SRI RAMACHANDRA

INSTITUTE OF HIGHER EDUCATION AND RESEARCH

(Category - I Deemed to be University) Porur, Chennai

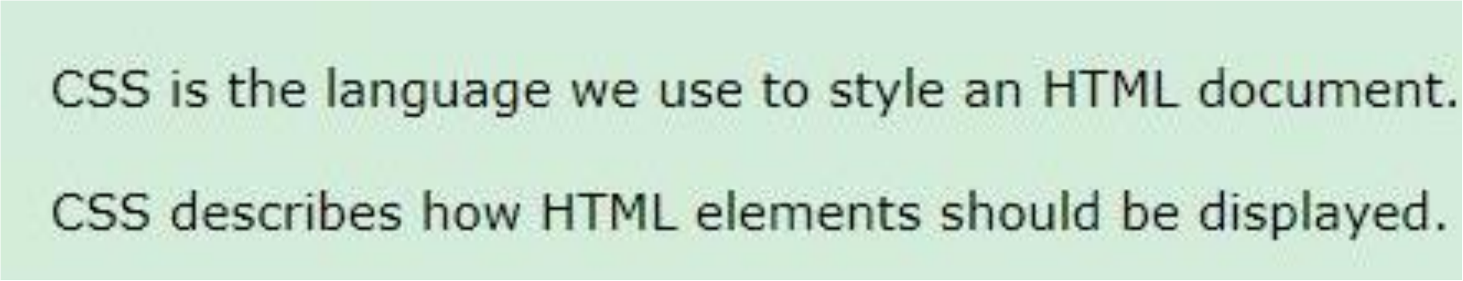
SRI RAMACHANDRA ENGINEERING AND TECHNOLOGY

CSE 150 WEB DESIGN – FRONTEND DEVELOPMENT

MODULE 2

Faculty : Dr. G. Shiyamalagowri

CSS



CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

CSS – Introduction

CSS is the language we use to style a Web page.

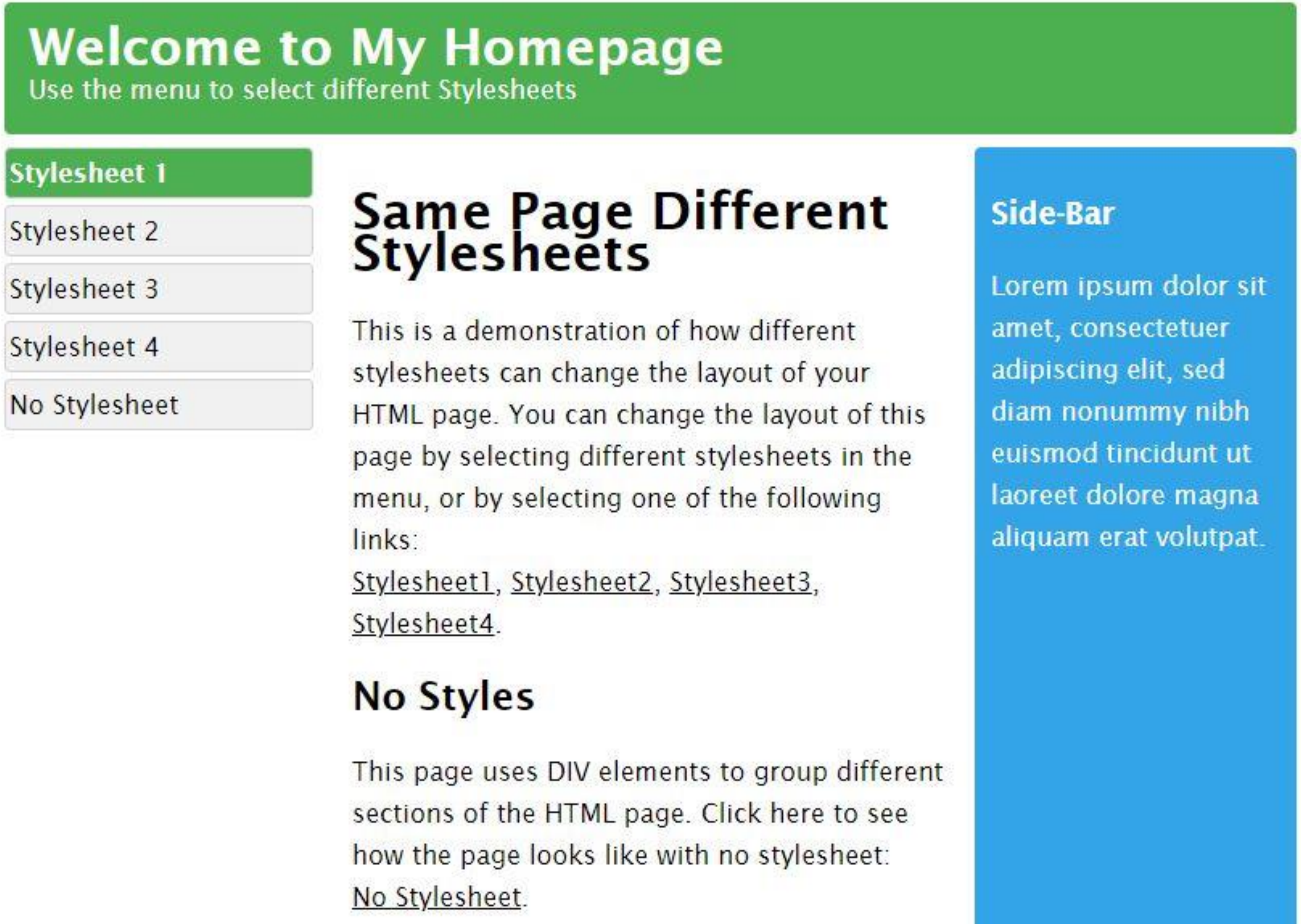
What is CSS?

- 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

CSS Demo One HTML Page – Multiple Sheets



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.

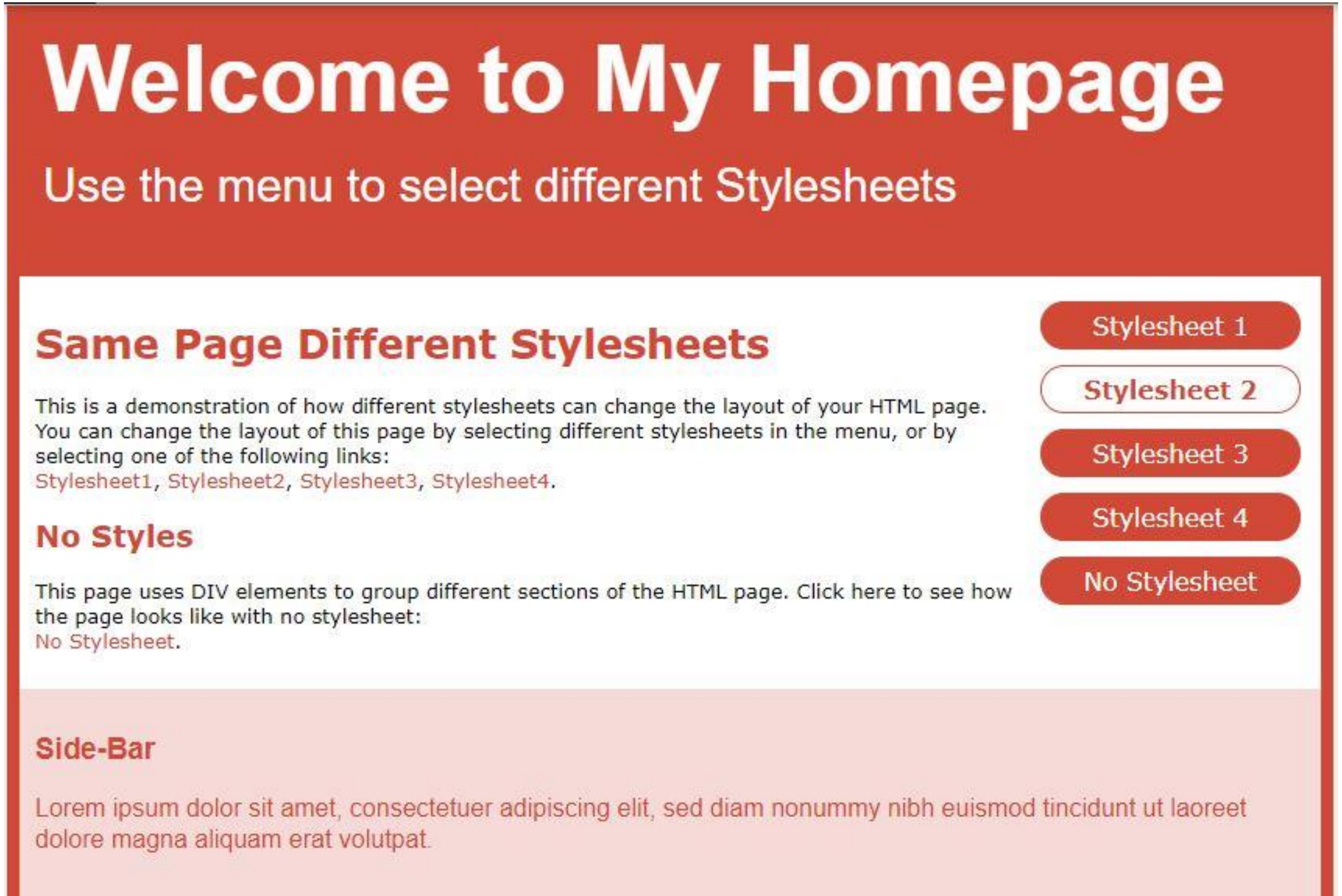
No Styles

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.

Side-Bar

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

CSS Demo One HTML Page – Multiple Sheets



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.

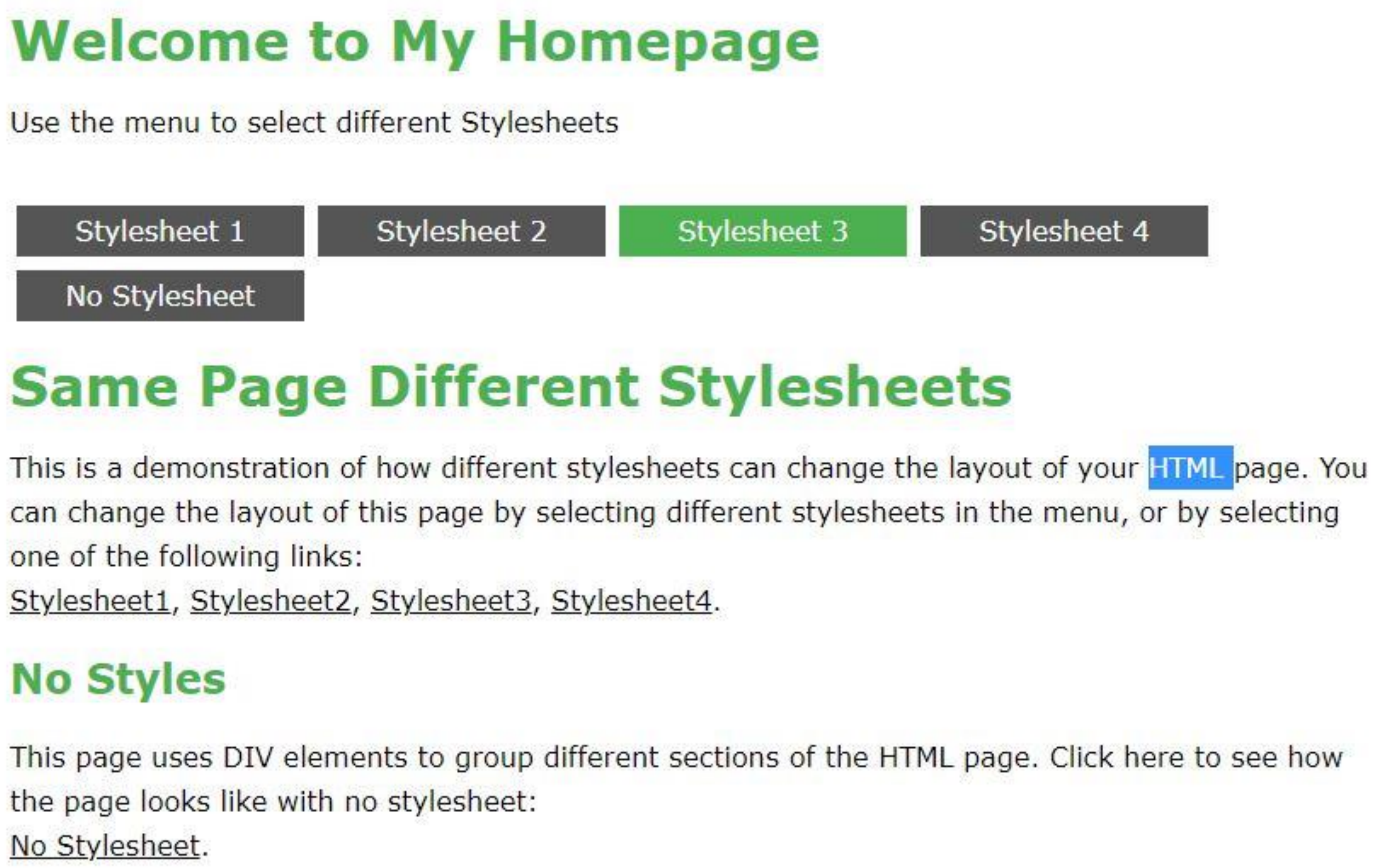
No Styles

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.

Side-Bar

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

CSS Demo One HTML Page – Multiple Sheets



Welcome to My Homepage

Use the menu to select different Stylesheets

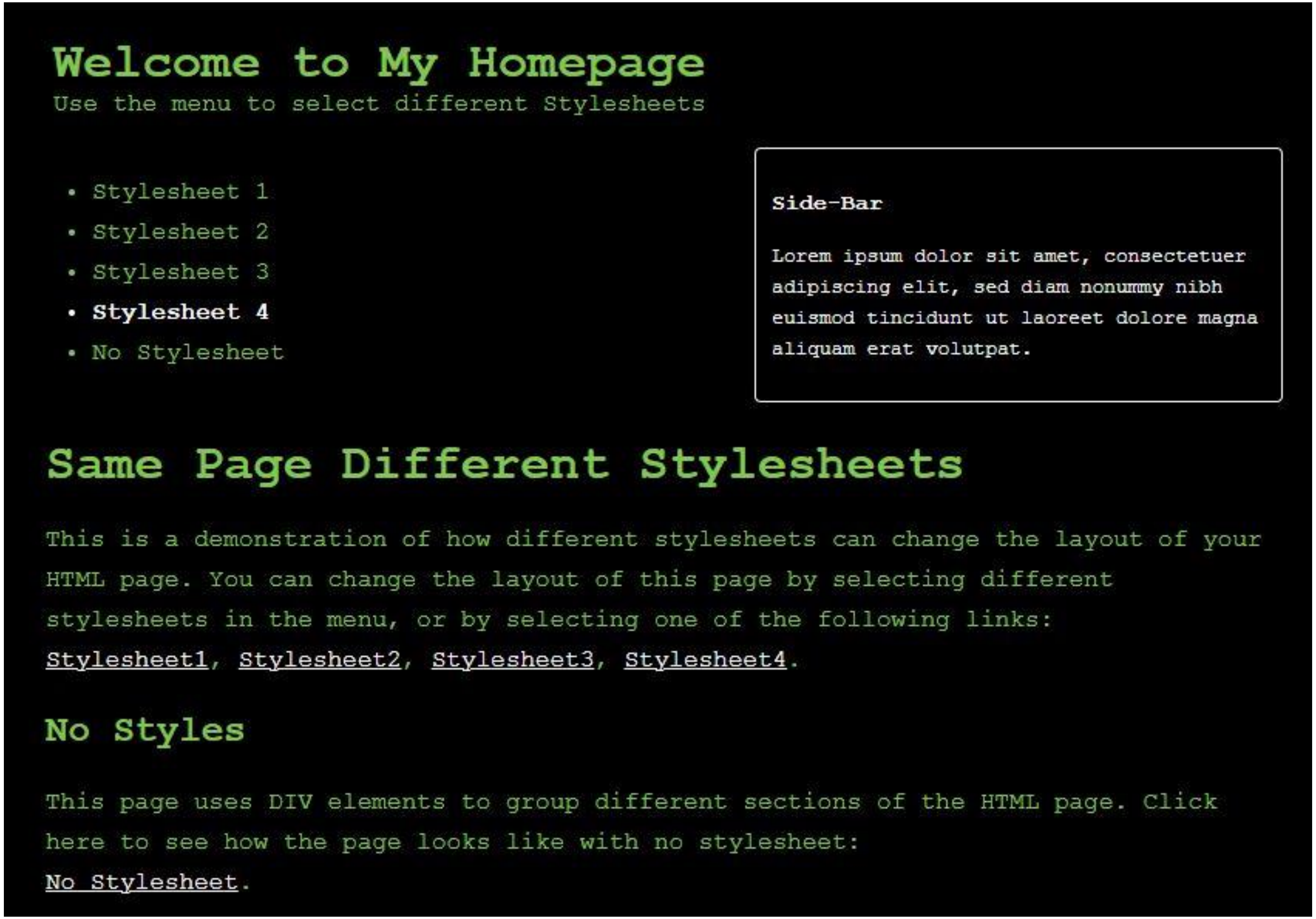
Same Page Different Stylesheets

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.

CSS Demo One HTML Page – Multiple Sheets



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.

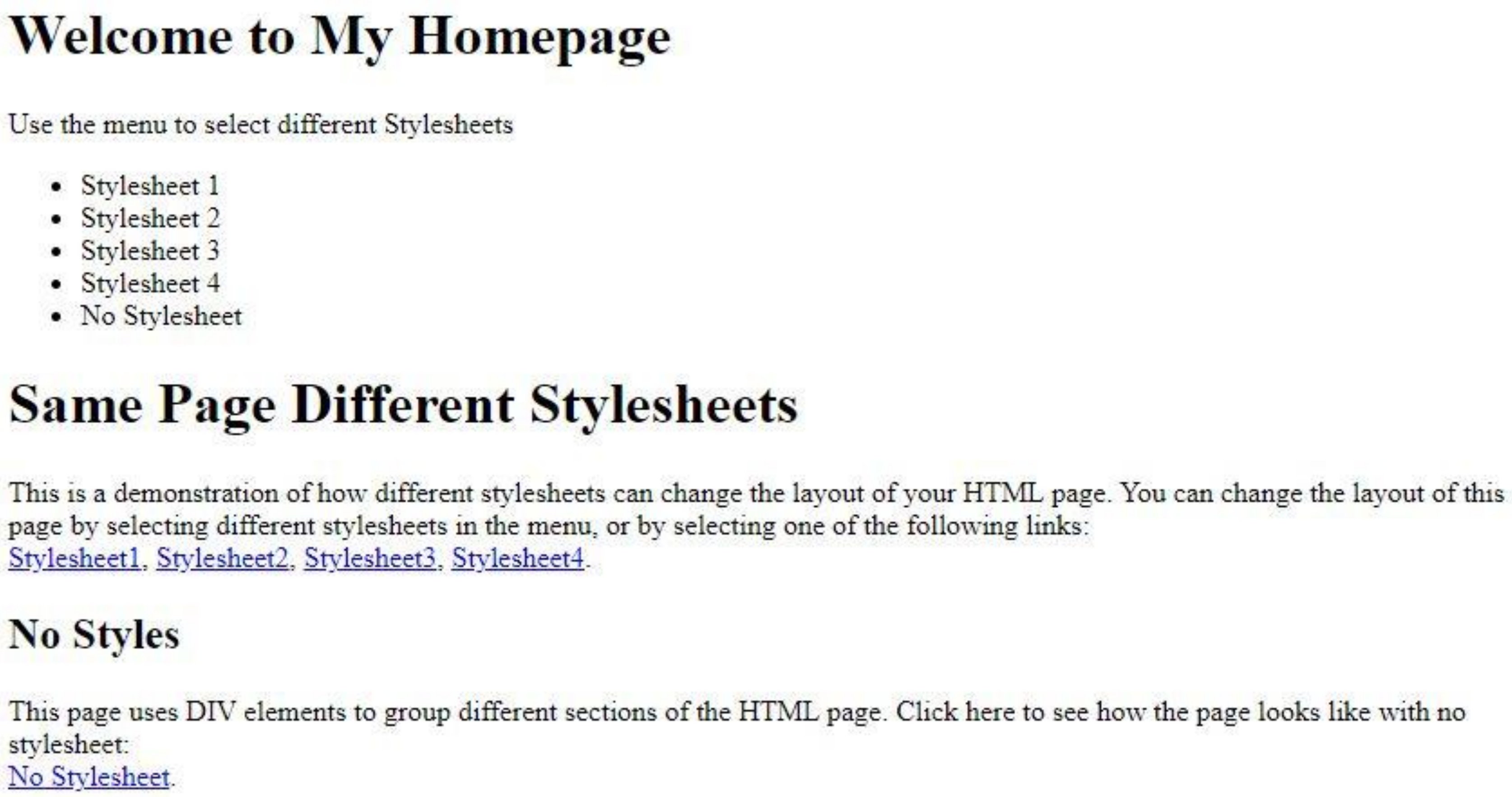
Same Page Different Stylesheets

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.

CSS Demo One HTML Page – Multiple Sheets



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.

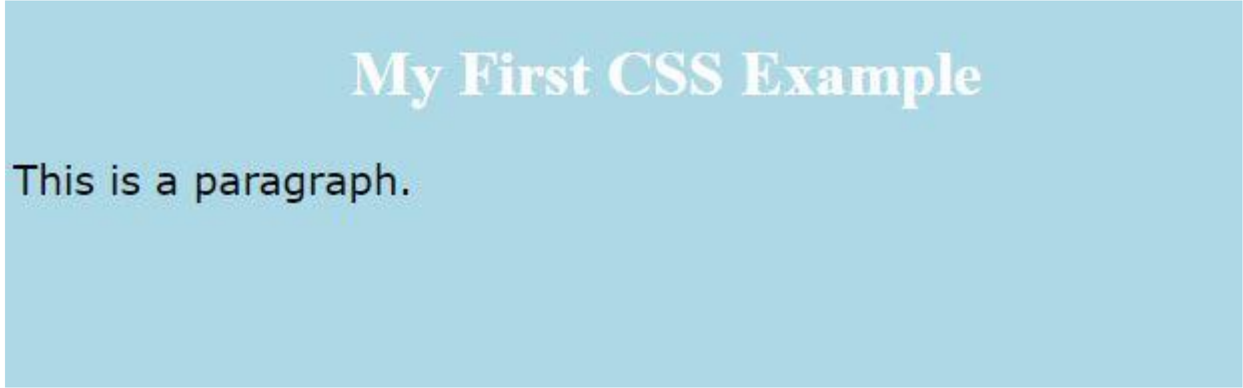
Why Use CSS?

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

CSS Example

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
```

```
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

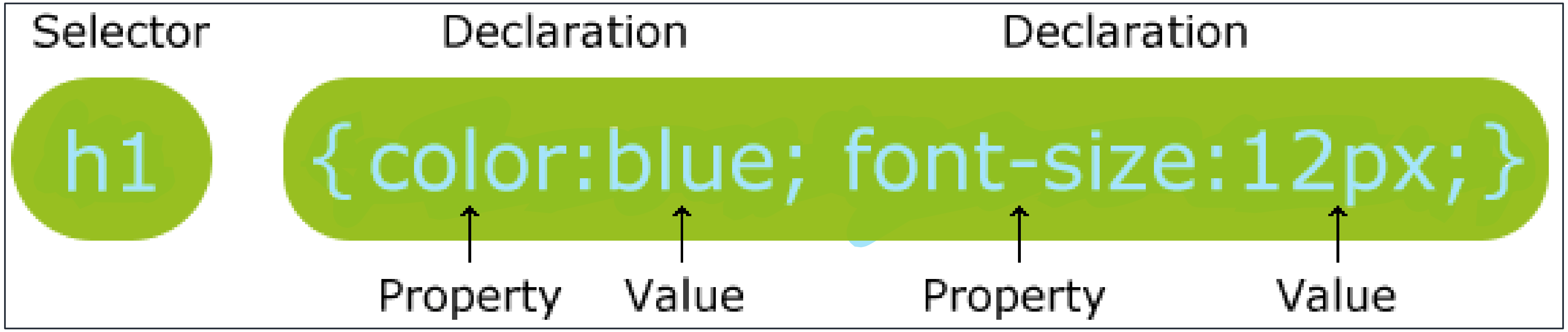


My First CSS Example

This is a paragraph.

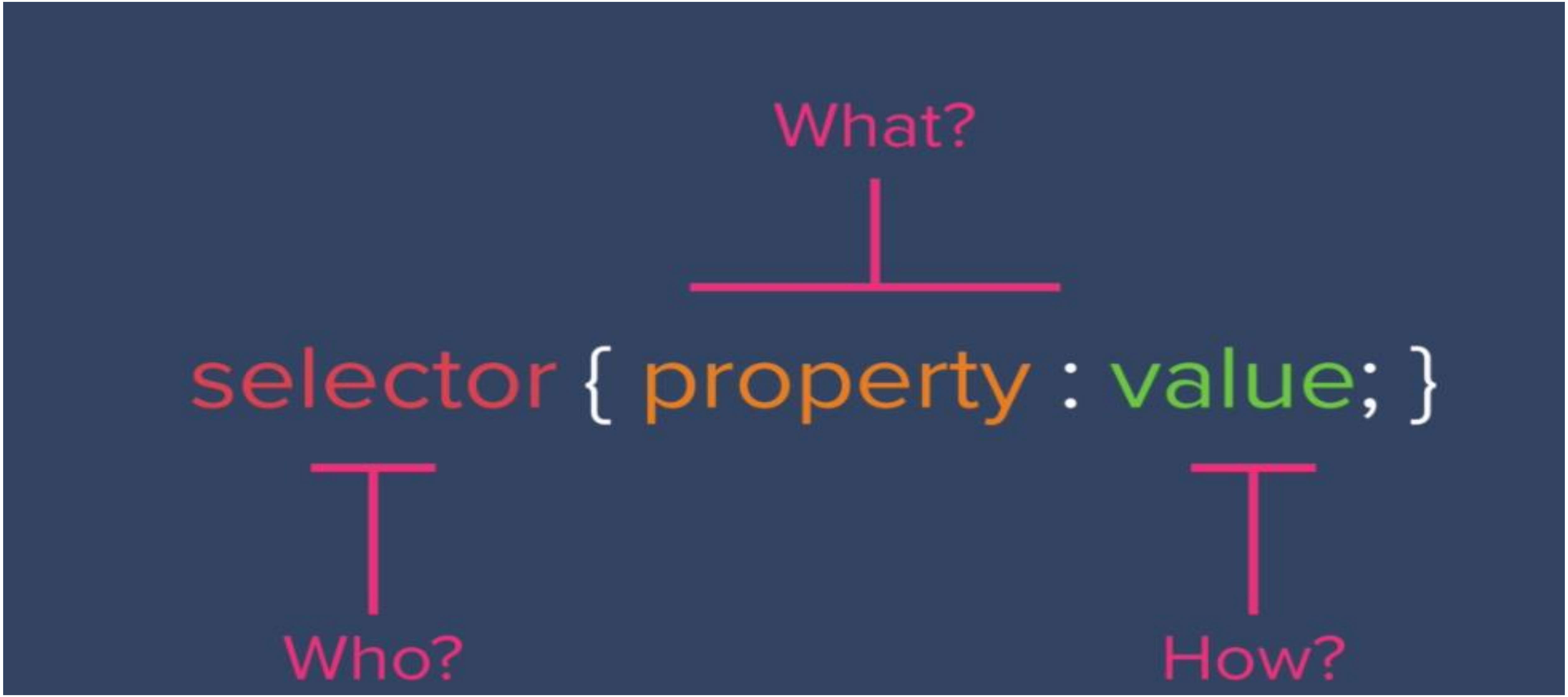
CSS Syntax

A CSS rule set 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 property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS Syntax



Challenge

- Create a new folder called CSS - My Site
- Create an index.html
- Add the HTML backbone and give your website a title of your name
- Create a css folder and a styles.css file
- Link up your html and css and give the background an ugly blue colour.

Using CSS

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the style attribute inside HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using a <link> element to link to an external CSS file

Inline CSS

- An inline CSS is used to apply a unique style to a single HTML element.
- An inline CSS uses the style attribute of an HTML element.
- The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

```
<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>
```

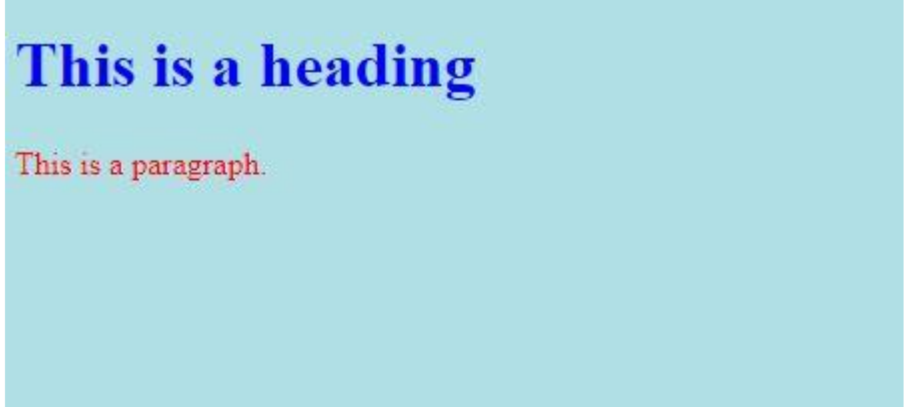
A Blue Heading

A red paragraph.

Internal CSS

- An internal CSS is used to define a style for a single HTML page.
- An internal CSS is defined in the <head> section of an HTML page, within a <style> element.
- The following example sets the text color of ALL the <h1> elements (on that page) to blue.
- The text color of ALL the <p> elements to red.
- In addition, the page will be displayed with a "powderblue" background color:

```
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a heading

This is a paragraph.

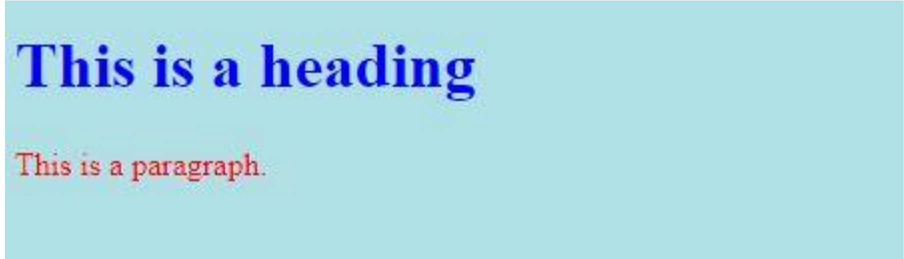
External CSS

- An external style sheet is used to define the style for many HTML pages.
- To use an external style sheet, add a link to it in the <head> section of each HTML page:

```html
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

"styles.css"

```css
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

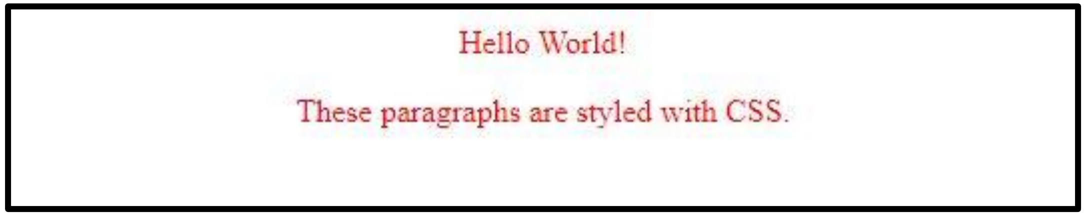


This is a heading

This is a paragraph.

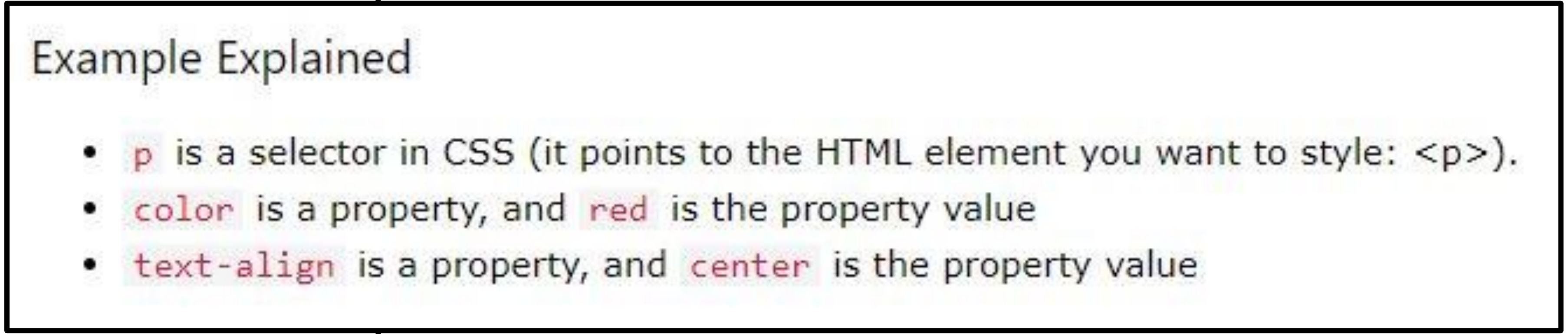
CSS Example

```
<!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>
```



Hello World!

These paragraphs are styled with CSS.



Example Explained

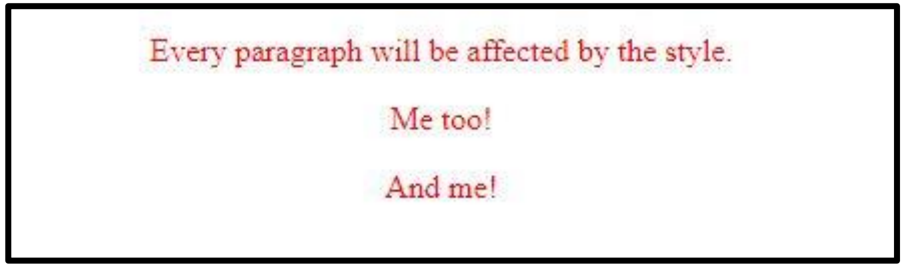
- `p` is a selector in CSS (it points to the HTML element you want to style: <p>).
- `color` is a property, and `red` is the property value
- `text-align` is a property, and `center` is the property value

CSS Selectors

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

The CSS element Selector

```
<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.

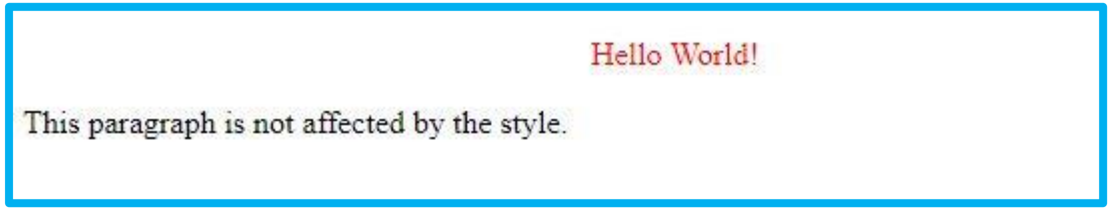
Me too!

And me!

CSS ID Selectors

- 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.

```
<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>
```



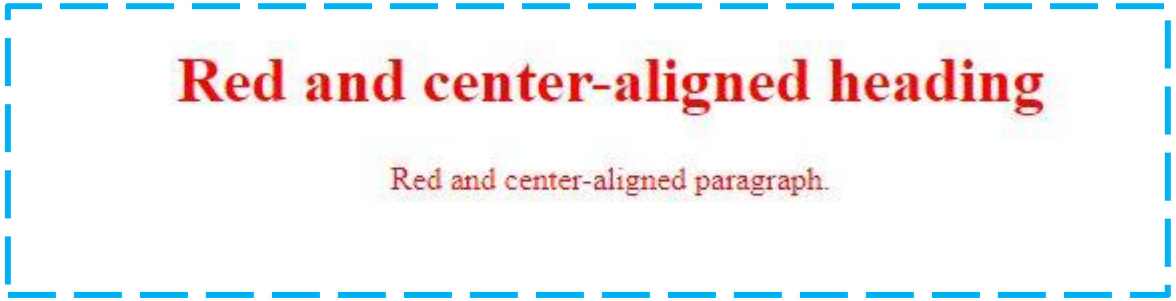
Hello World!

This paragraph is not affected by the style.

CSS Class Selectors

- 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.

```
<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>
```



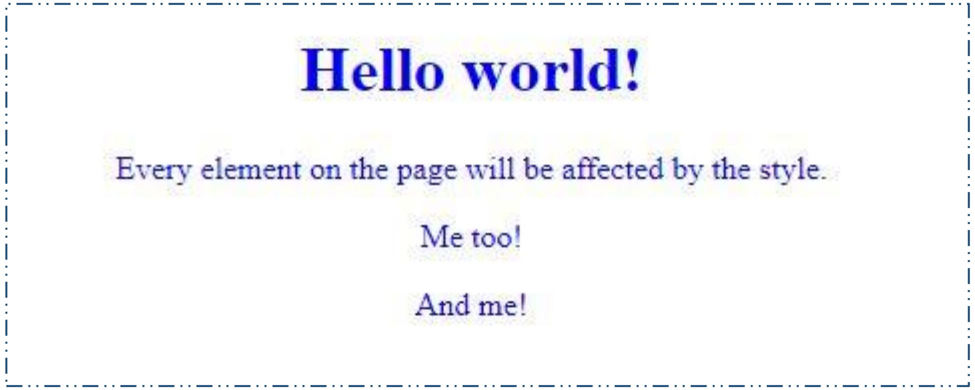
Red and center-aligned heading

Red and center-aligned paragraph.

CSS Universal Selectors

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

```
<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>
```



Hello world!

Every element on the page will be affected by the style.

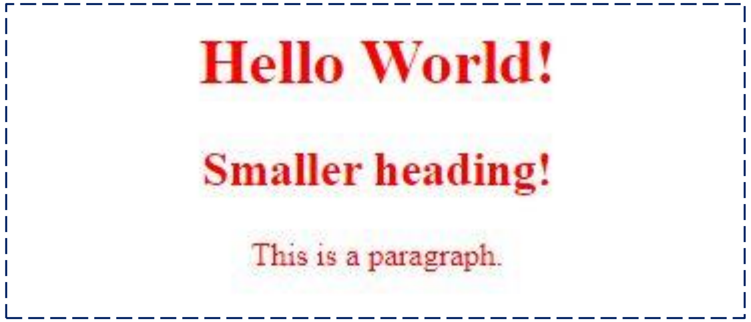
Me too!

And me!

CSS Grouping Selectors

- 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):

```
<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>
```



Hello World!

Smaller heading!

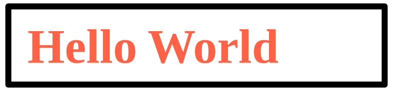
This is a paragraph.

ALL CSS Selectors

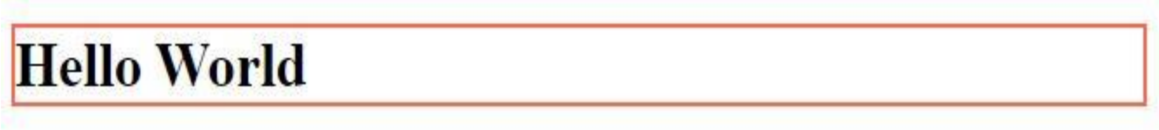
Selector	Example	Example description
#id	#firstname	Selects the element with id="firstname"
.class	.intro	Selects all elements with class="intro"
element.class	p.intro	Selects only <p> elements with class="intro"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element,..	div, p	Selects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

CSS COLORS

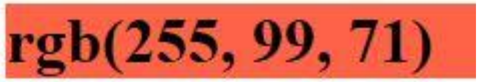
```
<h3 style="color:Tomato;">Hello World</h3>
```


Hello World

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

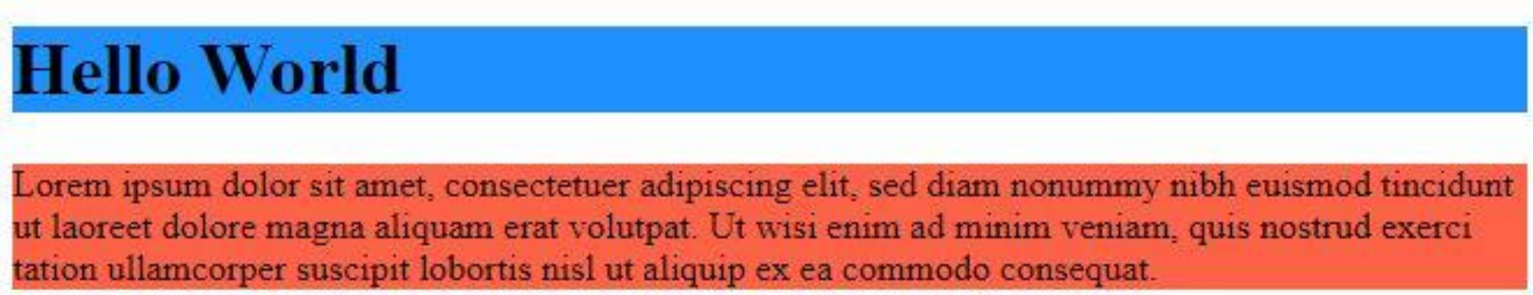

Hello World

```
<h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1>
```


rgb(255, 99, 71)

CSS COLORS

```
<html>

<body>

<h1 style="background-color:DodgerBlue;">Hello World</h1>

<p style="background-color:Tomato;">

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.

</p>

</body>

</html>
```



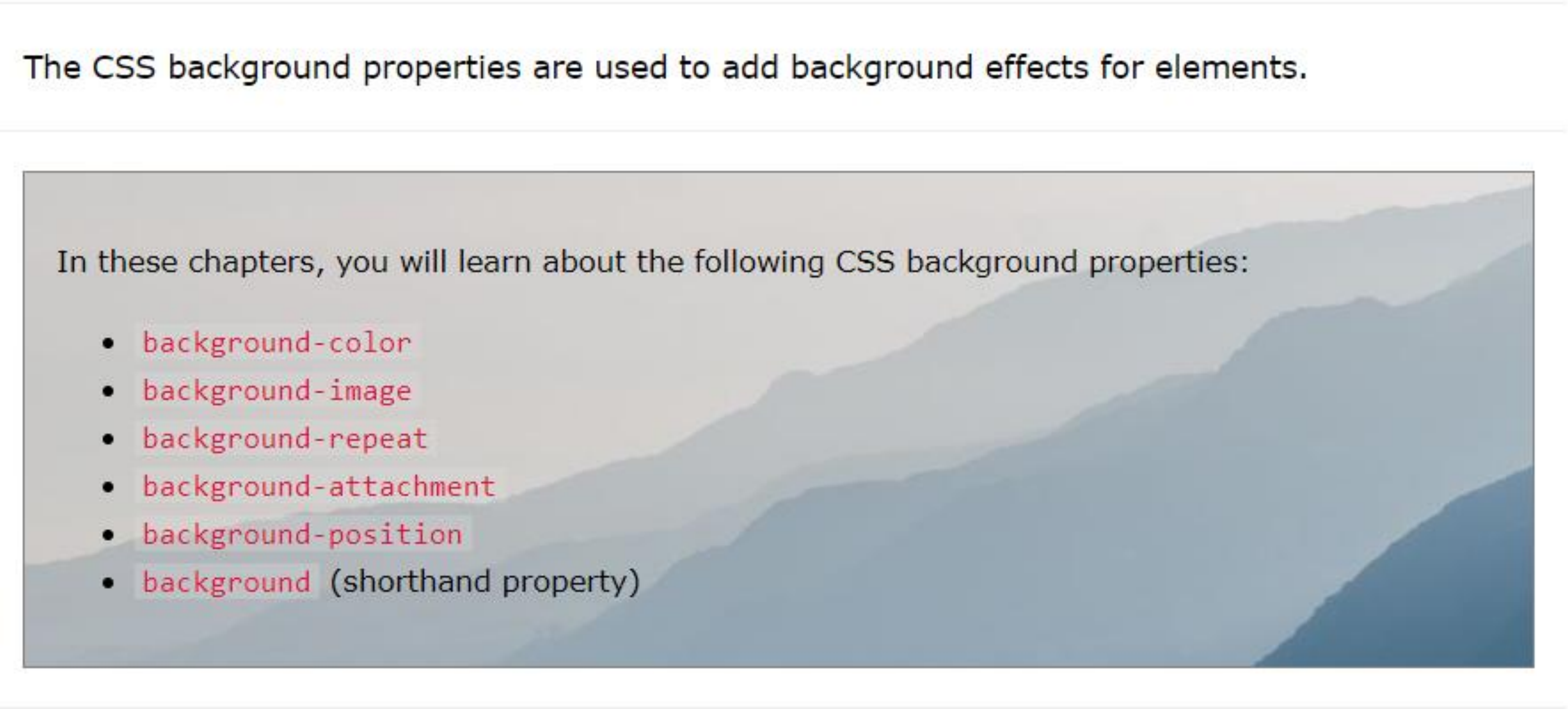
Hello World

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.

```
<h3 style="color:Tomato;">Hello World</h3>
```

Hello World

CSS BACKGROUNDS



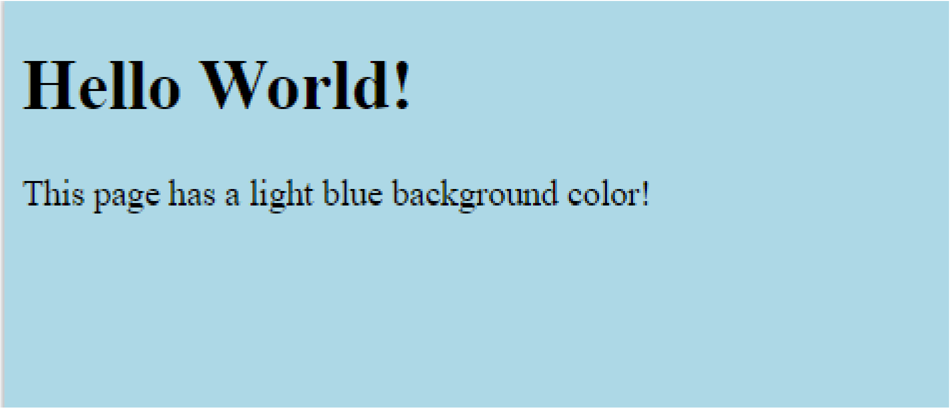
The CSS background properties are used to add background effects for elements.

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

```
<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>
```



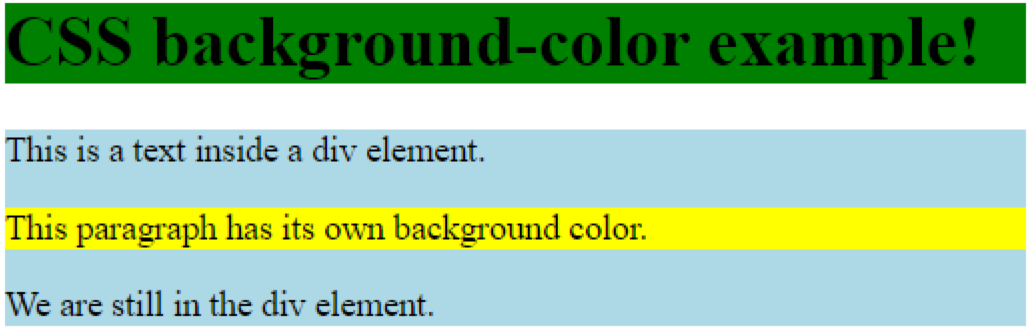
Hello World!

This page has a light blue background color!

CSS BACKGROUND COLOR

```
<html>
<head>
<style>
h1 {
  background-color: green;
}
div {
  background-color: lightblue;
}
p {
  background-color: yellow;
}
</style>
</head>
```

```
<body>
<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>
</body>
</html>
```



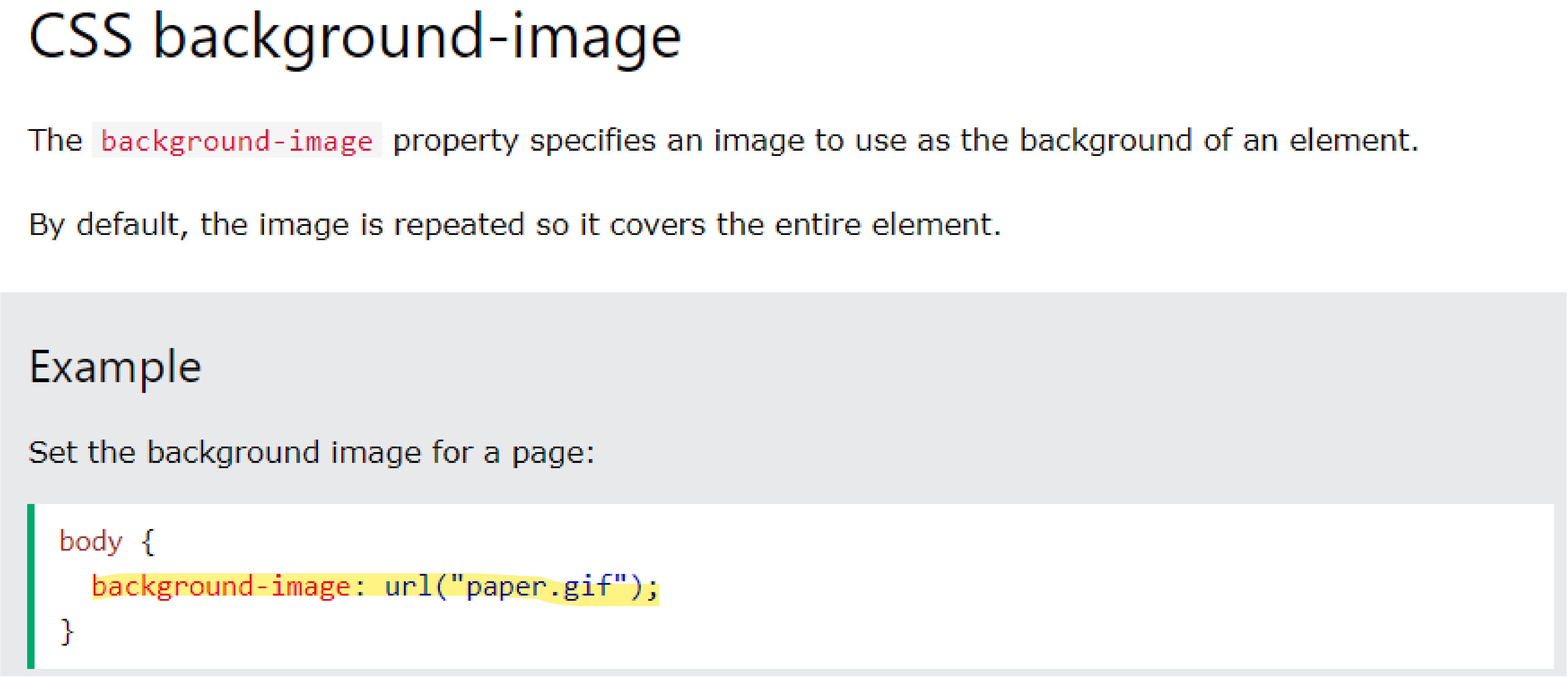
CSS background-color example!

This is a text inside a div element.

This paragraph has its own background color.

We are still in the div element.

CSS BACKGROUND IMAGE



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.

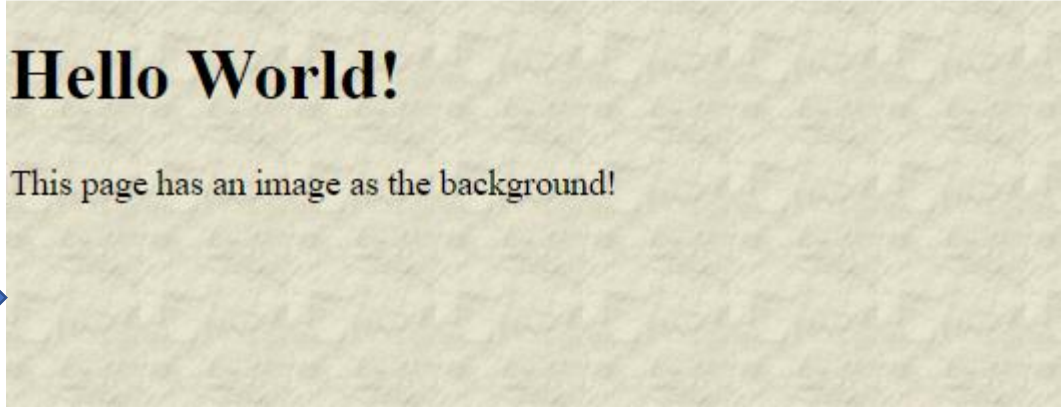
Example

Set the background image for a page:

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

CSS BACKGROUND IMAGE

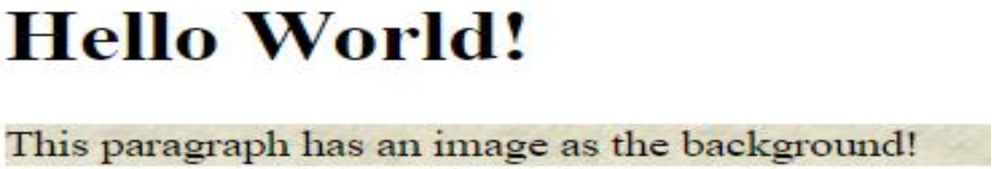
```
<html>
<head>
<style>
body {
  background-image: url("paper.gif");
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>This page has an image as the background!</p>
</body>
</html>
```



Hello World!

This page has an image as the background!

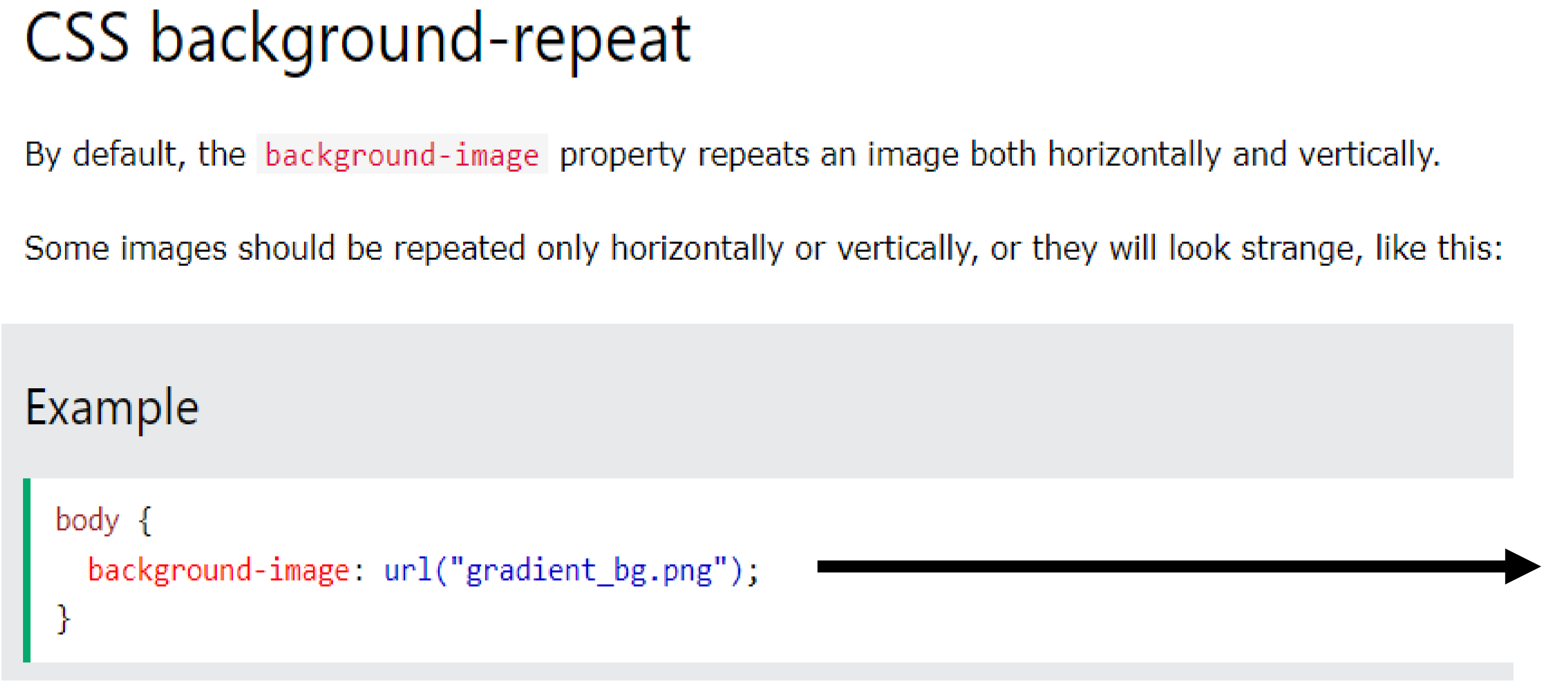
```
<style>
p {
  background-image: url("paper.gif");
}
</style>
```



Hello World!

This paragraph has an image as the background!

CSS BACKGROUND IMAGE REPEAT



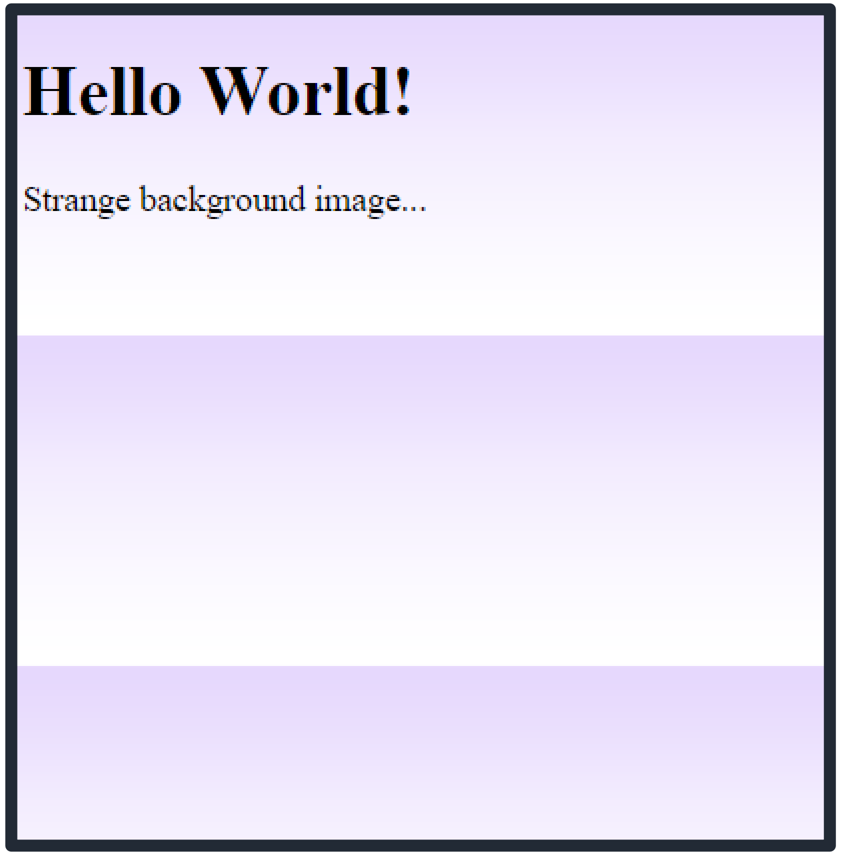
CSS background-repeat

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("gradient_bg.png");
}
```

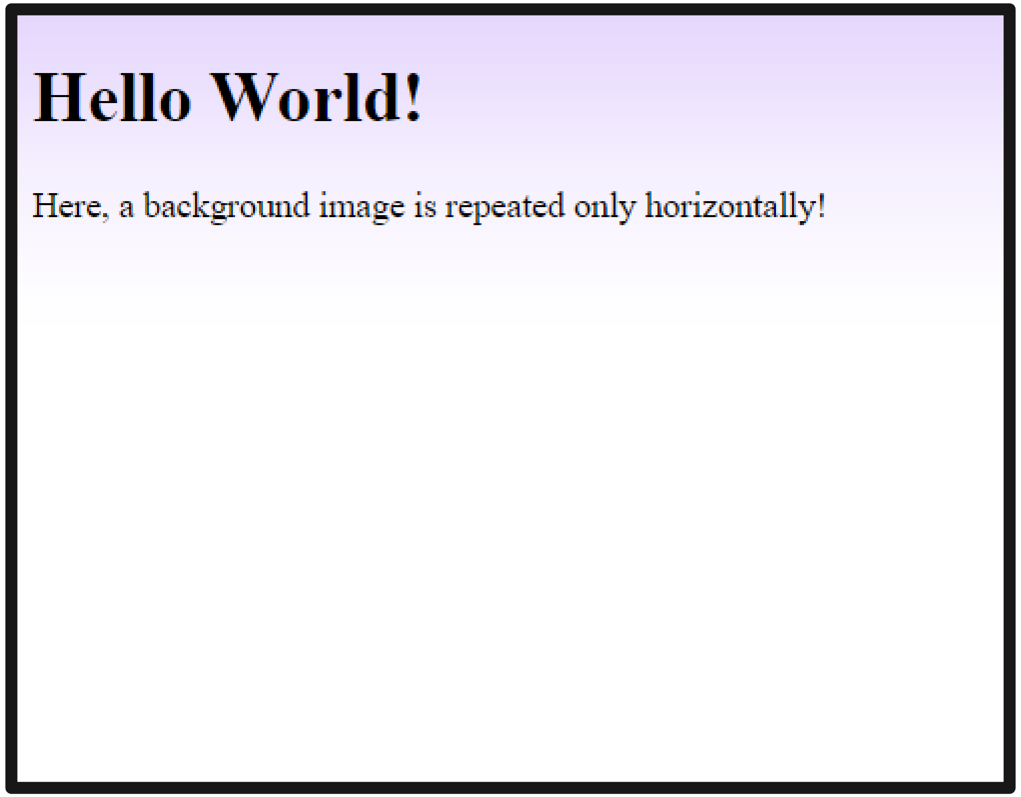


Hello World!

Strange background image...

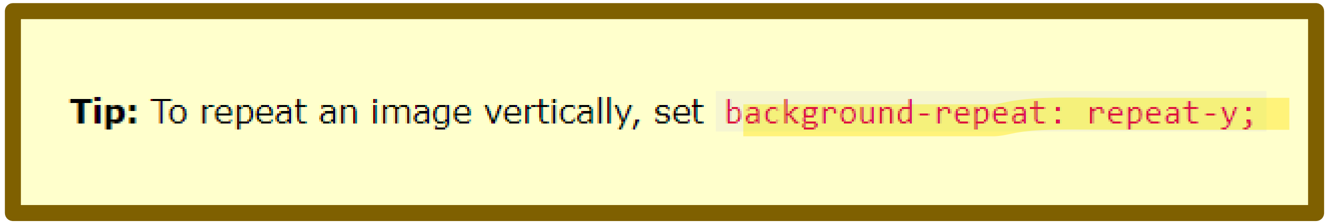
CSS BACKGROUND IMAGE REPEAT

```
<html>
<head>
<style>
body {
  background-image: url("gradient_bg.png");
  background-repeat: repeat-x;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>Here, a background image is repeated only
horizontally!</p>
</body>
</html>
```



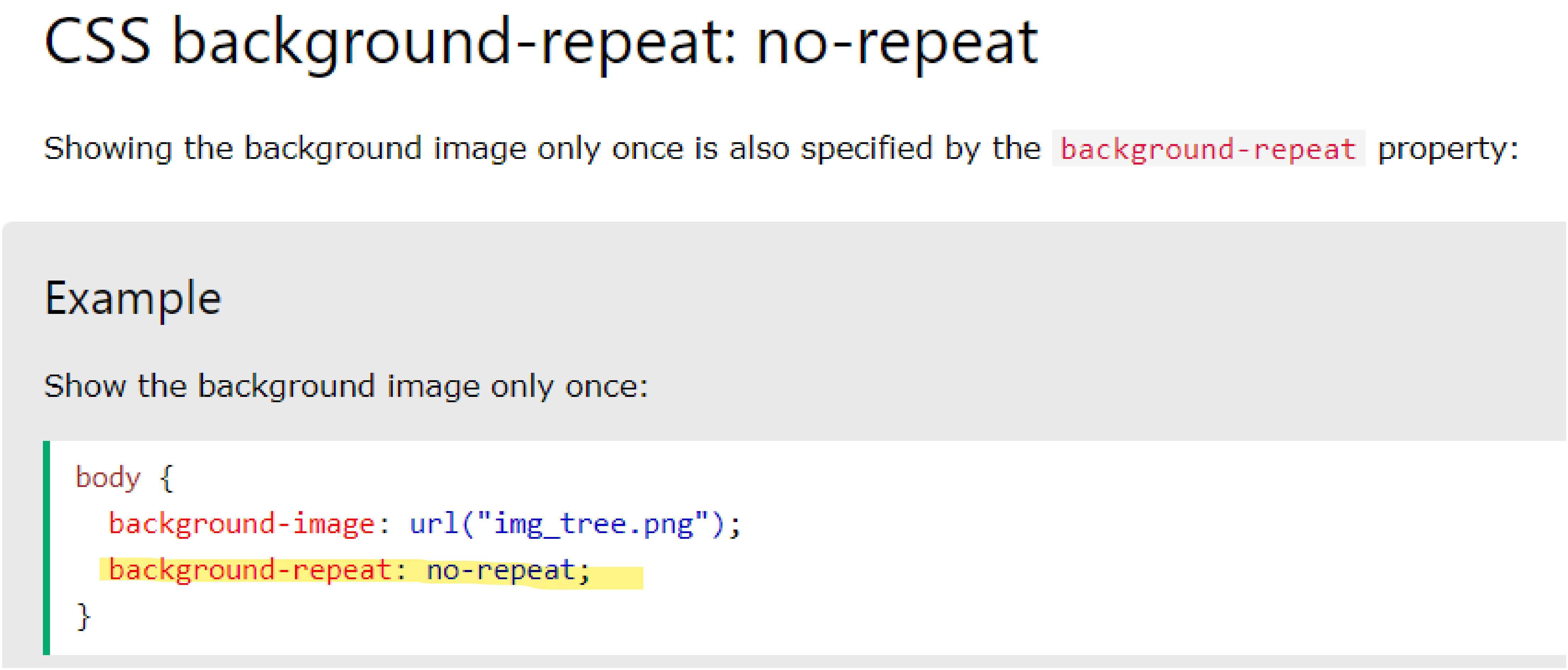
Hello World!

Here, a background image is repeated only horizontally!



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

CSS background-repeat: no-repeat



CSS background-repeat: no-repeat

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

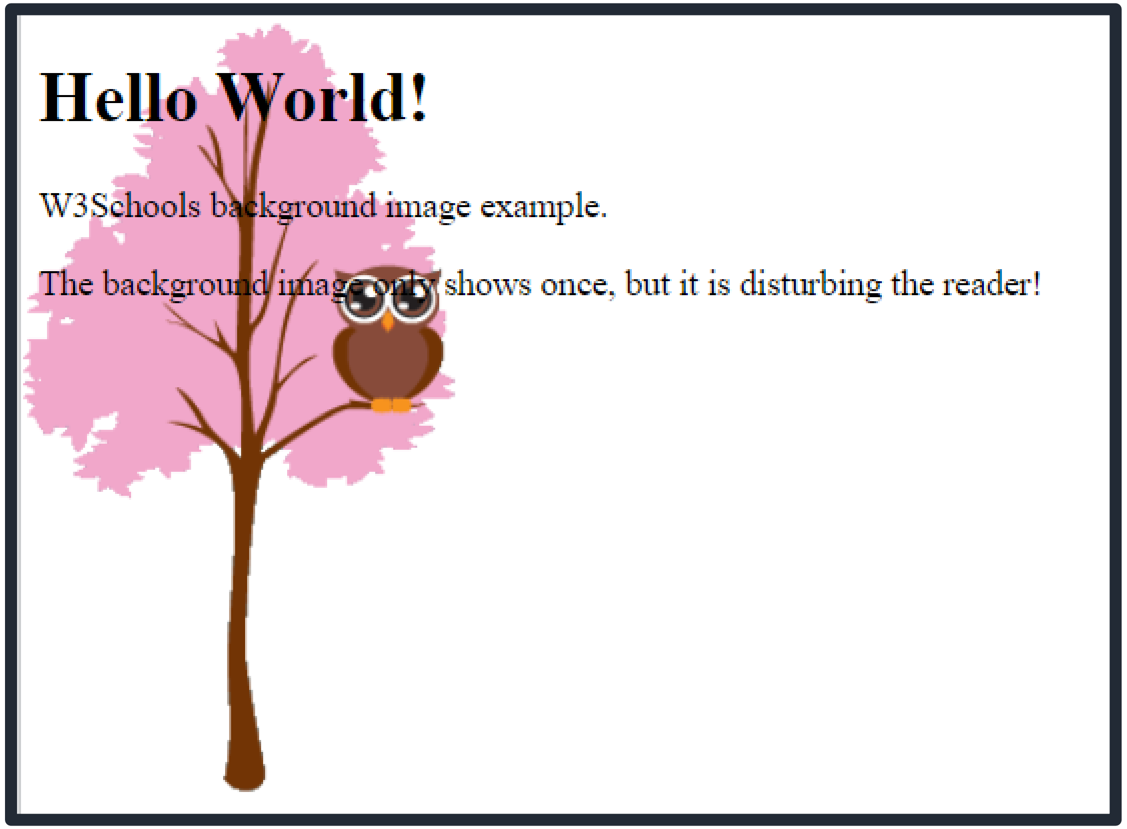
Example

Show the background image only once:

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

CSS background-repeat: no-repeat

```
<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>
```



Hello World!

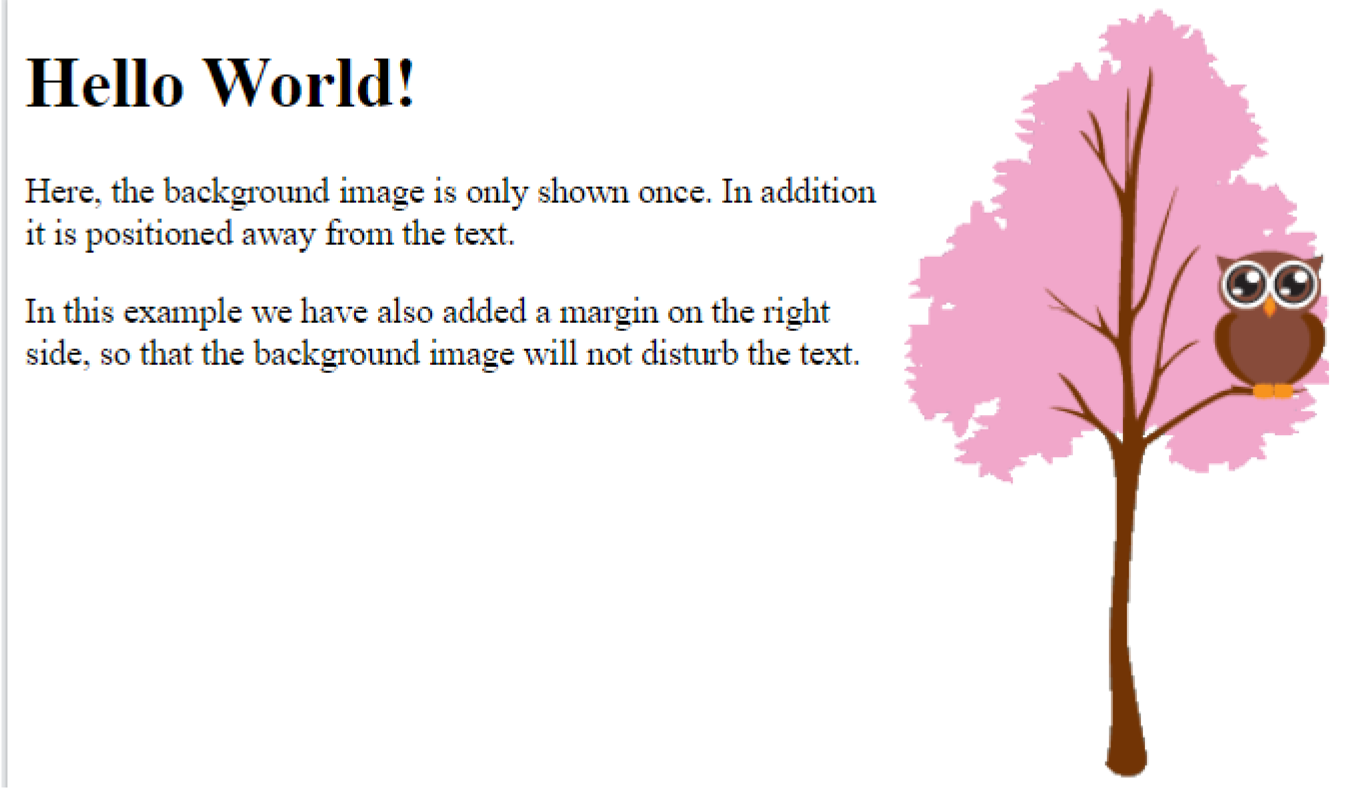
W3Schools background image example.

The background image only shows once, but it is disturbing the reader!

In the example above, the background image is placed in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

CSS background-repeat: no-repeat with Position

```
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  margin-right: 200px;
}
</style>
</head>
```

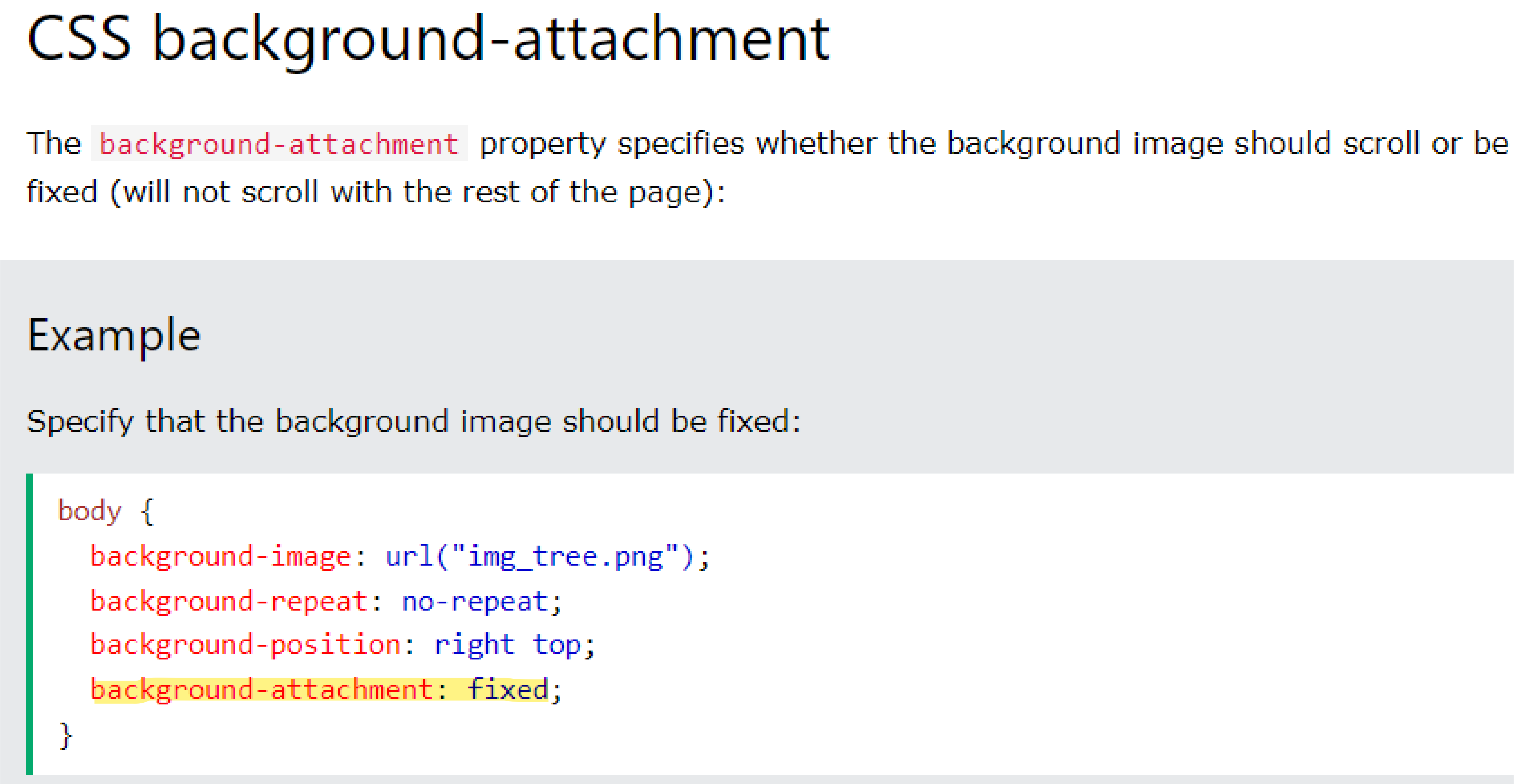


Hello World!

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.

CSS BACKGROUND - Attachment



CSS background-attachment

The `background-attachment` property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page):

Example

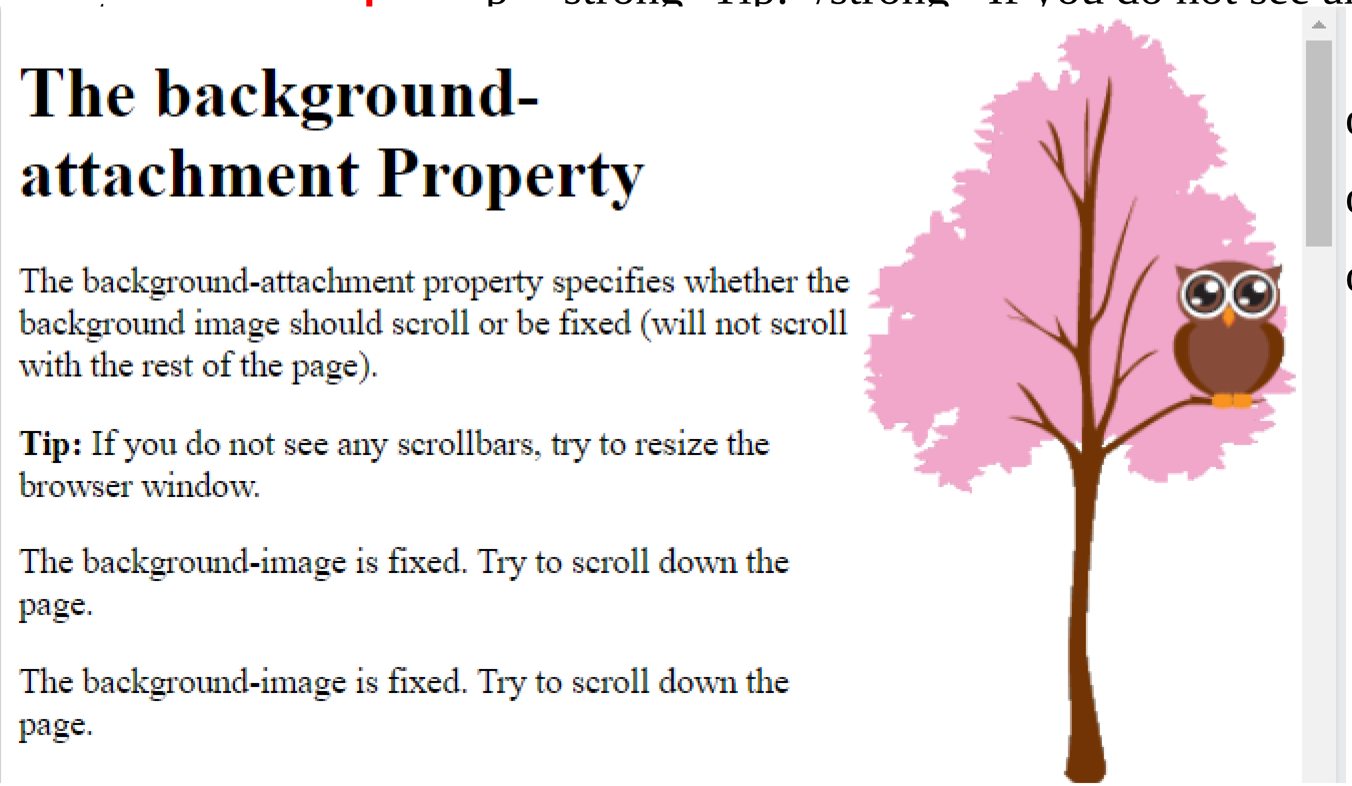
Specify that the background image should be fixed:

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

CSS BACKGROUND – Attachment Fixed

```
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-r
  background-position: ri
  margin-right: 200px;
  background-attachment
}
</style>
</head>
```

```
<body>
<h1>The background-attachment Property</h1>


<p>The background-attachment property specifies whether the background
image should scroll or be fixed (will not scroll with the rest of the page).</p>
<p><strong>Tip:</strong> If you do not see any scrollbars, try to resize the

oll down the page.</p>
oll down the page.</p>
oll down the page.</p>
```



The background-attachment Property

The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).

Tip: If you do not see any scrollbars, try to resize the browser window.

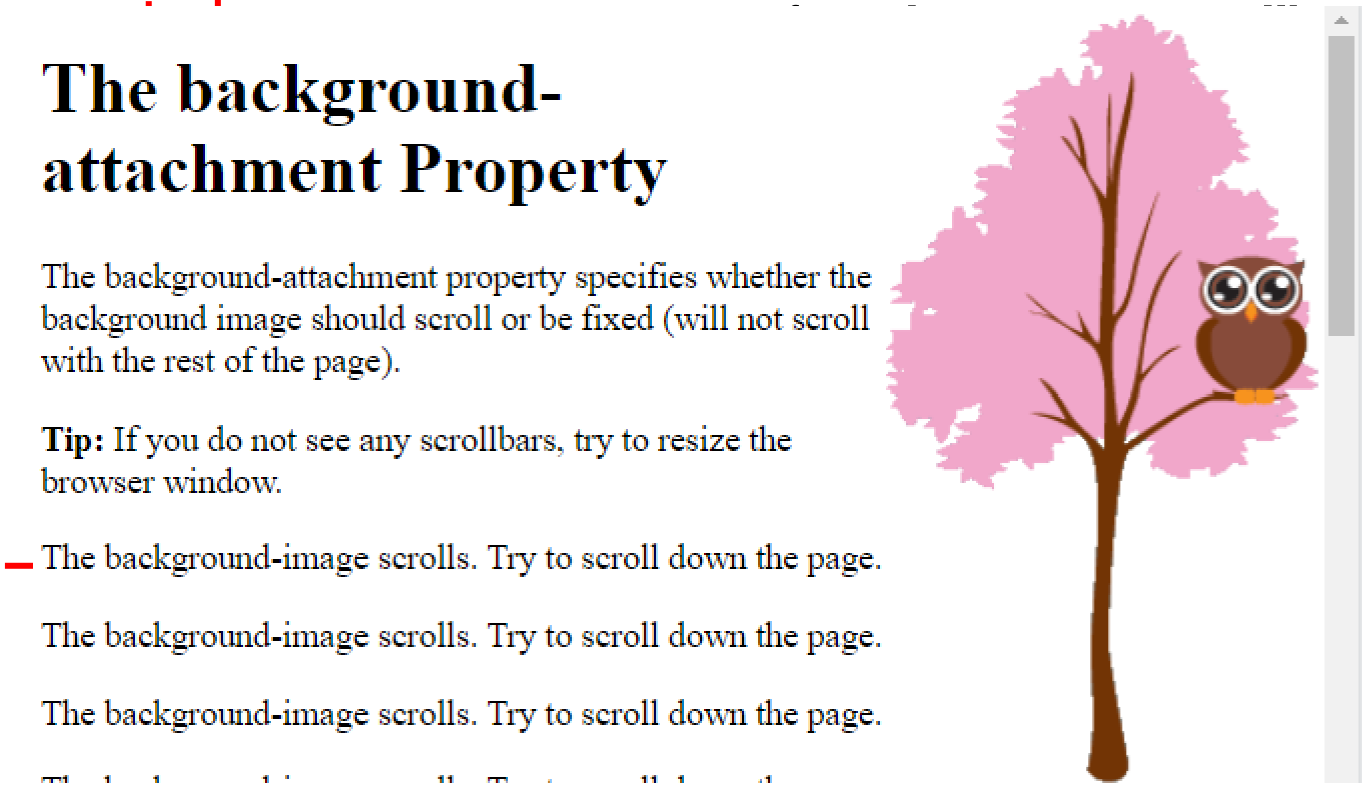
The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

CSS BACKGROUND – Attachment Scroll

```
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  margin-right: 200px;
  background-attachment: scroll;
}
</style>
</head>
```

```
<body>
<h1>The background-attachment Property</h1>


<p>The background-attachment property specifies whether the background
image should scroll or be fixed (will not scroll with the rest of the page).</p>
s, try to resize the

page.</p>
page.</p>
page.</p>
```



The background-attachment Property

The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).

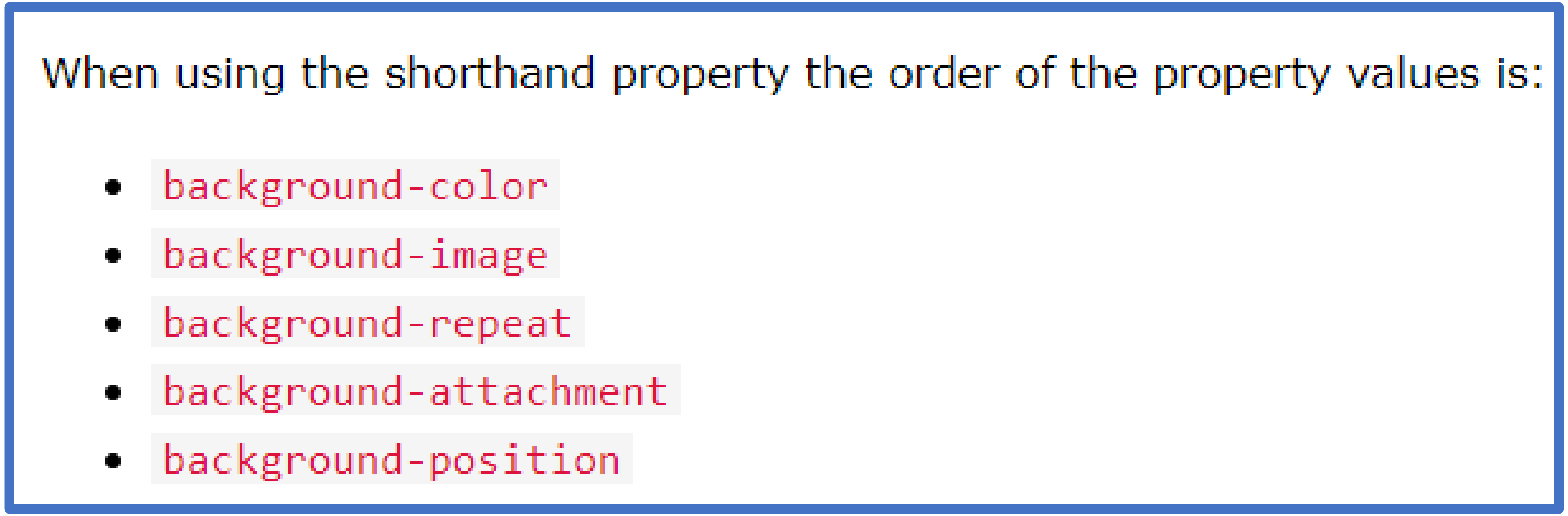
Tip: If you do not see any scrollbars, try to resize the browser window.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

CSS BACKGROUND – Shorthand property

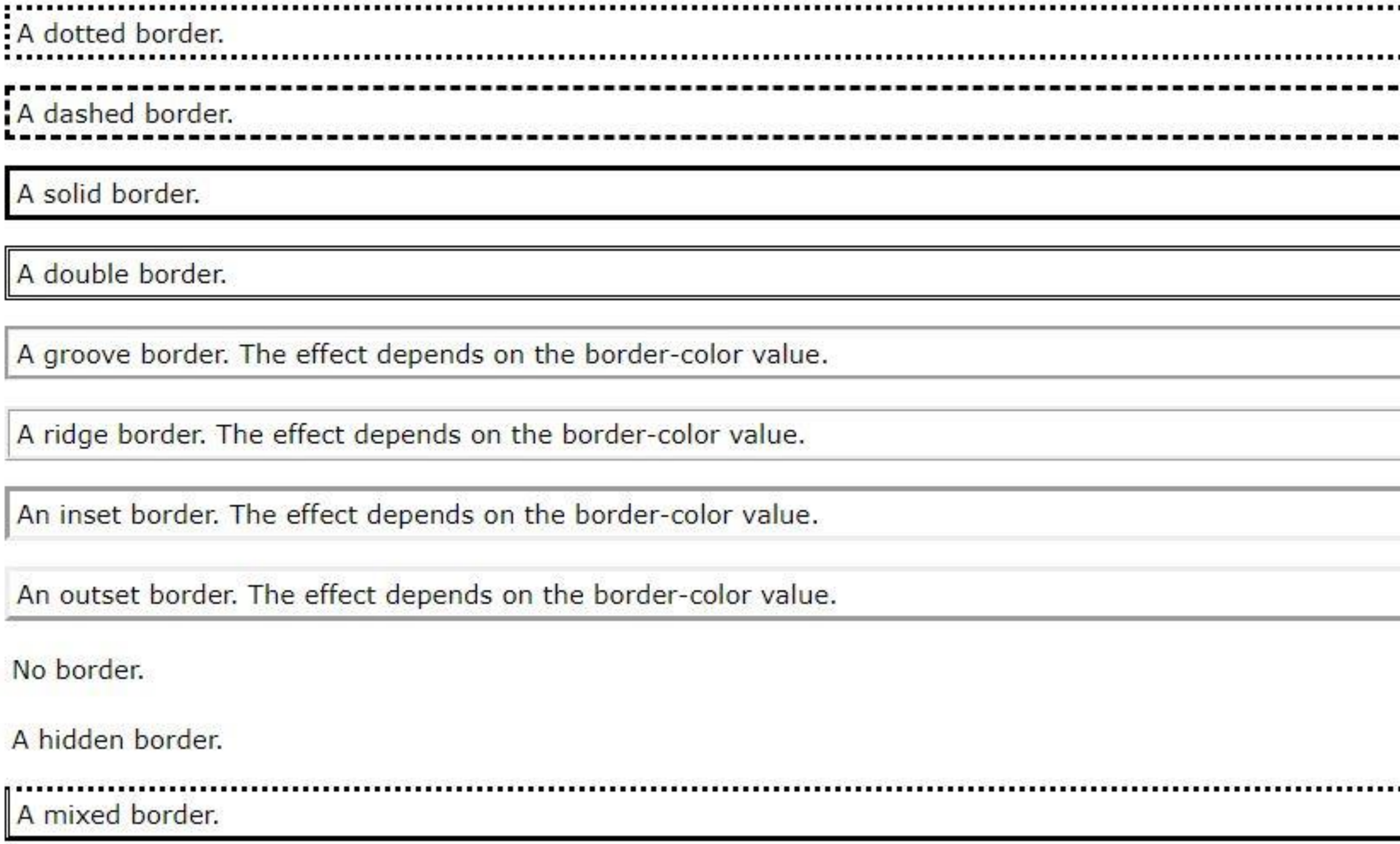


When using the shorthand property the order of the property values is:

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`

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

CSS BORDERS



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.

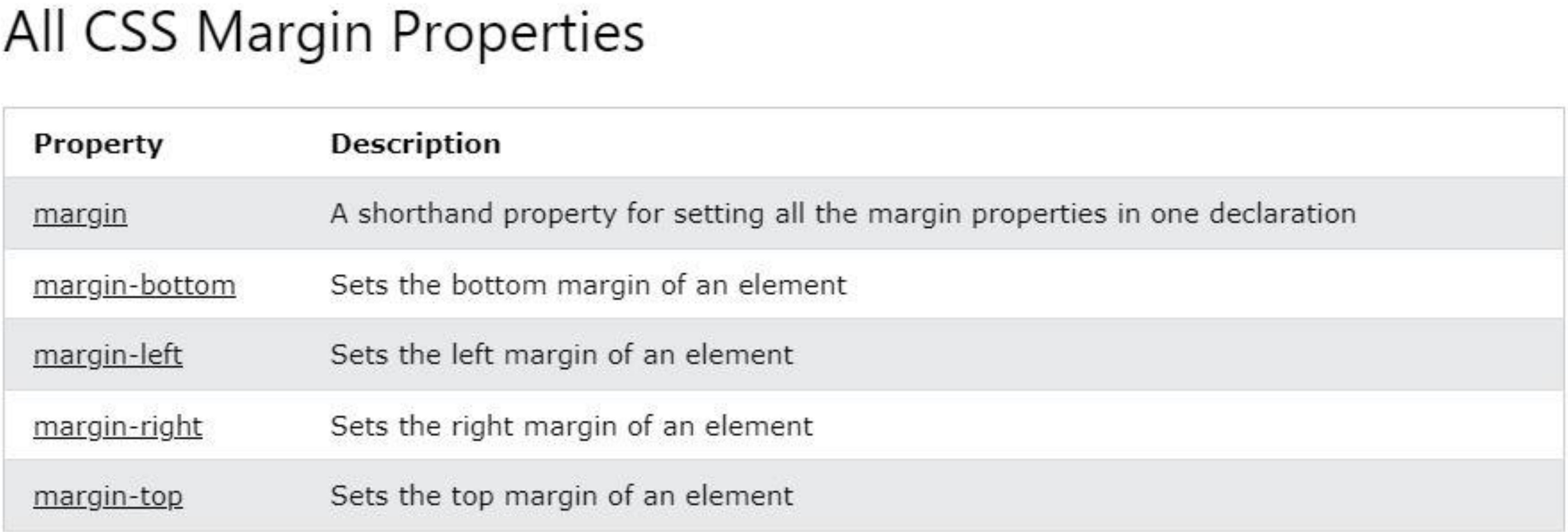
No border.

A hidden border.

A mixed border.

```
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;}
```

CSS Margins

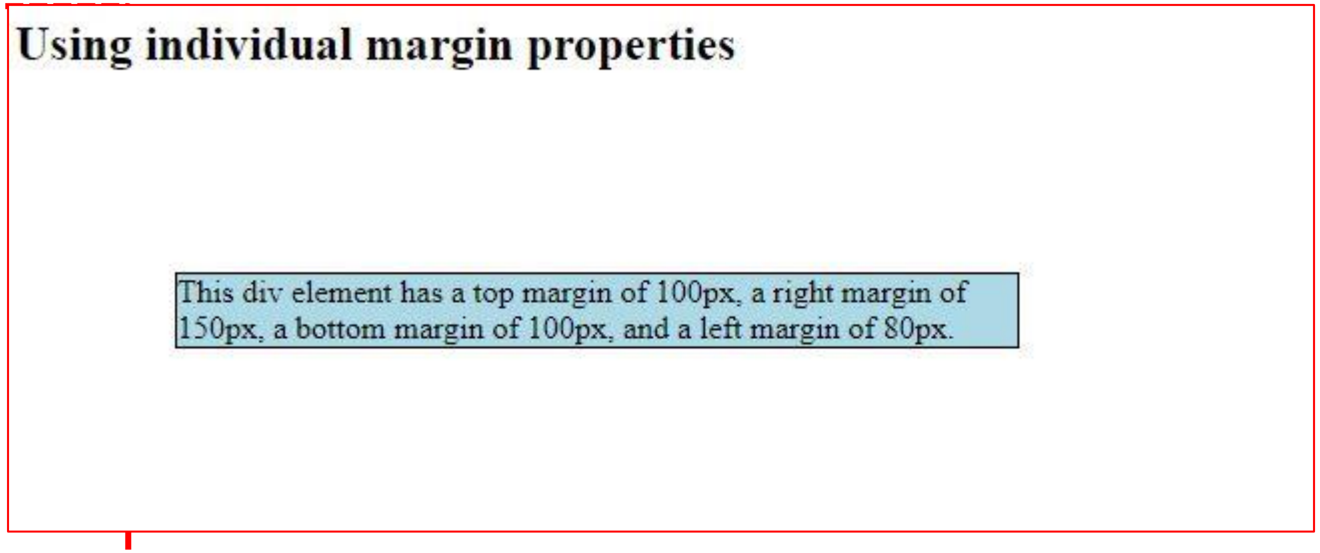


All CSS Margin Properties

Property	Description
margin	A shorthand property for setting all the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

CSS Margins

```
<style>
div {
  border: 1px solid black;
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>Using individual margin properties</h2>

<div>This div element has a top margin of 100px, a right margin
of 150px, a bottom margin of 100px, and a left margin of
80px.</div>

</body>
```

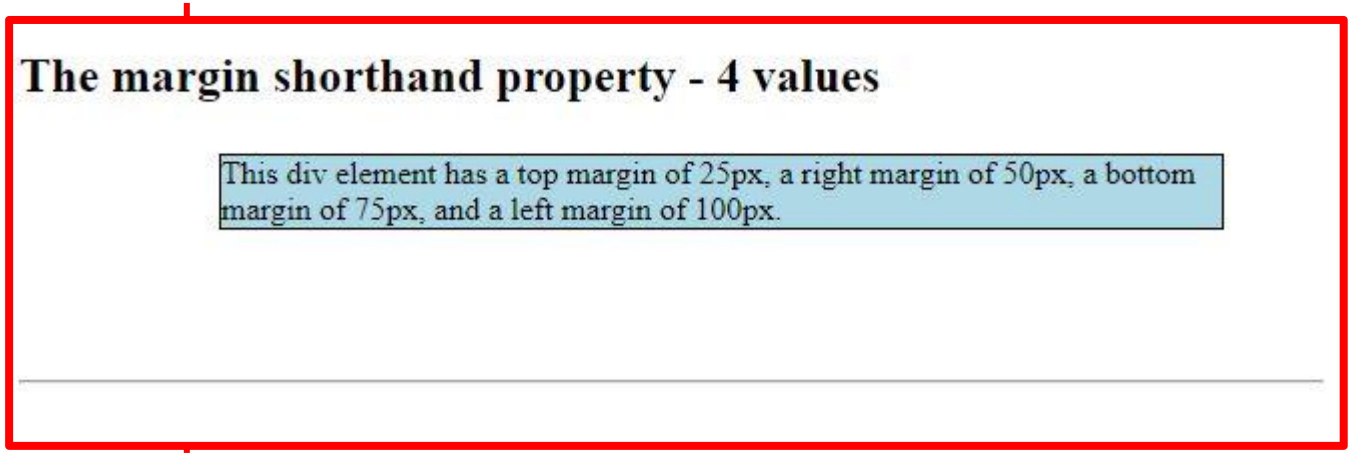


Using individual margin properties

This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

CSS Margins

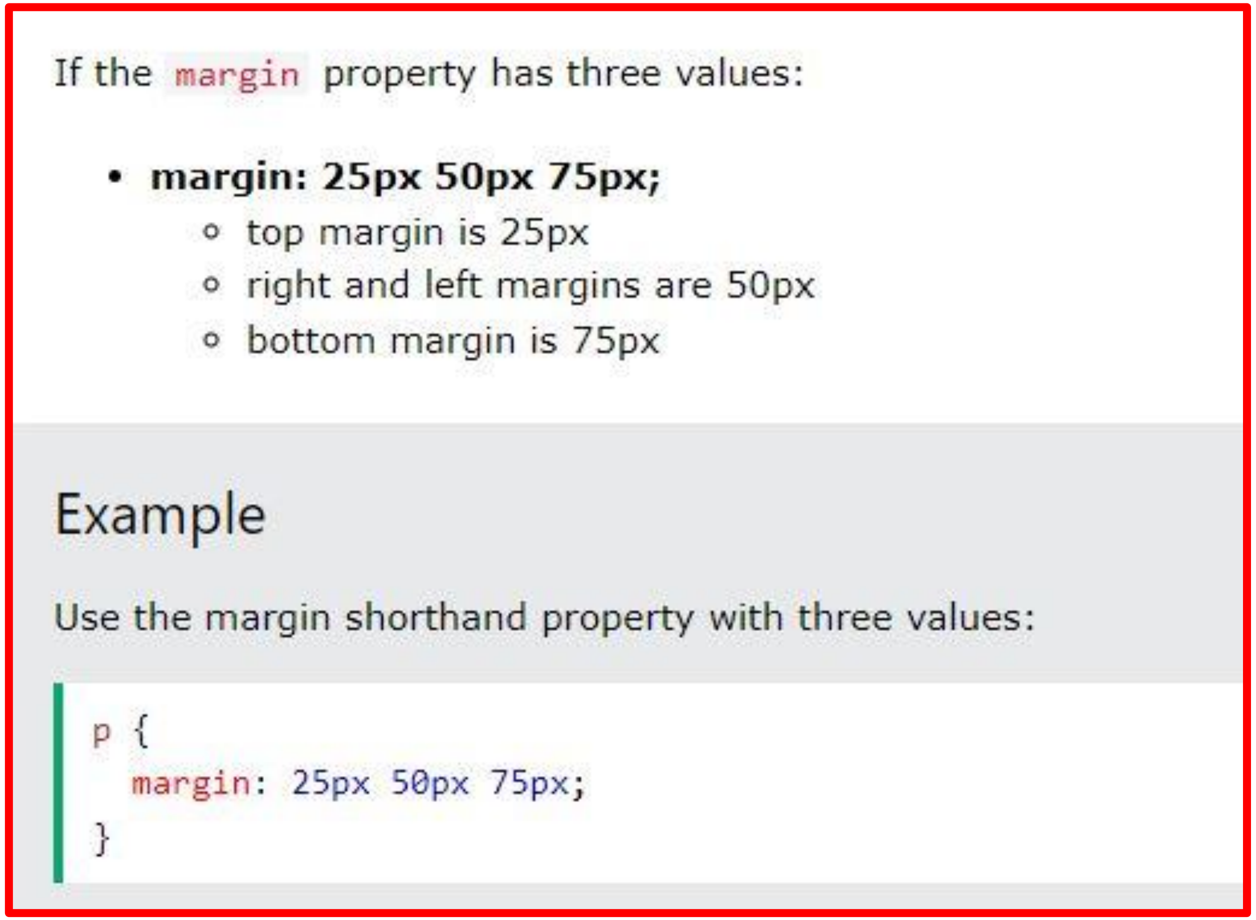
```
<style>
div {
  border: 1px solid black;
  margin: 25px 50px 75px 100px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>The margin shorthand property - 4 values</h2>

<div>This div element has a top margin of 25px, a right margin
of 50px, a bottom margin of 75px, and a left margin of
100px.</div>

<hr>

</body>
```



The margin shorthand property - 4 values

This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.

CSS Margins



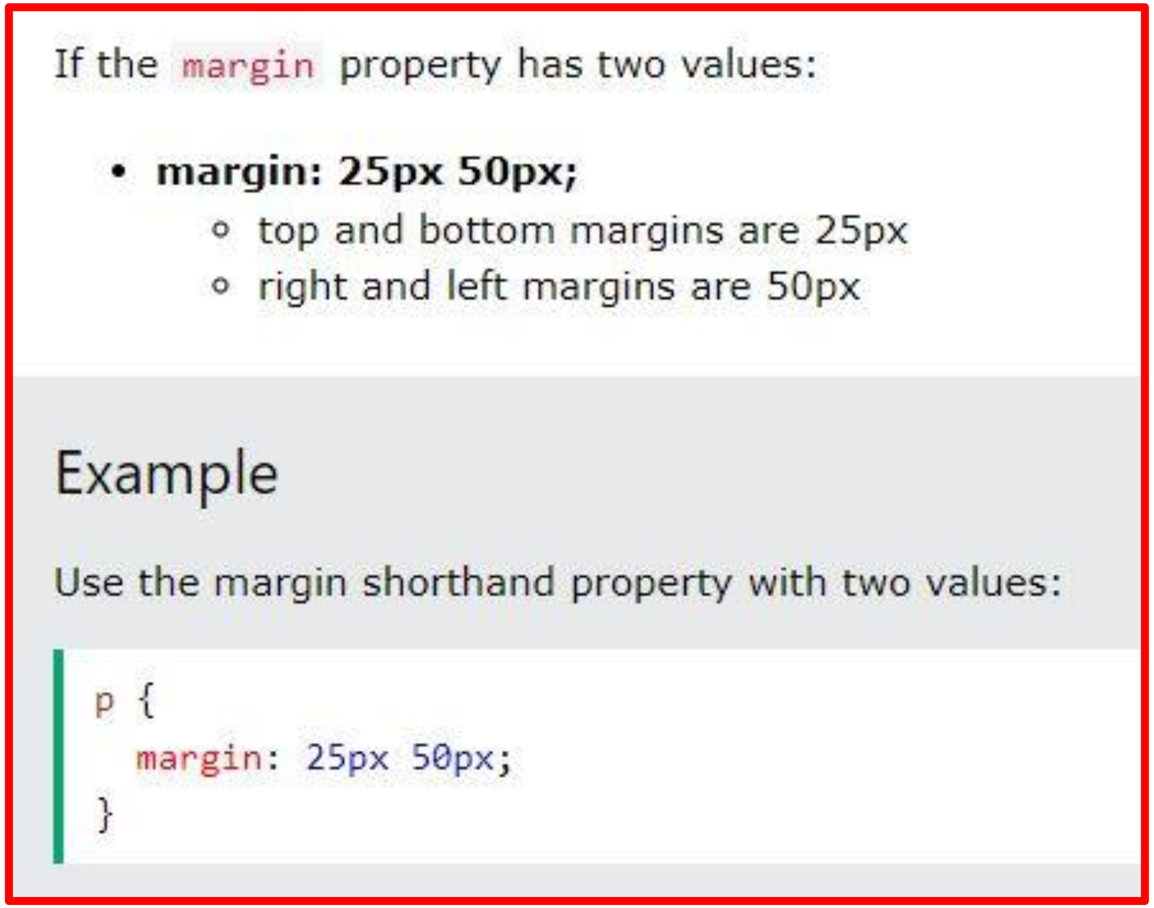
If the `margin` property has three values:

- **margin: 25px 50px 75px;**
  - top margin is 25px
  - right and left margins are 50px
  - bottom margin is 75px

Example

Use the margin shorthand property with three values:

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



If the `margin` property has two values:

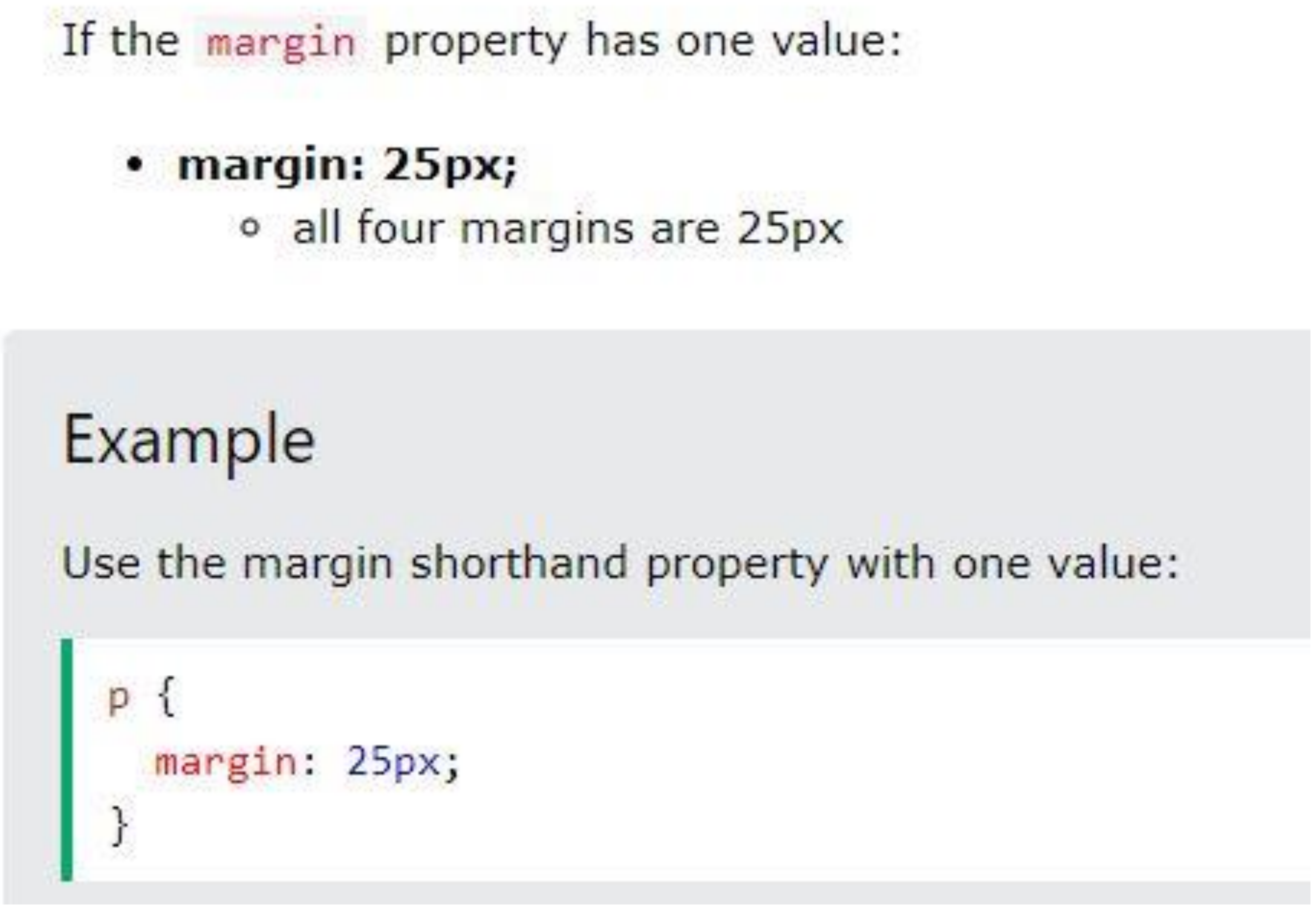
- **margin: 25px 50px;**
  - top and bottom margins are 25px
  - right and left margins are 50px

Example

Use the margin shorthand property with two values:

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

CSS Margins



If the `margin` property has one value:

- **margin: 25px;**
    - all four margins are 25px

Example

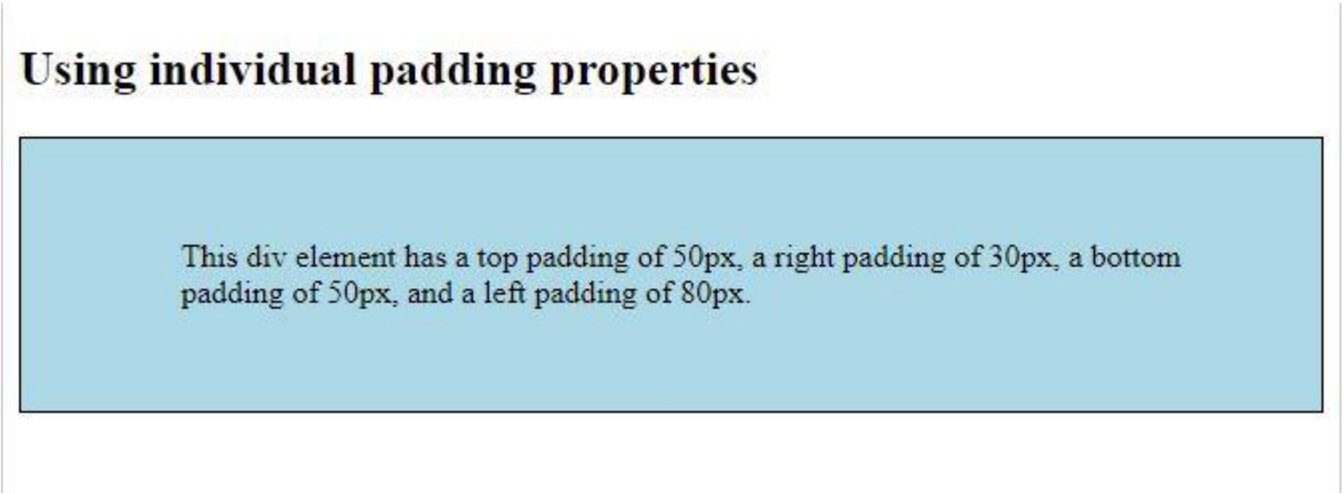
Use the margin shorthand property with one value:

```
p {
  margin: 25px;
}
```

CSS Padding

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- There are properties for setting the padding for each side of an element (top, right, bottom, and left).

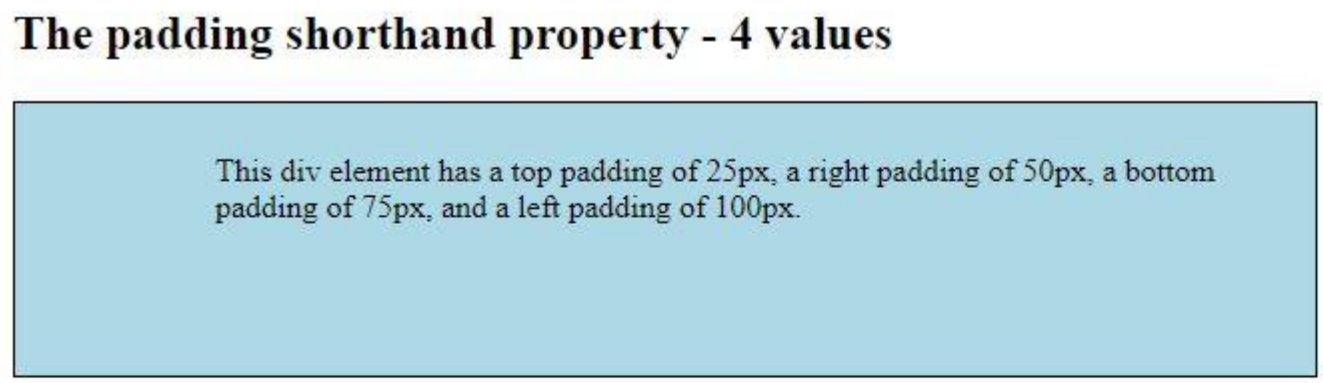
```
div {
   padding-top: 50px;
   padding-right: 30px;
   padding-bottom: 50px;
   padding-left: 80px;
}
```



Using individual padding properties

This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.

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

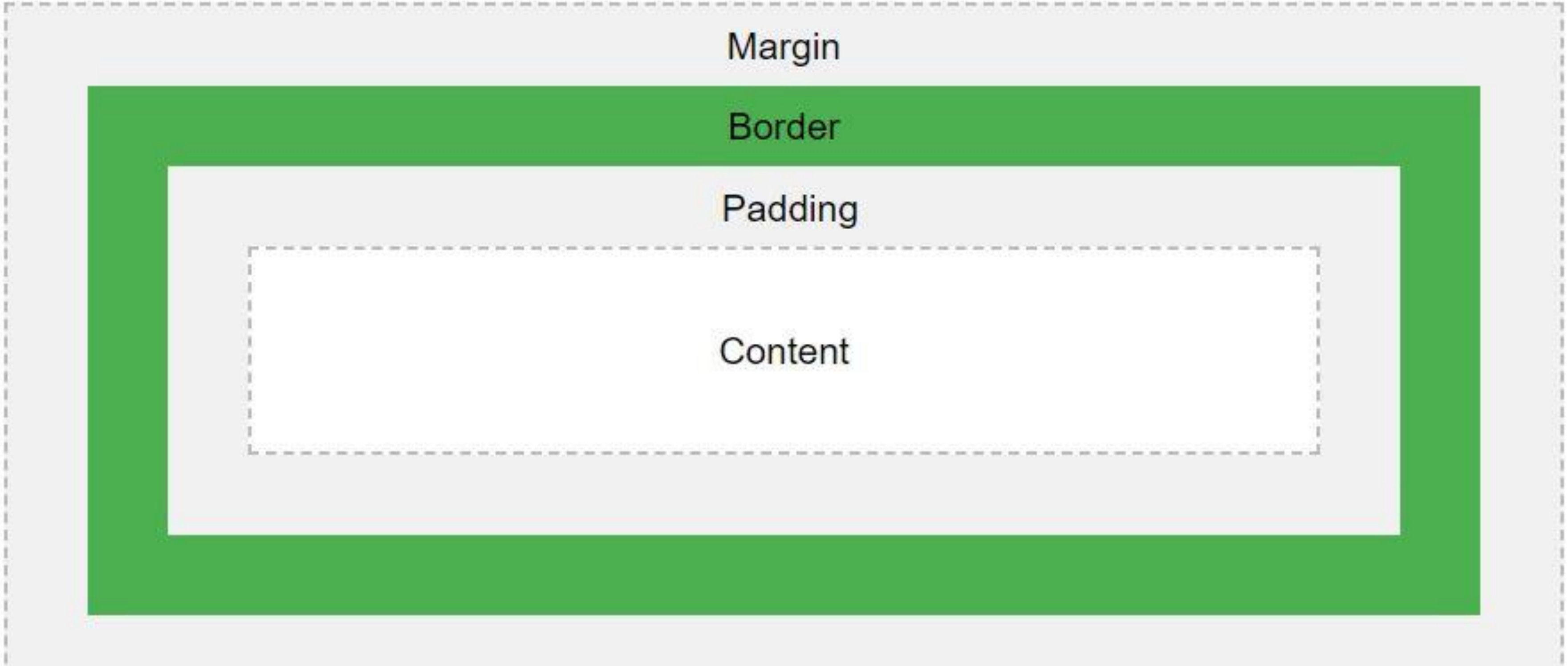


The padding shorthand property - 4 values

This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and a left padding of 100px.

CSS BOX MODEL

- 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 below illustrates the box model:

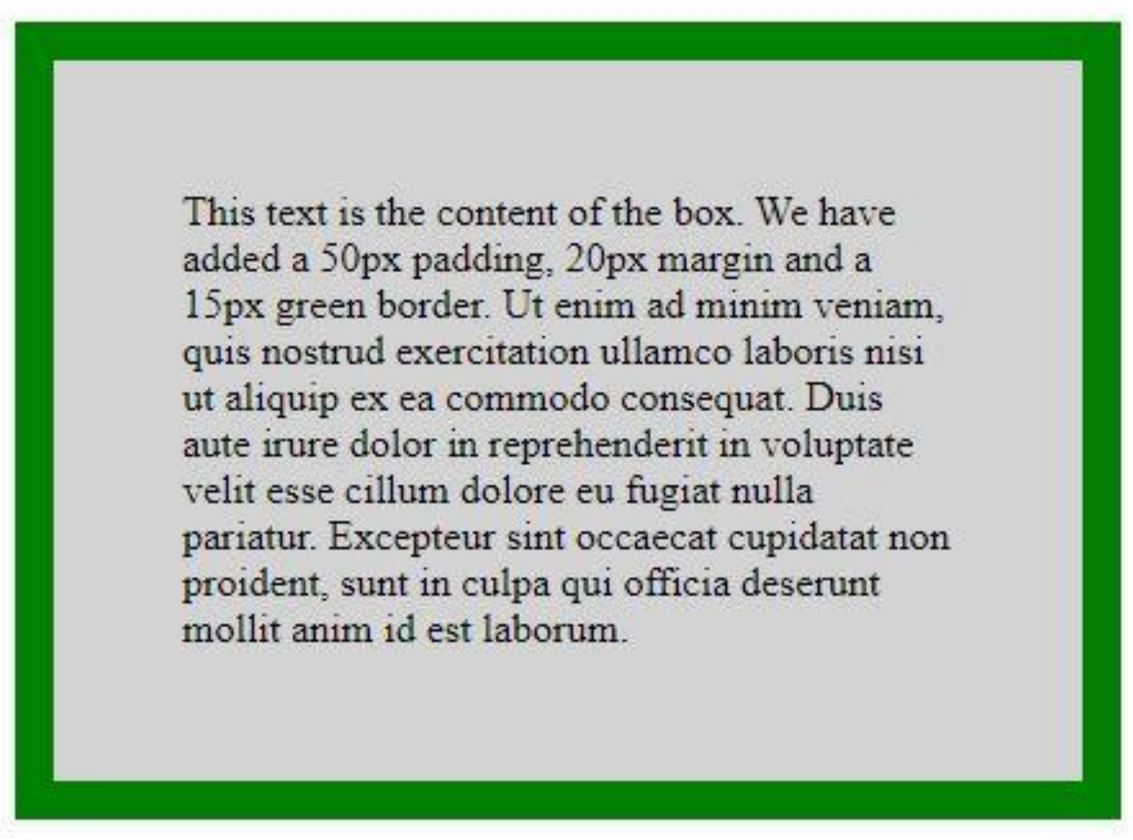


CSS BOX MODEL

- **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

CSS BOX MODEL

```
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
```

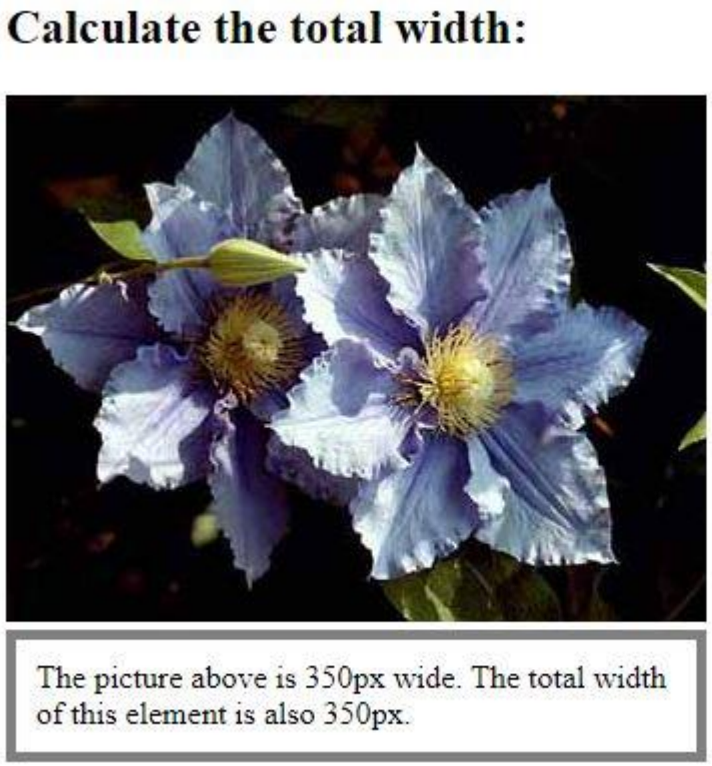


This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

CSS BOX MODEL

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

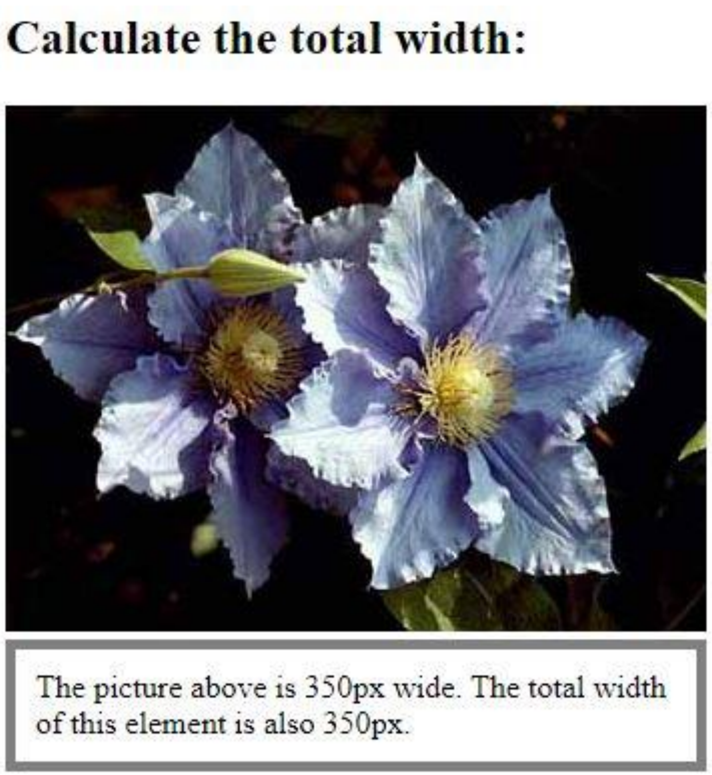
```
<body>
<h2>Calculate the total width:</h2>
<img src="klematis4_big.jpg" width="350" height="263" alt="Klematis">
<div>The picture above is 350px wide. The total width of this element is also 350px.</div>
</body>
```



Calculate the total width:

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

CSS BOX MODEL



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

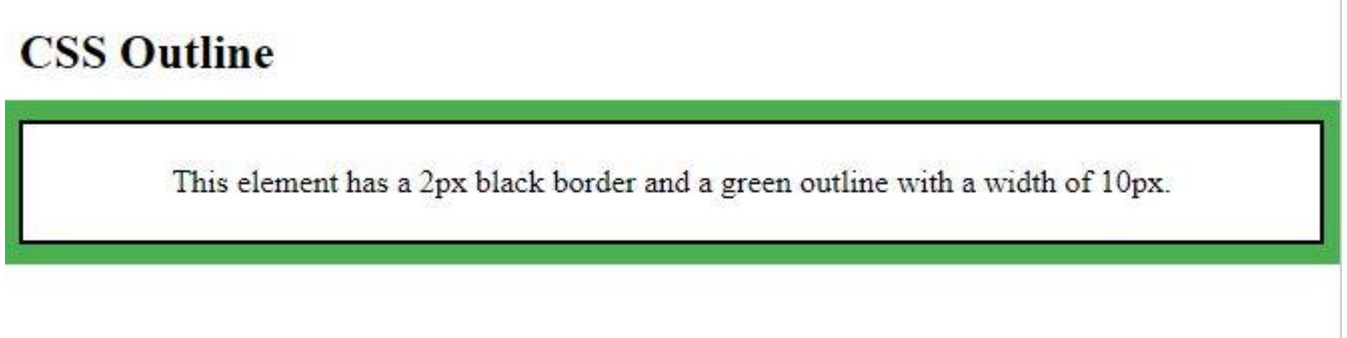
CSS OUTLINE

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

```
<head>
<style>
p {
  border: 2px solid black;
  outline: #4CAF50 solid 10px;
  margin: auto;
  padding: 20px;
  text-align: center;
}
</style>
</head>
```

```
<body>


<h2>CSS Outline</h2>
<p>This element has a 2px
black border and a green
outline with a width of
10px.</p>


</body>
```

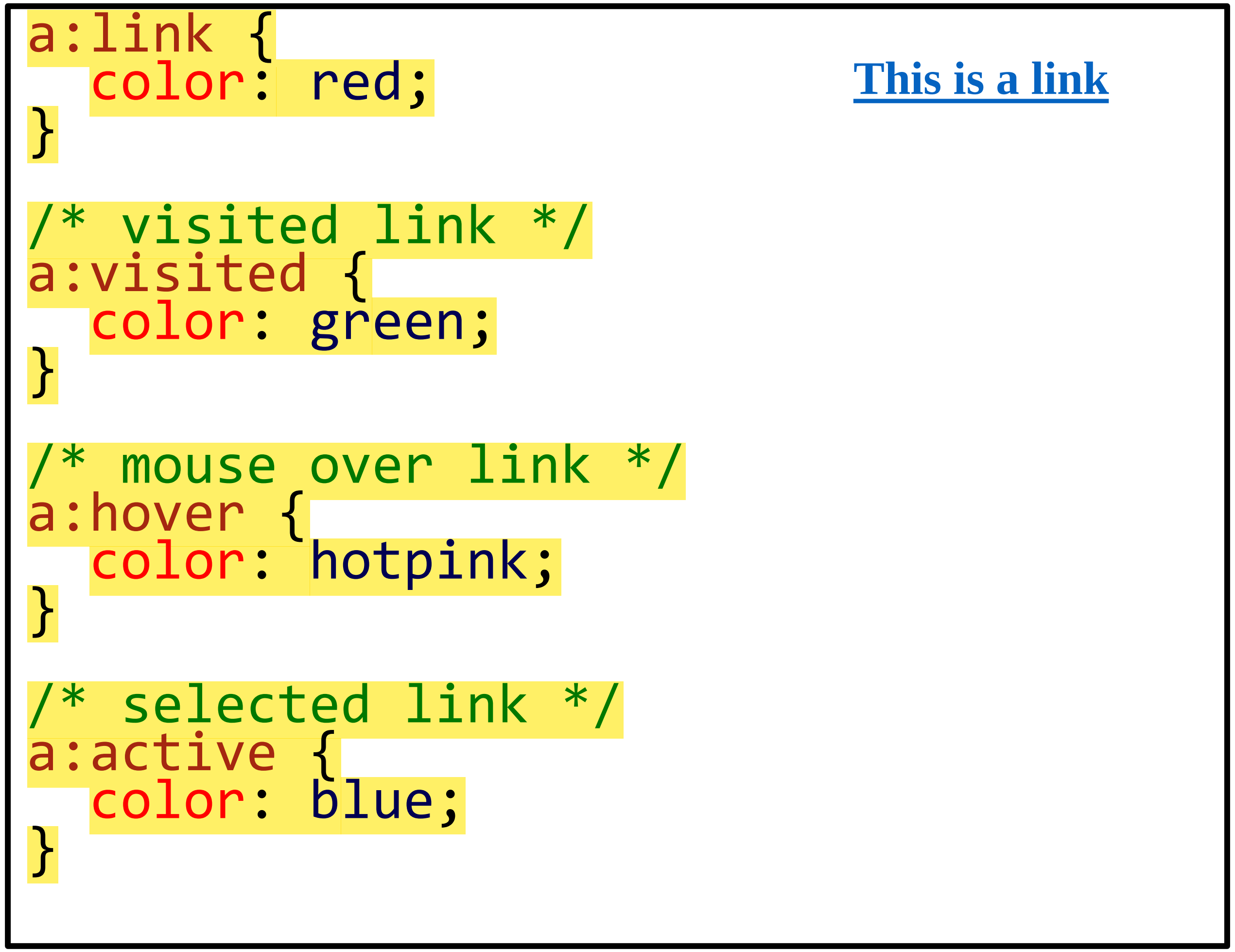
OUTPUT



CSS Outline

This element has a 2px black border and a green outline with a width of 10px.

CSS LINKS



```
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
```

This is a link

The four links states are:

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

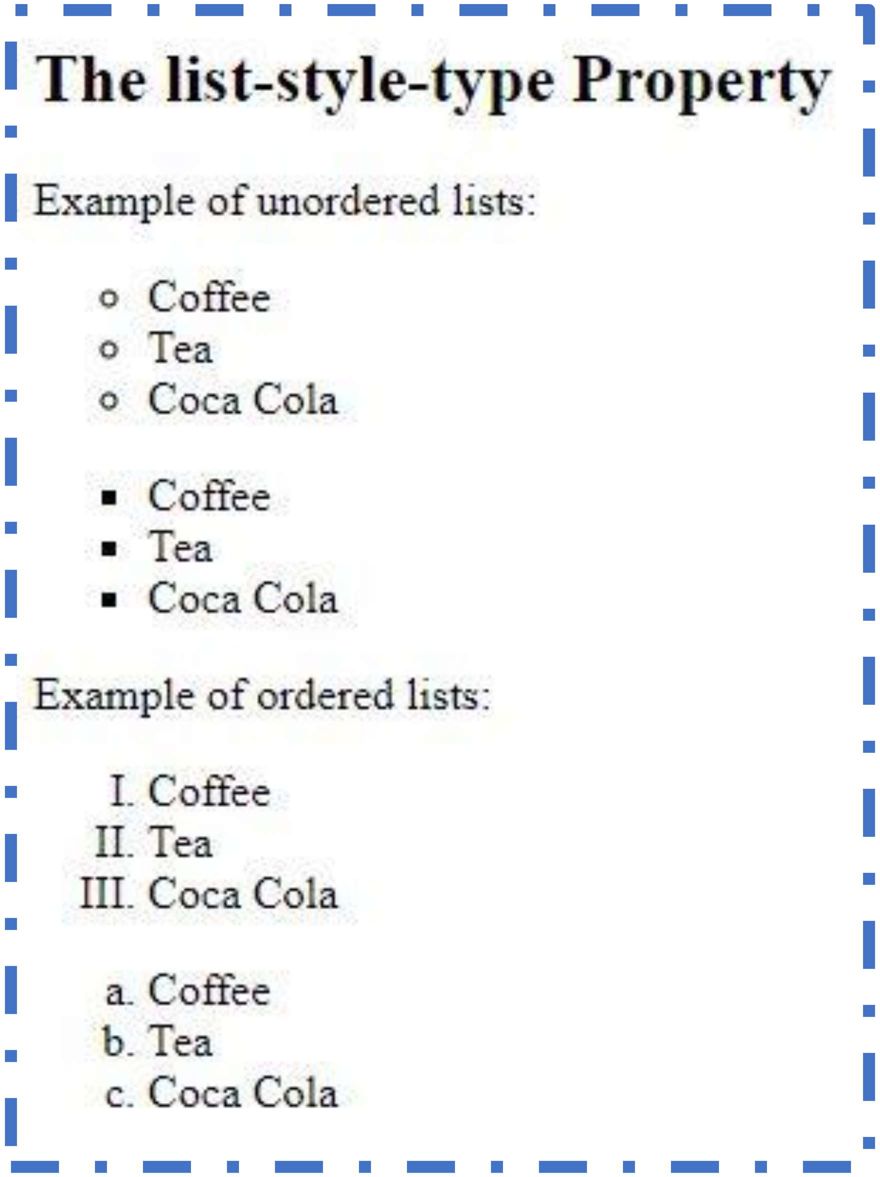
CSS LISTS

```
<style>
ul.a {
  list-style-type: circle;
}

ul.b {
  list-style-type: square;
}

ol.c {
  list-style-type: upper-roman;
}

ol.d {
  list-style-type: lower-alpha;
}
</style>
```

```
<h2>The list-style-type Property</h2>

<p>Example of unordered lists:</p>
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<ul class="b">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

```
<p>Example of ordered lists:</p>
<ol class="c">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>

<ol class="d">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
```

CSS LISTS - OUTPUT



The list-style-type Property

Example of unordered lists:

○ Coffee
○ Tea
○ Coca Cola

▪ Coffee
▪ Tea
▪ Coca Cola

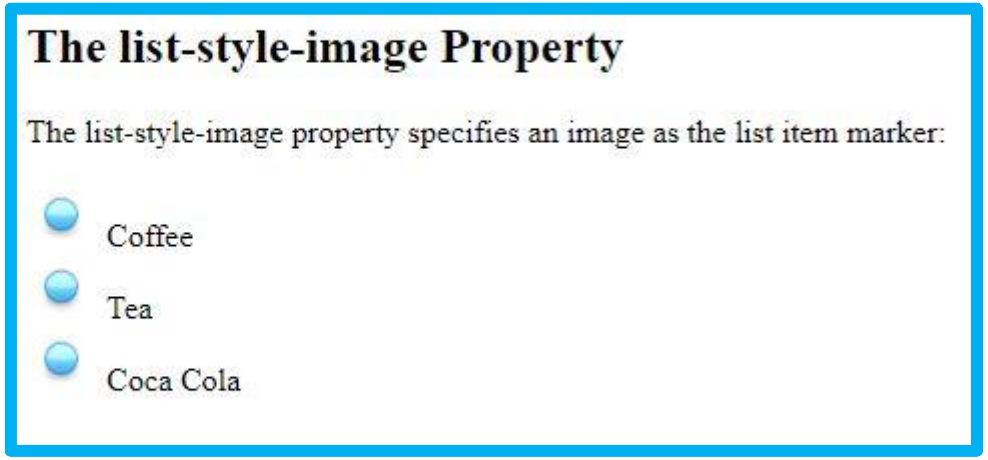
Example of ordered lists:

I. Coffee
II. Tea
III. Coca Cola

a. Coffee
b. Tea
c. Coca Cola

CSS LISTS AS AN IMAGE

```
<style>
ul {
  list-style-image:  url('bullet_blue.png');
}
</style>
```



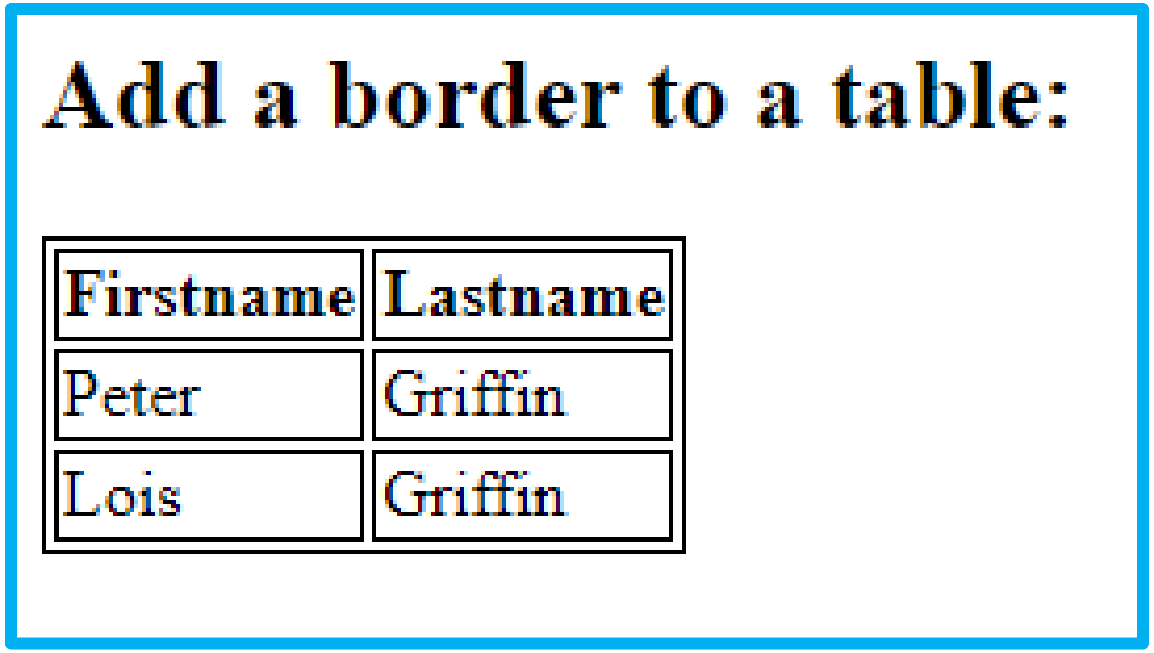
The list-style-image Property

The list-style-image property specifies an image as the list item marker:

- Coffee
- Tea
- Coca Cola

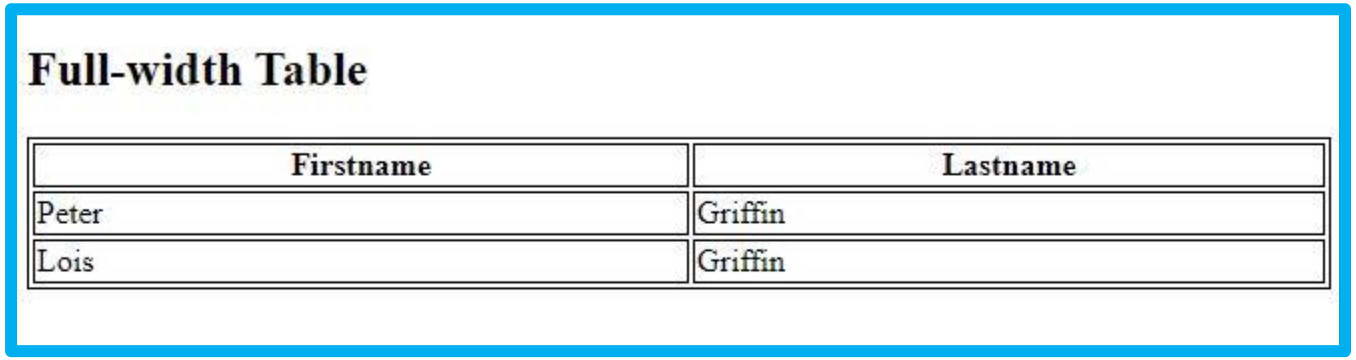
CSS TABLES

```
<style>
table, th, td {
  border: 1px
solid;
}

table {
  width: 100%;
}
</style>
```



Add a border to a table:

Firstname	Lastname
Peter	Griffin
Lois	Griffin

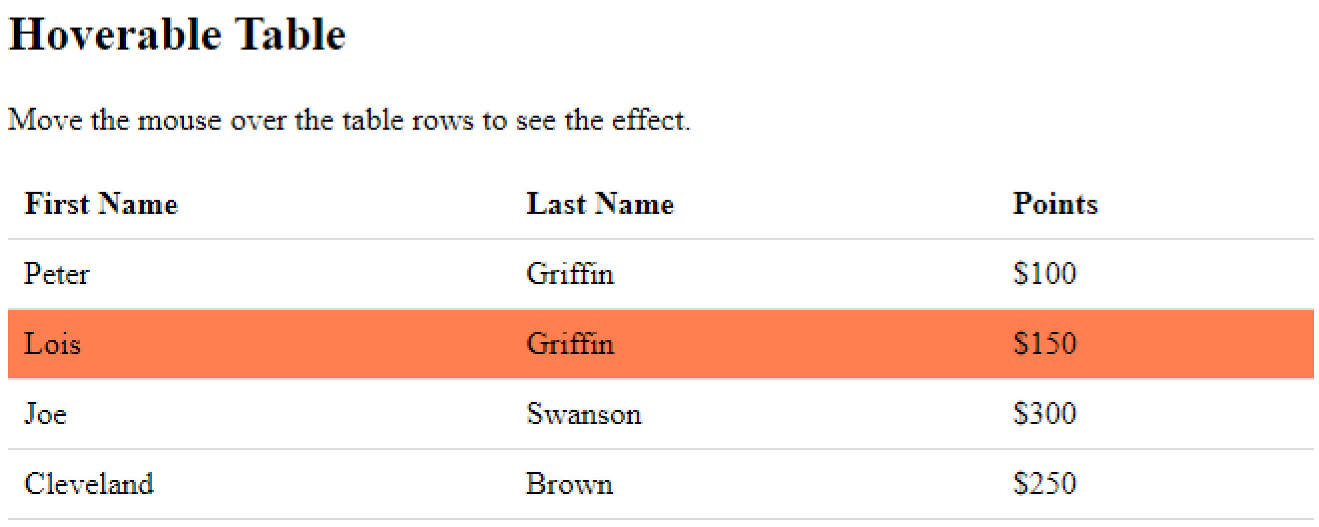


Full-width Table

Firstname	Lastname
Peter	Griffin
Lois	Griffin

CSS TABLES

```
<style>
table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  padding: 8px;
  text-align: left;
  border-bottom: 1px solid #ddd;
}

tr:hover {background-color: coral;}
</style>
```



Hoverable Table

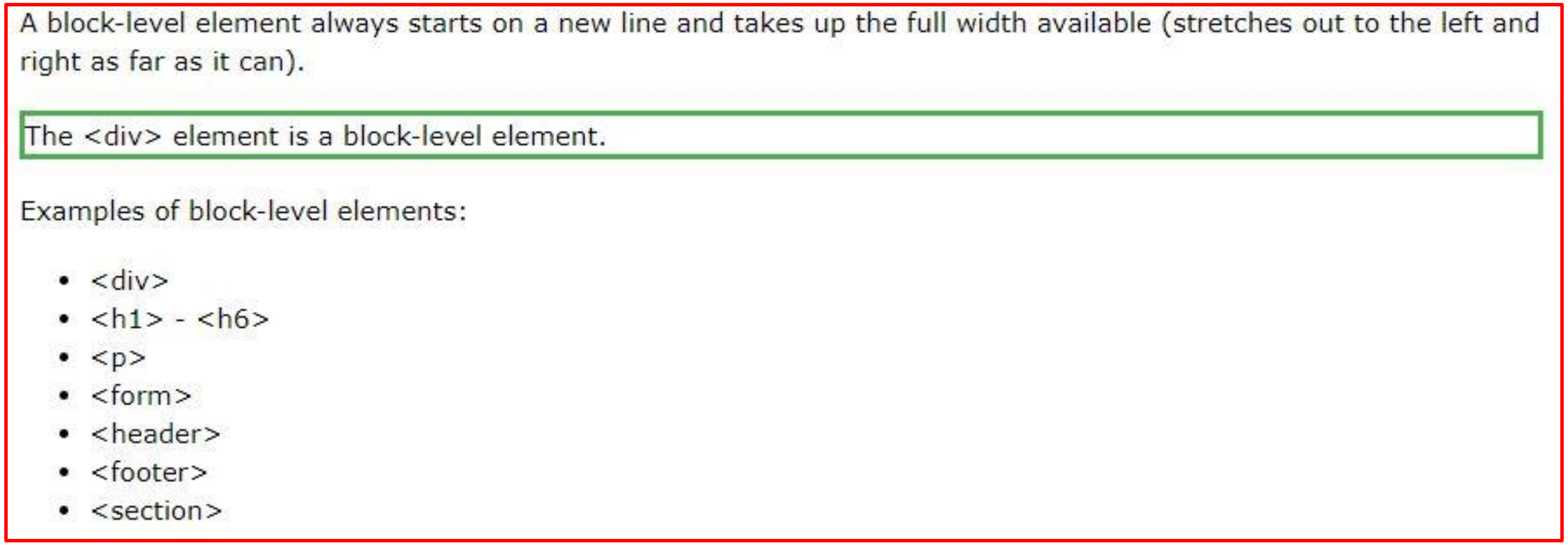
Move the mouse over the table rows to see the effect.

First Name	Last Name	Points
Peter	Griffin	$100
Lois	Griffin	$150
Joe	Swanson	$300
Cleveland	Brown	$250

CSS Layout – The display Property

- The display property is the most important CSS property for controlling layout.
- The display property specifies if/how an element is displayed.

Block-level Elements



A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

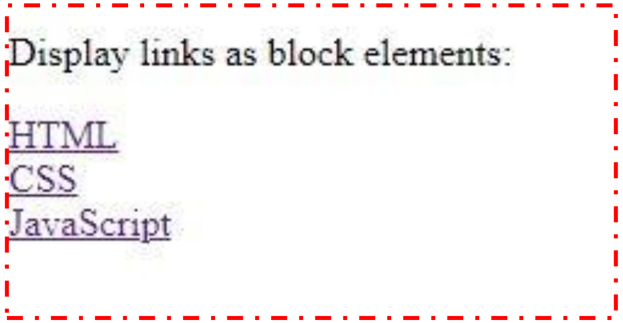
The <div> element is a block-level element.

Examples of block-level elements:

- <div>
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

CSS Layout – The display Property – Block Elements

```
<style>
a {
  display: block;
}
</style>



<p>Display links as block elements:</p>

<a href="/html/default.asp" target="_blank">HTML</a>
<a href="/css/default.asp" target="_blank">CSS</a>
<a href="/js/default.asp" target="_blank">JavaScript</a>
```

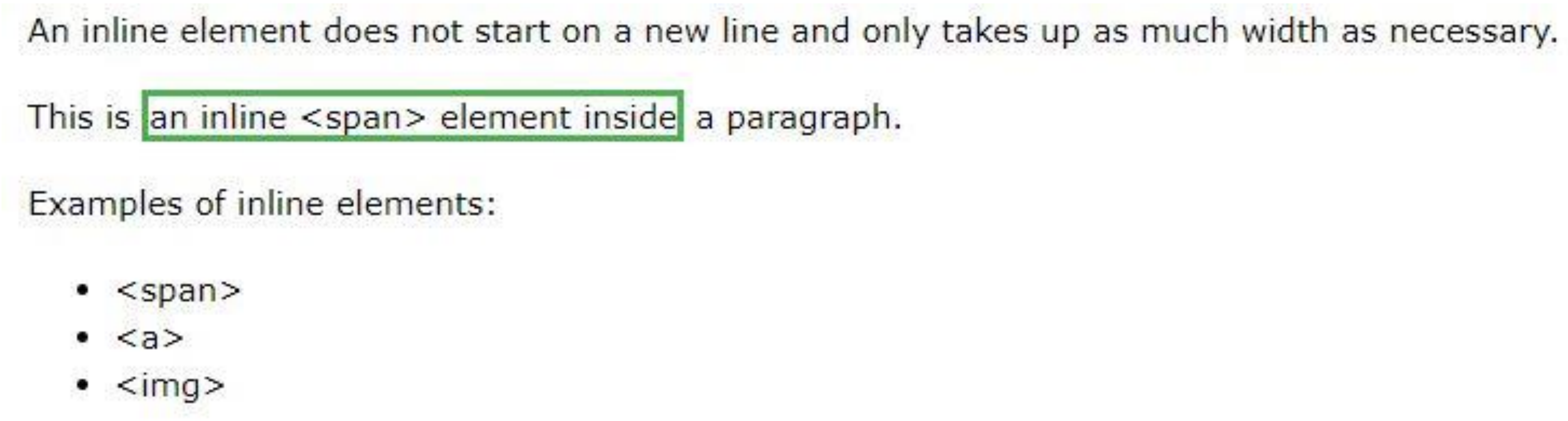


Display links as block elements:

HTML
CSS
JavaScript

CSS Layout – The display Property

Inline Elements



An inline element does not start on a new line and only takes up as much width as necessary.

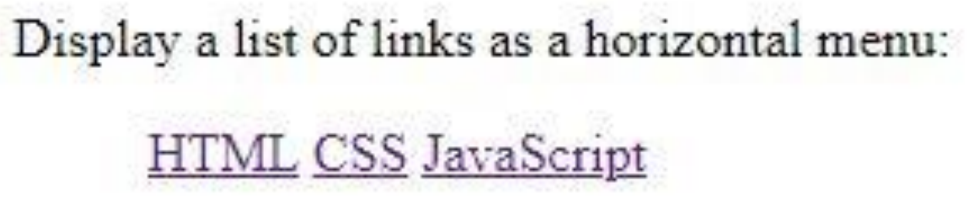
This is an inline <span> element inside a paragraph.

Examples of inline elements:

- <span>
- <a>
- <img>

CSS Layout – The display Property – Inline Elements

```
<style>
li {
  display: inline;
}
</style>
</head>
<body>

<p>Display a list of links as a horizontal menu:</p>

<ul>
  <li><a href="" target="_blank">HTML</a></li>
  <li><a href="" target="_blank">CSS</a></li>
  <li><a href="" target="_blank">JavaScript</a></li>
</ul>

</body>
```



Display a list of links as a horizontal menu:

HTML CSS JavaScript

CSS Layout – The display Property – None/Hidden

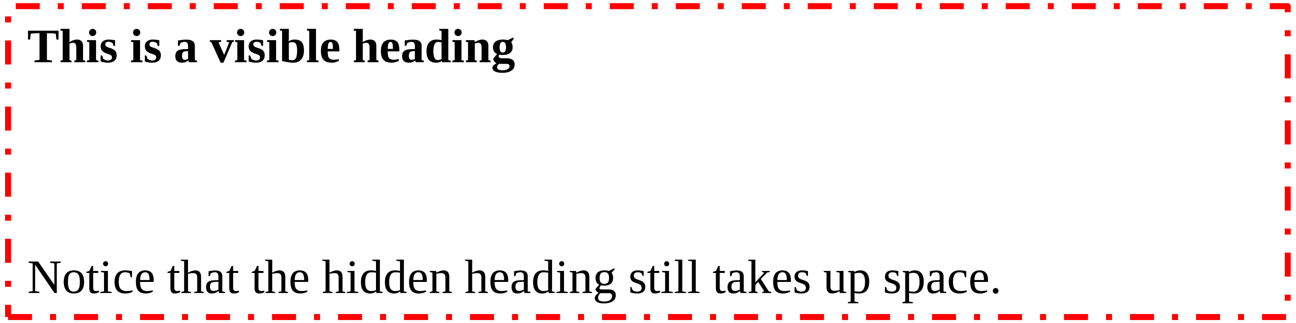
```
<style>
h1.hidden {
  display: none;
}
</style>

<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the h1 element with display: none;
does not take up any space.</p>

</body>
```

This is a visible heading

Notice that the h1 element with display: none; does not take up any space.

CSS Layout – The display Property – None/Hidden

```
<style>
h1.hidden {
  visibility: hidden;}
</style>

<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the h1 element with display: none;
does not take up any space.</p>

</body>
```



This is a visible heading

Notice that the hidden heading still takes up space.

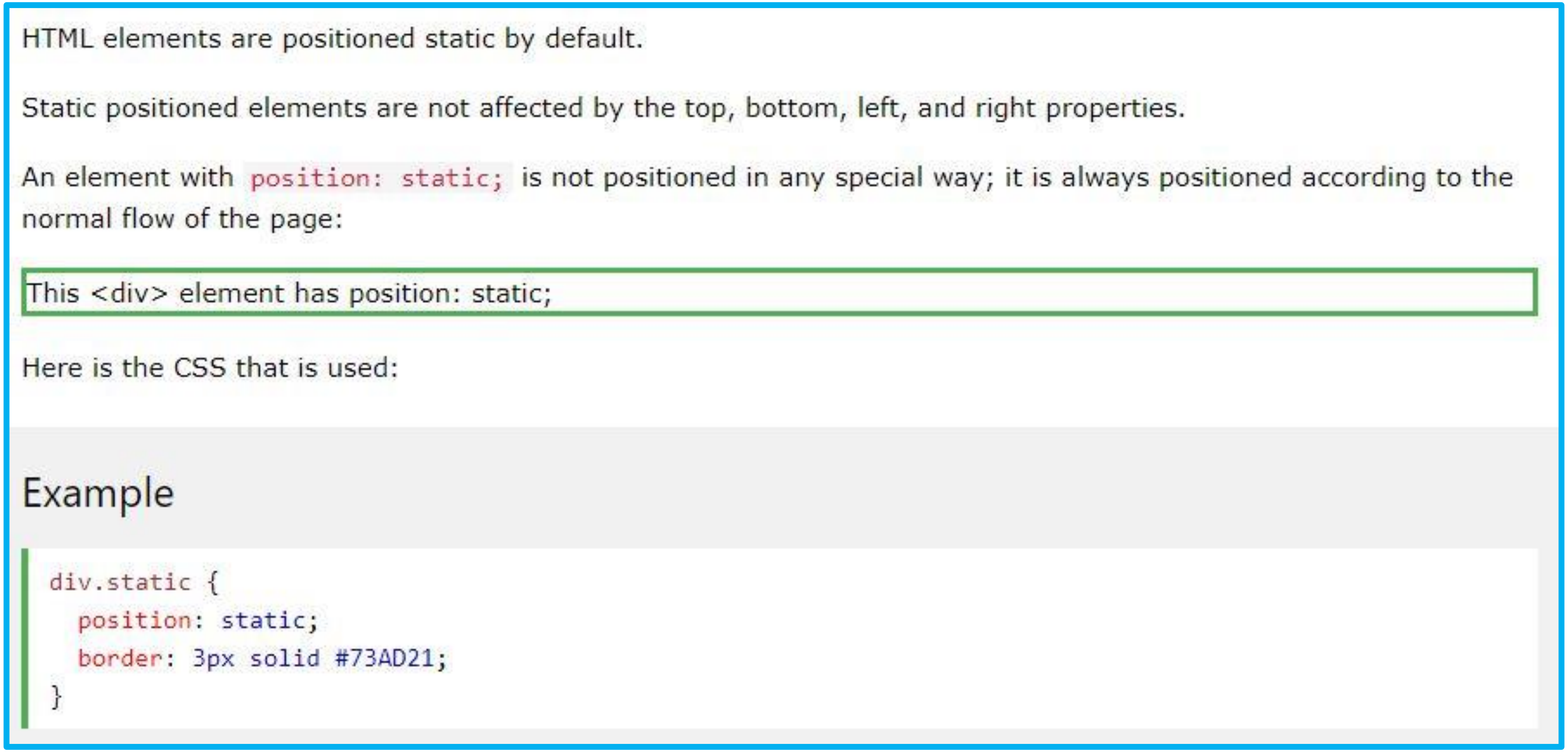
CSS Layout – The Position Property

The position property specifies the type of positioning method used for an element.

There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

CSS Layout – The Position Property - 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 <div> element has position: static;

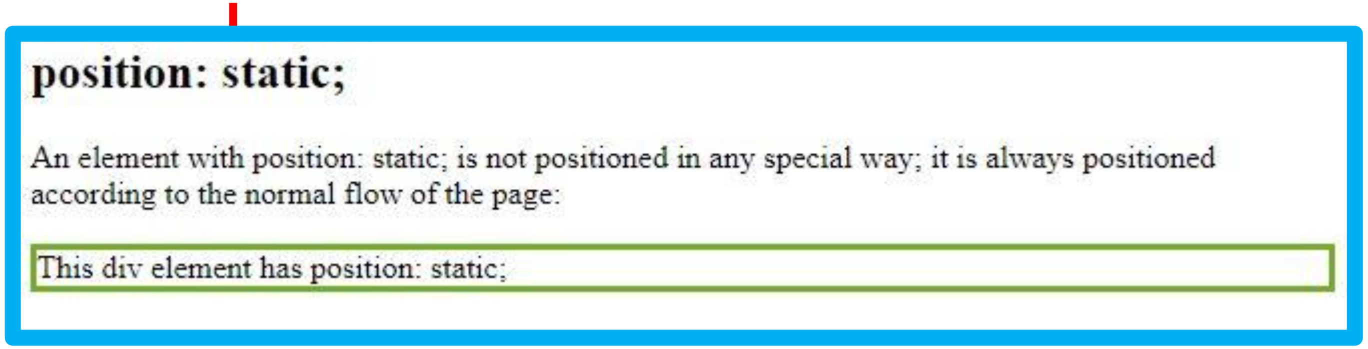
Here is the CSS that is used:

Example

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

Position Property - Static

```
<style>
div.static {
  position: static;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: static;</h2>

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

<div class="static">
  This div element has position: static;
</div>

</body>
```

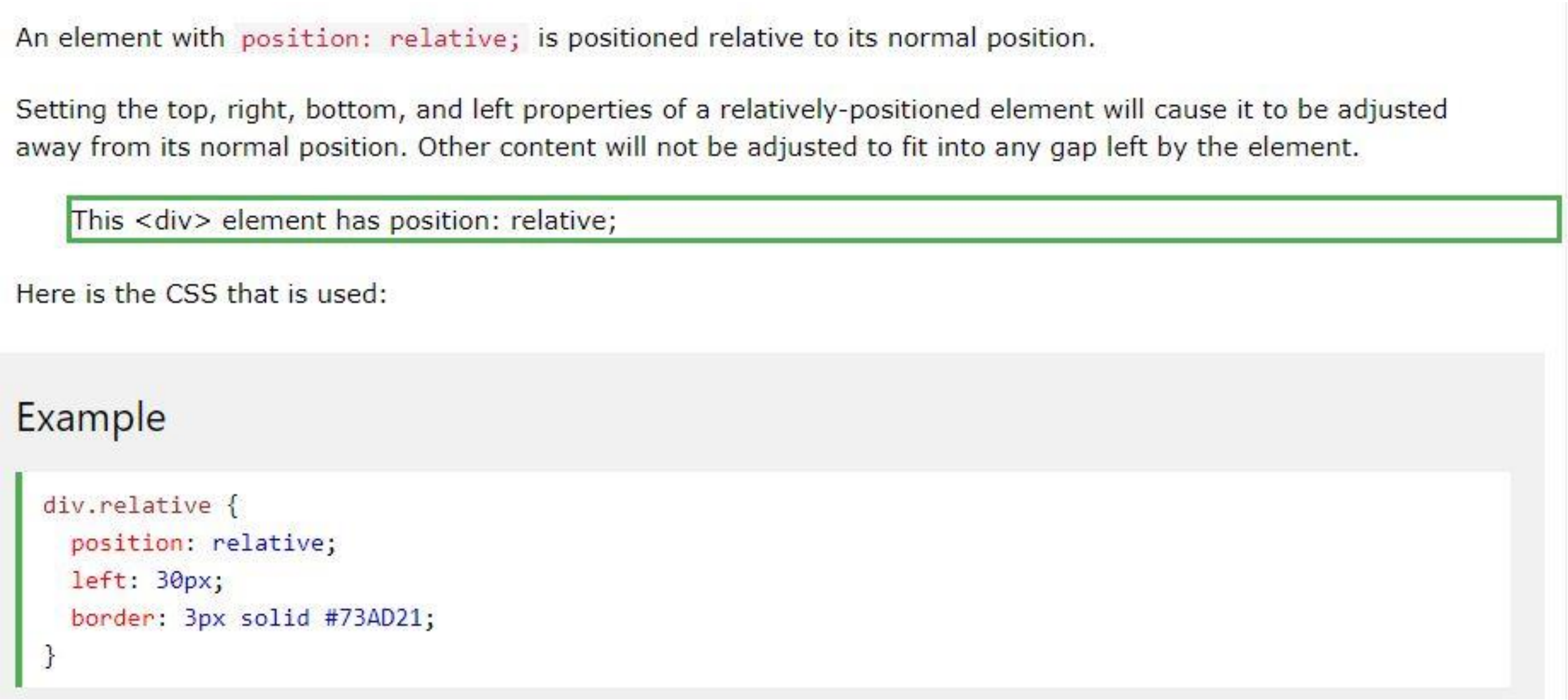


position: static;

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 div element has position: static;

CSS Layout – The Position Property - 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 <div> element has position: relative;

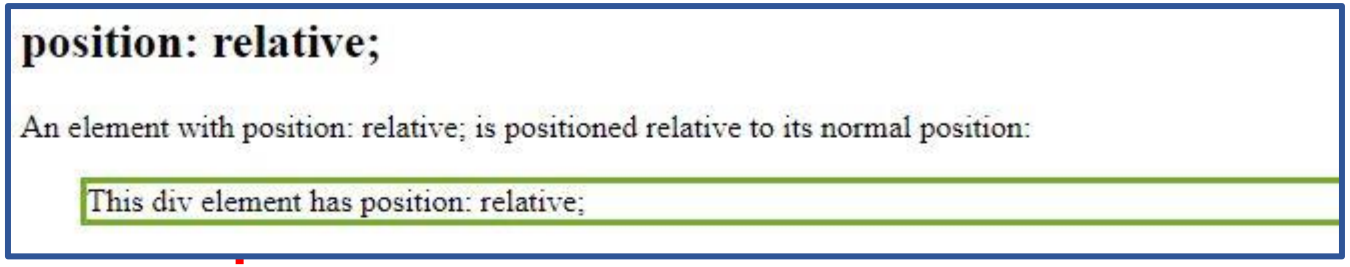
Here is the CSS that is used:

Example

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

Position Property - Relative

```
<style>
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: relative;</h2>

<p>An element with position: relative; is positioned relative to
its normal position:</p>

<div class="relative">
This div element has position: relative;
</div>

</body>
```

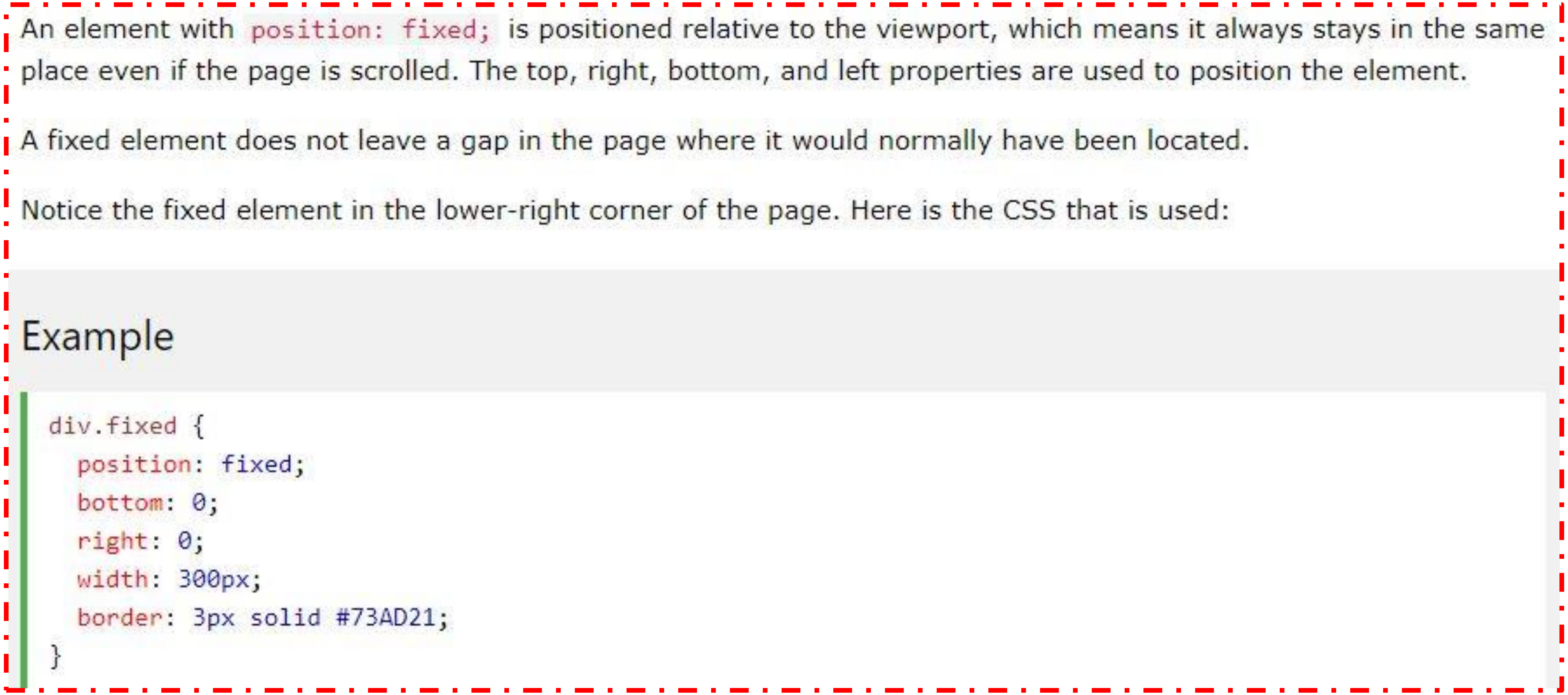


position: relative;

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

This div element has position: relative;

CSS Layout – The Position Property - Fixed



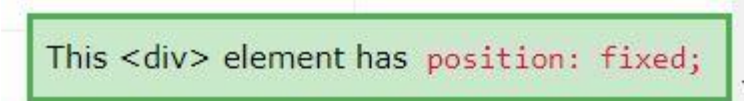
An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

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

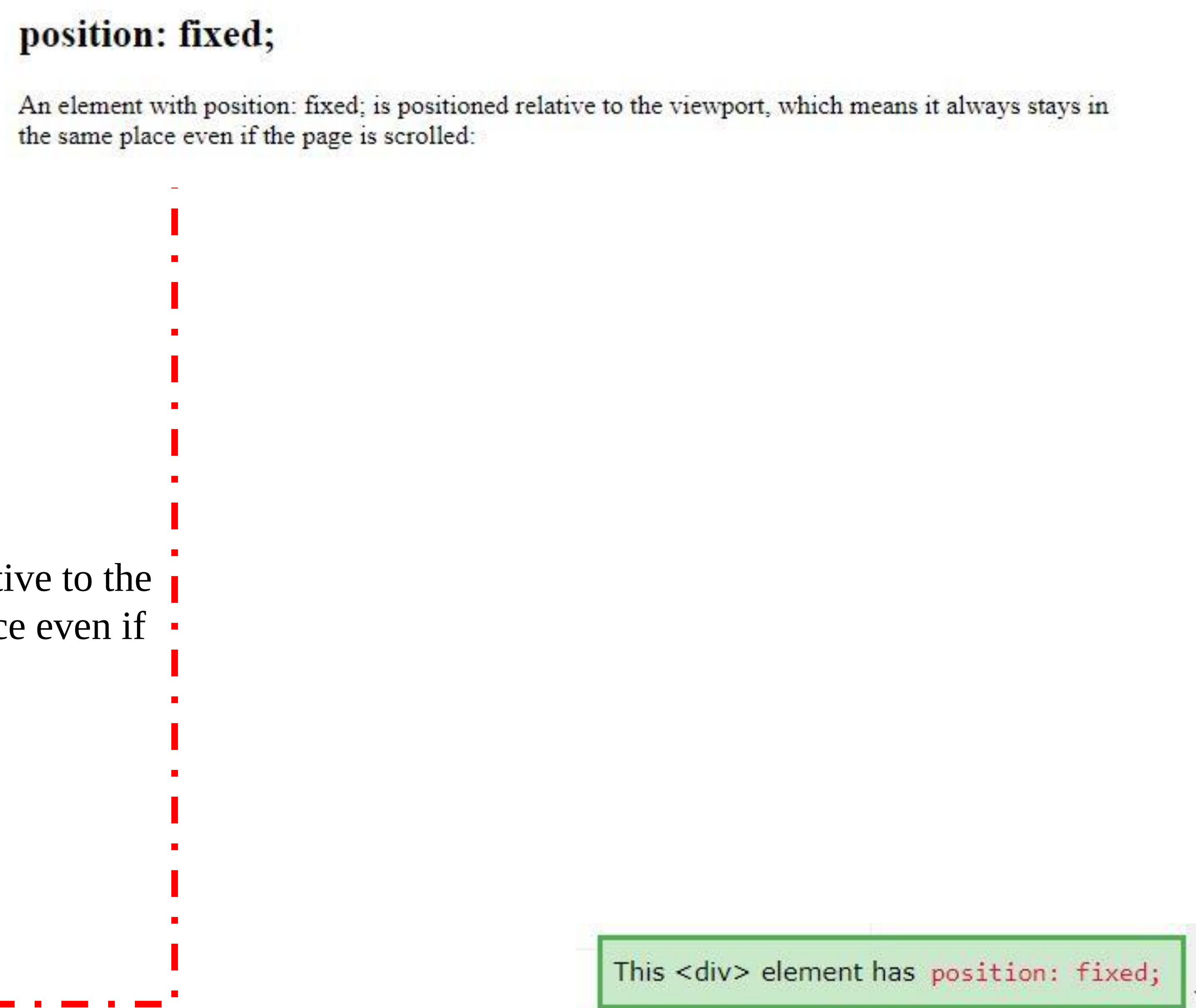
```
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
```



This <div> element has `position: fixed;`

Position Property - Fixed

```
<style>
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: fixed;</h2>

<p>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:</p>

<div class="fixed">
This div element has position: fixed;
</div>

</body>
```

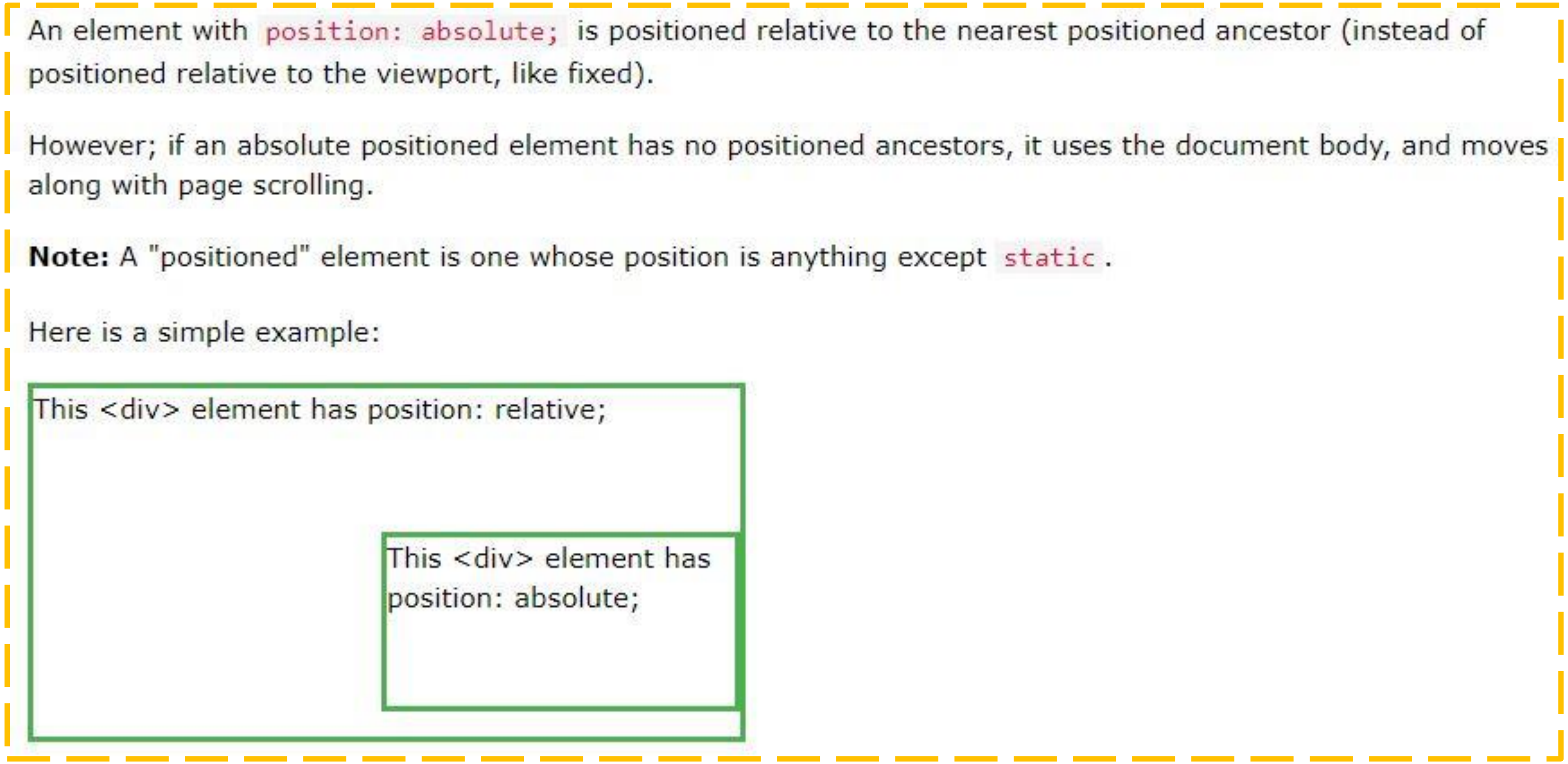


position: fixed;

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:

This <div> element has `position: fixed;`

CSS Layout – The Position Property - Absolute



An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

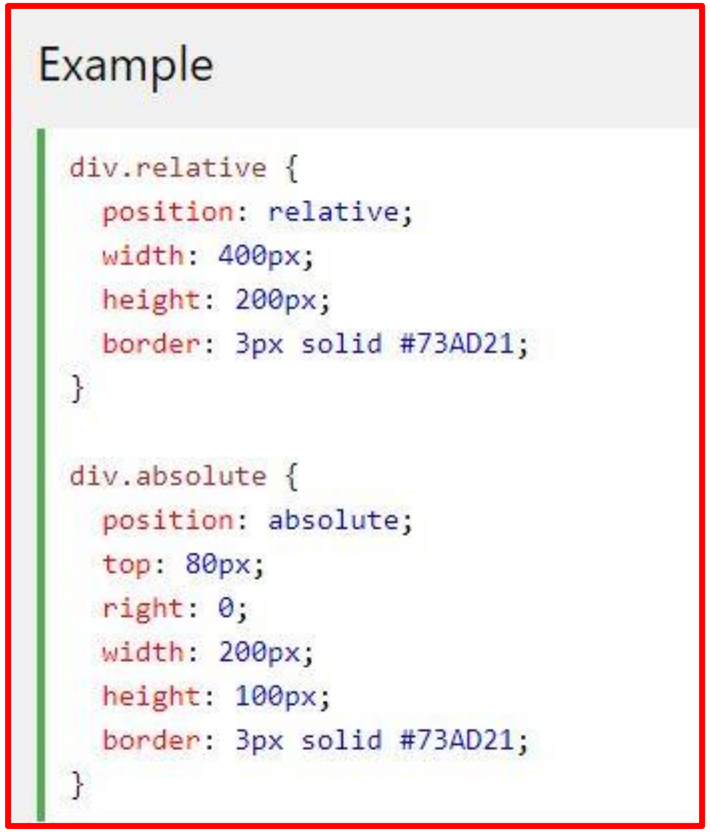
However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

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

Here is a simple example:

This <div> element has position: relative;

This <div> element has position: absolute;



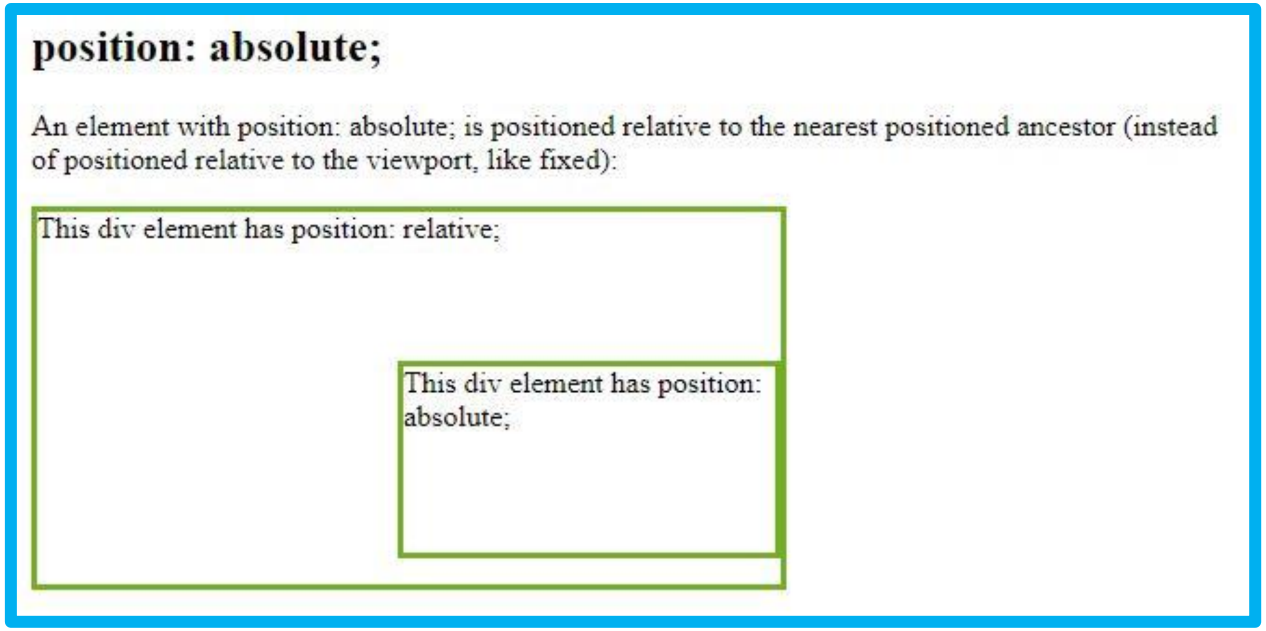
Example

```
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;
}
```

Position Property - Absolute

```
<style>
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;
}
</style>
</head>
```

```
<body>

<h2>position: absolute;</h2>

<p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>

<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
```



position: absolute;

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):

This div element has position: relative;

This div element has position: absolute;

CSS Float

- The CSS **float** property specifies how an element should float.
- The CSS **clear** property specifies what elements can float beside the cleared element and on which side.

The float property can have one of the following values:

•left - The element floats to the left of its container

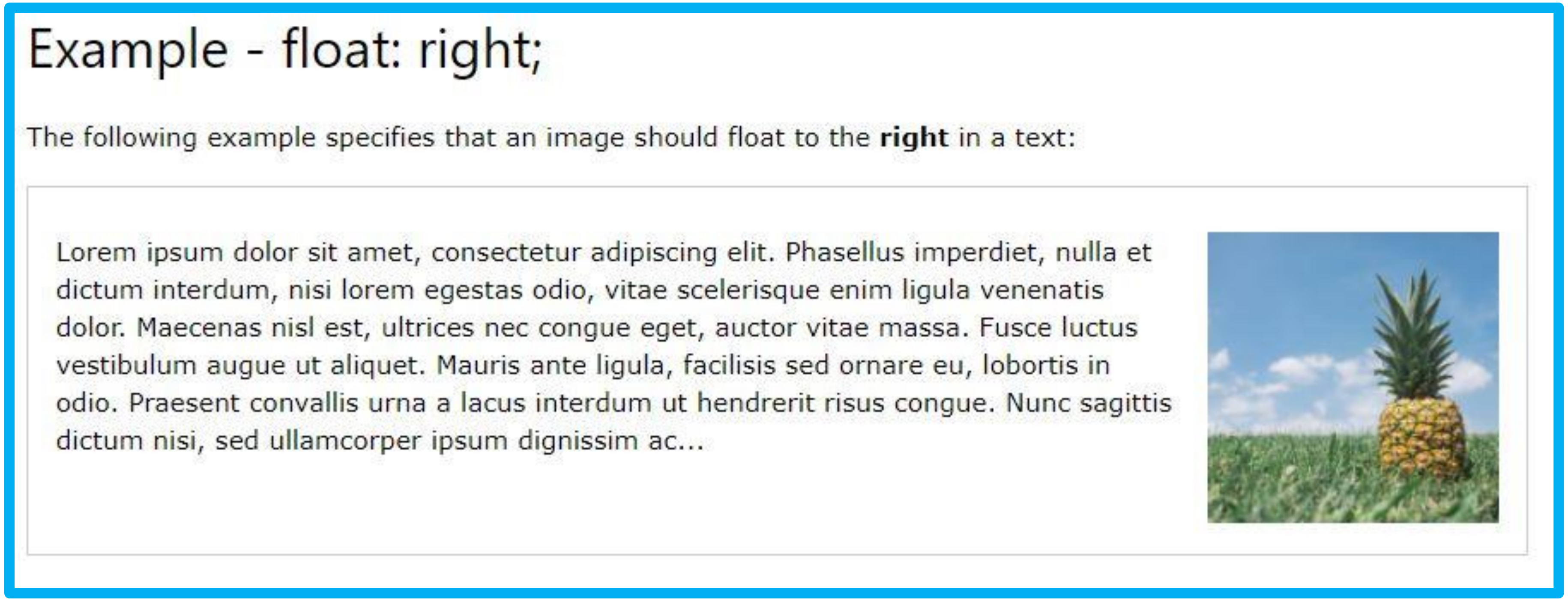
•right - The element floats to the right of its container

•none - The element does not float (will be displayed just where it occurs in the text). This is default

•inherit - The element inherits the float value of its parent

```
img {
  float: right;
}
```

CSS Float



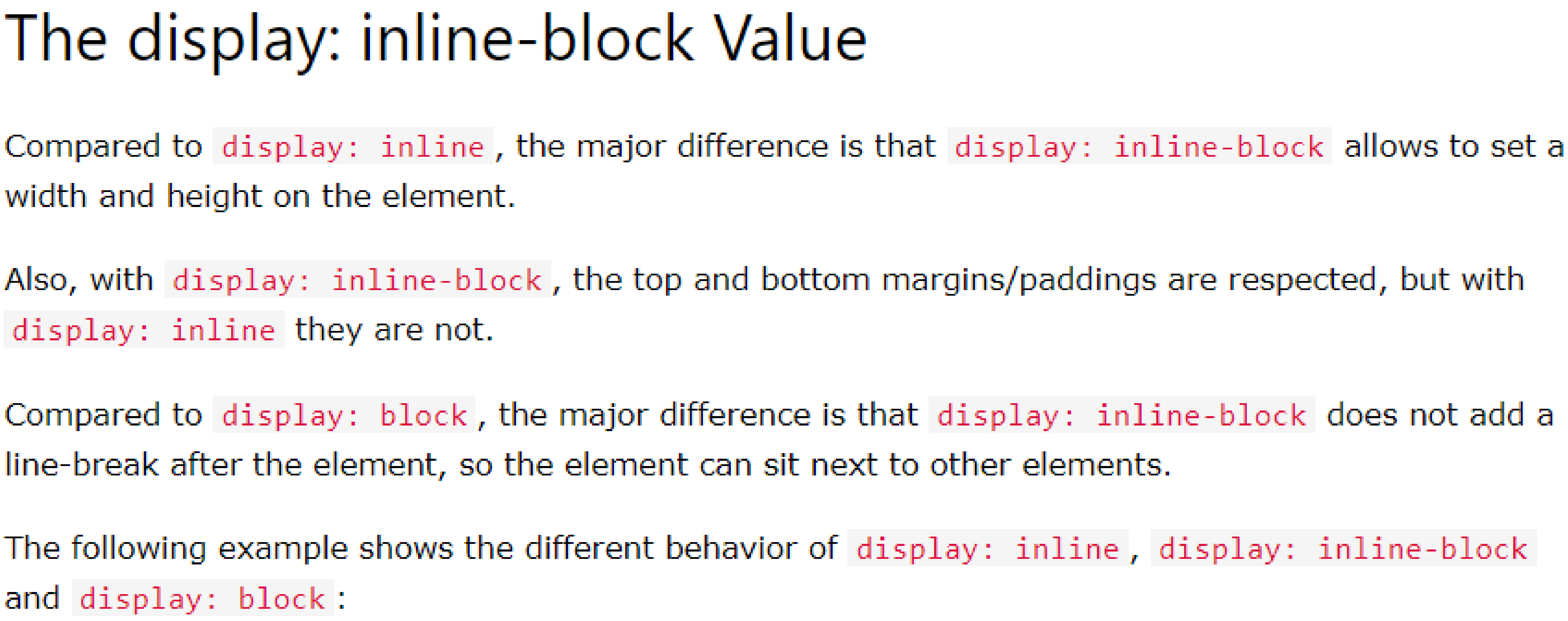
Example - float: right;

The following example specifies that an image should float to the **right** in a text:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

```css
img {
  float: right;
}
```

CSS Layout – display : inline-block



The display: inline-block Value

Compared to `display: inline`, the major difference is that `display: inline-block` allows to set a width and height on the element.

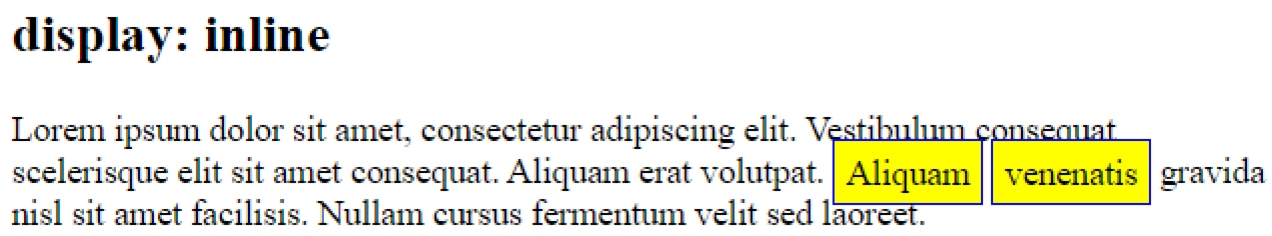
Also, with `display: inline-block`, the top and bottom margins/paddings are respected, but with `display: inline` they are not.

Compared to `display: block`, the major difference is that `display: inline-block` does not add a line-break after the element, so the element can sit next to other elements.

The following example shows the different behavior of `display: inline`, `display: inline-block` and `display: block`:

CSS Layout – display : inline-block

```
<html>
<head>
<style>
span.a {
  display: inline; /* the default for span */
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}
```

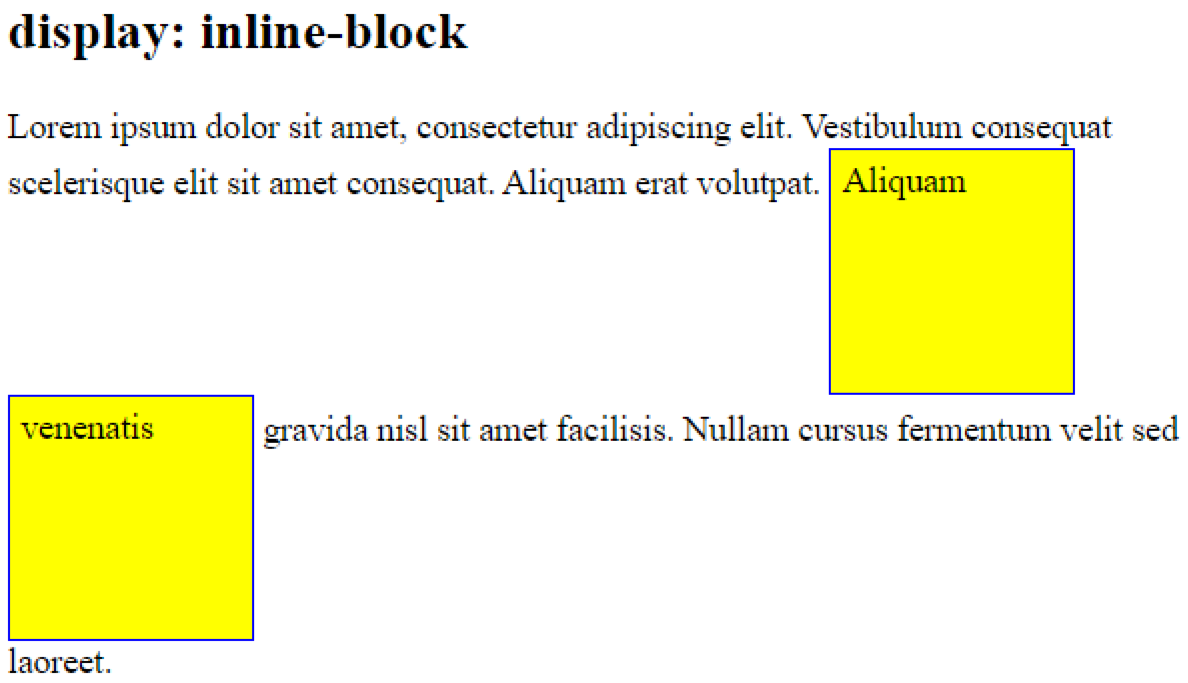


display: inline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

CSS Layout – display : inline-block

```
<html>
<head>
<style>
span.b {
  display: inline-block;
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}
```

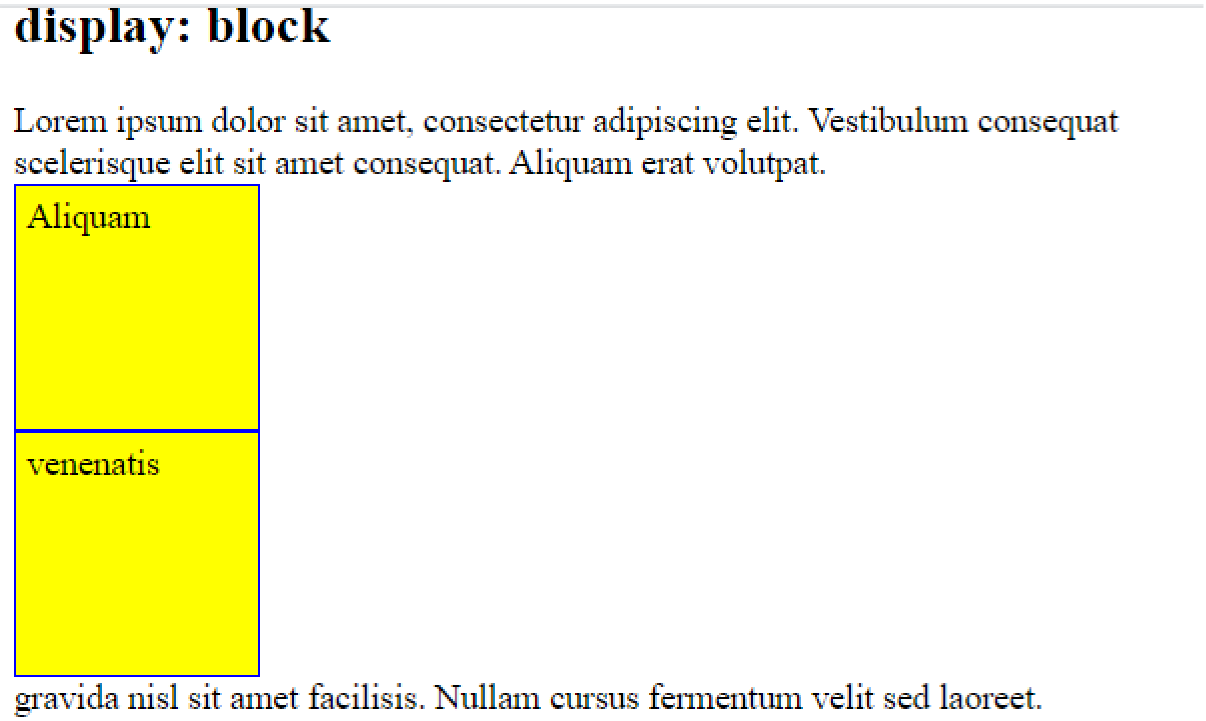


display: inline-block

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

CSS Layout – display : inline-block

```
<html>
<head>
<style>
span.c {
  display: block;
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}
</style>
</head>
```



display: block

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat.

Aliquam

venenatis

gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

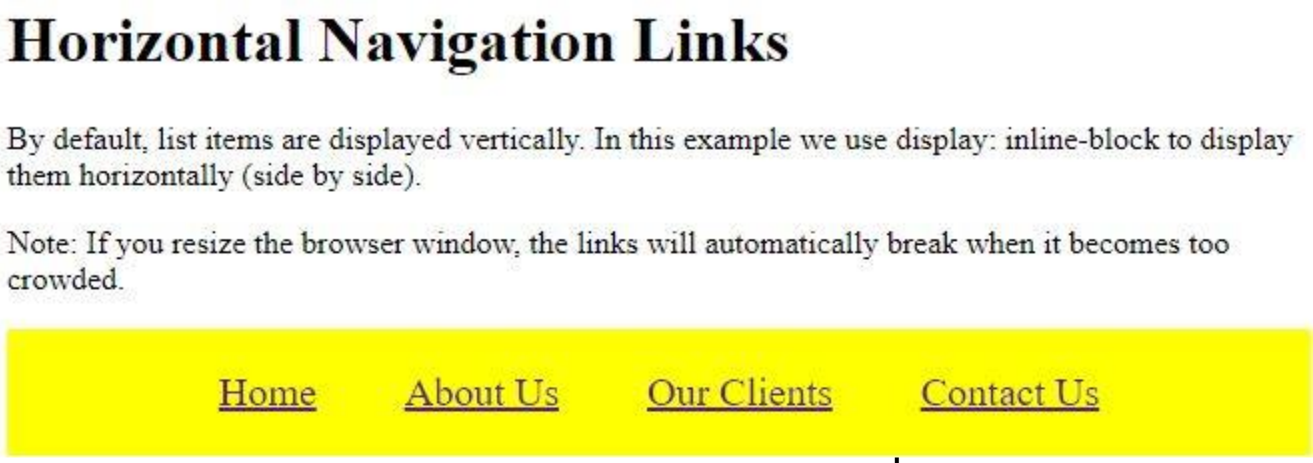
CSS Layout – display : inline-block

```
<body>

<h1>The display Property</h1>

<h2>display: inline</h2>
<div>Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Vestibulum
consequat scelerisque elit sit amet
consequat. Aliquam erat volutpat. <span
class="a">Aliquam</span> <span
class="a">venenatis</span> gravida nisl sit
amet facilisis. Nullam cursus fermentum
velit sed laoreet. </div>
```

```
<h2>display: inline-block</h2>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum
consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. <span
class="b">Aliquam</span> <span class="b">venenatis</span> gravida nisl sit
amet facilisis. Nullam cursus fermentum velit sed laoreet. </div>
```

```
<h2>display: block</h2>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum
consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. <span
class="c">Aliquam</span> <span class="c">venenatis</span> gravida nisl sit
amet facilisis. Nullam cursus fermentum velit sed laoreet. </div>

</body>
</html>
```

CSS Layout - display: inline-block



Horizontal Navigation Links

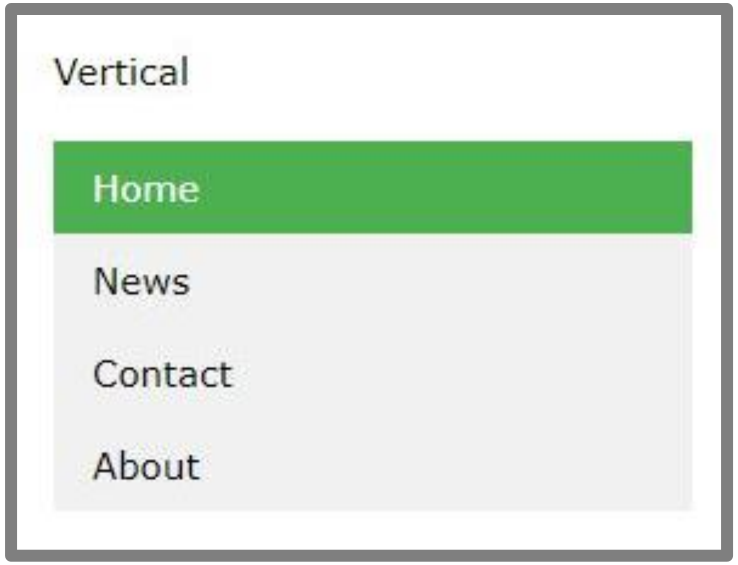
By default, list items are displayed vertically. In this example we use display: inline-block to display them horizontally (side by side).

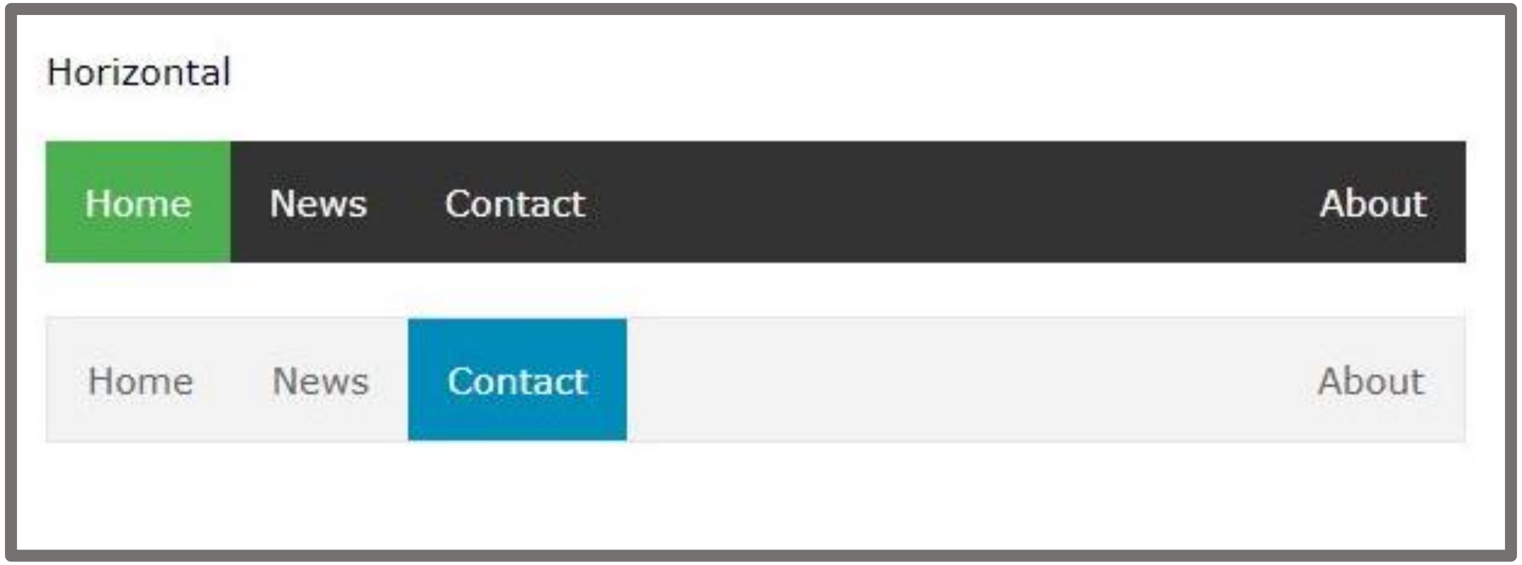
Note: If you resize the browser window, the links will automatically break when it becomes too crowded.

```
<head>
<style>
.nav {
  background-color: yellow;
  list-style-type: none;
  text-align: center;
  margin: 0;
  padding: 0;
}

.nav li {
  display: inline-block;
  font-size: 20px;
  padding: 20px;
}
</style>
<head>
```

```
<body>

<h1>Horizontal Navigation Links</h1>
<p>By default, list items are displayed vertically. In this example
we use display: inline-block to display them horizontally (side
by side).</p>
<p>Note: If you resize the browser window, the links will
automatically break when it becomes too crowded.</p>

<ul class="nav">
  <li><a href="#home">Home</a></li>
  <li><a href="#about">About Us</a></li>
  <li><a href="#clients">Our Clients</a></li>
  <li><a href="#contact">Contact Us</a></li>
</ul>

</body>
```

CSS Navigation bar





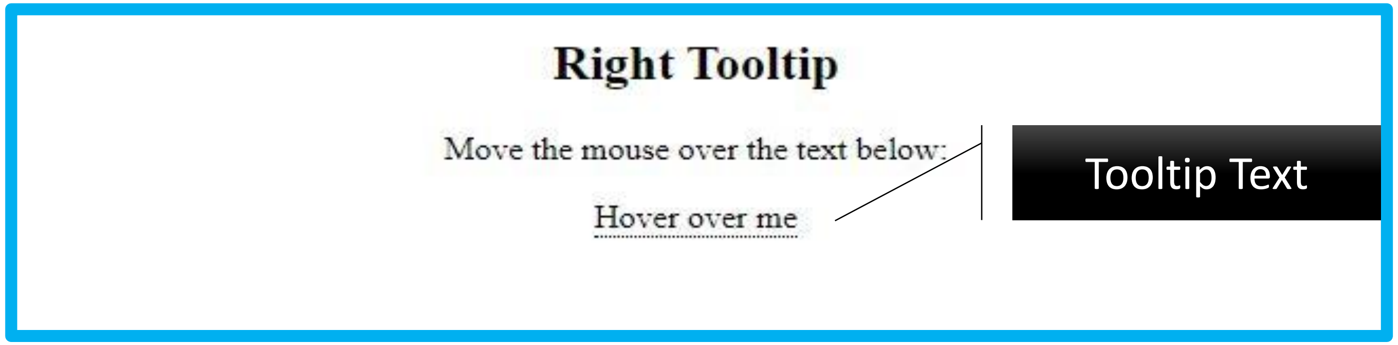
CSS TOOLTIP

A tooltip is often used to specify extra information about something when the user moves the mouse pointer over an element

```
<style>
.tooltip {
  position: relative;
  display: inline-block;
  border-bottom: 1px dotted black;
}

.tooltip .tooltiptext {
  visibility: hidden;
  width: 120px;
  background-color: black;
  color: #fff;
  text-align: center;
  border-radius: 6px;
  padding: 5px 0;
```

```
/* Position the tooltip */
  position: absolute;
  z-index: 1;
  top: -5px;
  left: 105%;
}

.tooltip:hover .tooltiptext {
  visibility: visible;
}
</style>
```

```

<body style="text-align:center;">

<h2>Right Tooltip</h2>
<p>Move the mouse over the text below:</p>

<div class="tooltip">Hover over me
  <span class="tooltiptext">Tooltip text</span>
</div>

</body>
```



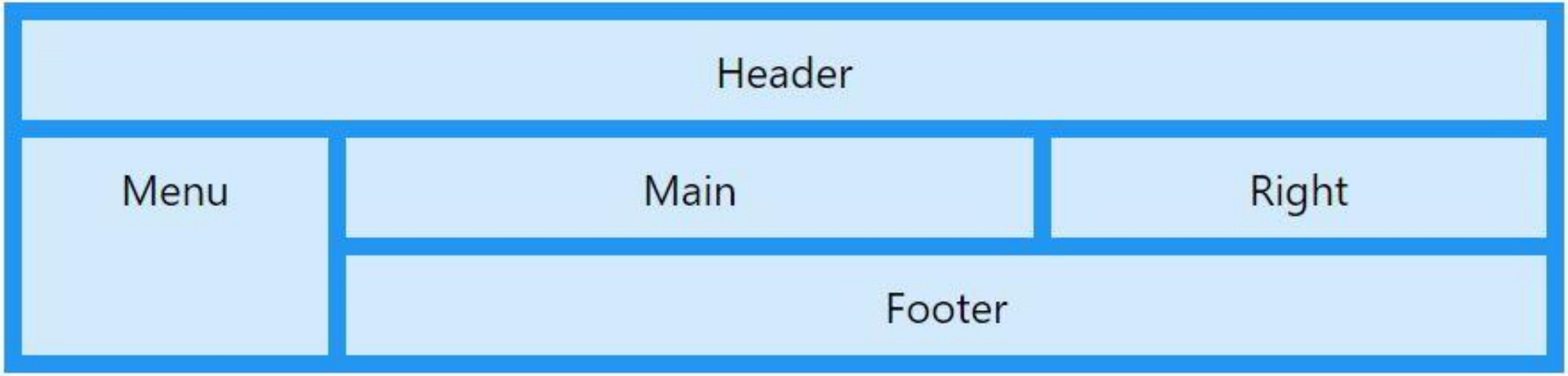
Right Tooltip

Move the mouse over the text below:

Hover over me

Tooltip Text

CSS GRID Layout Module



Header

Menu

Main

Right

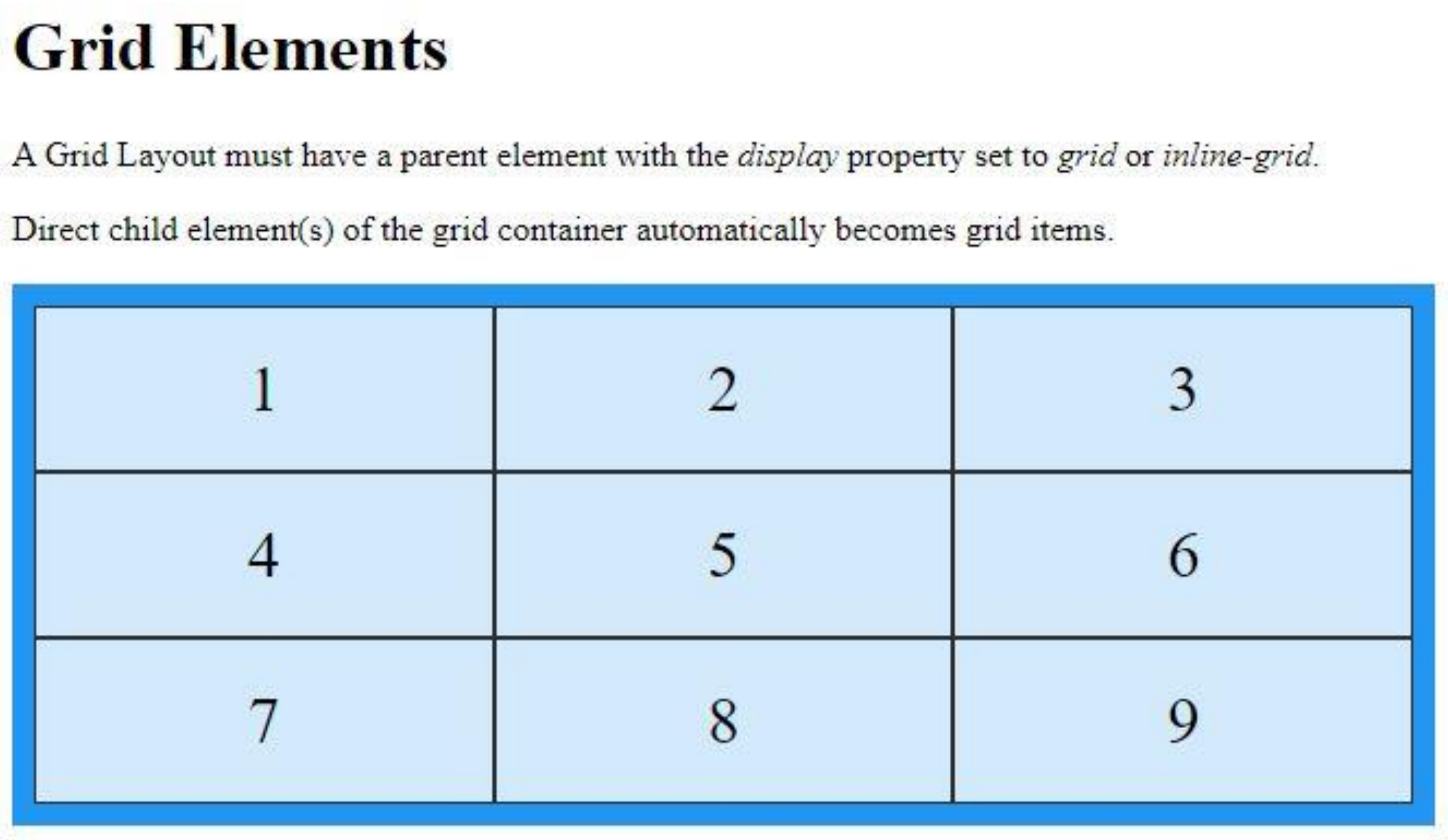
Footer

CSS GRID Layout Module

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
```

```
<h1>Grid Elements</h1>

<p>A Grid Layout must have a parent element with the
<em>display</em> property set to <em>grid</em> or
<em>inline-grid</em>.</p>

<p>Direct child element(s) of the grid container automatically
becomes grid items.</p>

<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

OUTPUT



Grid Elements

A Grid Layout must have a parent element with the *display* property set to *grid* or *inline-grid*.

Direct child element(s) of the grid container automatically becomes grid items.

1	2	3
4	5	6
7	8	9