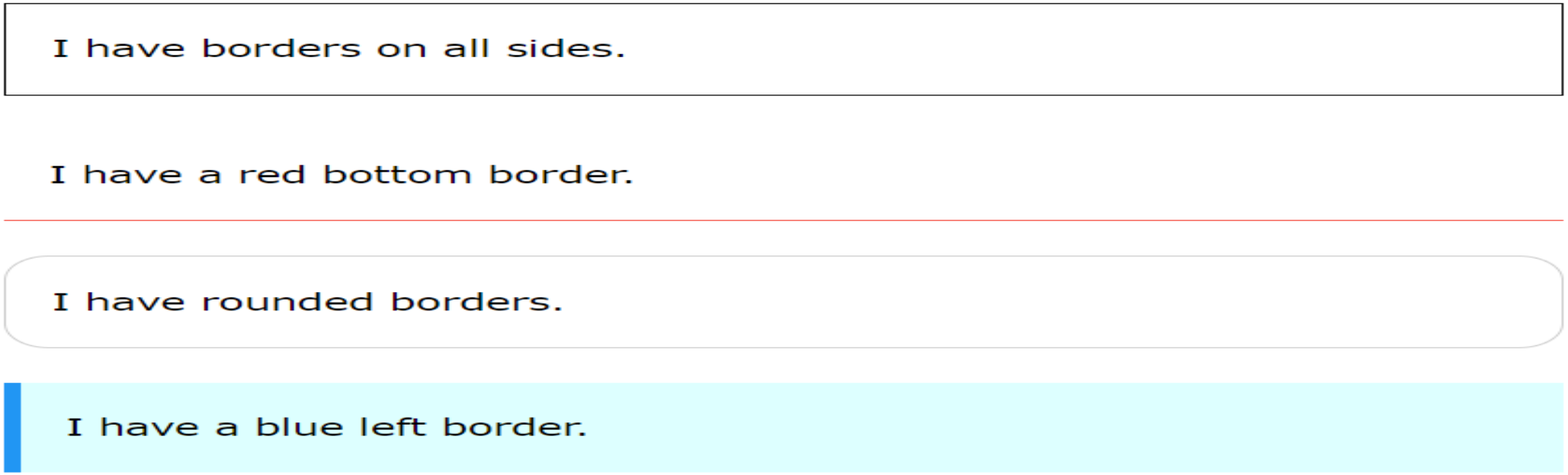
The background property is a shorthand property for specifying all the background properties in one declaration.

Here, the background image is only shown once, and it is also positioned in the top-right corner.

We have also added a right margin, so that the text will not write over the background image.

CSS Borders

The CSS border properties allow you to specify the style, width, and color of an element's border.



CSS Border Style

The border-style property specifies what kind of border to display.
The following values are allowed:

- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- groove - Defines a 3D grooved border. The effect depends on the border-color value

•ridge - Defines a 3D ridged border. The effect depends on the border-color value
•inset - Defines a 3D inset border. The effect depends on the border-color value
•outset - Defines a 3D outset border. The effect depends on the border-color value
•none - Defines no border
•hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example:

Demonstration of the different border styles:

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted
dashed solid double;}
```

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

CSS Border Color

The border-color property is used to set the color of the four borders.

The color can be set by:

- name - specify a color name, like "red"
- HEX - specify a HEX value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"

Note: If border-color is not set, it inherits the color of the element.

Example:

Demonstration of the different border colors:

```
p.one { border-style: solid; border-color: red; }

p.two { border-style: solid;  border-color: green; }

p.three { border-style: dotted;  border-color: blue; }
```

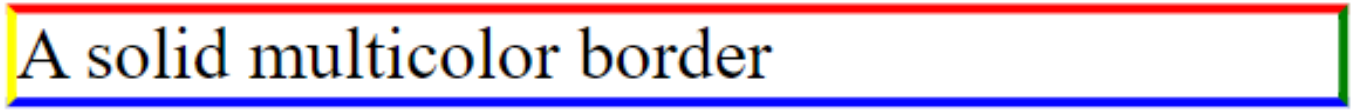

Red border
Green border
Blue border

Specific Side Colors:

The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example:

```
p.one { border-style: solid; border-color: red green blue yellow; /* red top, green right, blue bottom and yellow left */
}
```



CSS Border - Shorthand Property

Like you saw in the previous page, there are many properties to consider when dealing with borders.
To shorten the code, it is also possible to specify all the individual border properties in one property.
The border property is a shorthand property for the following individual border properties:

- border-width

border-style (required)

- border-color

Example

```
p {
  border: 5px solid red;
}
```

Result:

Some text

CSS Margins

Margins are used to create space around elements, outside of any defined borders.

The CSS margin properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

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

All the margin properties can have the following values:

- auto - the browser calculates the margin
- *length* - specifies a margin in px, pt, cm, etc.
- % - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

Tip: Negative values are allowed.

Example:

Set different margins for all four sides of a <p> element:

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

Margin - Shorthand Property

To shorten the code, it is possible to specify all the margin properties in one property.
The margin property is a shorthand property for the following individual margin properties:

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

margin: 25px 50px 75px 100px;

- top margin is 25px
- right margin is 50px
- bottom margin is 75px
- left margin is 100px

Example:

Use the margin shorthand property with four values:

```
p {
  margin: 25px 50px 75px 100px;
}
```

CSS Padding

Padding is used to create space around an element's content, inside of any defined borders.
The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

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

All the padding properties can have the following values:

- *length* - specifies a padding in px, pt, cm, etc.
- *%* - specifies a padding in % of the width of the containing element
- inherit - specifies that the padding should be inherited from the parent element

Note: Negative values are not allowed.

Example:

Set different padding for all four sides of a element:

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```

Padding - Shorthand Property

To shorten the code, it is possible to specify all the padding properties in one property.
The padding property is a shorthand property for the following individual padding properties:

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

padding: 25px 50px 75px 100px;

top padding is 25px
right padding is 50px
bottom padding is 75px
left padding is 100px

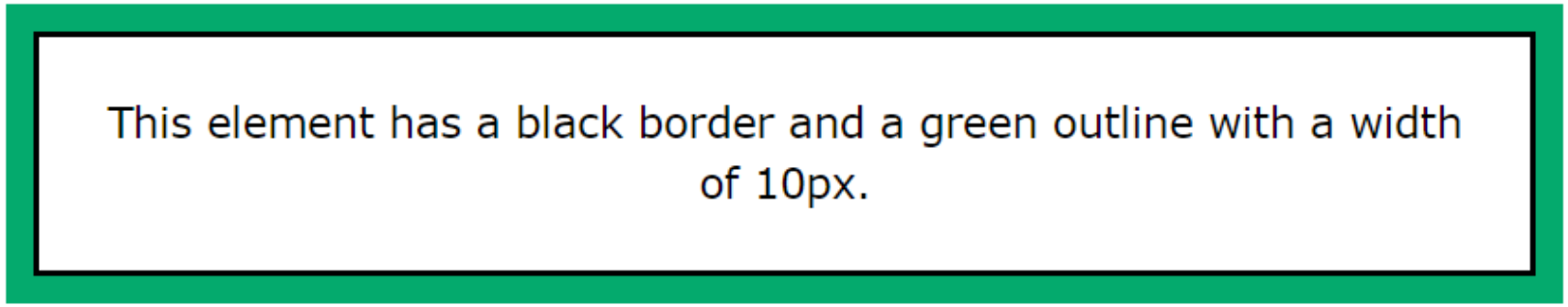
Example:

Use the padding shorthand property with four values:

```
div {
  padding: 25px 50px 75px 100px;
}
```

CSS Outline

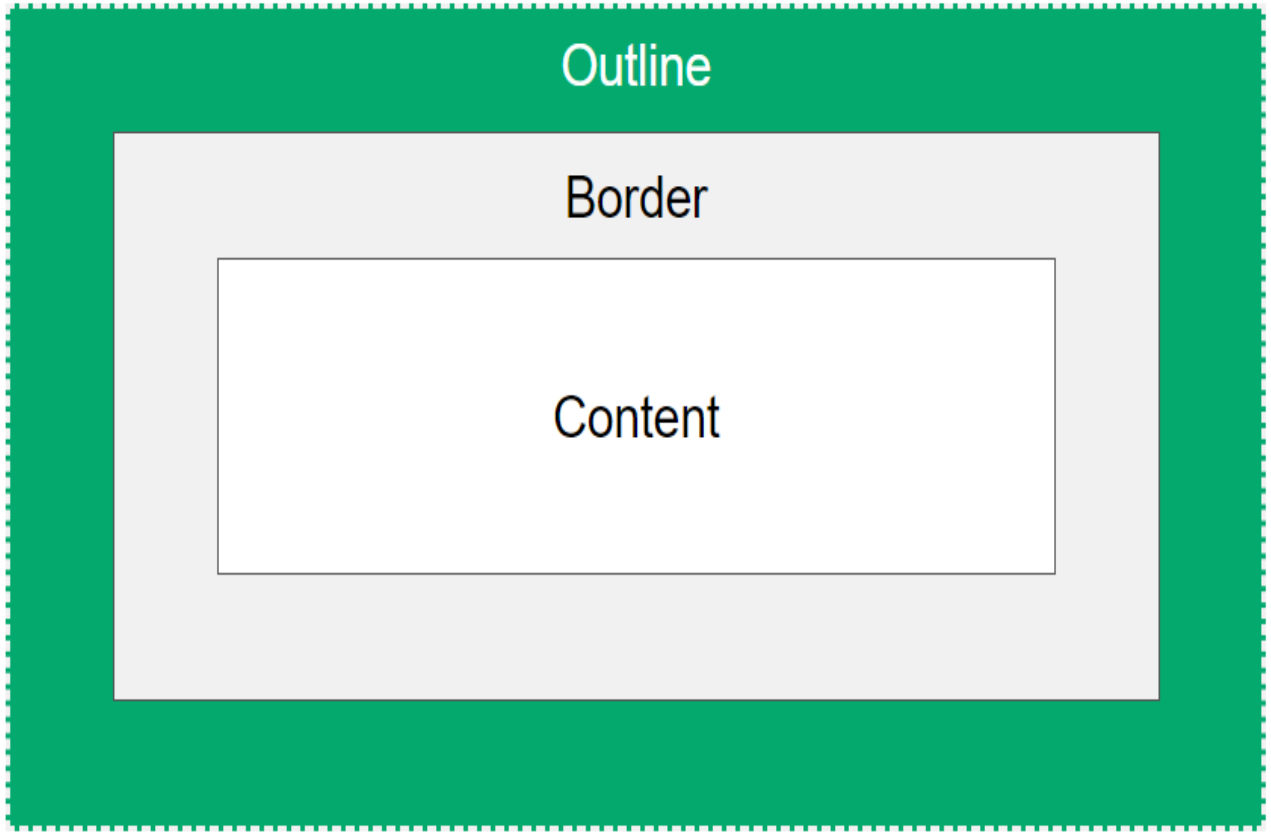
An outline is a line drawn outside the element's border.



An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".

CSS has the following outline properties:

- outline-style
- outline-color
- outline-width
- outline-offset
- outline



CSS Outline Style

The outline-style property specifies the style of the outline, and can have one of the following values:

- dotted - Defines a dotted outline
- dashed - Defines a dashed outline
- solid - Defines a solid outline
- double - Defines a double outline
- groove - Defines a 3D grooved outline
- ridge - Defines a 3D ridged outline
- inset - Defines a 3D inset outline
- outset - Defines a 3D outset outline
- none - Defines no outline
- hidden - Defines a hidden outline

The following example shows the different outline-style values:

Example:

Demonstration of the different outline styles:

```
p.dotted {outline-style: dotted;}
p.dashed {outline-style: dashed;}
p.solid {outline-style: solid;}
p.double {outline-style: double;}
p.groove {outline-style: groove;}
p.ridge {outline-style: ridge;}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
```

A dotted outline.

A dashed outline.

A solid outline.

A double outline.

A groove outline. The effect depends on the outline-color value.

A ridge outline. The effect depends on the outline-color value.

An inset outline. The effect depends on the outline-color value.

An outset outline. The effect depends on the outline-color value.

CSS Text

CSS has a lot of properties for formatting text.
This text is styled with some of the text formatting properties.
The heading uses the text-align, text-transform, and color properties.
The paragraph is indented, aligned, and the space between characters is specified.

Text Color

The color property is used to set the color of the text. The color is specified by:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

Look at CSS Color Values for a complete list of possible color values.

The default text color for a page is defined in the body selector.

Text Color and Background Color

In this example, we define both the background-color property and the color property:

Example

```
body {
  background-color: lightgrey;
  color: blue;
}
```

Text Alignment and Text Direction

In this chapter you will learn about the following properties:

- text-align
- text-align-last
- direction
- unicode-bidi
- vertical-align

Text Alignment

The text-align property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left)

Example

```
h1 {
  text-align: center;
}

h2 {
  text-align: left;
}

h3 {
  text-align: right;
}
```

The CSS Text Alignment/Direction Properties

Property	Description
direction	Specifies the text direction/writing direction
text-align	Specifies the horizontal alignment of text
text-align-last	Specifies how to align the last line of a text
unicode-bidi	Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document
vertical-align	Sets the vertical alignment of an element

CSS Fonts

CSS Font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more. You have already studied how to make text bold or underlined. Here, you will also know how to resize your font using percentage.

These are some important font attributes:

1. CSS Font color: This property is used to change the color of the text. (standalone attribute)
2. CSS Font family: This property is used to change the face of the font.
3. CSS Font size: This property is used to increase or decrease the size of the font.
4. CSS Font style: This property is used to make the font bold, italic or oblique.

CSS Font Color

- CSS font color is a standalone attribute in CSS although it seems that it is a part of CSS fonts.
- It is used to change the color of the text.
- There are three different formats to define a color:
  - By a color name
  - By hexadecimal value
  - By RGB

```
<style>
body {
    font-size: 100%;
}
h1 { color: red; }
h2 { color: #9000A1; }
p { color:rgb(0, 220, 98); }
}
</style>
```

CSS Font Family

- Choosing the right font has a huge impact on how the readers experience a website.
- The right font can create a strong identity for your brand.
- Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

Generic Font Families

In CSS there are five generic font families:

- Serif: fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
- Sans-serif: fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
- Monospace fonts: here all the letters have the same fixed width. They create a mechanical look.
- Cursive: fonts imitate human handwriting.
- Fantasy: fonts are decorative/playful fonts.

All the different font names belong to one of the generic font families.

Some Font Examples

Generic Font Family	Examples of Font Names
Serif	Times New Roman ; Georgia ; Garamond
Sans-serif	Arial ; Verdana ; Helvetica
Monospace	Courier New ; Lucida Console ; Monaco
Cursive	Brush Script MT ; Lucida Handwriting
Fantasy	Copperplate ; Papyrus

The CSS font-family Property:

In CSS, we use the font-family property to specify the font of a text.

Example:

Specify some different fonts for three paragraphs:

```
.p1 { font-family: "Times New Roman", Times, serif; }
.p2 { font-family: Arial, Helvetica, sans-serif; }
.p3 { font-family: "Lucida Console", "Courier New", monospace; }
```

Note: If the font name is more than one word, it must be in quotation marks, like: "Times New Roman".

CSS Web Safe Fonts

What are Web Safe Fonts?
Web safe fonts are fonts that are universally installed across all browsers and devices.

Fallback Fonts
However, there are no 100% completely web safe fonts. There is always a chance that a font is not found or is not installed properly.
Therefore, it is very important to always use fallback fonts.
This means that you should add a list of similar "backup fonts" in the font-family property. If the first font does not work, the browser will try the next one, and the next one, and so on. Always end the list with a generic font family name.

Best Web Safe Fonts for HTML and CSS

The following list are the best web safe fonts for HTML and CSS:

Arial (sans-serif)

Verdana (sans-serif)

Tahoma (sans-serif)

Trebuchet MS (sans-serif)

Times New Roman (serif)

Georgia (serif)

Garamond (serif)

Courier New (monospace)

Brush Script MT (cursive)

Note: Before you publish your website, always check how your fonts appear on different browsers and devices, and always use fallback fonts!

CSS Font Property

To shorten the code, it is also possible to specify all the individual font properties in one property.
The font property is a shorthand property for:

- font-style
- font-variant
- font-weight
- font-size/line-height
- font-family

Note: The font-size and font-family values are required. If one of the other values is missing, their default value are used.

Example:
Use font to set several font properties in one declaration:

```
p.a { font: 20px Arial, sans-serif; }
p.b { font: italic small-caps bold 12px/30px Georgia, serif; }
```

CSS Layout - The position Property

The position property specifies the type of positioning method used for an element.
There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

position: static;

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This element has position: static;

Example:

```
div.static {
    position: static;
    border: 3px solid #73AD21;
}
```

position: relative;

An element with position: relative; is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

This element has position: relative;

Example

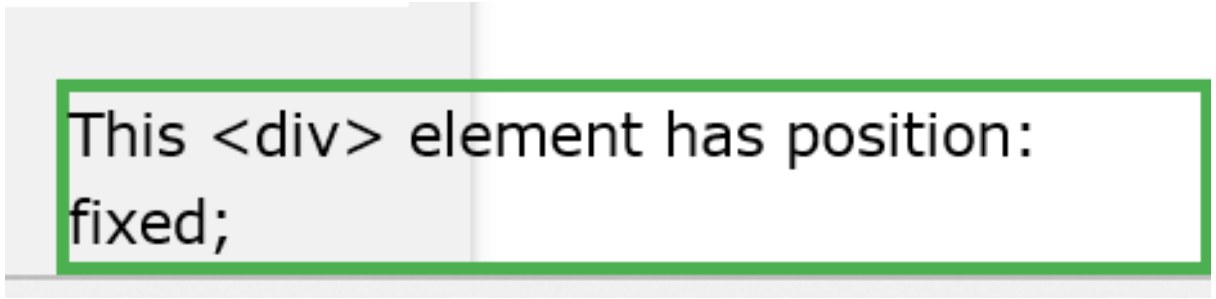
```
div.relative {
    position: relative;
    left: 30px;
    border: 3px solid #73AD21;
}
```

position: fixed;

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:



Example

```
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
}
```

position: absolute;

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed). However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Note: A "positioned" element is one whose position is anything except static.

Example

```
div.relative {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;
}

div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
}
```