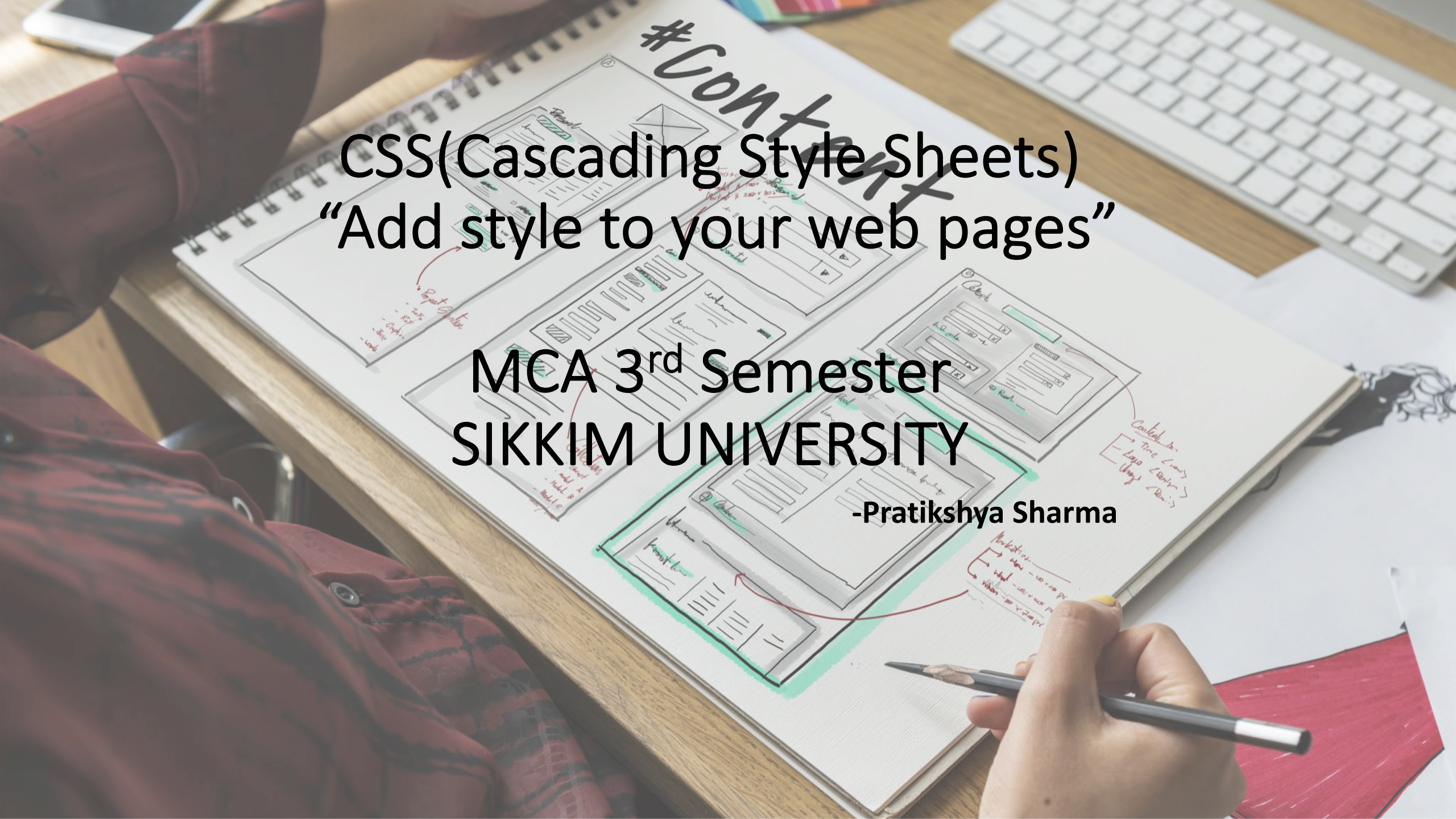
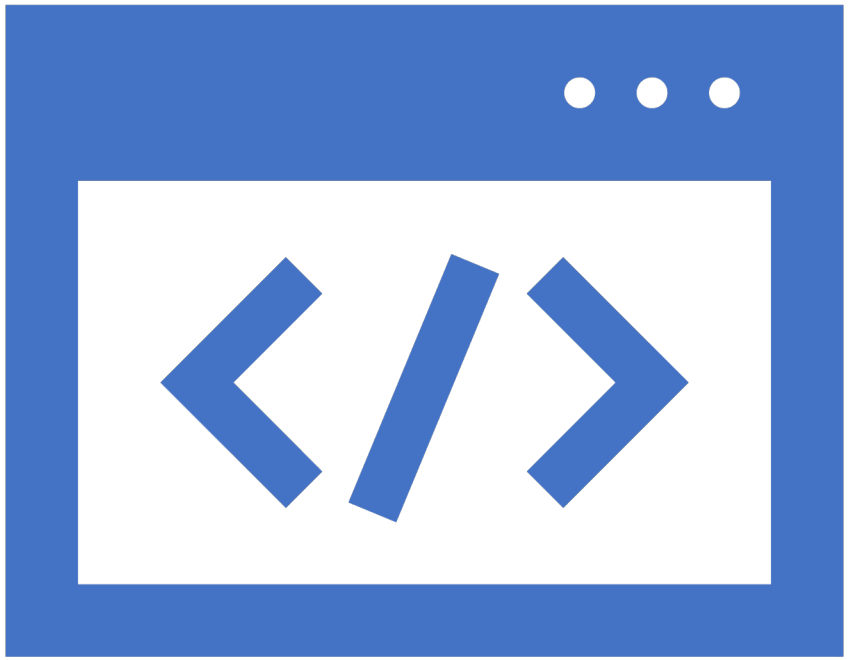


CSS(Cascading Style Sheets)
"Add style to your web pages"

MCA 3rd Semester
SIKKIM UNIVERSITY

-Pratikshya Sharma

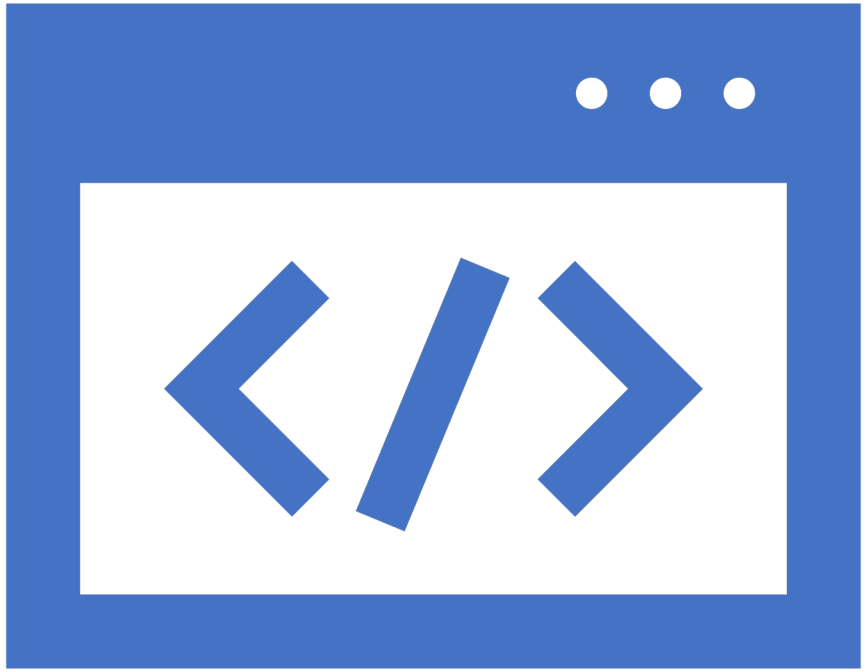


Introduction

- CSS is the language we use to style an HTML document (Web Page).
- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, 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

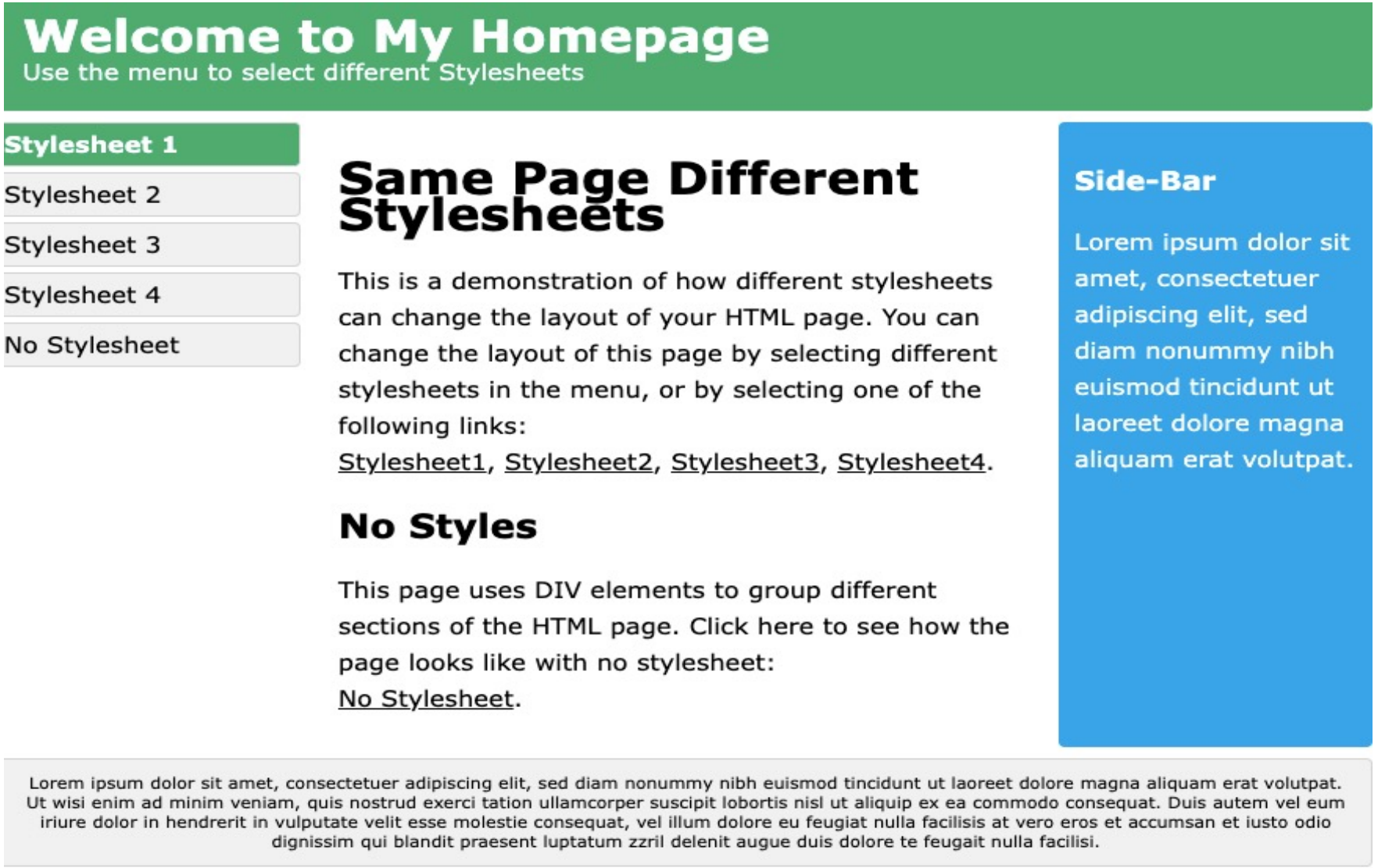
CSS Demo - One HTML Page - Multiple Styles!

- Here we will show one HTML page displayed with four different stylesheets. Click on the "Stylesheet 1", "Stylesheet 2", "Stylesheet 3", "Stylesheet 4" links below to see the different styles:

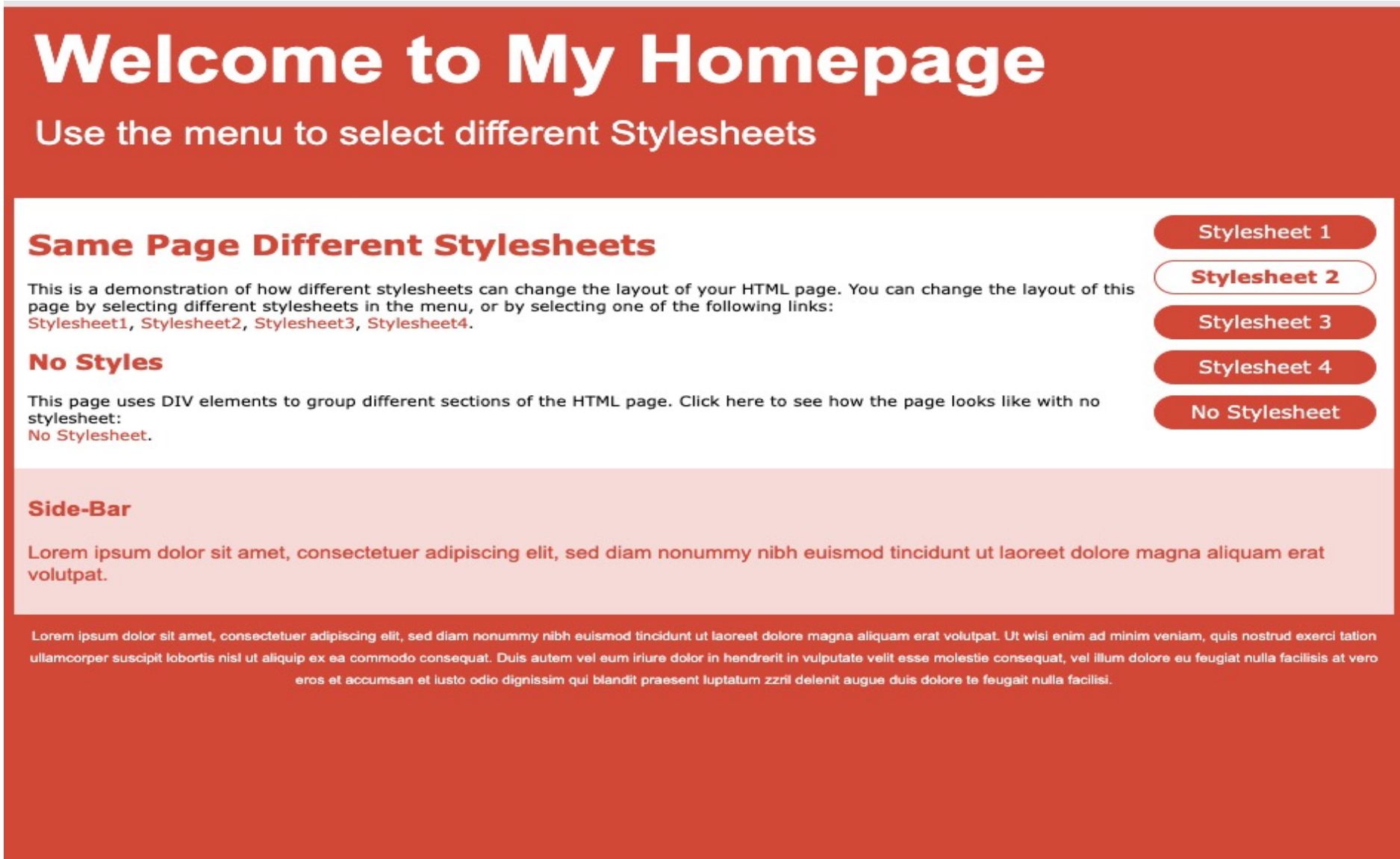


CSS Demo - One HTML Page - Multiple Styles!

- **Style 1**

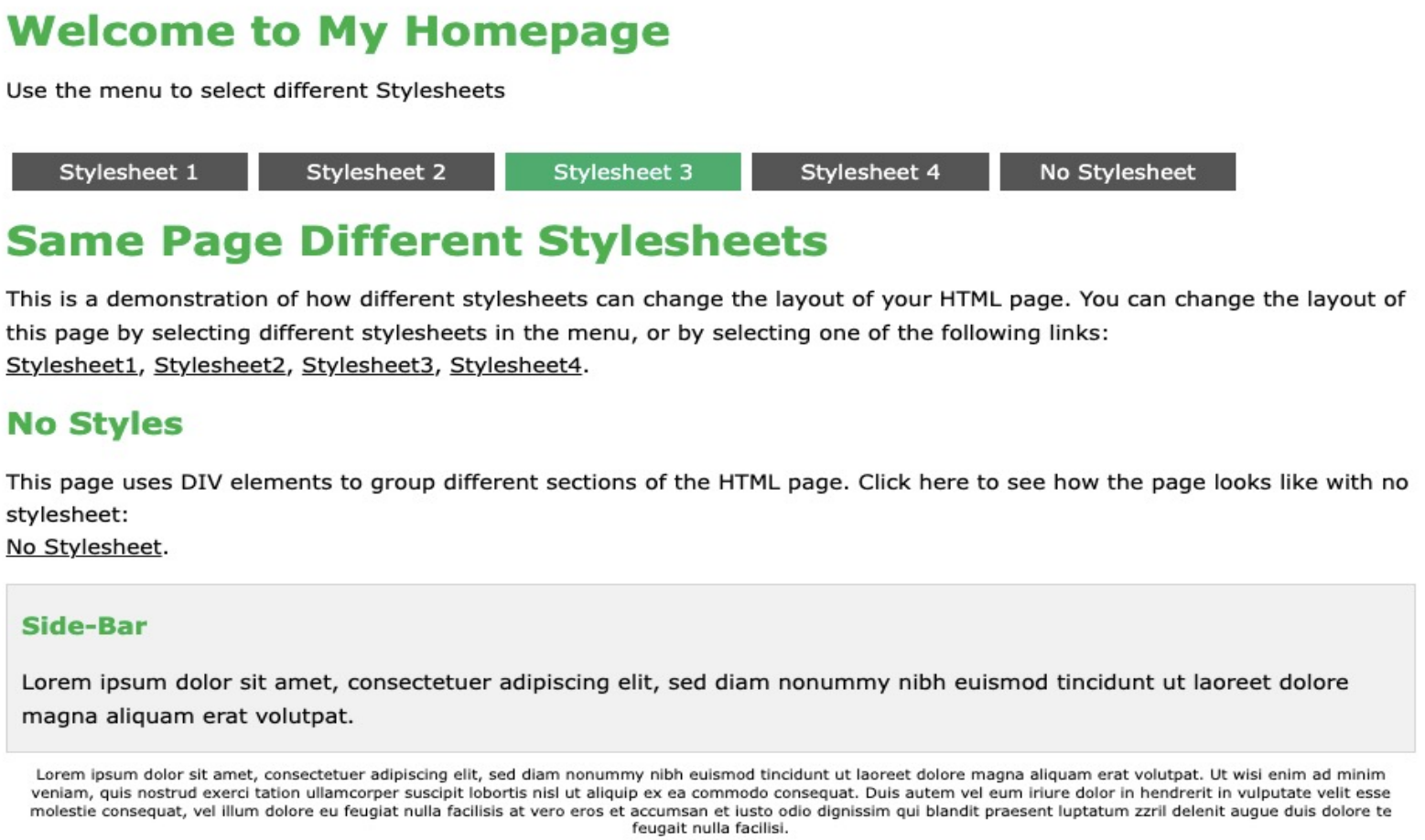


- Style 2


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.

CSS Demo - One HTML Page - Multiple Styles!

- **Style 3**

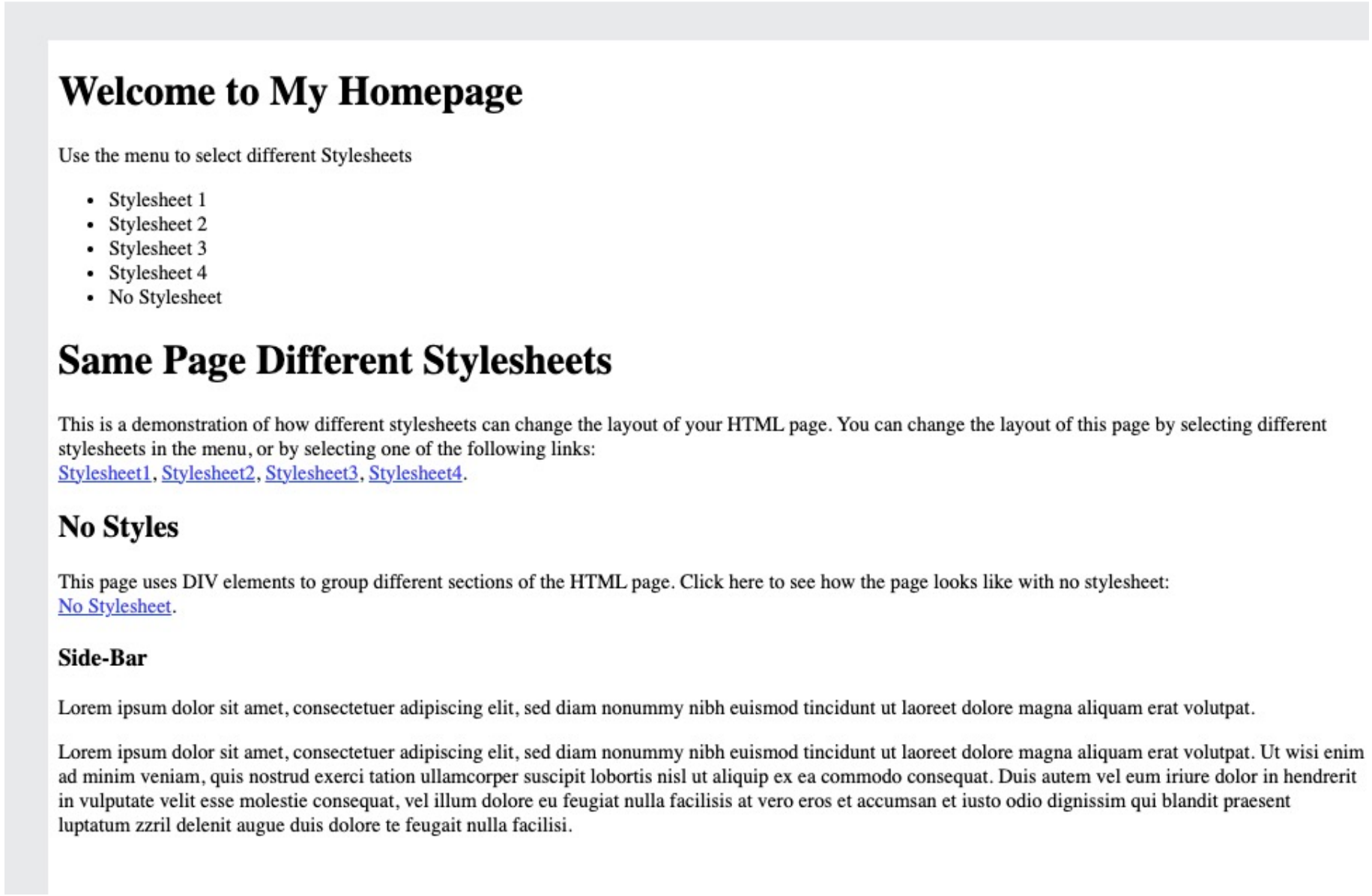


CSS Demo - One HTML Page - Multiple Styles!

- **Style 4**

CSS Demo - One HTML Page - Multiple Styles!

- **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 Example

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

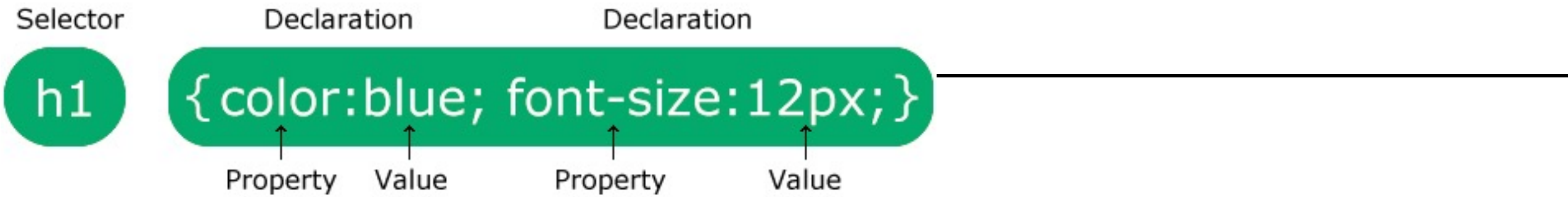
h1 {
  color: white;
  text-align: center;
}

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

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

</body>
</html>
```

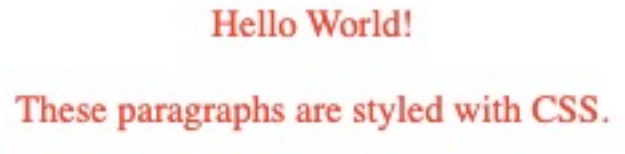
CSS Syntax


Selector
Declaration
Declaration
h1
{color:blue; font-size:12px;}
Property
Value
Property
Value

Example

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

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



CSS Selectors

- A CSS selector selects the HTML element(s) you want to style.
- CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

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":

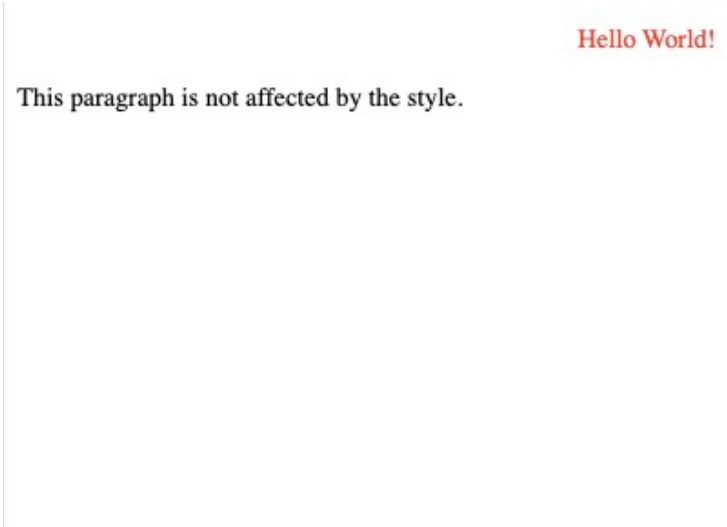
```
#para1 {
  text-align: center;
  color: red;
}
```

Example

```
<!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>
```



CSS class Selector

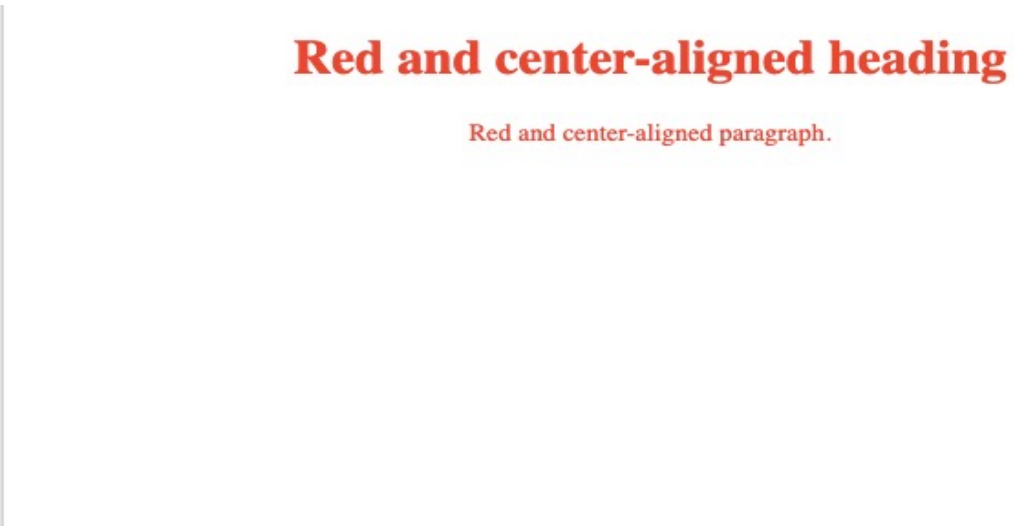
The class selector selects HTML elements with a specific class attribute.

- To select elements with a specific class, write a period (.) character, followed by the class name.
- Example
- In this example all HTML elements with class="center" will be red and center-aligned:
- ```
  .center {
    text-align: center;
    color: red;
  }
  ```

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
Red and center-aligned paragraph.

</body>
</html>
```


```

CSS class Selector

```
<!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>
```

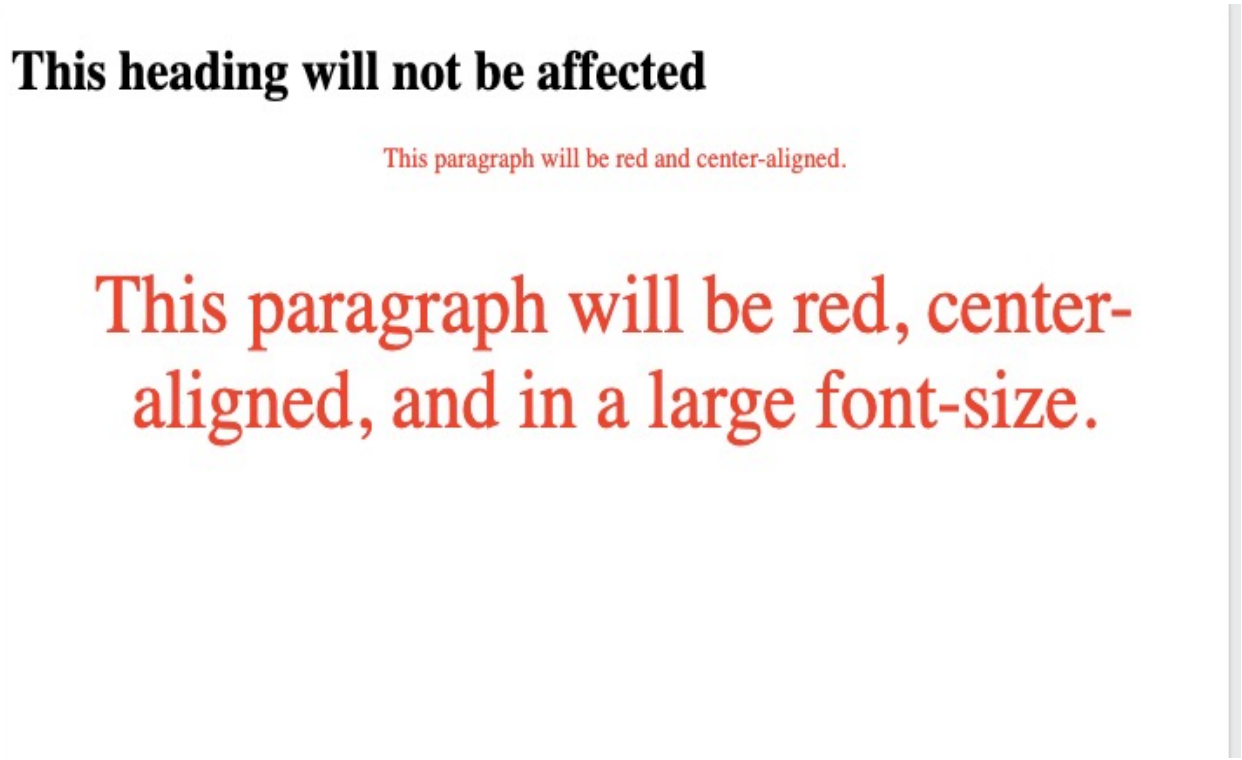
CSS class Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}

p.large {
  font-size: 300%;
}
</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

```
<!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

```
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.

```
<!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

How To Add CSS

- When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.

Three Ways to Insert CSS

- There are three ways of inserting a style sheet:
- External CSS
- Internal CSS
- Inline CSS

External CSS

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>

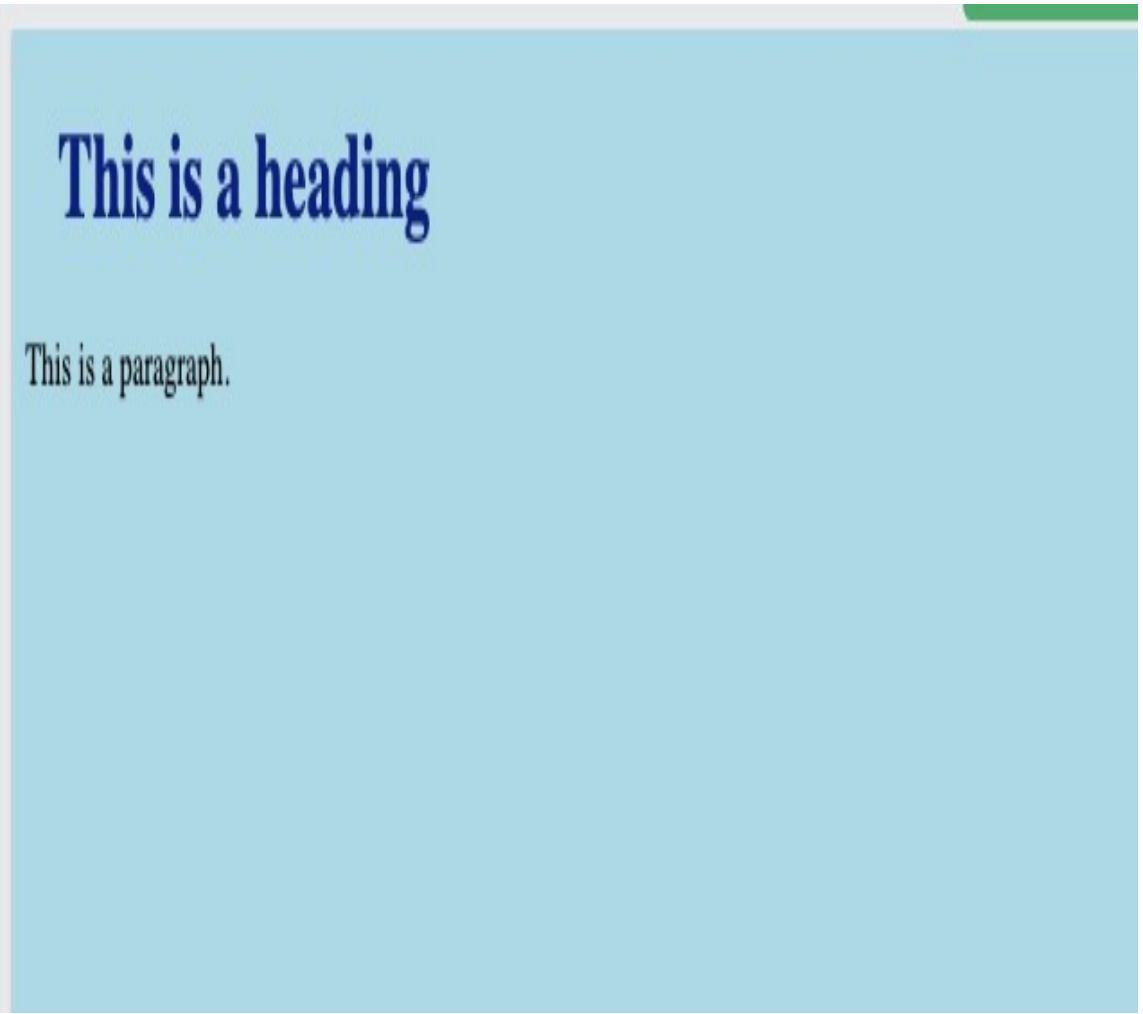
</body>
</html>
```

External CSS

```
<!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>
```



External CSS

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

- "mystyle.css"
- ```
  body {
    background-color: lightblue;
  }

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

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:

```
<!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

Example

- Inline styles are defined within the "style" attribute of the relevant element:
- ```
  <!DOCTYPE html>
  <html>
  <body>

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

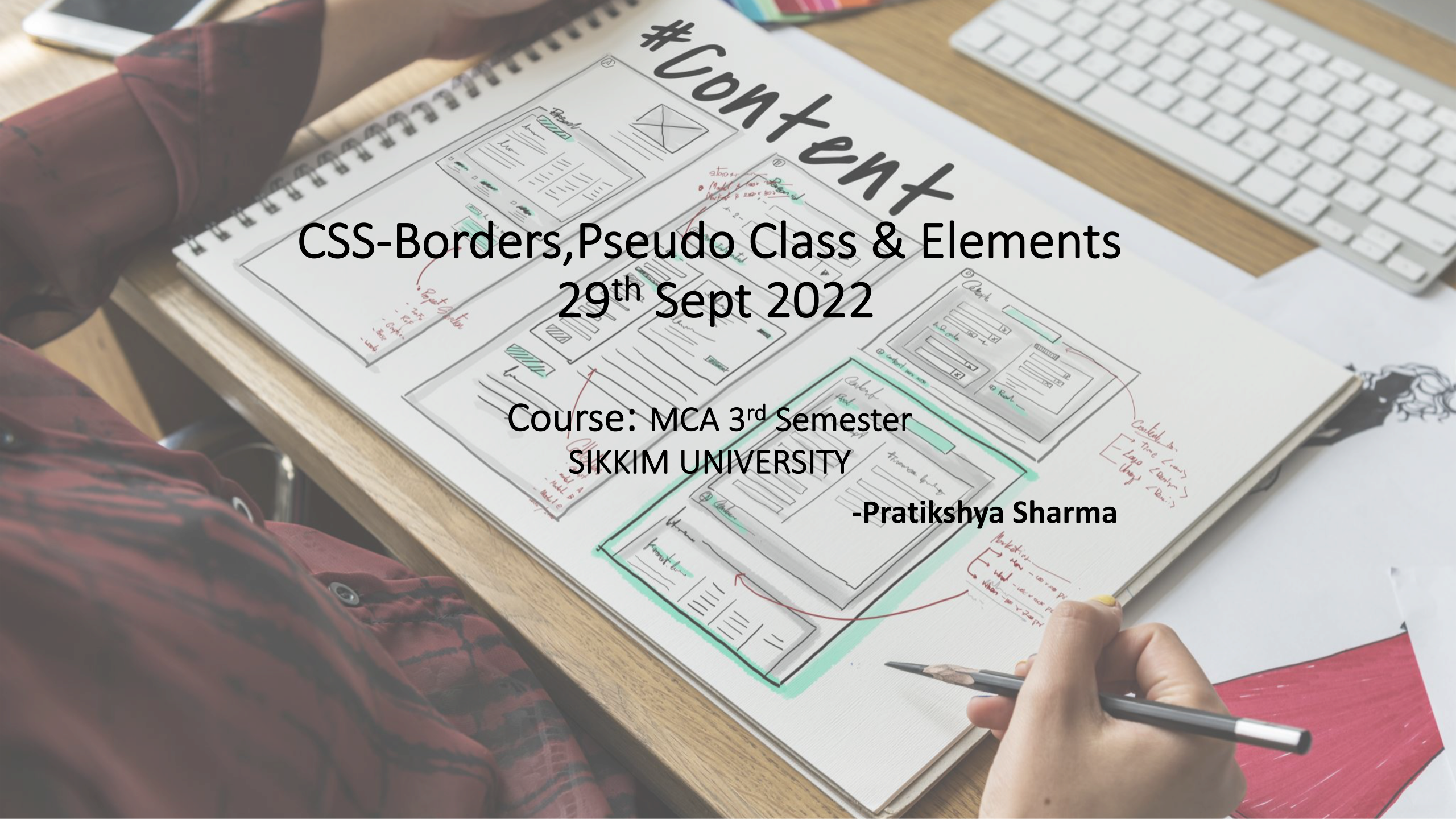
  </body>
  </html>
  ```
  ```

```html
<!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>
```



CSS-Borders,Pseudo Class & Elements

29th Sept 2022

Course: MCA 3rd Semester

SIKKIM UNIVERSITY

-Pratikshya Sharma

BORDER

- The *border* properties allow you to specify how the border of the box representing an element should look. There are three properties of a border you can change –
- The **border-color** specifies the color of a border.
- The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- The **border-width** specifies the width of a border.

The border-color Property

The border-color property allows you to change the color of the border surrounding an element. You can individually change the color of the bottom, left, top and right sides of an element's border using the properties –

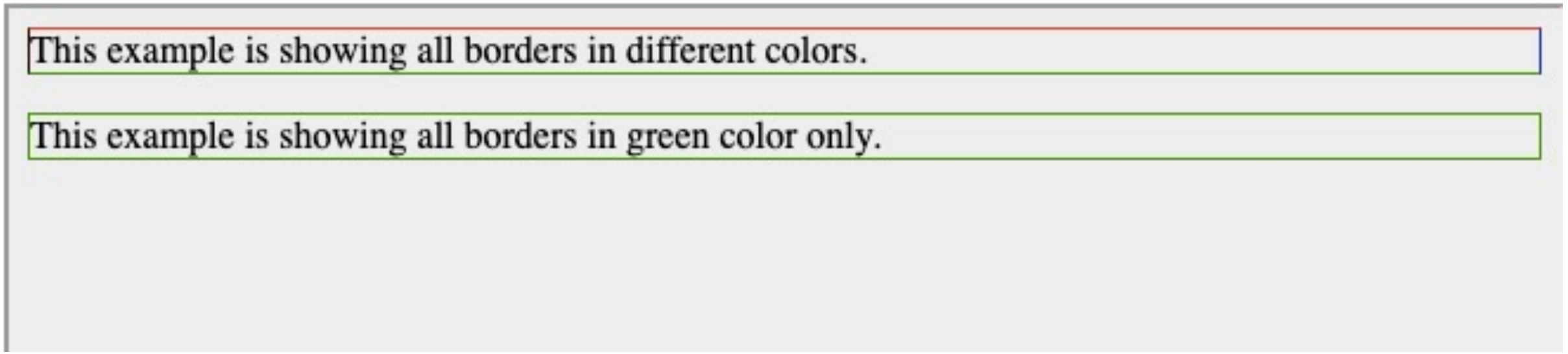
•**border-bottom-color** changes the color of bottom border.
•**border-top-color** changes the color of top border.
•**border-left-color** changes the color of left border.
•**border-right-color** changes the color of right border.

```
<html>
    <head>
        <style type = "text/css">
            p.example1 {
                border:1px solid;
                border-bottom-color:#009900; /* Green */
                border-top-color:#FF0000;    /* Red */
                border-left-color:#330000;   /* Black */
                border-right-color:#0000CC;  /* Blue */
            }
            p.example2 {
                border:1px solid;
                border-color:#009900;        /* Green */
            }
        </style>
    </head>

    <body>
        <p class = "example1">
            This example is showing all borders in different colors.
        </p>

        <p class = "example2">
            This example is showing all borders in green color only.
        </p>
    </body>
</html>
```

It will produce the following result –



The border-style Property

The border-style property allows you to select one of the following styles of border −

- **none** − No border. (Equivalent of border-width:0;)
- **solid** − Border is a single solid line.
- **dotted** − Border is a series of dots.
- **dashed** − Border is a series of short lines.
- **double** − Border is two solid lines.
- **groove** − Border looks as though it is carved into the page.
- **ridge** − Border looks the opposite of groove.
- **inset** − Border makes the box look like it is embedded in the page.
- **outset** − Border makes the box look like it is coming out of the canvas.
- **hidden** − Same as none, except in terms of border-conflict resolution for table elements.

You can individually change the style of the bottom, left, top, and right borders of an element using the following properties –

•**border-bottom-style** changes the style of bottom border.

•**border-top-style** changes the style of top border.

•**border-left-style** changes the style of left border.

•**border-right-style** changes the style of right border.

```
<html>
   <head>
   </head>

   <body>
      <p style = "border-width:4px; border-style:none;">
         This is a border with none width.
      </p>

      <p style = "border-width:4px; border-style:solid;">
         This is a solid border.
      </p>

      <p style = "border-width:4px; border-style:dashed;">
         This is a dashed border.
      </p>

      <p style = "border-width:4px; border-style:double;">
         This is a double border.
      </p>

      <p style = "border-width:4px; border-style:groove;">
         This is a groove border.
      </p>

      <p style = "border-width:4px; border-style:ridge">
         This is a ridge  border.
      </p>

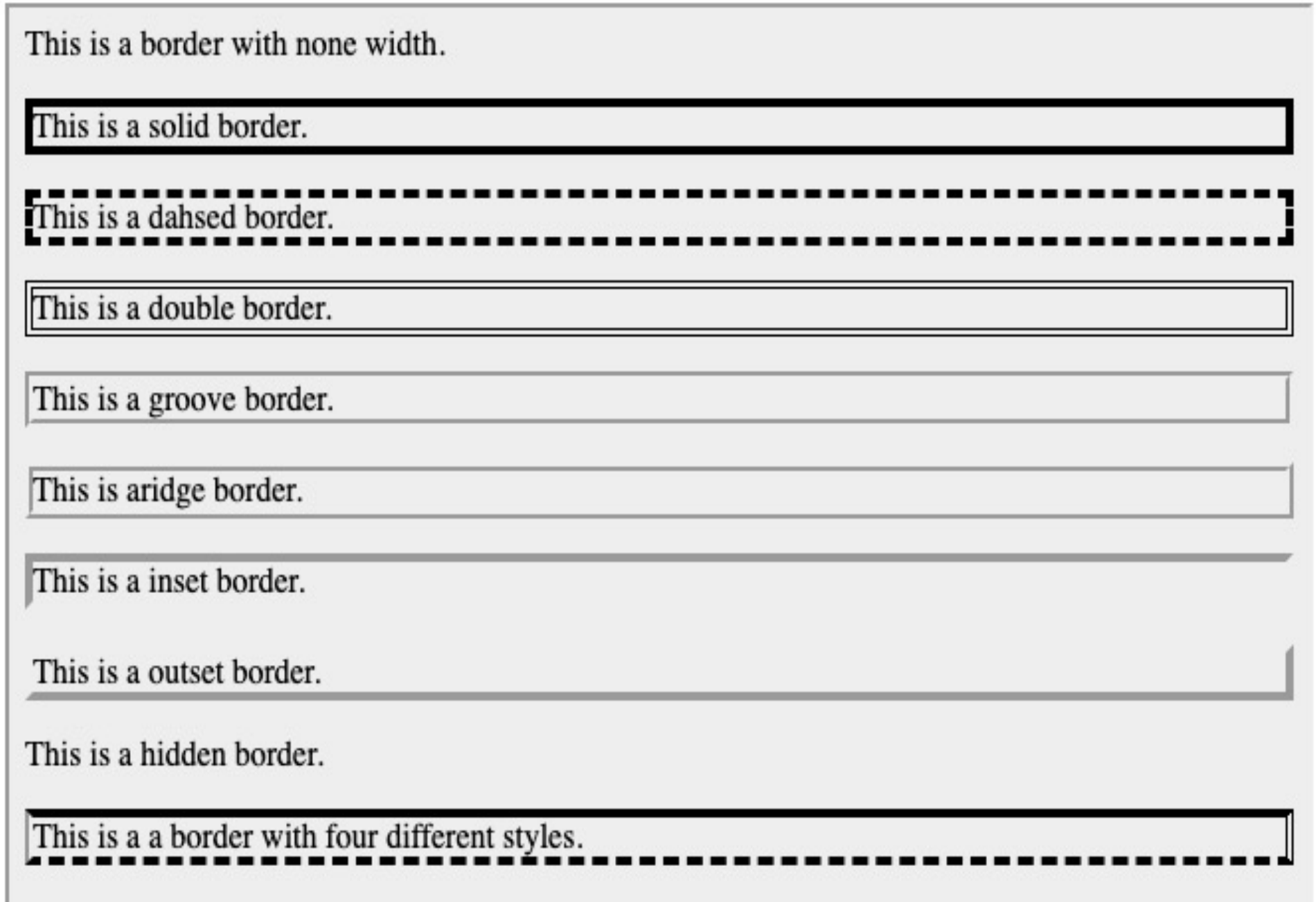
      <p style = "border-width:4px; border-style:inset;">
         This is a inset border.
      </p>

      <p style = "border-width:4px; border-style:outset;">
         This is a outset border.
      </p>

      <p style = "border-width:4px; border-style:hidden;">
         This is a hidden border.
      </p>

      <p style = "border-width:4px;
         border-top-style:solid;
         border-bottom-style:dashed;
         border-left-style:groove;
         border-right-style:double;">
         This is a a border with four different styles.
      </p>
   </body>
</html>
```

It will produce the following result –



The border-width Property

- The border-width property allows you to set the width of an element borders. The value of this property could be either a length in px, pt or cm or it should be set to *thin*, *medium or thick*.
- You can individually change the width of the bottom, top, left, and right borders of an element using the following properties –
- **border-bottom-width** changes the width of bottom border.
- **border-top-width** changes the width of top border.
- **border-left-width** changes the width of left border.
- **border-right-width** changes the width of right border.

```
<html>
   <head>
   </head>

   <body>
      <p style = "border-width:4px; border-style:solid;">
         This is a solid border whose width is 4px.
      </p>

      <p style = "border-width:4pt; border-style:solid;">
         This is a solid border whose width is 4pt.
      </p>

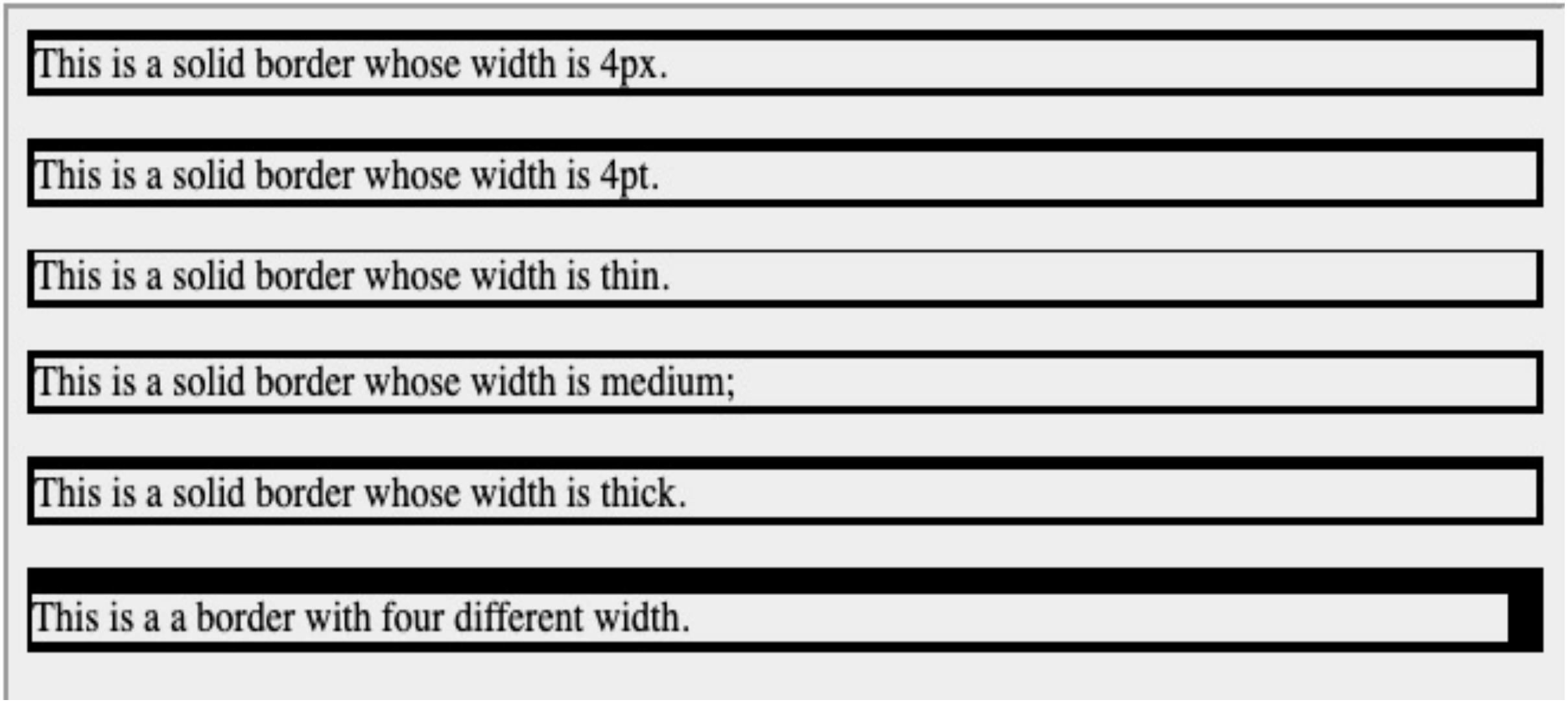
      <p style = "border-width:thin; border-style:solid;">
         This is a solid border whose width is thin.
      </p>

      <p style = "border-width:medium; border-style:solid;">
         This is a solid border whose width is medium;
      </p>

      <p style = "border-width:thick; border-style:solid;">
         This is a solid border whose width is thick.
      </p>

      <p style = "border-bottom-width:4px;border-top-width:10px;
         border-left-width: 2px;border-right-width:15px;border-style:solid
         This is a a border with four different width.
      </p>
   </body>
</html>
```

It will produce the following result –



CSS - Pseudo Classes

- CSS pseudo-classes are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-classes is as follows −

selector:pseudo-class {property: value}

CSS classes can also be used with pseudo-classes −

selector.class:pseudo-class {property: value}

The most commonly used pseudo-classes are as follows −

Sr.No.	Value & Description
1	**:link** Use this class to add special style to an unvisited link.
2	**:visited** Use this class to add special style to a visited link.
3	**:hover** Use this class to add special style to an element when you mouse over it.
4	**:active** Use this class to add special style to an active element.
5	**:focus** Use this class to add special style to an element while the element has focus.
6	**:first-child** Use this class to add special style to an element that is the first child of some other element.
7	**:lang** Use this class to specify a language to use in a specified element.

RULES

- While defining pseudo-classes in a <style>...</style> block, following points should be noted –
- a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.
- a:active MUST come after a:hover in the CSS definition in order to be effective.
- Pseudo-class names are not case-sensitive.
- Pseudo-class are different from CSS classes but they can be combined.

The :link pseudo-class

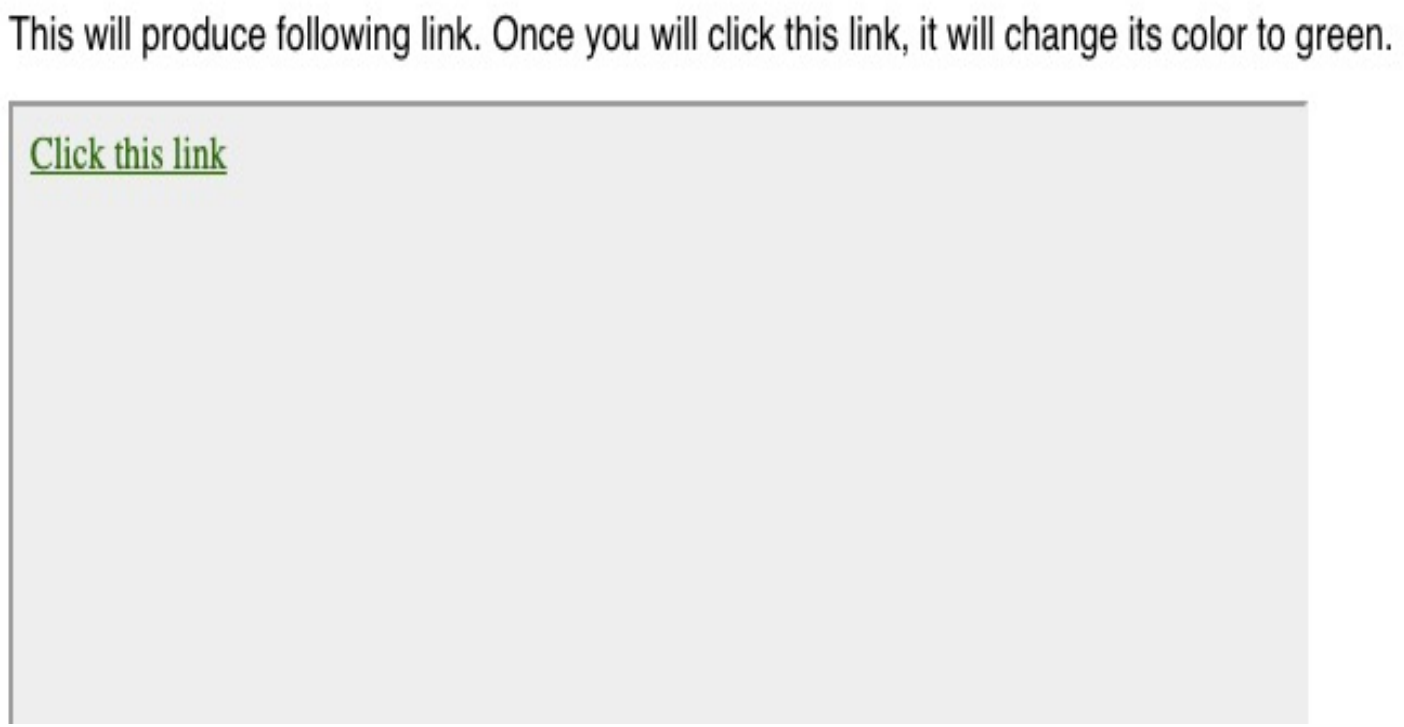
- The following example demonstrates how to use the *:link* class to set the link color. Possible values could be any color name in any valid format.

```
<html>
 <head>
        <style type = "text/css">
        a:link {color:#000000}
 </style>
</head>
<body>
        <a href = "">Black Link</a>
</body> </html>
```

The :visited pseudo-class

- The following is the example which demonstrates how to use the *:visited* class to set the color of visited links. Possible values could be any color name in any valid format.

```
<html>
<head>
<style type = "text/css">
        a:visited {color: #006600}
</style>
</head>
<body>
        <a href = "">Click this link</a>
</body>
</html>
```



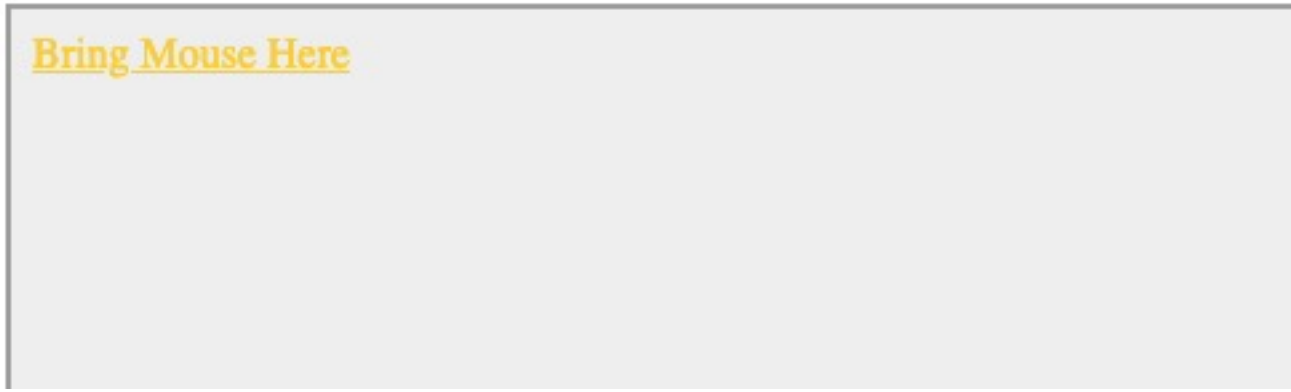
The :hover pseudo-class

- The following example demonstrates how to use the *:hover* class to change the color of links when we bring a mouse pointer over that link. Possible values could be any color name in any valid format.

```
<html>
        <head>
         <style type = "text/css">
         a:hover {color: #FFCC00}
        </style>
 </head>
 <body>
        <a href = "">Bring Mouse Here</a>
 </body>
 </html>
```

Before

After



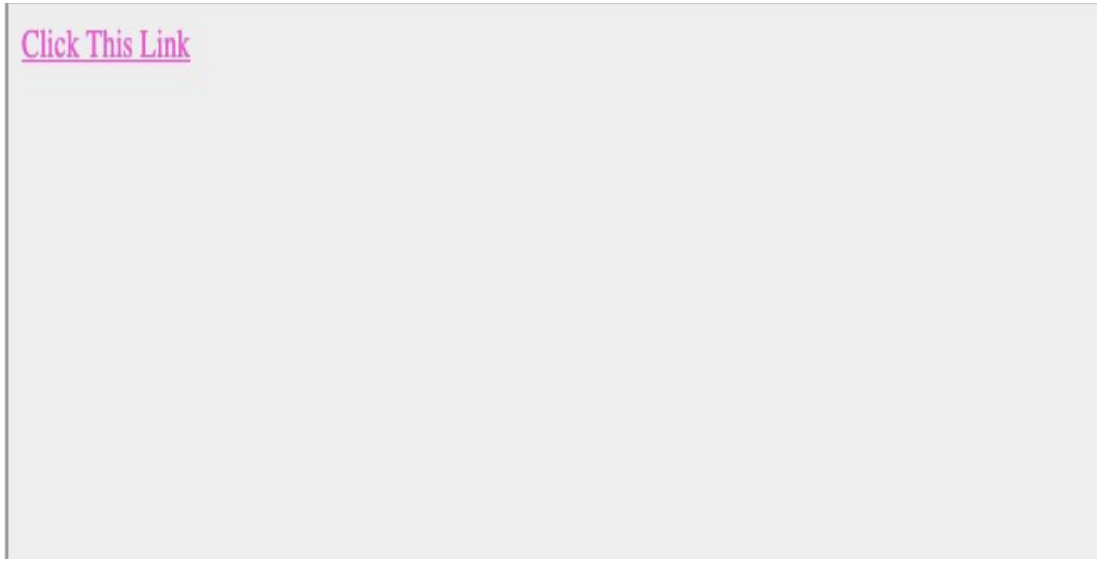
It will produce the following link. Now you bring your mouse over this link and you will see that it changes its color to yellow.

The :active pseudo-class

The following example demonstrates how to use the *:active* class to change the color of active links. Possible values could be any color name in any valid format.

```
<html>
        <head>
        <style type = "text/css">
                a:active {color: #FF00CC}
</style>
</head>
        <body> <a href = "">Click This Link</a>
 </body>
</html>
```

It will produce the following link. When a user clicks it, the color changes to pink.



The :lang pseudo-class

- The language pseudo-class *:lang*, allows constructing selectors based on the language setting for specific tags.
- This class is useful in documents that must appeal to multiple languages that have different conventions for certain language constructs. For example, the French language typically uses angle brackets (< and >) for quoting purposes, while the English language uses quote marks (' and ').
- In a document that needs to address this difference, you can use the :lang pseudo-class to change the quote marks appropriately.

The following code changes the <blockquote> tag appropriately for the language being used –

```
<html>
        <head>
         <style type = "text/css">
 /* Two levels of quotes for two languages*/
        :lang(en) { quotes: "'" "'" "'" "'"; }
        :lang(fr) { quotes: "<<" ">>" "<" ">"; }
</style>
 </head>
<body> <p>...<q lang = "fr">A quote in a paragraph</q>...</p>
</body>
</html>
```

The :lang selectors will apply to all the elements in the document. However, not all elements make use of the quotes property, so the effect will be transparent for most elements.

It will produce the following result –

...<<A quote in a paragraph>>...

CSS - Pseudo Elements

- CSS pseudo-elements are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-element is as follows −

selector:pseudo-element {property: value} CSS classes can also be used with pseudo-elements −

selector.class:pseudo-element {property: value}

- A CSS pseudo-element is used to style specified parts of an element.
- For example, it can be used to:
- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

Syntax

The syntax of pseudo-elements:

```
selector::pseudo-element {
  property: value;
}
```

The most commonly used pseudo-elements are as follows −

Sr.No.	Value & Description
1	**:first-line** Use this element to add special styles to the first line of the text in a selector.
2	**:first-letter** Use this element to add special style to the first letter of the text in a selector.
3	**:before** Use this element to insert some content before an element.
4	**:after** Use this element to insert some content after an element.

The :first-line pseudo-element

- The following example demonstrates how to use the *:first-line* element to add special effects to the first line of elements in the document.
- **Note:** The ::first-line pseudo-element can only be applied to block-level elements.
- The following properties apply to the ::first-line pseudo-element:
- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

```
<html>
   <head>
      <style type = "text/css">
         p:first-line { text-decoration: underline; }
         p.noline:first-line { text-decoration: none; }
      </style>
   </head>

   <body>
      <p class = "noline">
         This line would not have any underline because this belongs to nl
      </p>

      <p>
         The first line of this paragraph will be underlined as defined in
         CSS rule above. Rest of the lines in this paragraph will remain n
         This example shows how to use :first-line pseduo element to give
         to the first line of any HTML element.
      </p>
   </body>
</html>
```

It will produce the following link –

This line would not have any underline because this belongs to nline class.

The first line of this paragraph will be underlined as defined in the CSS rule above. Rest of the lines in this paragraph will remain normal. This example shows how to use :first-line pseduo element to give effect to the first line of any HTML element.

The :first-letter pseudo-element

The following example demonstrates how to use the *:first-letter* element to add special effects to the first letter of elements in the document.

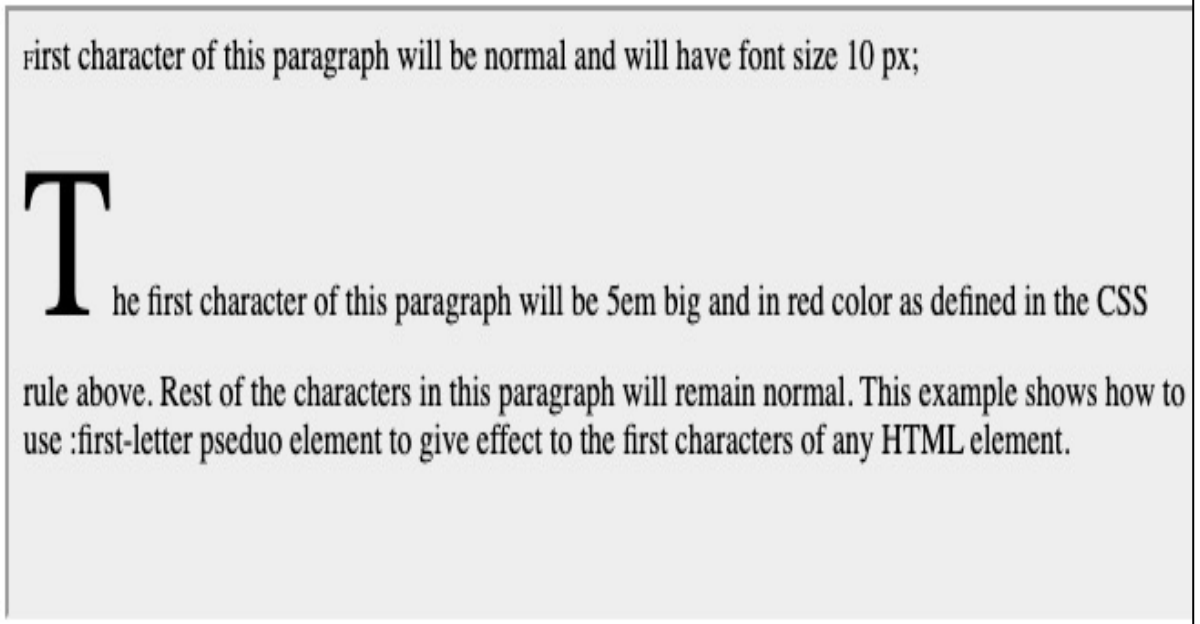
Live Demo

```
<html>
   <head>
      <style type = "text/css">
         p:first-letter { font-size: 5em; }
         p.normal:first-letter { font-size: 10px; }
      </style>
   </head>

   <body>
      <p class = "normal">
         First character of this paragraph will be normal and will have fo
      </p>

      <p>
         The first character of this paragraph will be 5em big as defined
         CSS rule above. Rest of the characters in this paragraph will rem
         normal. This example shows how to use :first-letter pseduo elemen
         to give effect to the first characters  of any HTML element.
      </p>
   </body>
</html>
```

It will produce the following black link –



Note:
The ::first-letter pseudo-element can only be applied to block-level elements.

The following properties apply to the ::first-letter pseudo- element:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

CSS - The ::before Pseudo-element

- The ::before pseudo-element can be used to insert some content before the content of an element.
- The following example inserts an image before the content of each <h1> element:

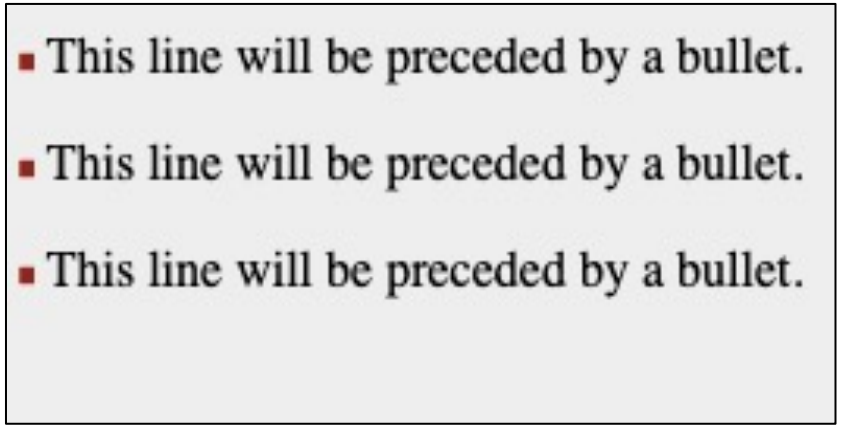
Example

```
h1::before {
  content: url(smiley.gif);
}
```

The :before pseudo-element

The following example demonstrates how to use the *:before* element to add some content before any element.

```
<html>
        <head>
        <style type = "text/css">
         p:before { content: url(/images/bullet.gif) }
</style>
 </head>
<body>
 <p> This line will be preceded by a bullet.</p>
 <p> This line will be preceded by a bullet.</p>
 <p> This line will be preceded by a bullet.</p>
</body>
 </html>
```



CSS - The ::after Pseudo-element

- The ::after pseudo-element can be used to insert some content after the content of an element.
- The following example inserts an image after the content of each <h1> element:

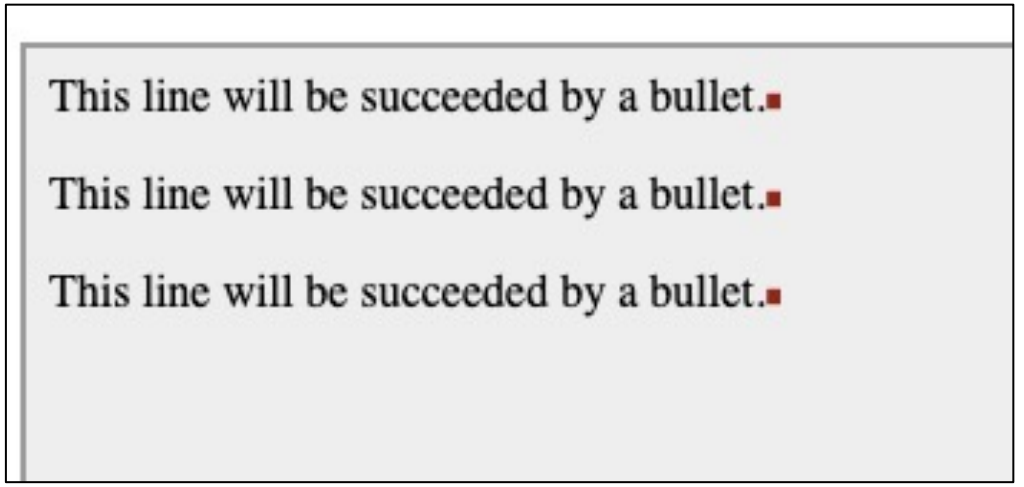
Example

```
h1::after {
  content: url(smiley.gif);
}
```

The :after pseudo-element

The following example demonstrates how to use the *:after* element to add some content after any element.

```
<html>
        <head>
         <style type = "text/css">
         p:after { content: url(/images/bullet.gif) }
        </style>
        </head>
 <body>
<p> This line will be succeeded by a bullet.</p>
<p> This line will be succeeded by a bullet.</p>
 <p> This line will be succeeded by a bullet.</p>
 </body>
 </html>
```



Universal Selector

- **CSS universal selectors** select any type of elements in an HTML page. It matches a single element. An asterisk (i.e. "*") is used to denote a **CSS universal selector**. An asterisk can also be followed by a selector. This is useful when you want to set a style for of all the elements of an HTML page or for all of the elements within an element of an HTML page.

Syntax:

- * { CSS-Property: value; ..... }

Example of CSS universal selectors

```
* { color: blue; /* color of all the elements should be blue */
    background: silver; /* silver background is set for all the elements */
}
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN”
"http://www.w3.org/TR/html4/strict.dtd">
 <html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<title>Simple Example of CSS universal selectors
 </title>
<link rel='stylesheet' href='simple-example-of-CSS-universal-selectors.css'
type='text/css’>
 </head>
 <body>
<h1>Example of CSS.</h1>
<h2>Example of grouping of CSS universal selectors.</h2>
 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a urna elit.
Integer malesuada tempus enim nec rhoncus.</p>
</body>
 </html>
```

Example of CSS.

Example of grouping of CSS universal selectors.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a urna elit. Integer malesuada tempus enim nec rhoncus.

CSS Combinators

- A combinator is something that explains the relationship between the selectors.
- A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.
- There are four different combinators in CSS:
- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

Descendant Selector

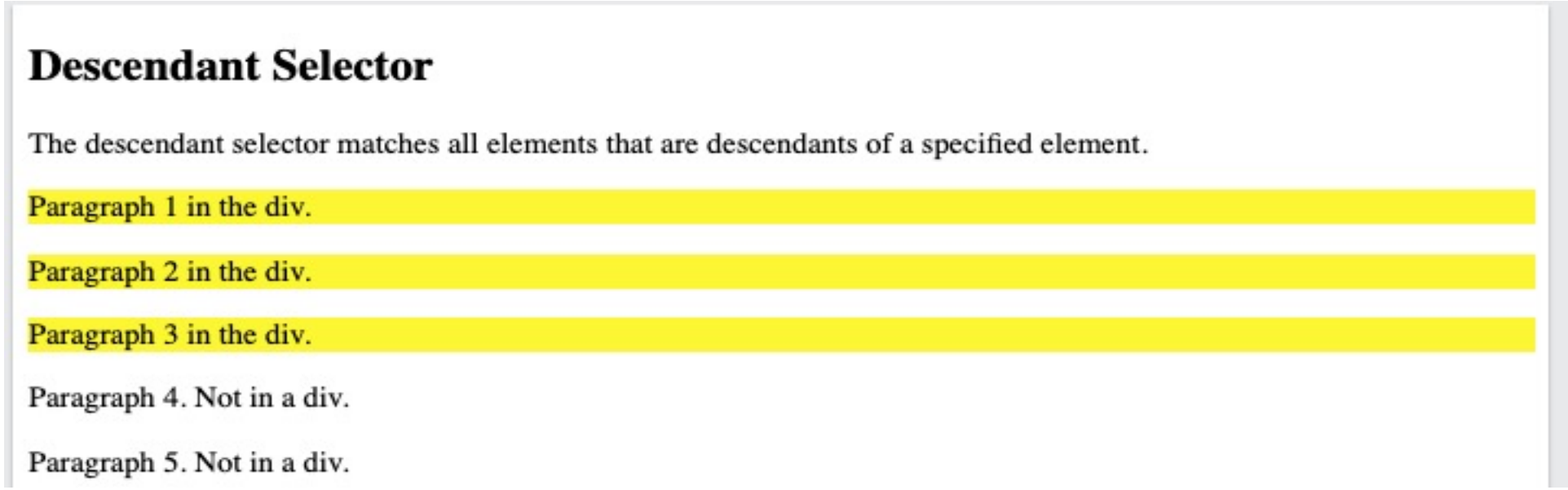
- The descendant selector matches all elements that are descendants of a specified element.
- The following example selects all <p> elements inside elements:

Example

```
div p {
  background-color: yellow;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>Descendant Selector</h2>
<p>The descendant selector matches all elements that are descendants of a specified element.</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>
<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>

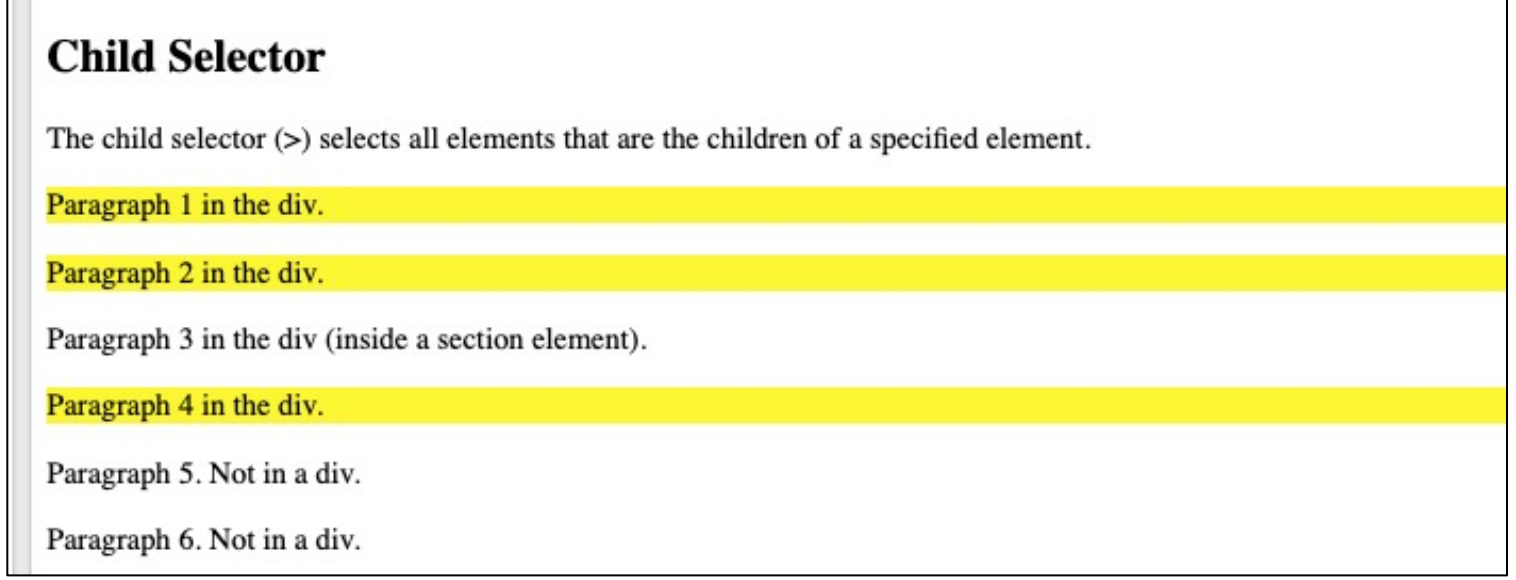
</body>
</html>
```



Child Selector (>)

- The child selector selects all elements that are the children of a specified element.
- The following example selects all <p> elements that are children of a element:
- Example
- ```
  div > p {
    background-color: yellow;
  }
  ```
  ```

```html
<!DOCTYPE html>
<html>
<head>
<style>
div > p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Child Selector</h2>

<p>The child selector (>) selects all elements that are the children of a specified
element.</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section>
    <!-- not Child but Descendant -->
    <p>Paragraph 3 in the div (inside a section element).</p>
  </section>
  <p>Paragraph 4 in the div.</p>
</div>

<p>Paragraph 5. Not in a div.</p>
<p>Paragraph 6. Not in a div.</p>

</body>
</html>
```



Adjacent Sibling Selector (+)

- The adjacent sibling selector is used to select an element that is directly after another specific element.
- Sibling elements must have the same parent element, and "adjacent" means "immediately following".
- The following example selects the first <p> element that are placed immediately after elements:

Example

```
div + p {
  background-color: yellow;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Adjacent Sibling Selector</h2>

<p>The + selector is used to select an element that is directly after another
specific element.</p>
<p>The following example selects the first p element that are placed immediately
after div elements:</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. After a div.</p>
<p>Paragraph 4. After a div.</p>

<div>
  <p>Paragraph 5 in the div.</p>
  <p>Paragraph 6 in the div.</p>
</div>

<p>Paragraph 7. After a div.</p>
<p>Paragraph 8. After a div.</p>

</body>
</html>
```

Adjacent Sibling Selector

The + selector is used to select an element that is directly after another specific element.

The following example selects the first p element that are placed immediately after div elements

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. After a div.

Paragraph 4. After a div.

Paragraph 5 in the div.

Paragraph 6 in the div.

Paragraph 7. After a div.

Paragraph 8. After a div.

General Sibling Selector (~)

- The general sibling selector selects all elements that are next siblings of a specified element.
- The following example selects all <p> elements that are next siblings of elements:

Example

```
div ~ p {
  background-color: yellow;
}
```

```html
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>General Sibling Selector</h2>

<p>The general sibling selector (~) selects all elements that are next siblings of a specified element.</p>

<p>Paragraph 1.</p>

<div>
  <p>Paragraph 2.</p>
</div>

<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>

</body>
</html>
```

General Sibling Selector

The general sibling selector (~) selects all elements that are next siblings of a specified element.

Paragraph 1.

Paragraph 2.

Paragraph 3.

`Some code.`

Paragraph 4.

CSS RGB Colors

- An RGB color value represents RED, GREEN, and BLUE light sources.
- RGB Value
- In CSS, a color can be specified as an RGB value, using this formula:

rgb(*red, green, blue*)

- Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.
- For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255) and the others are set to 0.
- To display black, set all color parameters to 0, like this: rgb(0, 0, 0).
- To display white, set all color parameters to 255, like this: rgb(255, 255, 255).
- Experiment by mixing the RGB values is presented in next slide.

Experiment by mixing the RGB values below:

rgb(255, 99, 71)

RED 255

GREEN 99

BLUE 71

Example

rgb(255, 0, 0)

rgb(0, 0, 255)

rgb(60, 179, 113)

rgb(238, 130, 238)

rgb(255, 165, 0)

rgb(106, 90, 205)

```
<!DOCTYPE html>
<html>
<body>
<h1>Specify colors using RGB values</h1>
<h2 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h2>
<h2 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255)</h2>
<h2 style="background-color:rgb(60, 179, 113);">rgb(60, 179, 113)</h2>
<h2 style="background-color:rgb(238, 130, 238);">rgb(238, 130, 238)</h2>
<h2 style="background-color:rgb(255, 165, 0);">rgb(255, 165, 0)</h2>
<h2 style="background-color:rgb(106, 90, 205);">rgb(106, 90, 205)</h2>


</body>
</html>
```

Specify colors using RGB values

rgb(255, 0, 0)

rgb(0, 0, 255)

rgb(60, 179, 113)

rgb(238, 130, 238)

rgb(255, 165, 0)

rgb(106, 90, 205)

CSS HEX Colors

- A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.
- HEX Value
- In CSS, a color can be specified using a hexadecimal value in the form:

  #rrggbb

- Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).
- For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).
- To display black, set all values to 00, like this: #000000.
- To display white, set all values to ff, like this: #ffffff.

Experiment by mixing the HEX values below:

#ff6347

RED

ff

GREEN

63

BLUE

47

Example

#ff0000

#0000ff

#3cb371

#ee82ee

#ffa500

#6a5acd

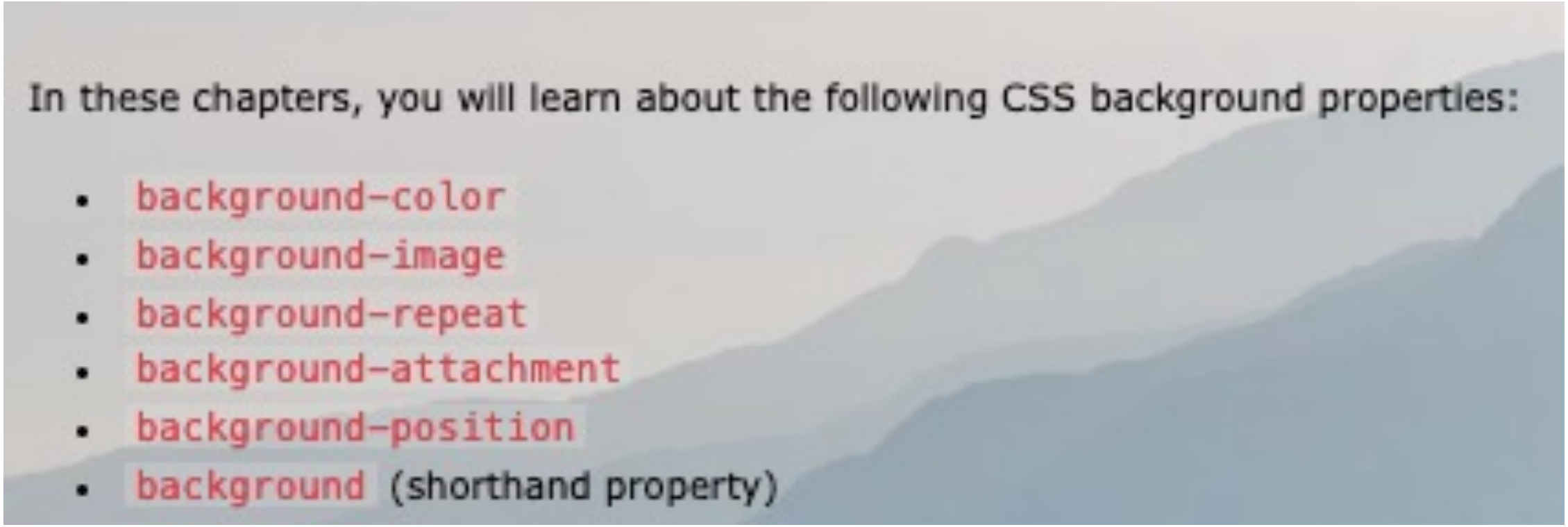
CSS
18th Oct 2022

Course: MCA 3rd Semester
SIKKIM UNIVERSITY

-Pratikshya Sharma

BACKGROUND

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



CSS background-color

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

With CSS, a color is most often specified by:

- a valid color name - like "red"

-a HEX value - like "#ff0000"

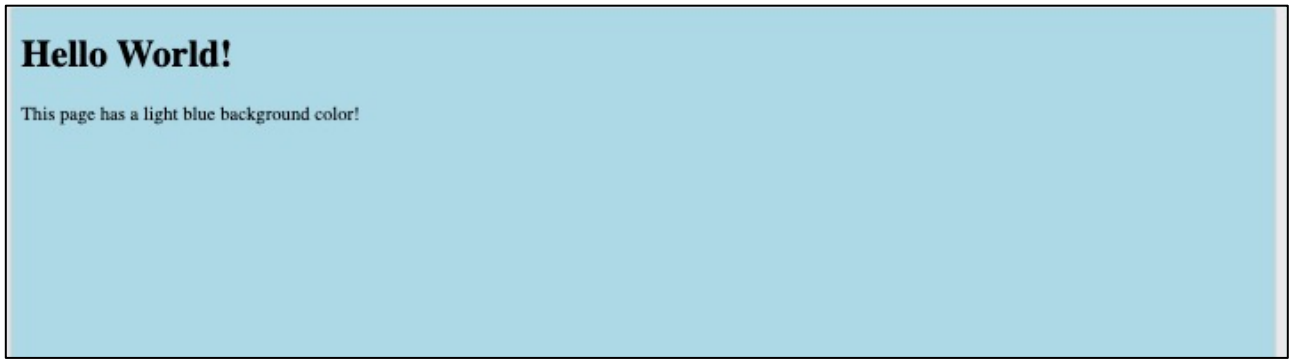
- an RGB value - like "rgb(255,0,0)"

CSS Colors

Colors in CSS can be specified by the following methods:

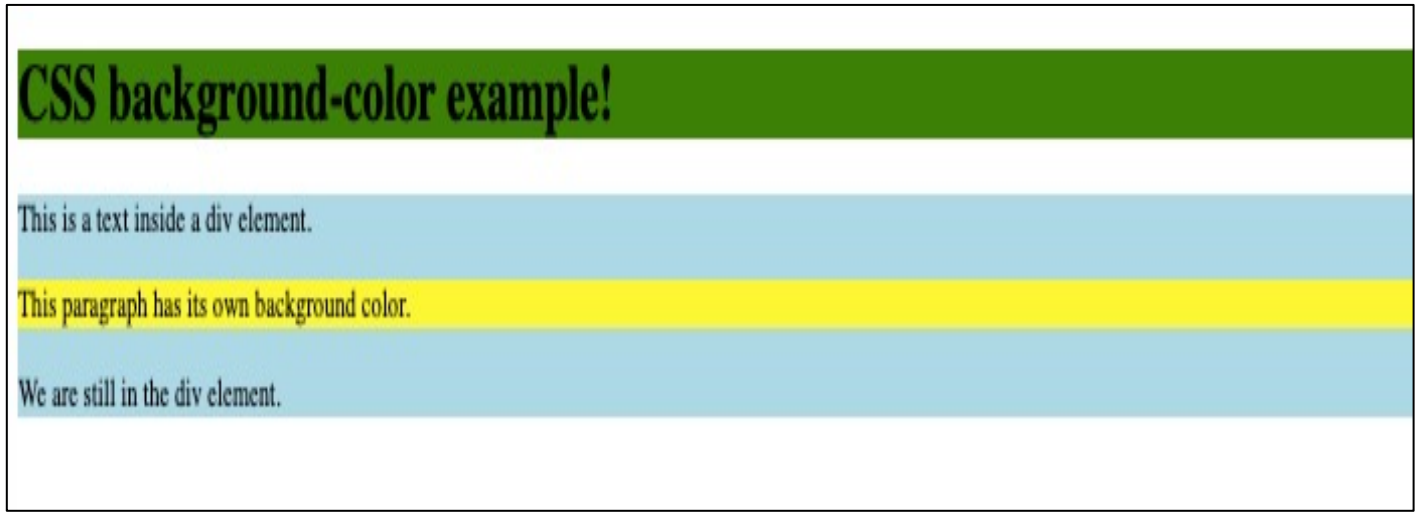
•Hexadecimal colors
•Hexadecimal colors with transparency
•RGB colors
•RGBA colors
•HSL colors
•HSLA colors
•Predefined/Cross-browser color names
•With the currentcolor keyword

```html
<!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

```
<!DOCTYPE html>
<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_colour_example

CSS background-image

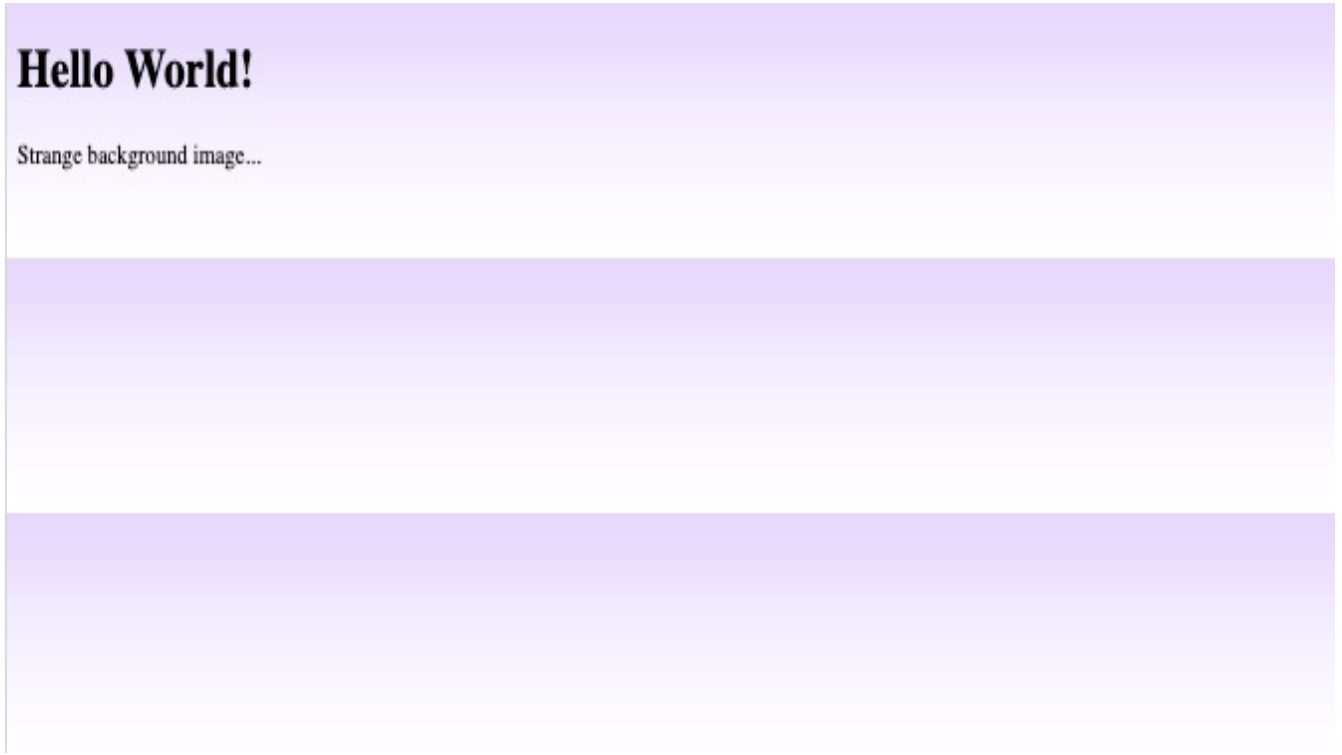
- The background-image property specifies an image to use as the background of an element.
- By default, the image is repeated so it covers the entire element.


```
<!DOCTYPE html>
<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>
```

css_background_image.html

CSS Background Image 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 as shown here.



```html
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("gradient_bg.png");
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Strange background image...</p>

</body>
</html>
```

background_image_repeat


```
<!DOCTYPE html>
<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>
```

backgound_image_horizontal_repeat.html

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

CSS Background Image Repeat

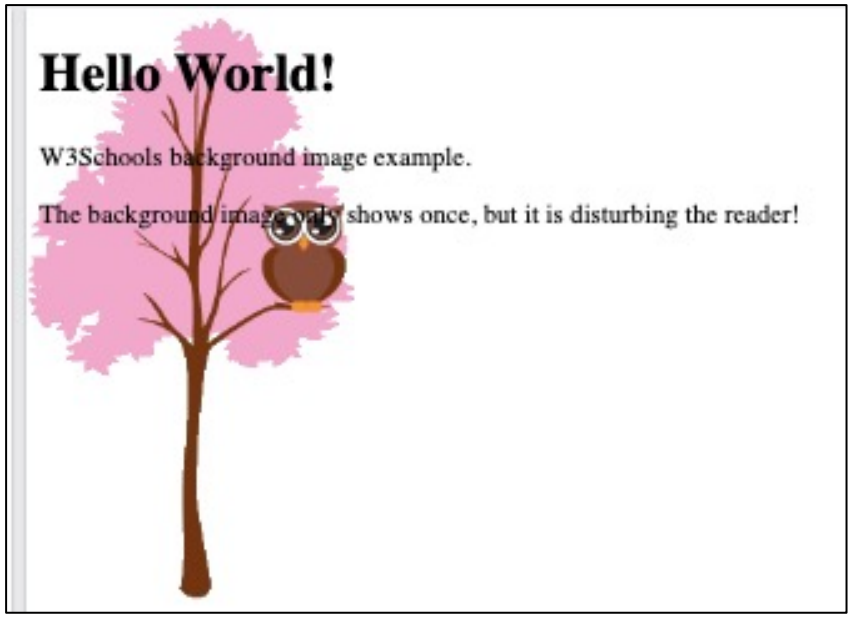
CSS Syntax `background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit;`

Value	Description
repeat	The background image is repeated both vertically and horizontally. The last image will be clipped if it does not fit. This is default
repeat-x	The background image is repeated only horizontally
repeat-y	The background image is repeated only vertically
no-repeat	The background-image is not repeated. The image will only be shown once
space	The background-image is repeated as much as possible without clipping. The first and last image is pinned to either side of the element, and whitespace is distributed evenly between the images
round	The background-image is repeated and squished or stretched to fill the space (no gaps)
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

CSS background-repeat: no-repeat

```
<!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>
```

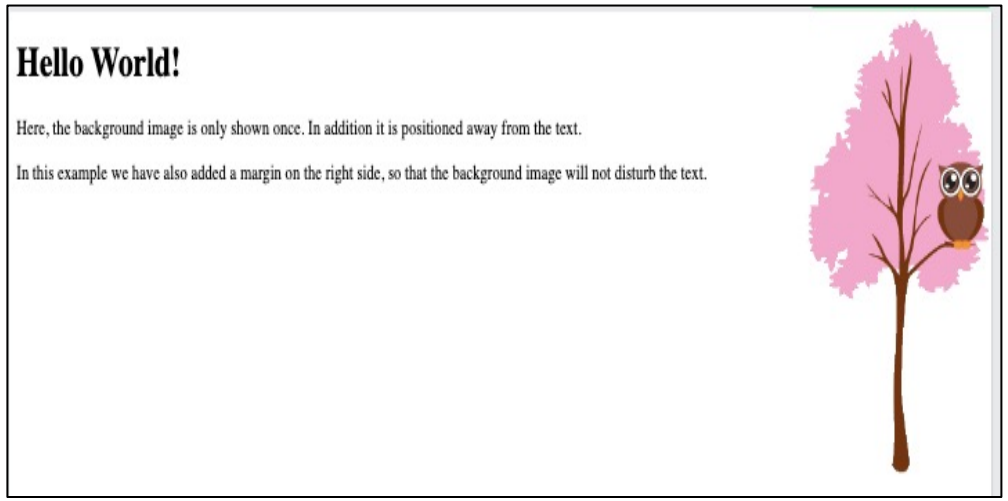
Showing the background image only once is also specified by the background-repeat property.



background_image_no_repeat

CSS background-position

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




```
<!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>

<h1>Hello World!</h1>
<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>
```

background_image_position.html

CSS background-attachment Property

- The background-attachment property sets whether a background image scrolls with the rest of the page, or is fixed.

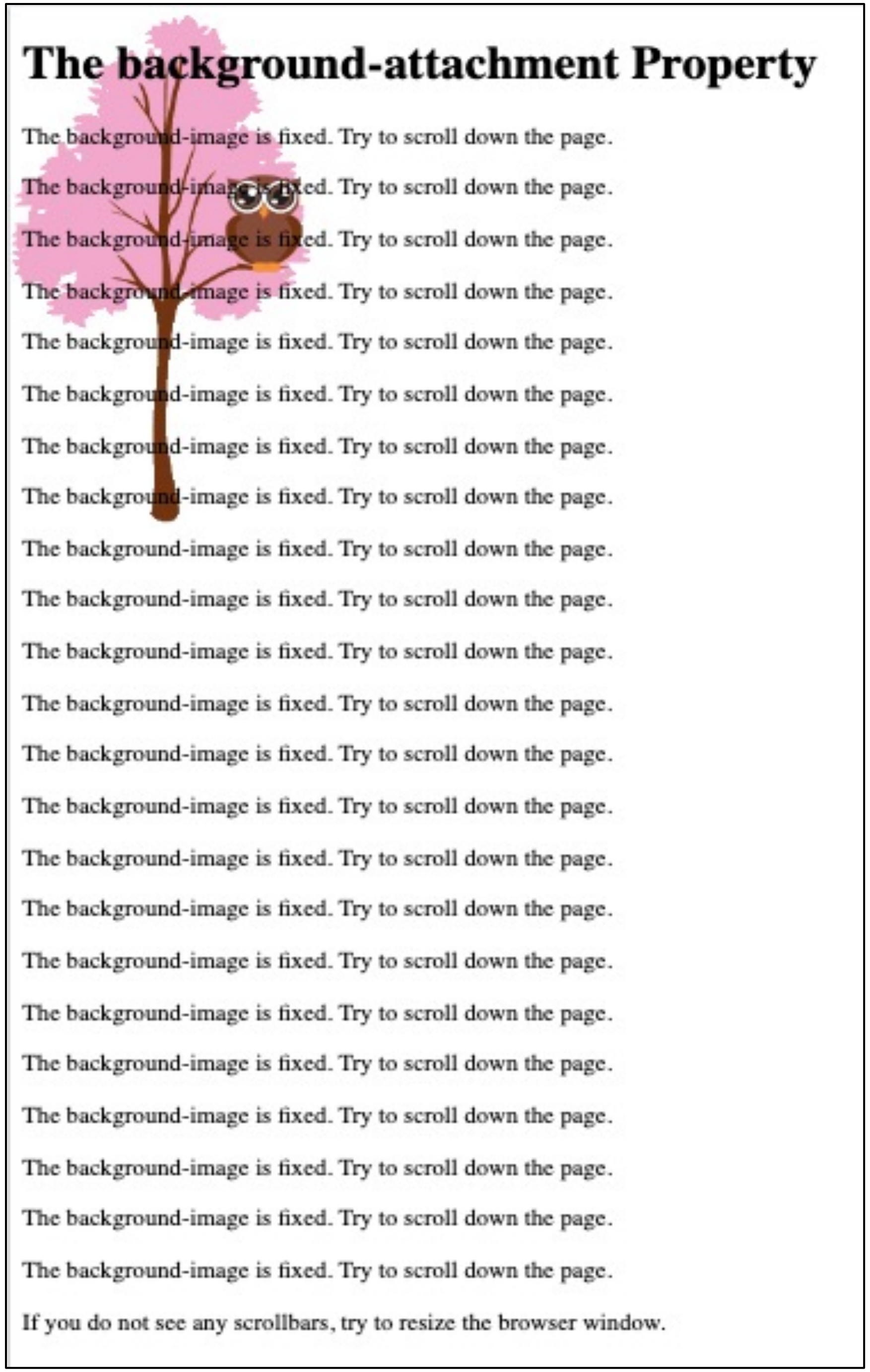
CSS Syntax

```
background-attachment: scroll|fixed|local|initial|inherit;
```

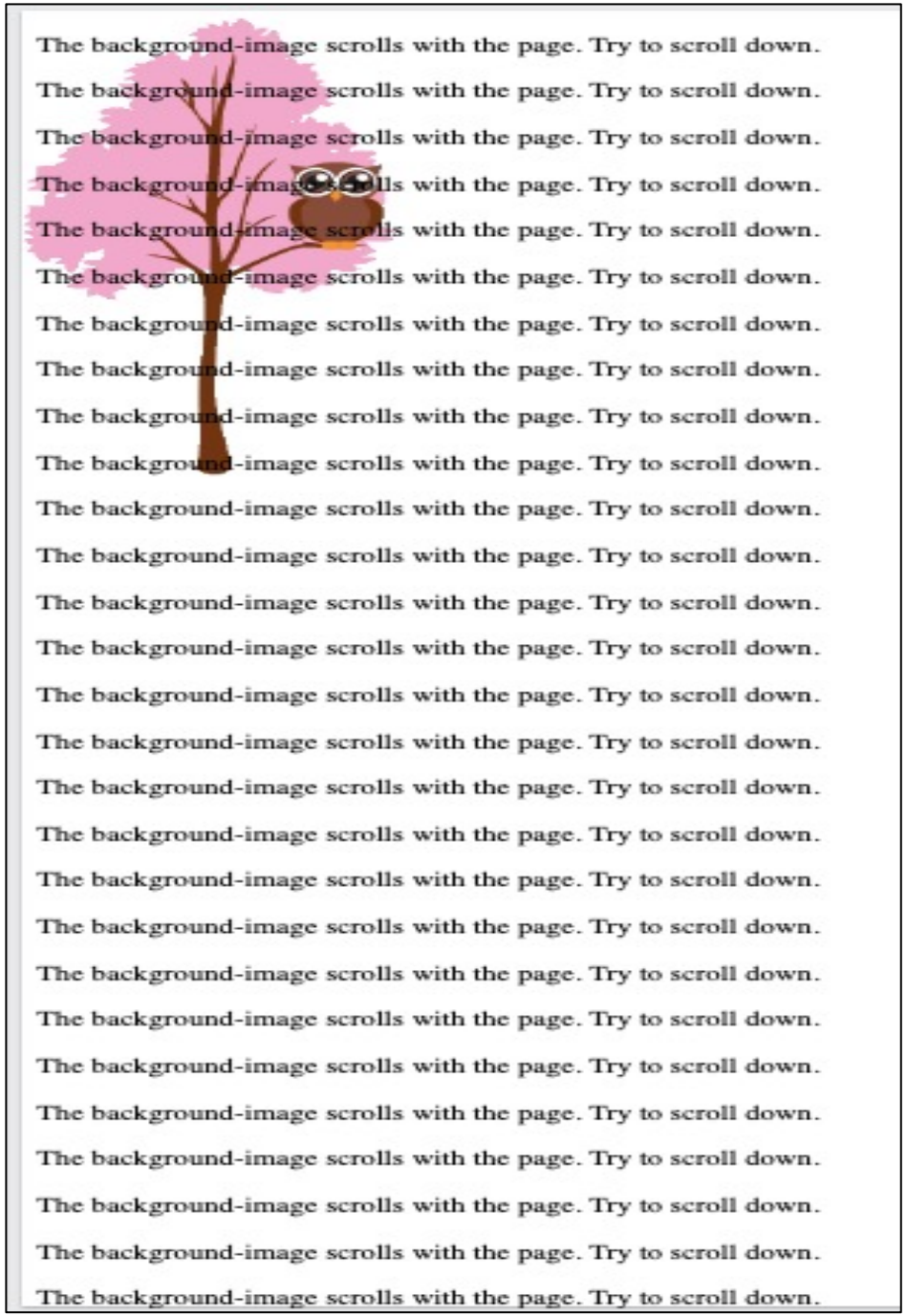
Property Values

Value	Description
scroll	The background image will scroll with the page. This is default
fixed	The background image will not scroll with the page
local	The background image will scroll with the element's contents
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

```html
<!DOCTYPE html>
<html>
<head>
<style>
body  {
  background-image: url("img_tree.gif");
  background-repeat: no-repeat;
  background-attachment: fixed;
}
</style>
</head>
<body>

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

<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>If you do not see any scrollbars, try to resize the browser window.</p>

</body>
</html>
```



background_image_fixed. html



```
<!DOCTYPE html>
<html>
<head>
<style>
body  {
  background-image: url("img_tree.gif");
  background-repeat: no-repeat;
  background-attachment: scroll;
}
</style>
</head>
<body>

<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>The background-image scrolls with the page. Try to scroll down.</p>
<p>If you do not see any scrollbars, try to resize the browser window.</p>

</body>
</html>
```

background_image_scroll.html

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

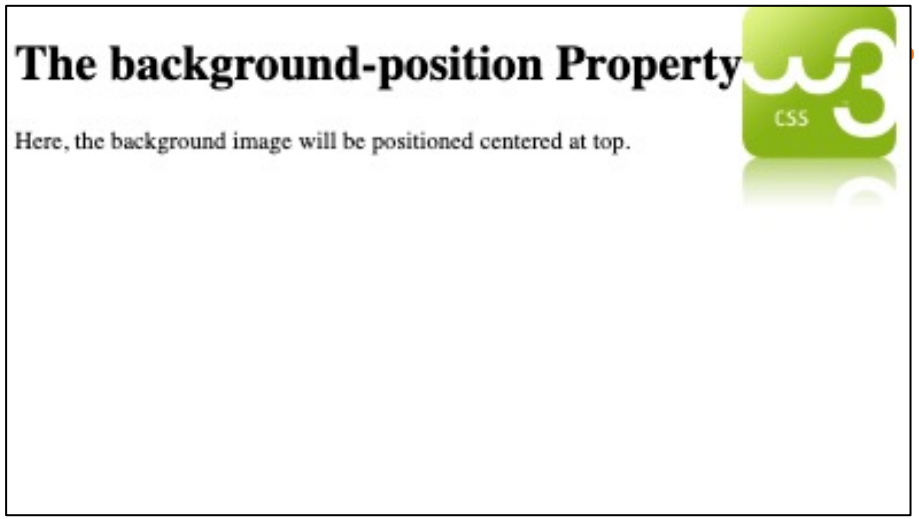
CSS background-position Property

- The background-position property sets the starting position of a background image.
- **Tip:** By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.
- CSS Syntax
- `background-position: value;`

CSS background-position Property

Value	Description
left top ,left center, left bottom,right top right center,right bottom center top center center center bottom	If you only specify one keyword, the other value will be "center"
x% y%	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. Default value is: 0% 0%
xpos ypos	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other CSS units. If you only specify one value, the other value will be 50%. You can mix % and positions
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

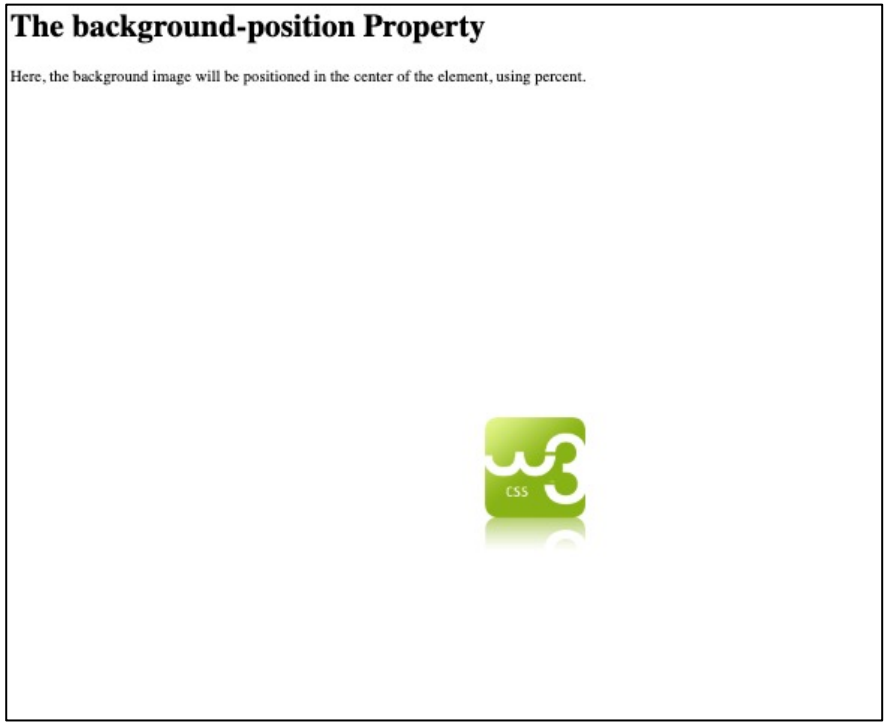
```html
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url('w3css.gif');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center top;
}
</style>
</head>
<body>

<h1>The background-position Property</h1>
<p>Here, the background image will be positioned centered at top.</p>

</body>
</html>
```



background_image_position_center_top.html

POSITIONING A BACKGROUND-IMAGE USING PERCENT

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url('w3css.gif');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: 50% 50%;
}
</style>
</head>
<body>

<h1>The background-position Property</h1>
<p>Here, the background image will be positioned in the center of the element, using percent.</p>

</body>
</html>
```

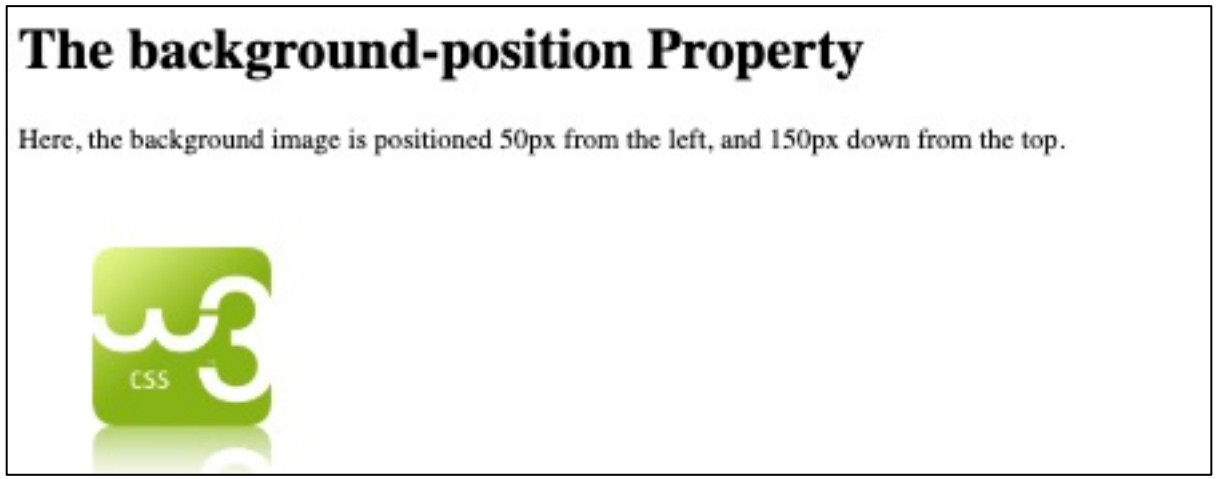


background_image_using_percentage.html

POSITIONING A BACKGROUND-IMAGE USING PIXELS

Example

```
body {
  background-image: url('w3css.gif');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: 50px 150px;
}
```



CSS background - Shorthand property

Example

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

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

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

- ✓ background-color
- ✓ background-image
- ✓ background-repeat
- ✓ background-attachment
- ✓ background-position

- It does not matter if one of the property values is missing, as long as the other ones are in this order. Note that we do not use the background-attachment property in the examples above, as it does not have a value.

CSS Units

- CSS has several different units for expressing a length.
- Many CSS properties take "length" values, such as width, margin, padding, font-size, etc.
- **Length** is a number followed by a length unit, such as 10px, 2em, etc.
- Example
- Set different length values, using px (pixels):

```
h1 {
  font-size: 60px;
}

p {
  font-size: 25px;
  line-height: 50px;
}
```

- **Note:** A whitespace cannot appear between the number and the unit. However, if the value is 0, the unit can be omitted.

- For some CSS properties, negative lengths are allowed.

- There are two types of length units: **absolute** and **relative**.

```html
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  font-size: 60px;
}

p {
  font-size: 25px;
  line-height: 50px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

This is heading 1

This is heading 2

This is a paragraph.

This is another paragraph.

Absolute Lengths

- The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.
- Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.
- * Pixels (px) are relative to the viewing device. For low-dpi devices, 1px is one device pixel (dot) of the display. For printers and high resolution screens 1px implies multiple device pixels.

Absolute Lengths

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

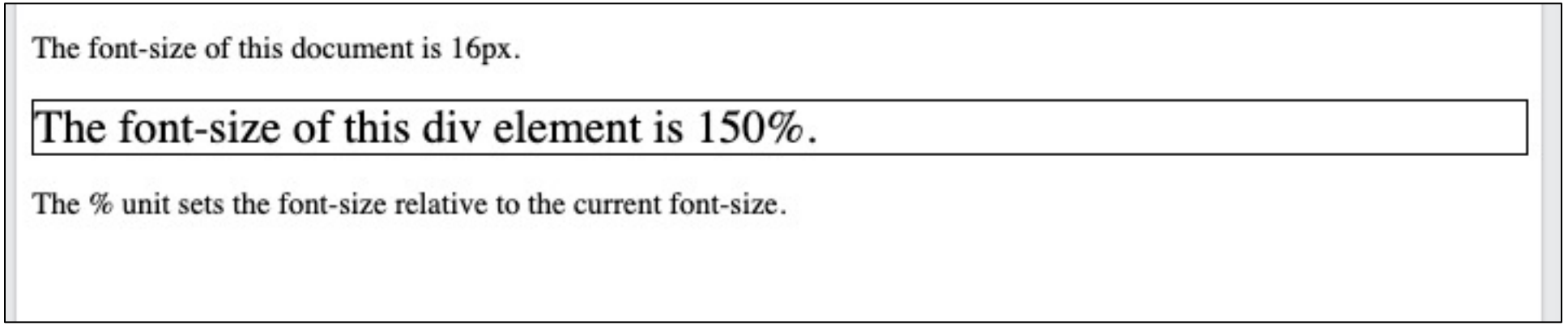
Note: Zero can be used with no unit: `border: 0;`

Relative Lengths

Relative length units specify a length relative to another length property. Relative length units scale better between different rendering medium.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
 font-size:16px;
}
div {
  font-size: 150%;
  border: 1px solid black;
}
```




```
</style>
</head>
<body>
<p>The font-size of this document is 16px.</p>

<div>The font-size of this div element is 150%.</div>

<p>The % unit sets the font-size relative to the current
font-size.</p>

</body>
</html>
```

CSS Margins

- 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

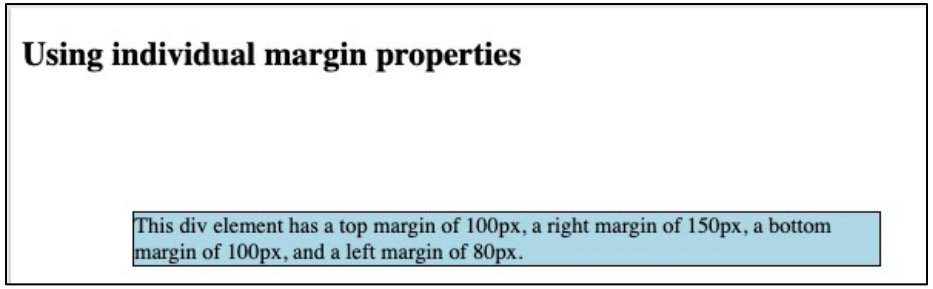
All the margin properties can have the following values:

- 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

```
<!DOCTYPE html>
<html>
<head>
<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>
</html>
```

(Margin1.html)



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

Example: If the margin property has four values:

margin: 25px 50px 75px 100px;

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

Margin - Shorthand Property

- 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

- If the margin property has two values:

margin: 25px 50px;

  - top and bottom margins are 25px
  - right and left margins are 50px

If the margin property has one value:

margin: 25px;

  - all four margins are 25px

All CSS Margin Properties

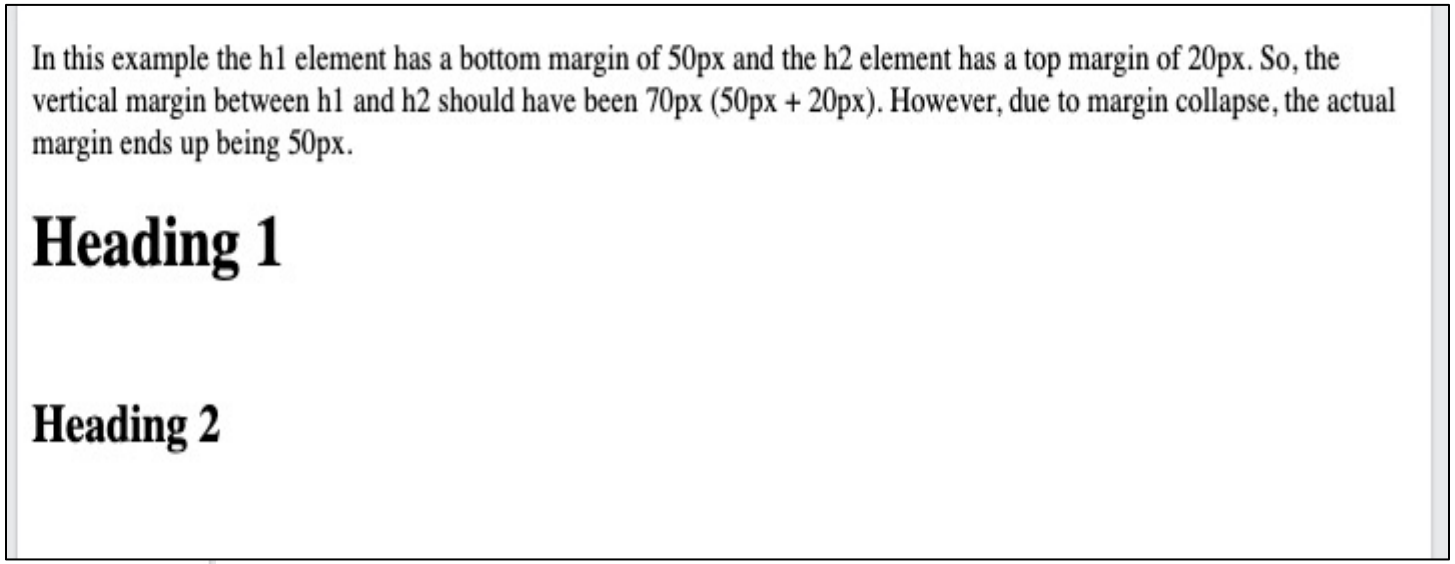
Property	Description
margin	A shorthand property for setting all the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

Margin Collapse

- Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.

- This does not happen on left and right margins! Only top and bottom margins!

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  margin: 0 0 50px 0;
}

h2 {
  margin: 20px 0 0 0;
}
</style>
</head>
<body>

<p>In this example the h1 element has a bottom margin of 50px and the h2 element has
a top margin of 20px. So, the vertical margin between h1 and h2 should have been 70px
(50px + 20px). However, due to margin collapse, the actual margin ends up being 50px.
</p>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

</body>
</html>
```



margin_collapse.txt

The Box Model

CSS Padding

- Padding is used to create space around an element's content, inside of any defined borders.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  padding: 70px;
  border: 1px solid #4CAF50;
}
</style>
</head>
<body>

<h2>CSS Padding</h2>
<div>This element has a padding of 70px.</div>

</body>
</html>
```

Padding1.txt

CSS Padding

- 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).

Note: Negative values are not allowed.

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:

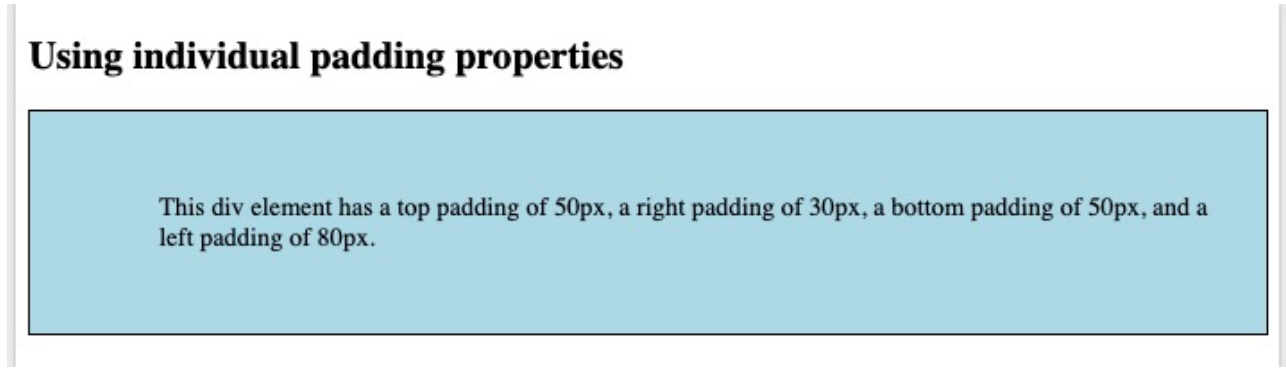
length - specifies a padding in px, pt, cm, etc.

% - specifies a padding in % of the width of the containing element

inherit - specifies that the padding should be inherited from the parent element

padding2.html

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
</style>
</head>
<body>

<h2>Using individual padding properties</h2>

<div>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>

</body>
</html>
```



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 - Shorthand Property

If the padding property has four values:

padding: 25px 50px 75px 100px;

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

If the padding property has three values:

padding: 25px 50px 75px;

- top padding is 25px
- right and left paddings are 50px
- bottom padding is 75px

If the padding property has two values:

padding: 25px 50px;

- top and bottom paddings are 25px
- right and left paddings are 50px

If the padding property has one value:

padding: 25px;

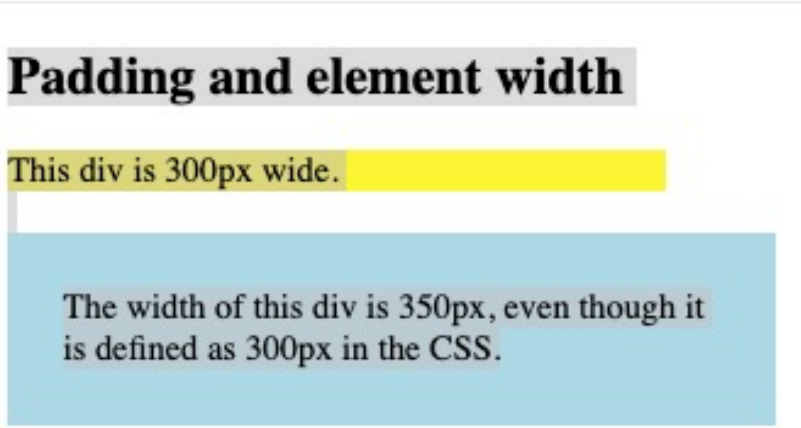
- all four paddings are 25px

Padding and Element Width

- The CSS width property specifies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element (the box model).

- So, if an element has a specified width, the padding added to that element will be added to the total width of the element. This is often an undesirable result.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
  width: 300px;
  background-color: yellow;
}

div.ex2 {
  width: 300px;
  padding: 25px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>Padding and element width</h2>

<div class="ex1">This div is 300px wide.</div>
<br>

<div class="ex2">The width of this div is 350px, even though it is defined as 300px
in the CSS.</div>

</body>
</html>
```

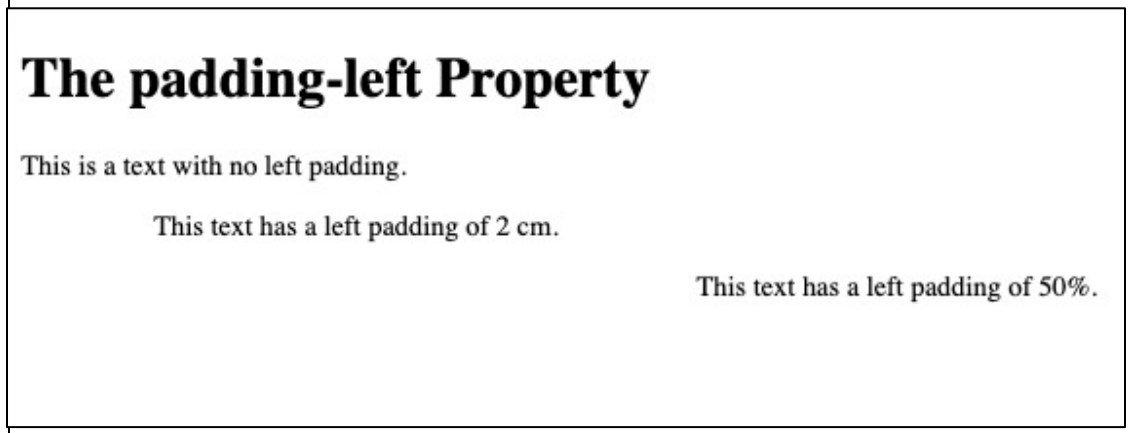


Padding_width1.html

Here, the element is given a width of 300px. However, the actual width of the element will be 350px (300px + 25px of left padding + 25px of right padding):

Left Padding Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.padding {
  padding-left: 2cm;
}
p.padding2 {
  padding-left: 50%;
}
</style>
</head>
<body>

<h1>The padding-left Property</h1>

<p>This is a text with no left padding.</p>
<p class="padding">This text has a left padding of 2 cm.</p>
<p class="padding2">This text has a left padding of 50%.</p>

</body>
</html>
```



Paddingz-left.html

All CSS Padding Properties

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

CSS Height, Width and Max-width

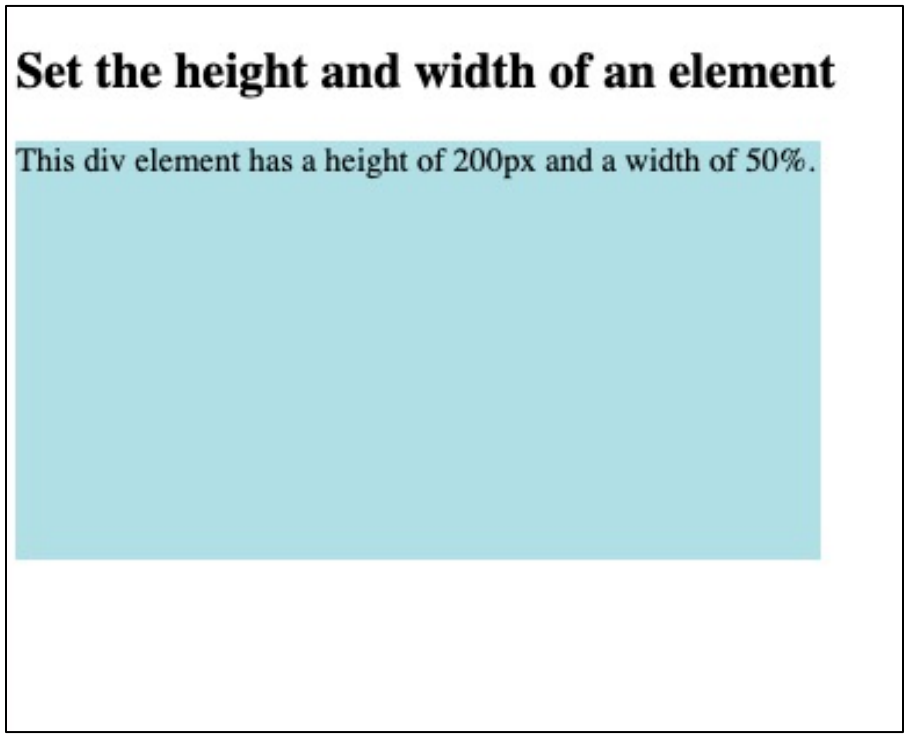
- The CSS height and width properties are used to set the height and width of an element.
- The CSS max-width property is used to set the maximum width of an element.

CSS height and width Values

The height and width properties may have the following values:

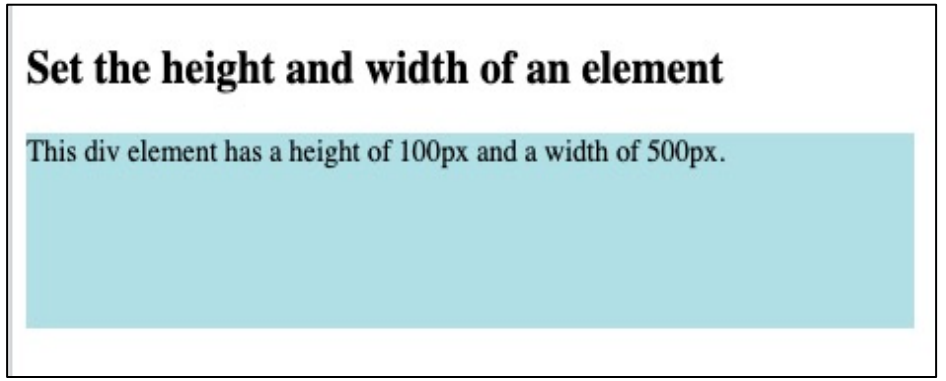
auto - This is default. The browser calculates the height and width
length - Defines the height/width in px, cm, etc.
% - Defines the height/width in percent of the containing block
initial - Sets the height/width to its default value
inherit - The height/width will be inherited from its parent value

```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2>Set the height and width of an element</h2>

<div>This div element has a height of 200px and a width of 50%.</div>

</body>
</html>
```



Height_width1.html

```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
  height: 100px;
  width: 500px;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2>Set the height and width of an element</h2>

<div>This div element has a height of 100px and a width of 500px.</div>

</body>
</html>
```



Height_width2.html

Note:
Remember that the height and width properties do not include padding, borders, or margins! They set the height/width of the area inside the padding, border, and margin of the element!

Setting max-width

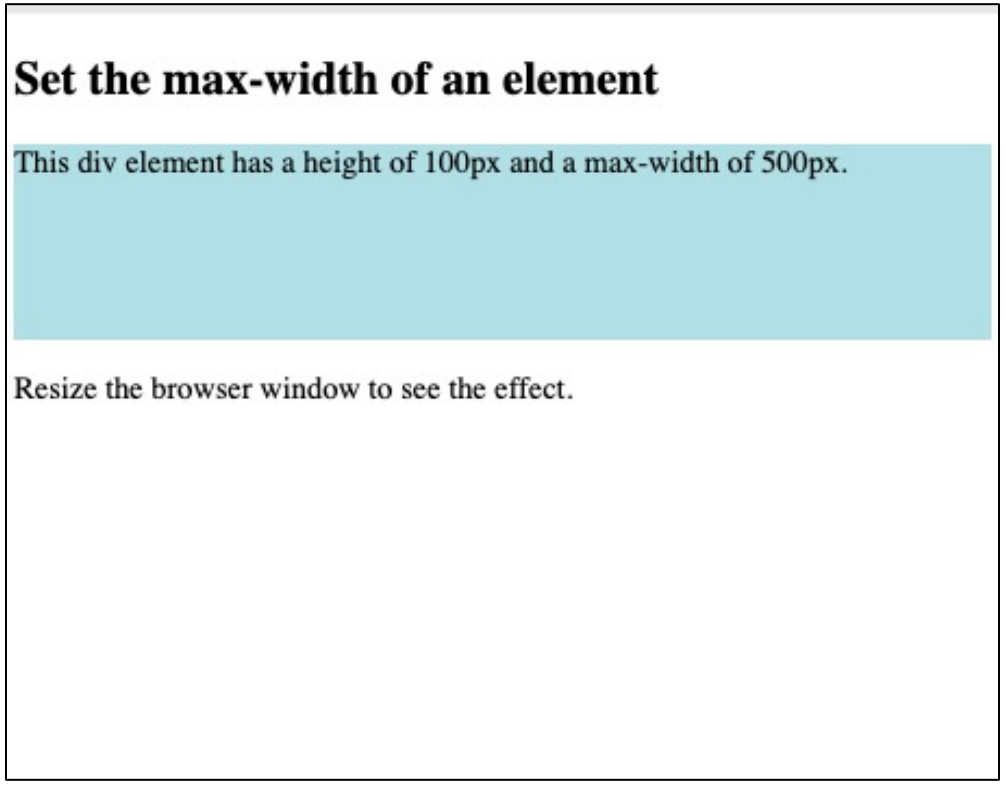
- The max-width property is used to set the maximum width of an element.
- The max-width can be specified in *length values*, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).
- The problem with the above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.
- Using max-width instead, in this situation,

will improve the browser's handling of small windows.

Tip: Drag the browser window to smaller than 500px wide, to see the difference between the two divs!

Note: If you for some reason use both the width property and the max-width property on the same element, and the value of the width property is larger than the max-width property; the max-width property will be used (and the width property will be ignored).

```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
  max-width: 500px;
  height: 100px;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2>Set the max-width of an element</h2>

<div>This div element has a height of 100px and a max-width of 500px.</div>

<p>Resize the browser window to see the effect.</p>

</body>
</html>
```



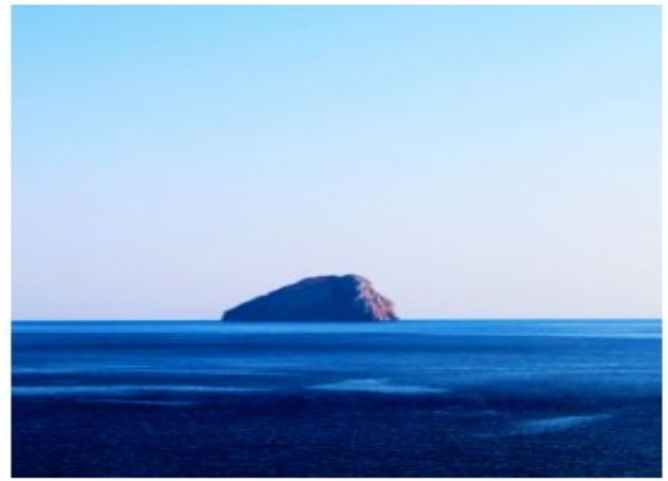
max_width.html

height_width_different_elements.html

```
<!DOCTYPE html>
<html>
<head>
<style>
img.one {
  height: auto;
}

img.two {
  height: 200px;
  width: 200px;
}

div.three {
  height: 300px;
  width: 300px;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2>Set the height and width of elements</h2>

<p>Original image:</p>
<img class="one" src="ocean.jpg" width="300" height="300"><br>

<p>Sized image (200x200 pixels):</p>
<img class="two" src="ocean.jpg" width="300" height="300"><br>

<p>The height and width of this div element is 300px:</p>
<div class="three"></div>

</body>
</html>
```

Set the height and width of elements

Original image:

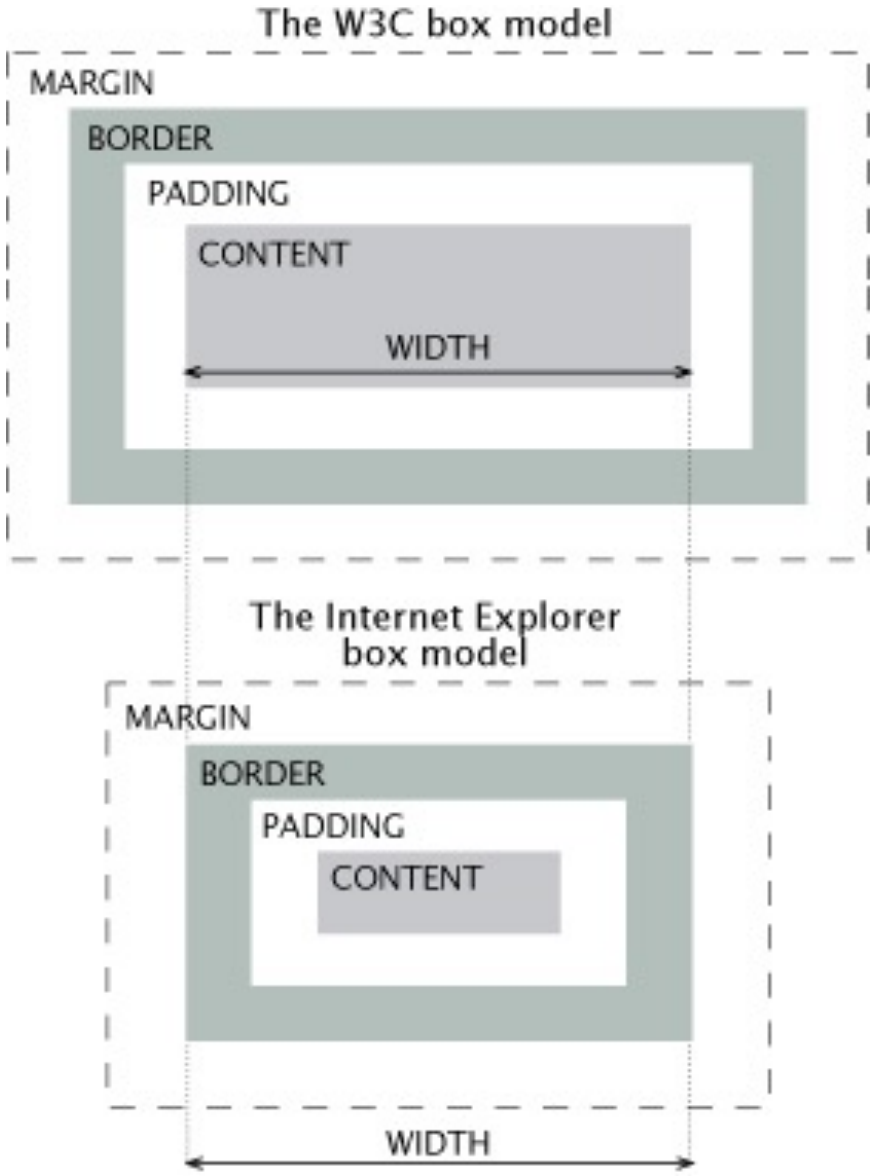


Sized image (200x200 pixels):

The height and width of this div element is 300px:

IE Quirks Mode

- When using quirks mode (pages with no DOCTYPE or with a HTML 4 Transitional DOCTYPE), Internet Explorer violates the box model standard



CSS Layout - The position Property

The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

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

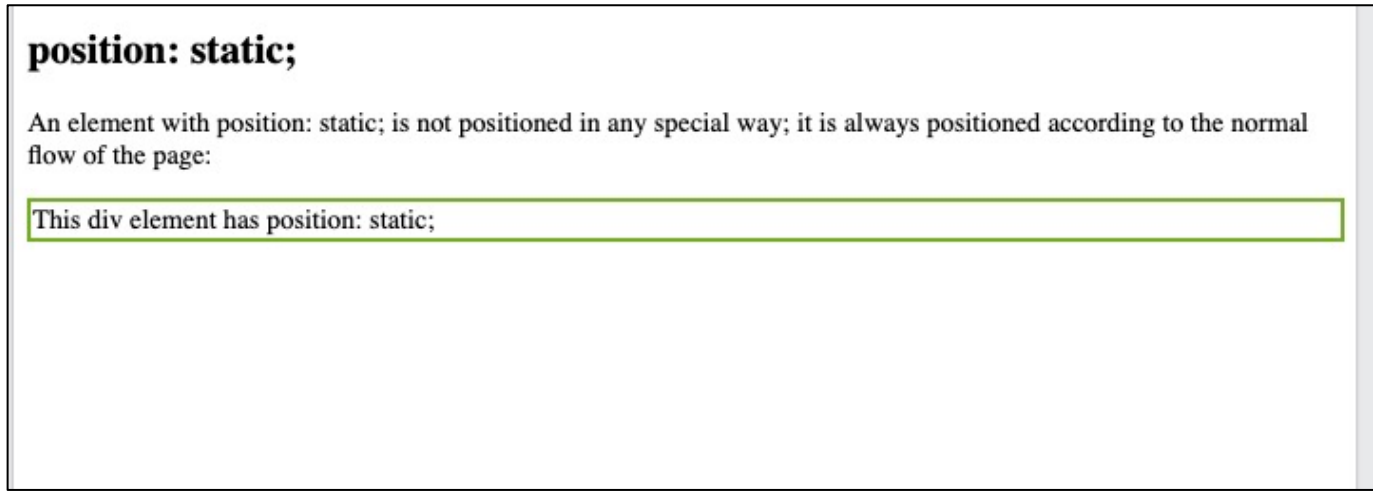
Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

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:

```
<!DOCTYPE html>
<html>
<head>
<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>
</html>
```

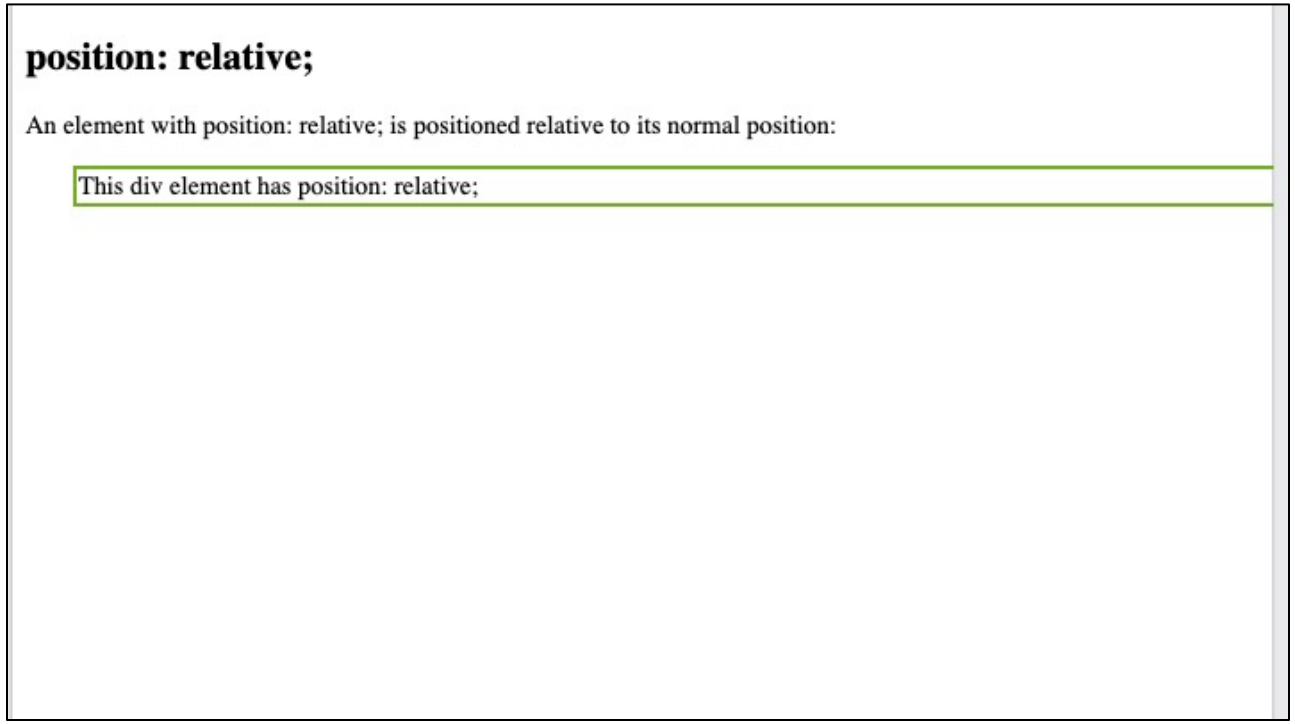
position_static.html



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.

```html
<!DOCTYPE html>
<html>
<head>
<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>
</html>
```



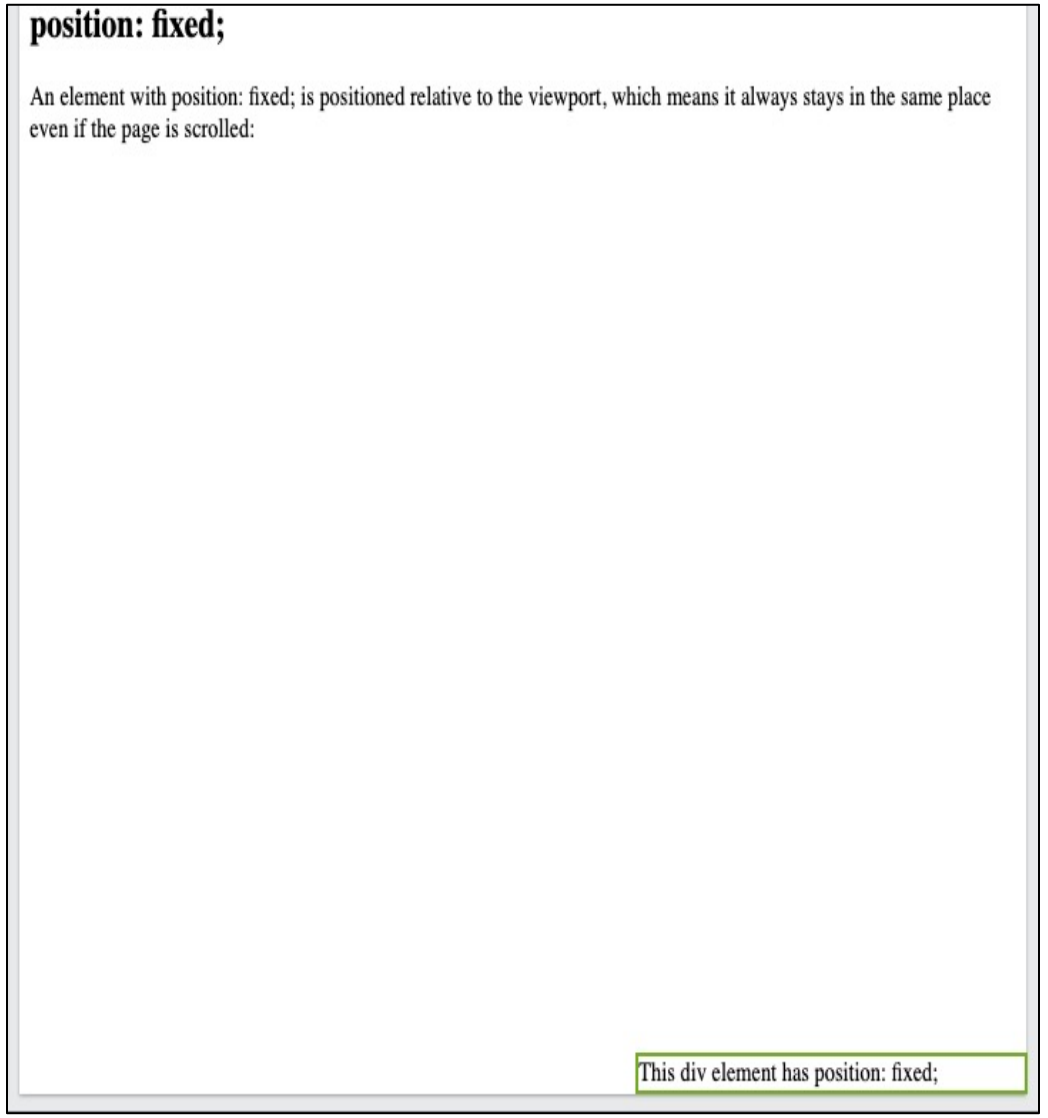
position_relative.html

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.

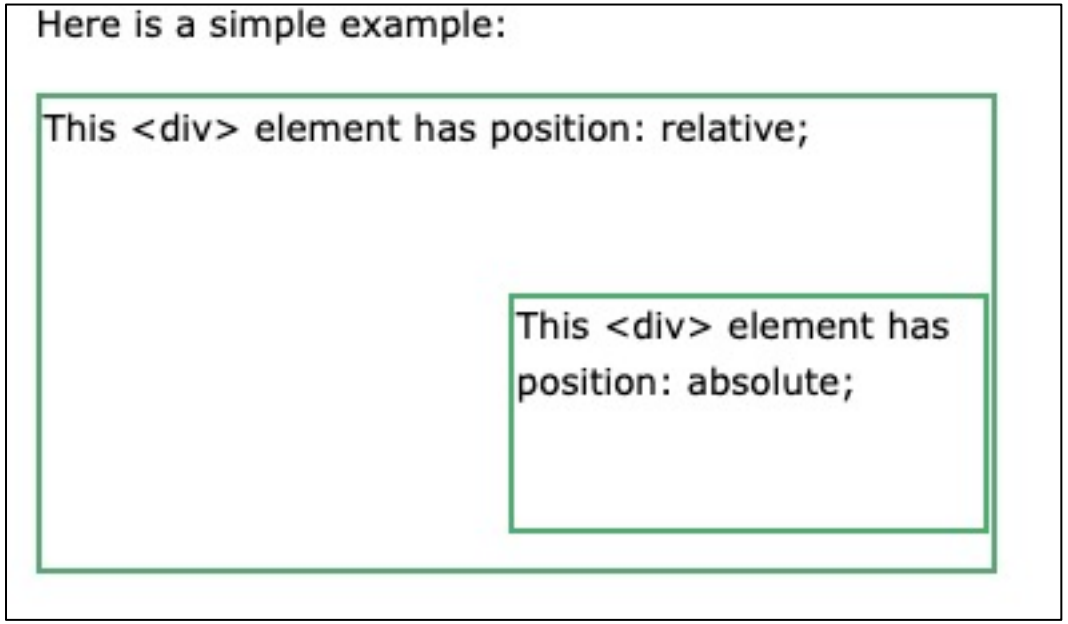
```html
<!DOCTYPE html>
<html>
<head>
<style>
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<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>
</html>
```

position_fixed.html



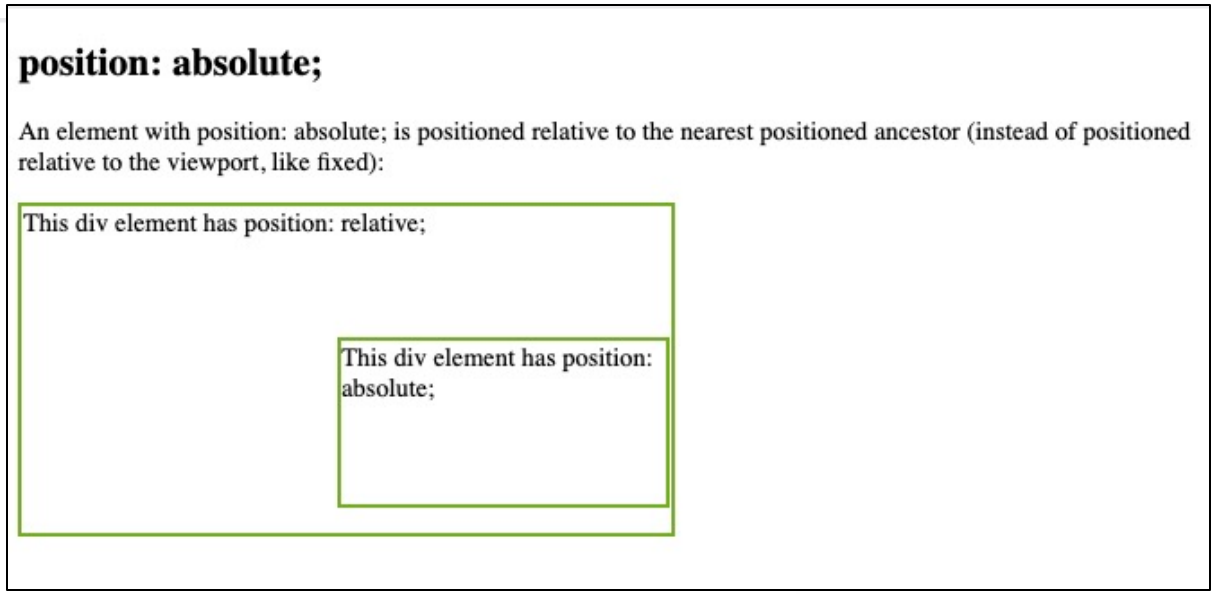
position: absolute

- 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: Absolute positioned elements are removed from the normal flow, and can overlap elements.

```html
<!DOCTYPE html>
<html>
<head>
<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>

</body>
</html>
```



Position_absolute.html

position: sticky

- An element with position: sticky; is positioned based on the user's scroll position.
- A sticky element toggles between relative and fixed, depending on the scroll position.
- It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

Note: Internet Explorer does not support sticky positioning. Safari requires a -webkit- prefix (see example below). You must also specify at least one of top, right, bottom or left for sticky positioning to work.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.sticky {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: #cae8ca;
  border: 2px solid #4CAF50;
}
</style>
</head>
<body>

<p>Try to <b>scroll</b> inside this frame to understand how sticky positioning works.
</p>

<div class="sticky">I am sticky!</div>

<div style="padding-bottom:2000px">
  <p>In this example, the sticky element sticks to the top of the page (top: 0), when
you reach its scroll position.</p>
  <p>Scroll back up to remove the stickyness.</p>
  <p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones
no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae
gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et.
Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>
  <p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones
no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae
gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et.
Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>
</div>

</body>
</html>
```

position_sticky.html

Try to **scroll** inside this frame to understand how sticky positioning works.

I am sticky!

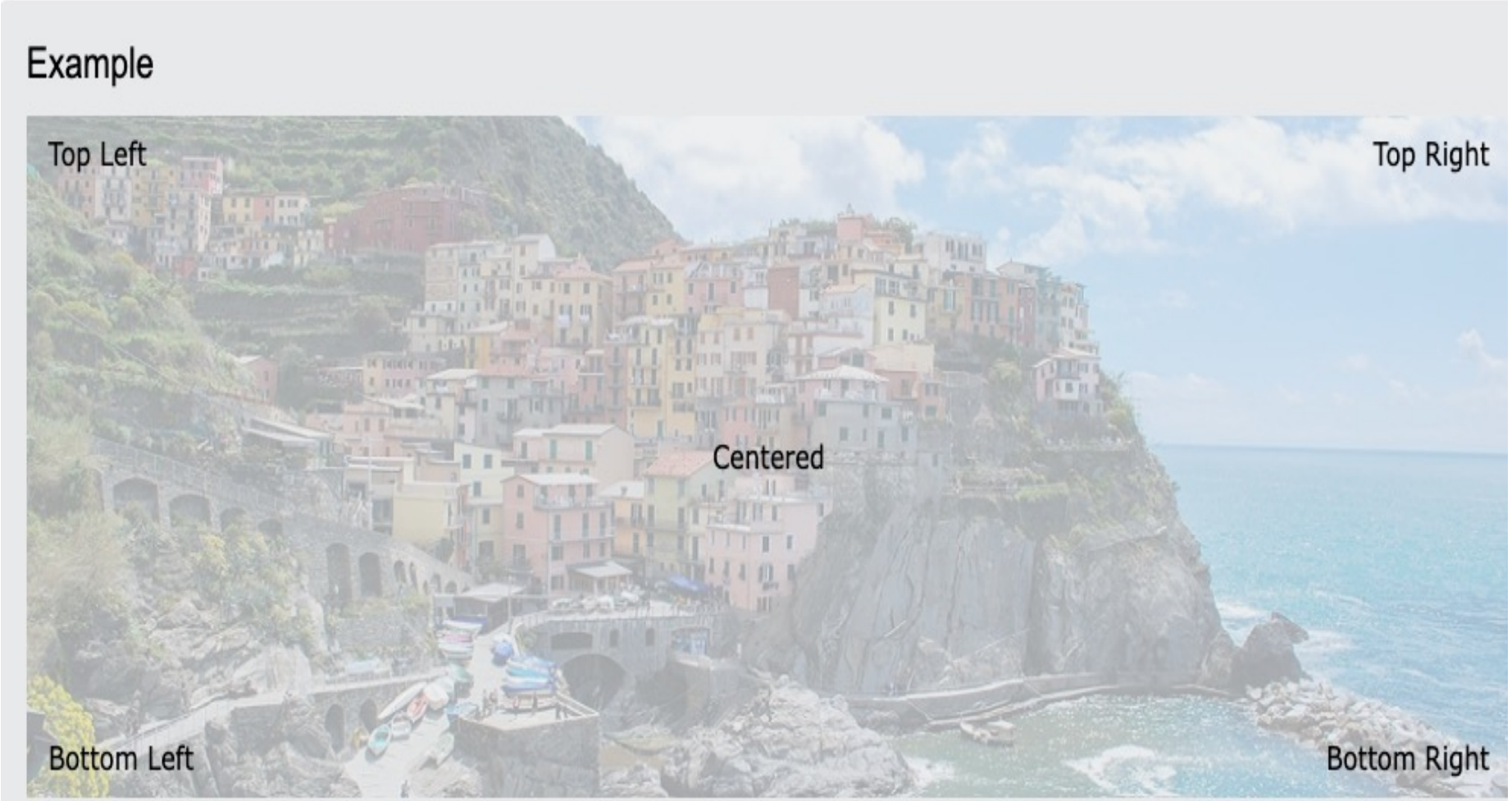
In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.

Scroll back up to remove the stickyness.

Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

Positioning Text In an Image



```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.topleft {
  position: absolute;
  top: 8px;
  left: 16px;
  font-size: 18px;
}

img {
  width: 100%;
  height: auto;
  opacity: 0.3;
}
</style>
</head>
<body>

<h2>Image Text</h2>
<p>Add some text to an image in the top left corner:</p>

<div class="container">
  <img src="img_5terre_wide.jpg" alt="Cinque Terre" width="1000" height="300">
  <div class="topleft">Top Left</div>
</div>

</body>
</html>
```

image_top_left_text.html

Try text positioning with following :

- Bottom Left
- Bottom Right
- Centered

All CSS Positioning Properties

Property	Description
bottom	Sets the bottom margin edge for a positioned box
clip	Clips an absolutely positioned element
left	Sets the left margin edge for a positioned box
position	Specifies the type of positioning for an element
right	Sets the right margin edge for a positioned box
top	Sets the top margin edge for a positioned box

z-index

The z-index property specifies the stack order of an element.

- When elements are positioned, they can overlap other elements.
- The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).
- An element can have a positive or negative stack order

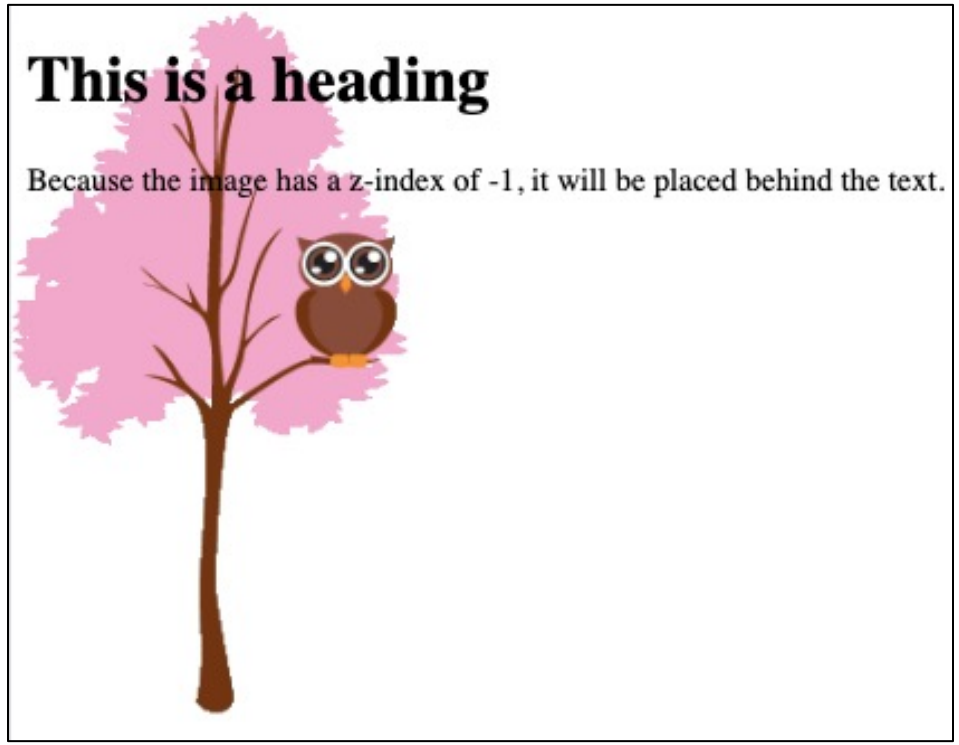
This is a heading

Because the image has a z-index of -1, it will be placed behind the text.

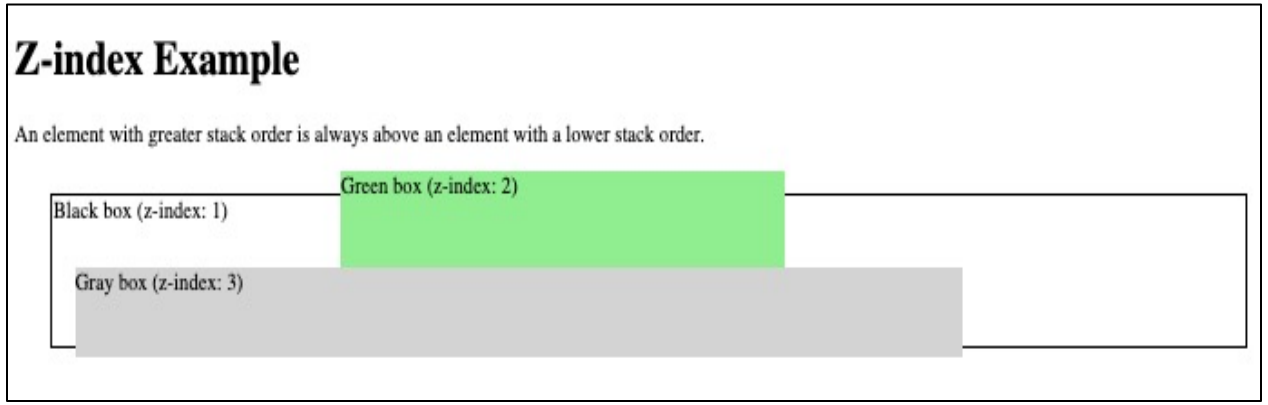
Note: z-index only works on positioned elements (position: absolute, position: relative, position: fixed, or position: sticky) and flex items (elements that are direct children of display: flex elements).

Z_index_example1.html

```html
<!DOCTYPE html>
<html>
<head>
<style>
img {
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<img src="img_tree.png">
<p>Because the image has a z-index of -1, it will be placed behind the text.</p>

</body>
</html>
```



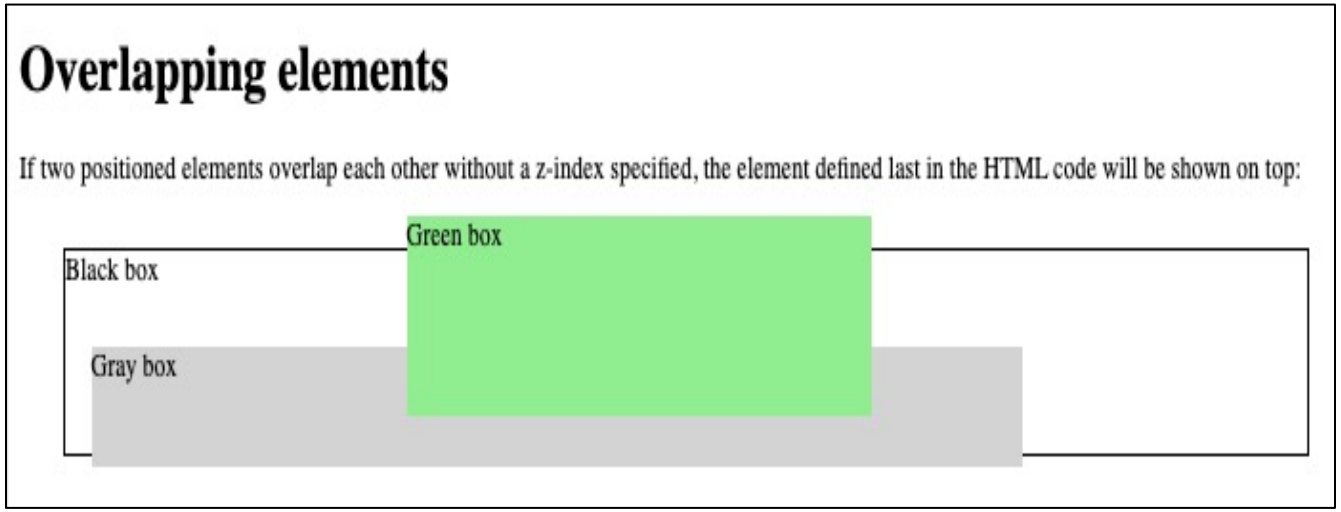
```html
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.black-box {
  position: relative;
  z-index: 1;
  border: 2px solid black;
  height: 100px;
  margin: 30px;
}

.gray-box {
  position: absolute;
  z-index: 3; /* gray box will be above both green and black box */
  background: lightgray;
  height: 60px;
  width: 70%;
  left: 50px;
  top: 50px;
}

.green-box {
  position: absolute;
  z-index: 2; /* green box will be above black box */
  background: lightgreen;
  width: 35%;
  left: 270px;
  top: -15px;
  height: 100px;
}
</style>
</head>
<body>

<h1>Z-index Example</h1>

<p>An element with greater stack order is always above an element with a lower stack order.</p>

<div class="container">
  <div class="black-box">Black box (z-index: 1)</div>
  <div class="gray-box">Gray box (z-index: 3)</div>
  <div class="green-box">Green box (z-index: 2)</div>
</div>

</body>
</html>
```



Note:
An element with greater stack order is always above an element with a lower stack order

Without z-index

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.black-box {
  position: relative;
  border: 2px solid black;
  height: 100px;
  margin: 30px;
}

.gray-box {
  position: absolute;
  background: lightgray;
  height: 60px;
  width: 70%;
  left: 50px;
  top: 50px;
}

.green-box {
  position: absolute;
  background: lightgreen;
  width: 35%;
  left: 270px;
  top: -15px;
  height: 100px;
}
</style>
</head>
<body>

<h1>Overlapping elements</h1>

<p>If two positioned elements overlap each other without a z-index specified,
the element defined last in the HTML code will be shown on top:</p>

<div class="container">
  <div class="black-box">Black box</div>
  <div class="gray-box">Gray box</div>
  <div class="green-box">Green box</div>
</div>

</body>
```



Without z-index.html

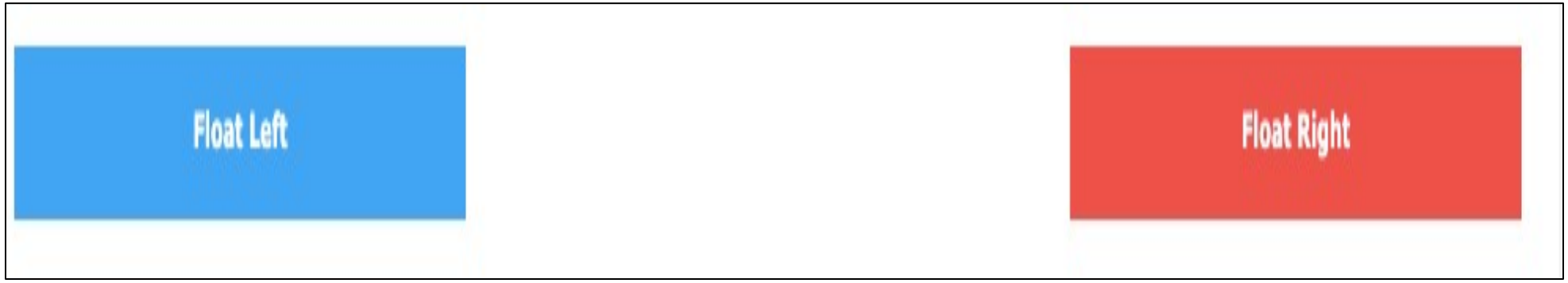
If two positioned elements overlap each other without a z-index specified, the element defined **last in the HTML code** will be shown on top.

Overlapping elements

If two positioned elements overlap each other without a z-index specified, the element defined last in the HTML code will be shown on top:

CSS Layout - float and clear

- 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

- The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

- 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

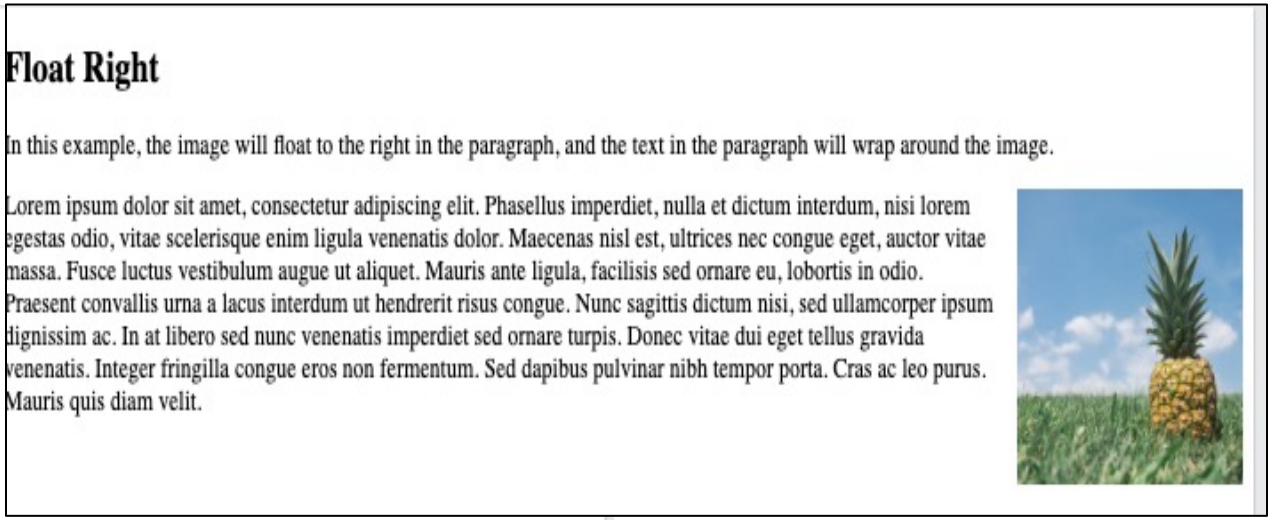
- In its simplest use, the float property can be used to wrap text around images.

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...

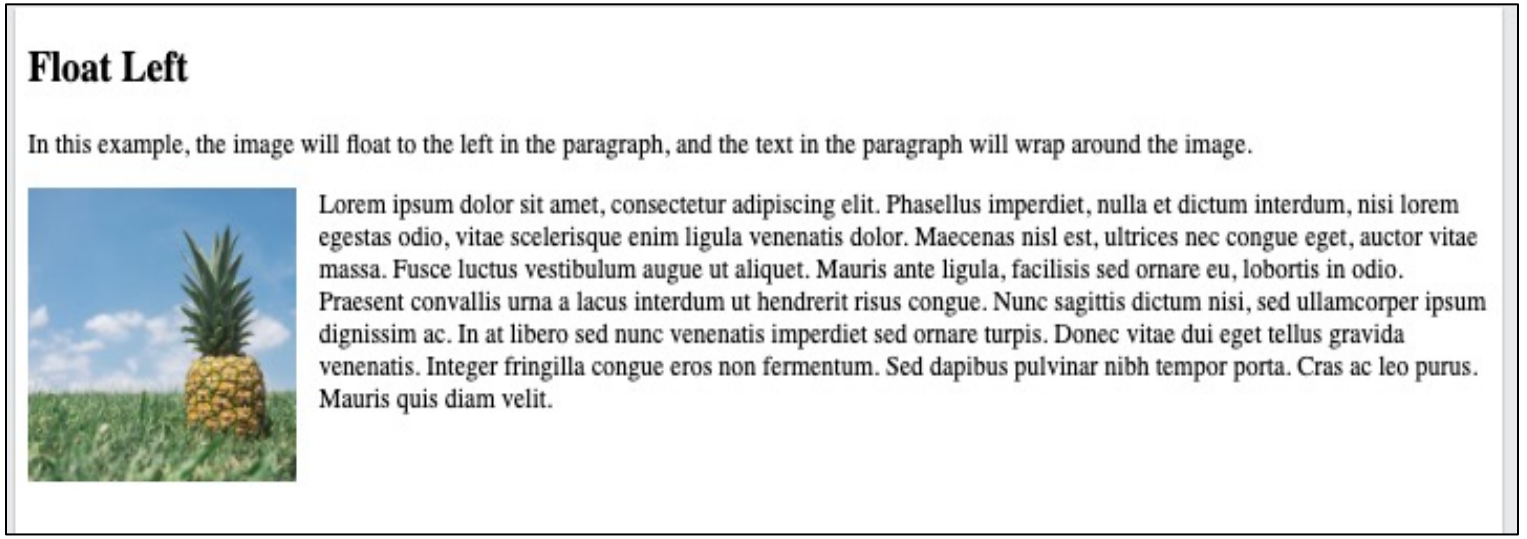
```html
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: right;
}
</style>
</head>
<body>

<h2>Float Right</h2>

<p>In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.</p>

<p><img src="pineapple.jpg" alt="Pineapple" style="width:170px;height:170px;margin-left:15px;">
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. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.</p>

</body>
</html>
```



float_right.html

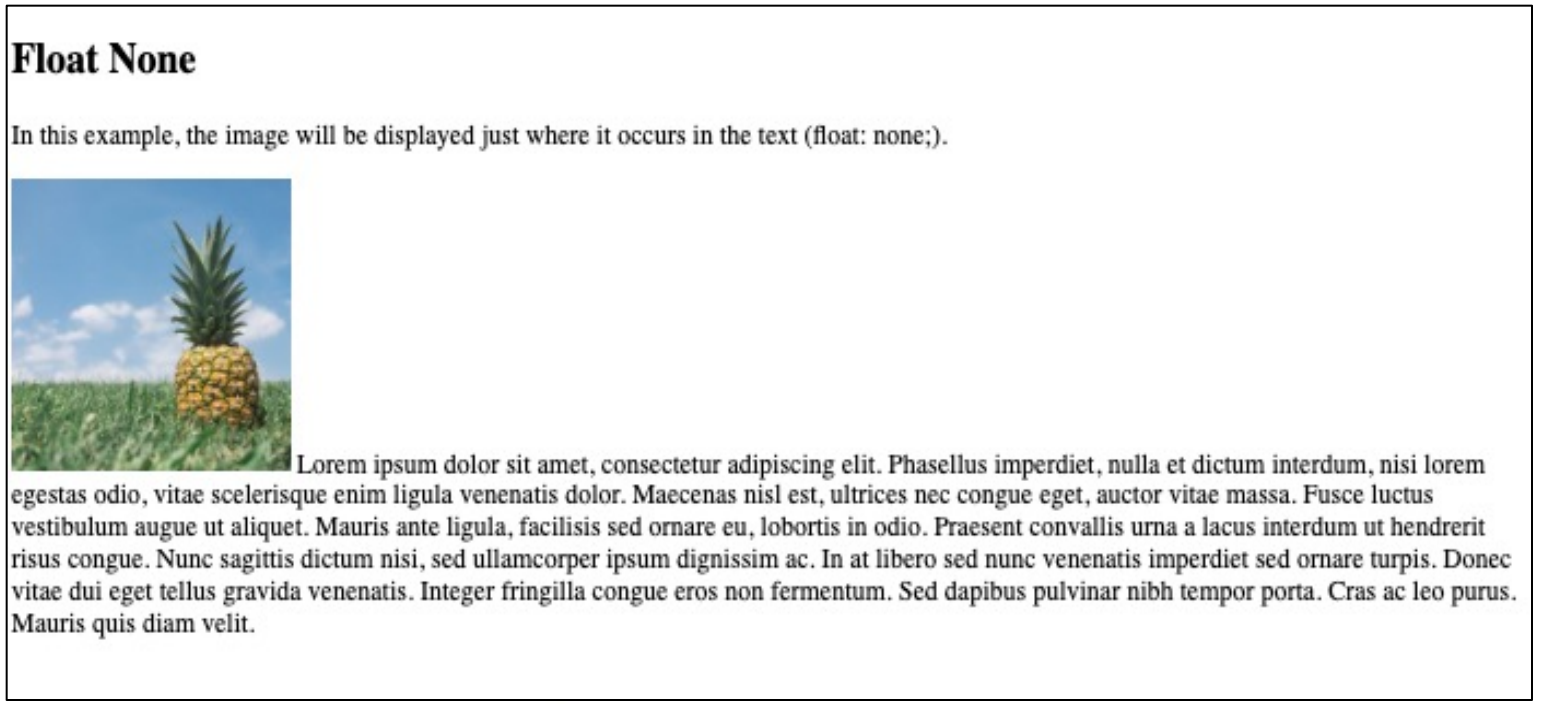
Float_left.html

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: left;
}
</style>
</head>
<body>

<h2>Float Left</h2>

<p>In this example, the image will float to the left in the paragraph, and the text in the paragraph
will wrap around the image.</p>

<p><img src="pineapple.jpg" alt="Pineapple" style="width:170px;height:170px;margin-right:15px;">
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. In at libero
sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer
fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris
quis diam velit.</p>

</body>
</html>
```



No float

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: none;
}
</style>
</head>
<body>

<h2>Float None</h2>

<p>In this example, the image will be displayed just where it occurs in the text (float: none;).</p>

<p><img src="pineapple.jpg" alt="Pineapple" style="width:170px;height:170px;">
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. In at libero
sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer
fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris
quis diam velit.</p>

</body>
</html>
```



float_none.html

The clear Property

- When we use the float property, and we want the next element below (not on right or left), we will have to use the clear property.
- The clear property specifies what should happen with the element that is next to a floating element.
- The clear property can have one of the following values:

none - The element is not pushed below left or right floated elements. This is default

left - The element is pushed below left floated elements

right - The element is pushed below right floated elements

both - The element is pushed below both left and right floated elements

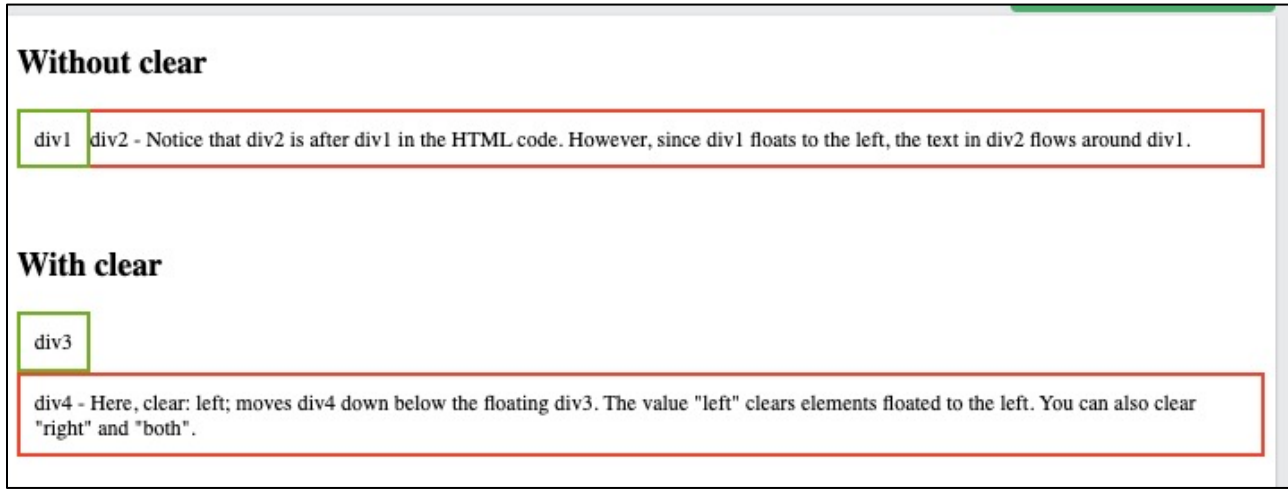
inherit - The element inherits the clear value from its parent

- When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left. Your floated element will continue to float, but the cleared element will appear below it on the web page.

Example

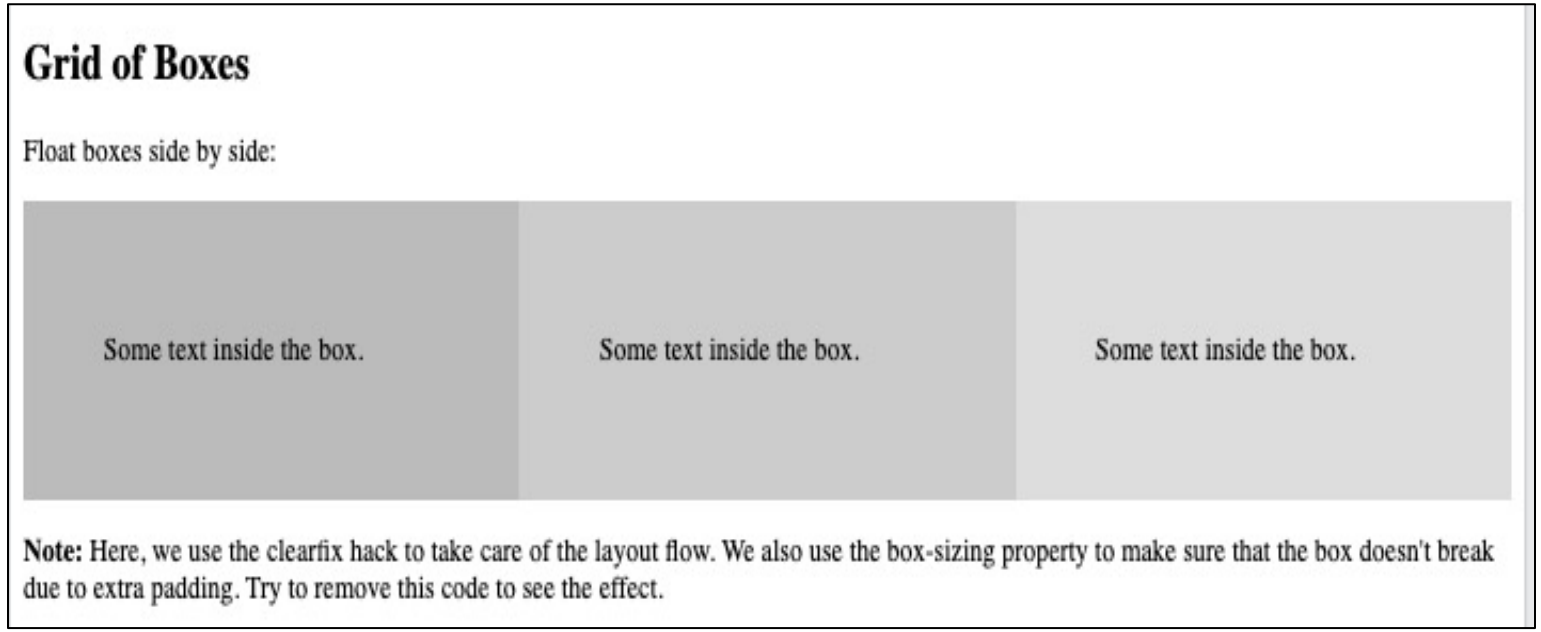
This example clears the float to the left. Here, it means that the <div2> element is pushed below the left floated <div1> element.

```
<!DOCTYPE html>
<html>
<head>
<style>
.div1 {
  float: left;
  padding: 10px;
  border: 3px solid #73AD21;
}

.div2 {
  padding: 10px;
  border: 3px solid red;
}

.div3 {
  float: left;
  padding: 10px;
  border: 3px solid #73AD21;
}

.div4 {
  padding: 10px;
  border: 3px solid red;
  clear: left;
}
</style>
</head>
<body>

<h2>Without clear</h2>
<div class="div1">div1</div>
<div class="div2">div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats
to the left, the text in div2 flows around div1.</div>
<br><br>

<h2>With clear</h2>
<div class="div3">div3</div>
<div class="div4">div4 - Here, clear: left; moves div4 down below the floating div3. The value "left"
clears elements floated to the left. You can also clear "right" and "both".</div>

</body>
</html>
```



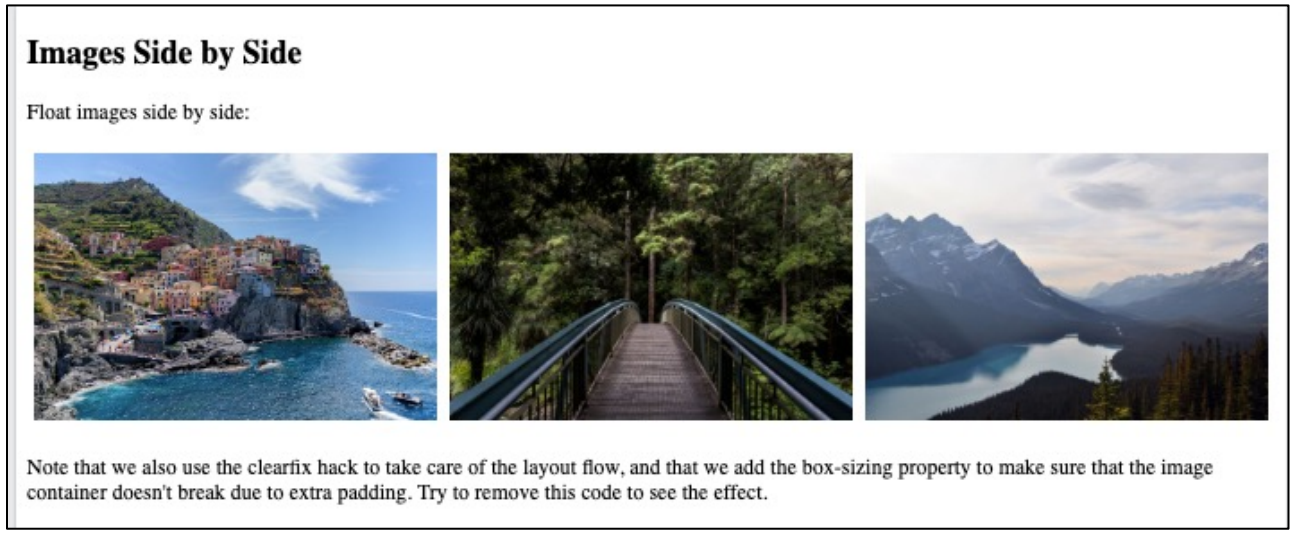
clear_example1.html

```html
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.box {
  float: left;
  width: 33.33%;
  padding: 50px;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
</style>
</head>
<body>

<h2>Grid of Boxes</h2>
<p>Float boxes side by side:</p>

<div class="clearfix">
  <div class="box" style="background-color:#bbb">
  <p>Some text inside the box.</p>
  </div>
  <div class="box" style="background-color:#ccc">
  <p>Some text inside the box.</p>
  </div>
  <div class="box" style="background-color:#ddd">
  <p>Some text inside the box.</p>
  </div>
</div>

<p><strong>Note:</strong> Here, we use the clearfix hack to take care of the layout flow.
We also use the box-sizing property to make sure that the box doesn't break due to extra padding. Try
to remove this code to see the effect.</p>

</body>
</html>
```



Float_example1.html

```html
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.img-container {
  float: left;
  width: 33.33%;
  padding: 5px;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
</style>
</head>
<body>

<h2>Images Side by Side</h2>
<p>Float images side by side:</p>

<div class="clearfix">
  <div class="img-container">
  <img src="img_5terre.jpg" alt="Italy" style="width:100%">
  </div>
  <div class="img-container">
  <img src="img_forest.jpg" alt="Forest" style="width:100%">
  </div>
  <div class="img-container">
  <img src="img_mountains.jpg" alt="Mountains" style="width:100%">
  </div>
</div>

<p>Note that we also use the clearfix hack to take care of the layout flow, and that we add the box-
sizing property to make sure that the image container doesn't break due to extra padding. Try to
remove this code to see the effect.</p>

</body>
</html>
```

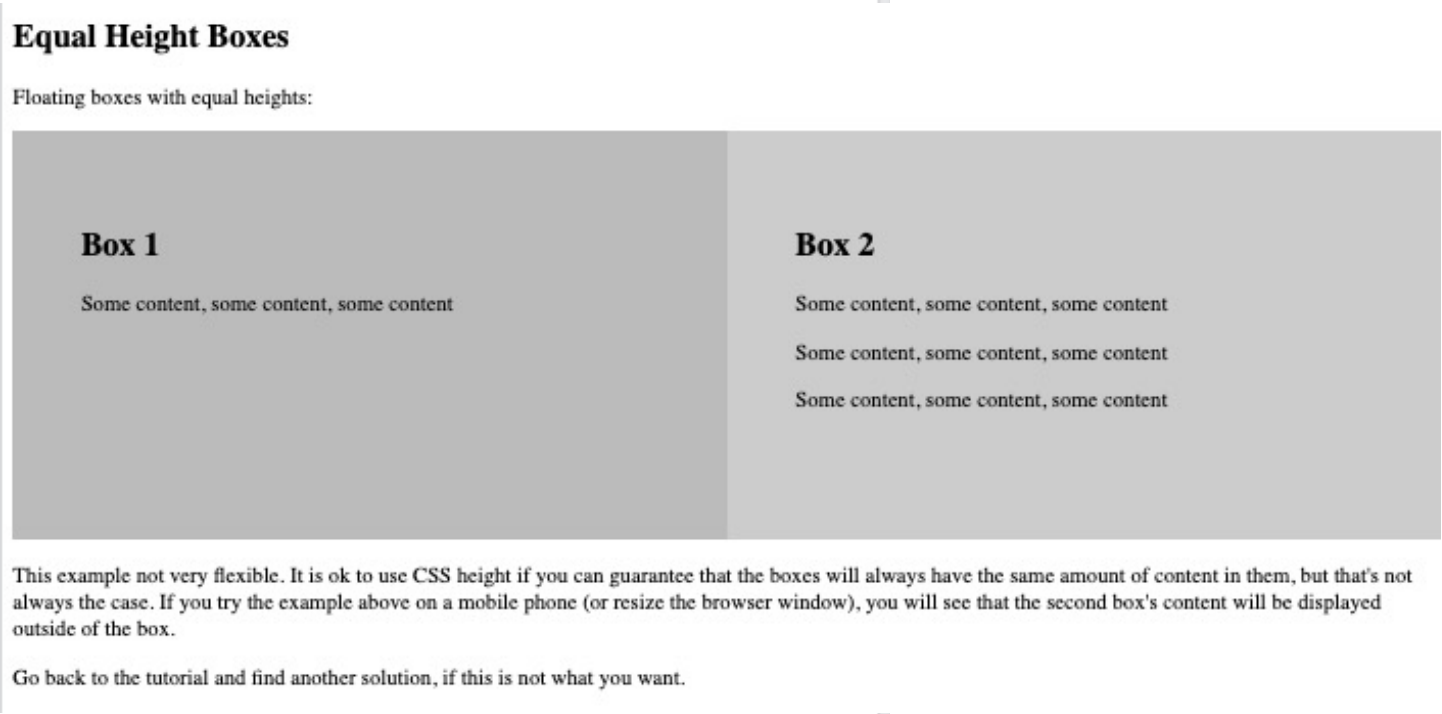


image_side_by_side

Equal Height Boxes

- In the previous example, you learned how to float boxes side by side with an equal width.
- However, it is not easy to create floating boxes with equal heights.
- A quick fix however, is to set a fixed height, like in the example below:

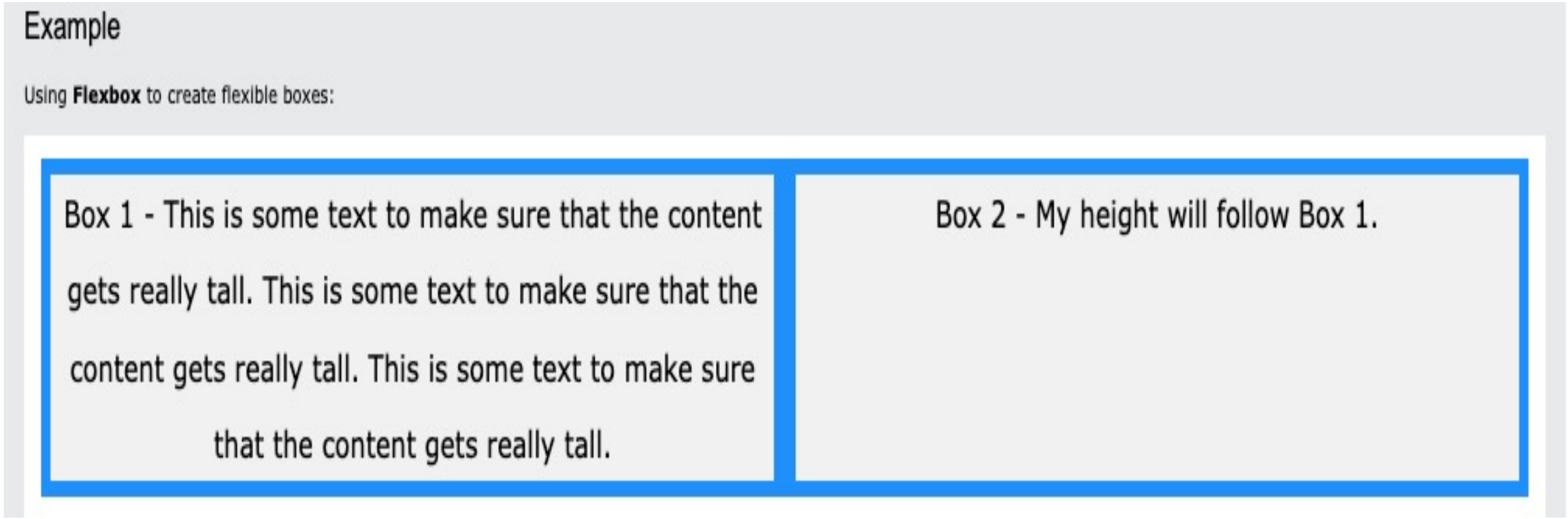
```html
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.box {
  float: left;
  width: 50%;
  padding: 50px;
  height: 300px;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
</style>
</head>
<body>

<h2>Equal Height Boxes</h2>
<p>Floating boxes with equal heights:</p>

<div class="clearfix">
  <div class="box" style="background-color:#bbb">
  <h2>Box 1</h2>
  <p>Some content, some content, some content</p>
  </div>
  <div class="box" style="background-color:#ccc">
  <h2>Box 2</h2>
  <p>Some content, some content, some content</p>
  <p>Some content, some content, some content</p>
  <p>Some content, some content, some content</p>
  </div>
</div>

<p>This example not very flexible. It is ok to use CSS height if you can guarantee that the boxes will always have the same amount of content in them, but that's not always the case. If you try the example above on a mobile phone (or resize the browser window), you will see that the second box's content will be displayed outside of the box.</p>
<p>Go back to the tutorial and find another solution, if this is not what you want.</p>

</body>
</html>
```



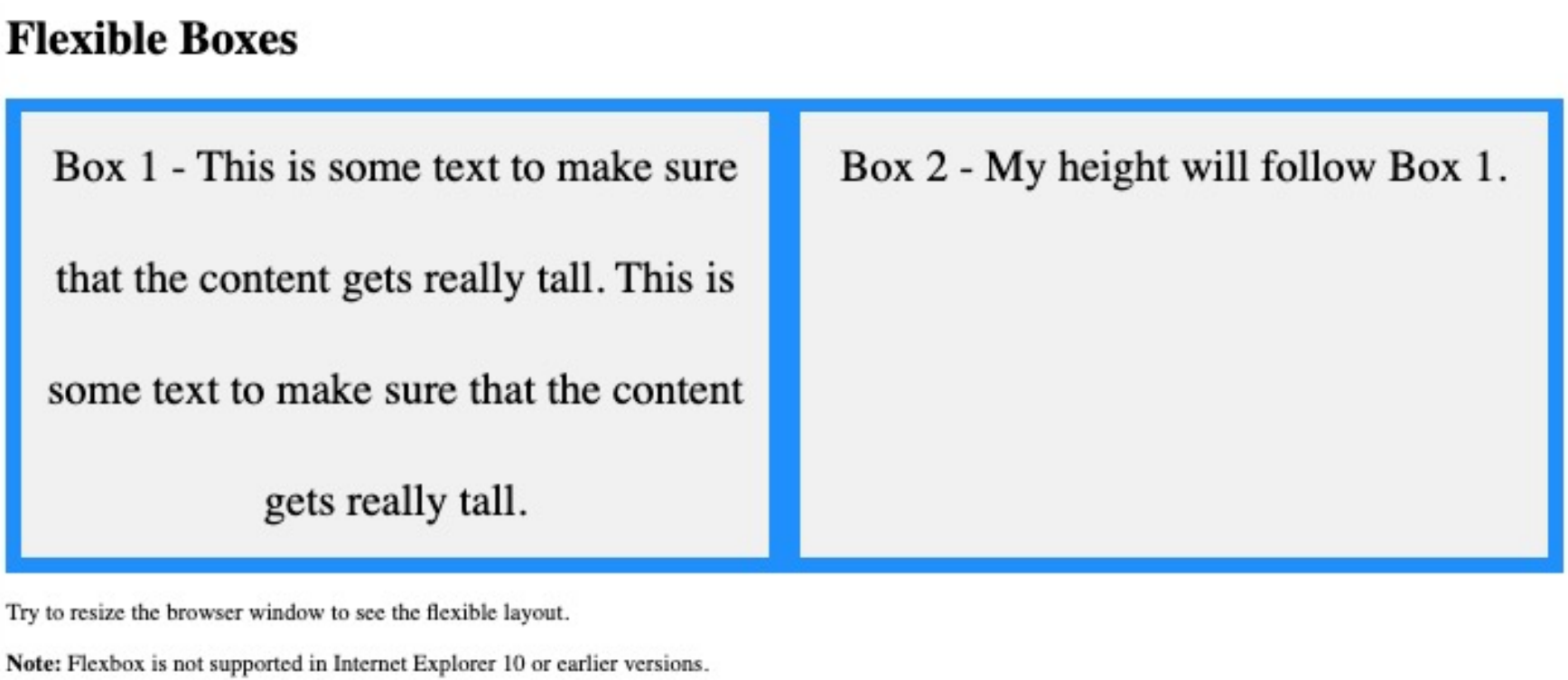
FLEXBOX

- **Previous approach** is not very flexible.
- It is ok if you can guarantee that the boxes will always have the same amount of content in them.
- But many times, the content is not the same.
- If you try the example above on a mobile phone, you will see that the second box's content will be displayed outside of the box.
- This is where CSS3 Flexbox comes in handy - as it can automatically stretch boxes to be as long as the longest box.

FLEXBOX


Example
Using Flexbox to create flexible boxes:
Box 1 - This is some text to make sure that the content gets really tall. This is some text to make sure that the content gets really tall. This is some text to make sure that the content gets really tall.
Box 2 - My height will follow Box 1.

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-wrap: nowrap;
  background-color: DodgerBlue;
}

.flex-container .box {
  background-color: #f1f1f1;
  width: 50%;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>Flexible Boxes</h1>

<div class="flex-container">
  <div class="box">Box 1 - This is some text to make sure that the content gets really tall. This is some text to
make sure that the content gets really tall.</div>
  <div class="box">Box 2 - My height will follow Box 1.</div>
</div>

<p>Try to resize the browser window to see the flexible layout.</p>
<p><strong>Note:</strong> Flexbox is not supported in Internet Explorer 10 or earlier versions.</p>

</body>
</html>
```

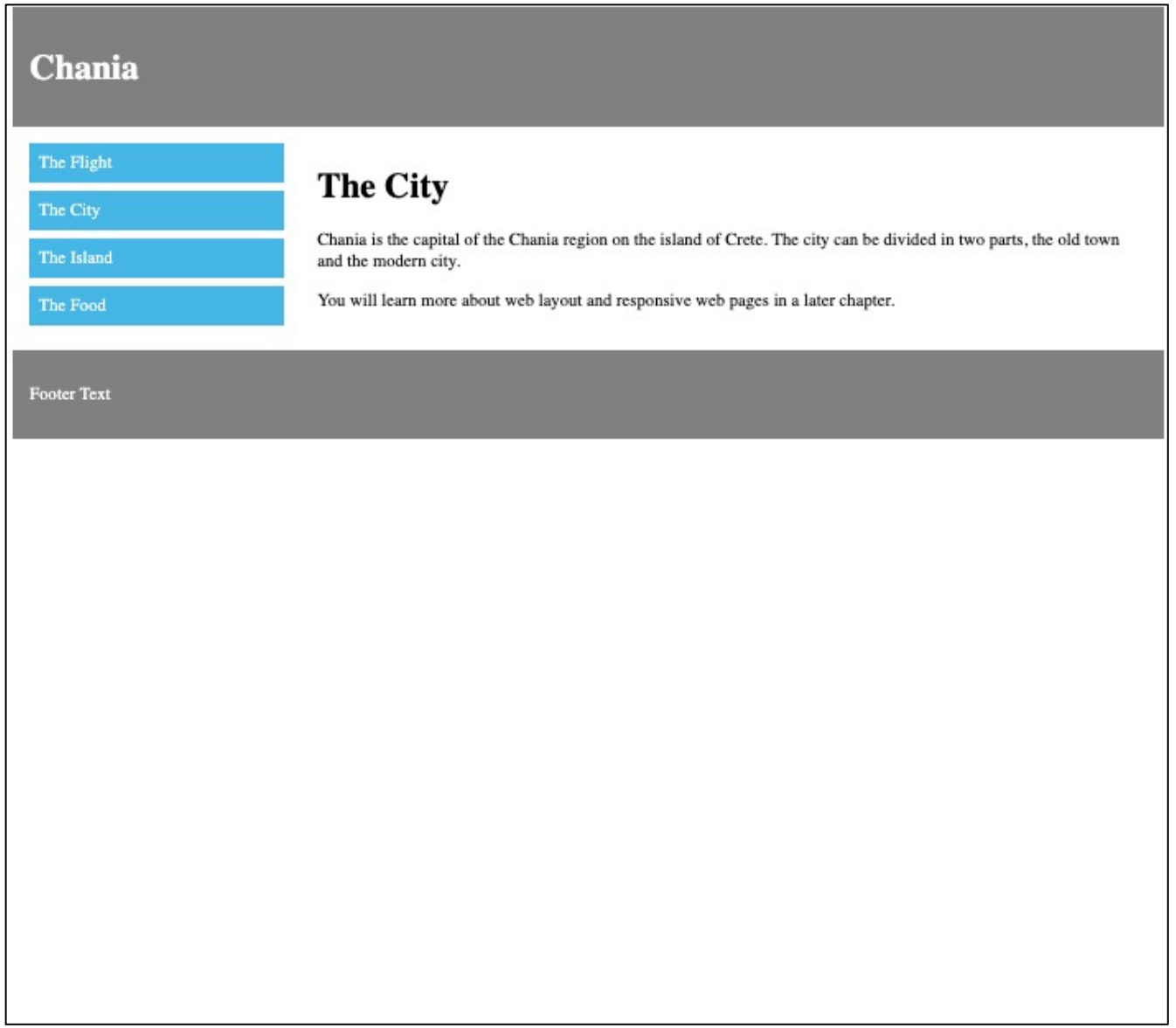


Flexbox.html

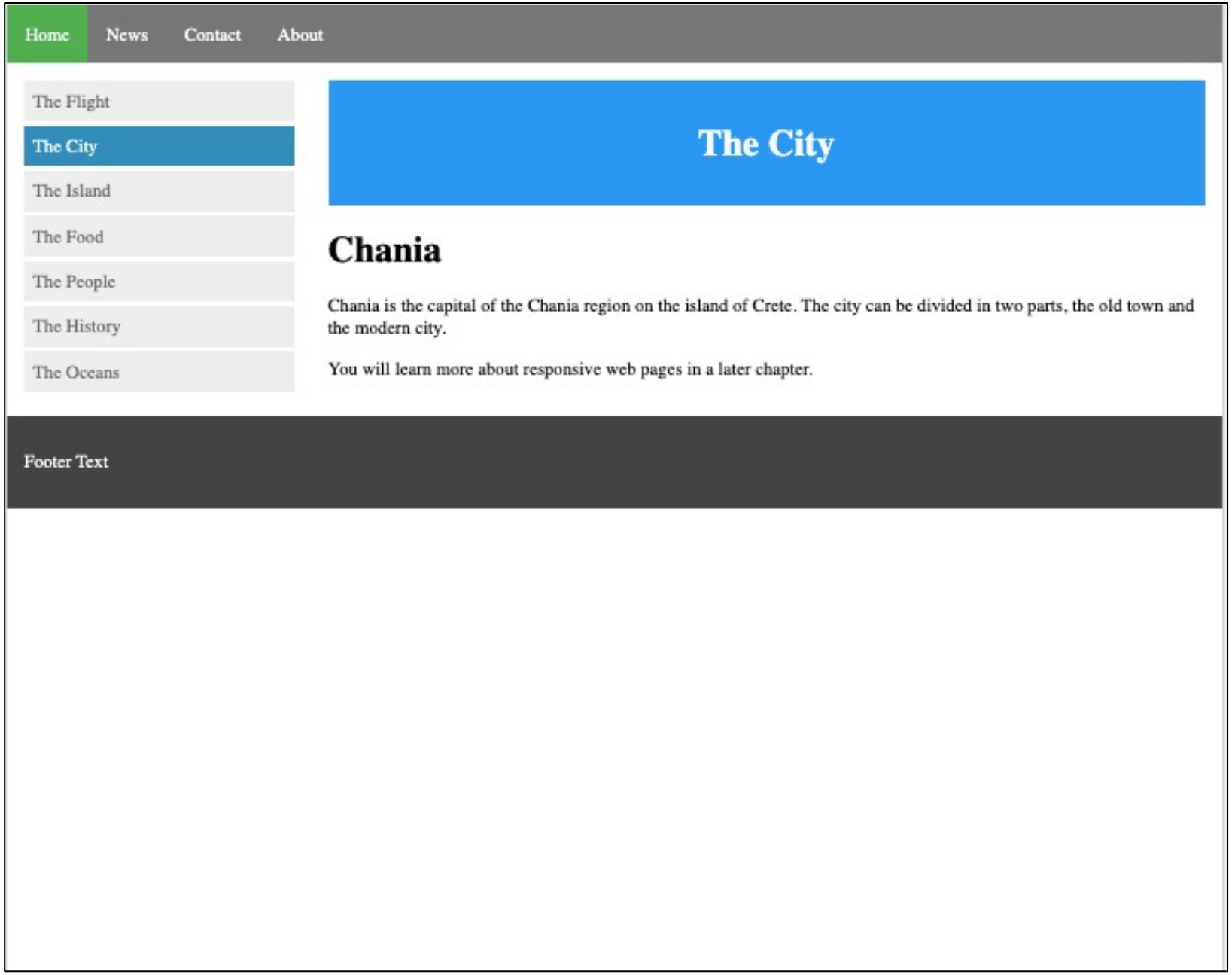
All CSS Float Properties

Property	Description
box-sizing	Defines how the width and height of an element are calculated: should they include padding and borders, or not
clear	Specifies what should happen with the element that is next to a floating element
float	Specifies whether an element should float to the left, right, or not at all
overflow	Specifies what happens if content overflows an element's box
overflow-x	Specifies what to do with the left/right edges of the content if it overflows the element's content area
overflow-y	Specifies what to do with the top/bottom edges of the content if it overflows the element's content area

Question:
Create this layout using float concepts.



Question:
Create this layout using float concepts.



OPACITY/ TRANSPARENCY

- The opacity property specifies the opacity/transparency of an element.
- The opacity property can take a value from 0.0 - 1.0. The lower the value, the more transparent.

EXAMPLE:

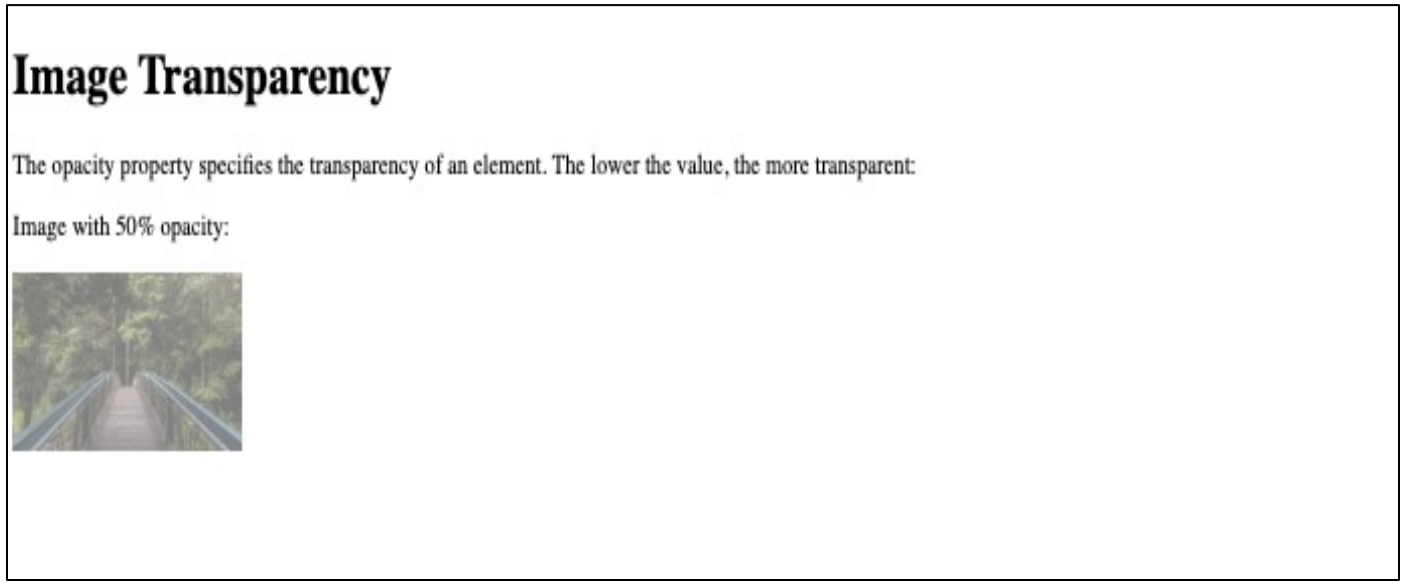
```
img {
     opacity: 0.5;
     }
```


opacity 0.2

opacity 0.5

opacity 1
(default)

```html
<!DOCTYPE html>
<html>
<head>
<style>
img {
  opacity: 0.5;
}
</style>
</head>
<body>

<h1>Image Transparency</h1>
<p>The opacity property specifies the transparency of an element. The lower the value, the more transparent:</p>

<p>Image with 50% opacity:</p>
<img src="img_forest.jpg" alt="Forest" width="170" height="100">

</body>
</html>
```



Opacity_example1.html

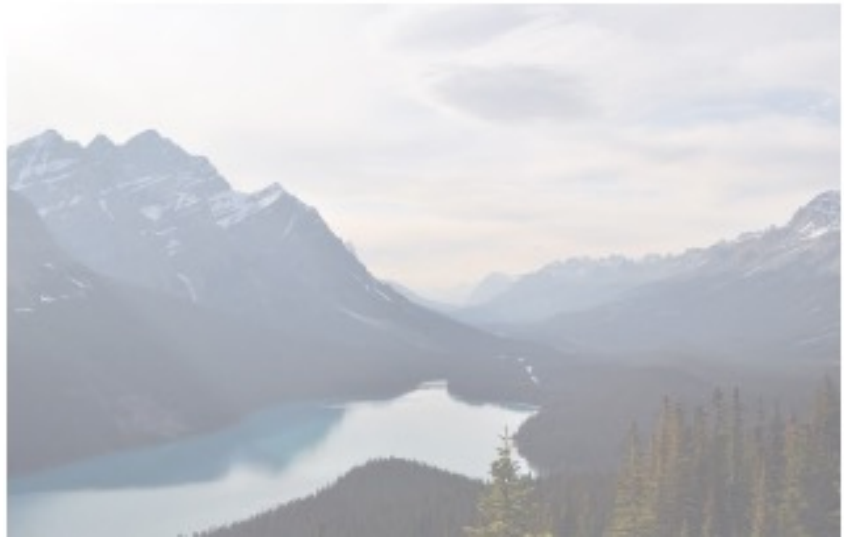
Transparent Hover Effect

- The opacity property is often used together with the :hover selector to change the opacity on mouse-over.

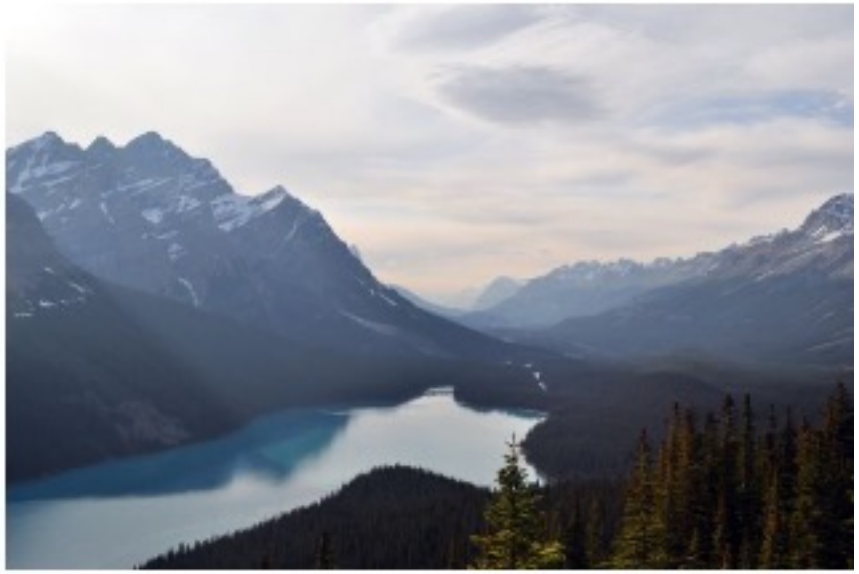
Before hover

After hover

Before hover



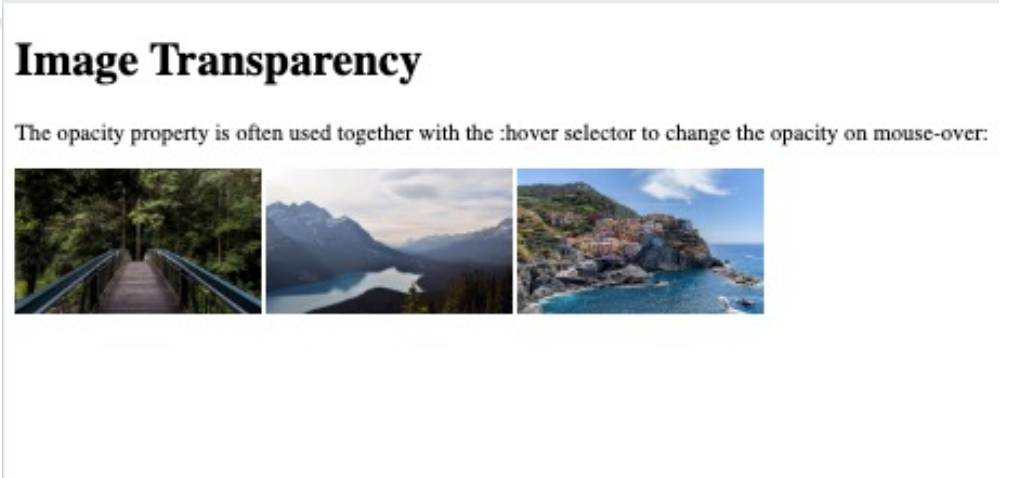
After hover



Before hover

After hover

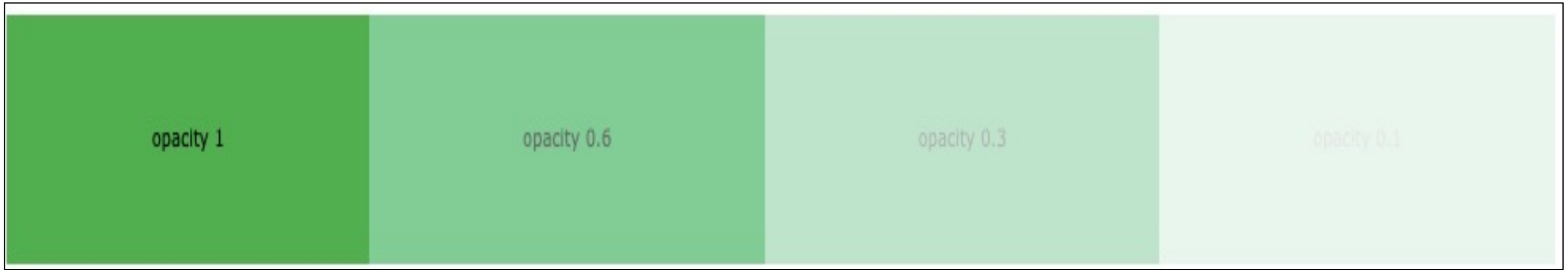
```html
<!DOCTYPE html>
<html>
<head>
<style>
img:hover {
  opacity: 0.5;
}
</style>
</head>
<body>

<h1>Image Transparency</h1>
<p>The opacity property is often used together with the :hover selector to change the opacity on mouse-over:</p>
<img src="img_forest.jpg" alt="Forest" width="170" height="100">
<img src="img_mountains.jpg" alt="Mountains" width="170" height="100">
<img src="img_5terre.jpg" alt="Italy" width="170" height="100">

</body>
</html>
```



image_hover_transparent

Transparent Box

- When using the opacity property to add transparency to the background of an element, all of its child elements inherit the same transparency.
- This can make the text inside a fully transparent element hard to read.



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: #4CAF50;
  padding: 10px;
}

div.first {
  opacity: 0.1;
}

div.second {
  opacity: 0.3;
}

div.third {
  opacity: 0.6;
}
</style>
</head>
<body>

<h1>Transparent Box</h1>
<p>When using the opacity property to add transparency to the background of an element, all of its child elements
become transparent as well. This can make the text inside a fully transparent element hard to read:</p>

<div class="first"><p>opacity 0.1</p></div>
<div class="second"><p>opacity 0.3</p></div>
<div class="third"><p>opacity 0.6</p></div>
<div><p>opacity 1 (default)</p></div>

</body>
</html>
```

transparent_box

Transparent Box

When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read:

opacity 0.1

opacity 0.3

opacity 0.6

opacity 1 (default)

Question:

Create this layout.

visibility

- Opacity
- Visibility
- display

DISPLAY

- The display property is the most important CSS property for controlling layout.
- The display property specifies if/how an element is displayed.
- Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.
- 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 element is a block-level element.

Examples of block-level elements:
-
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

DISPLAY

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

Examples of inline elements:

-
- <a>

DISPLAY

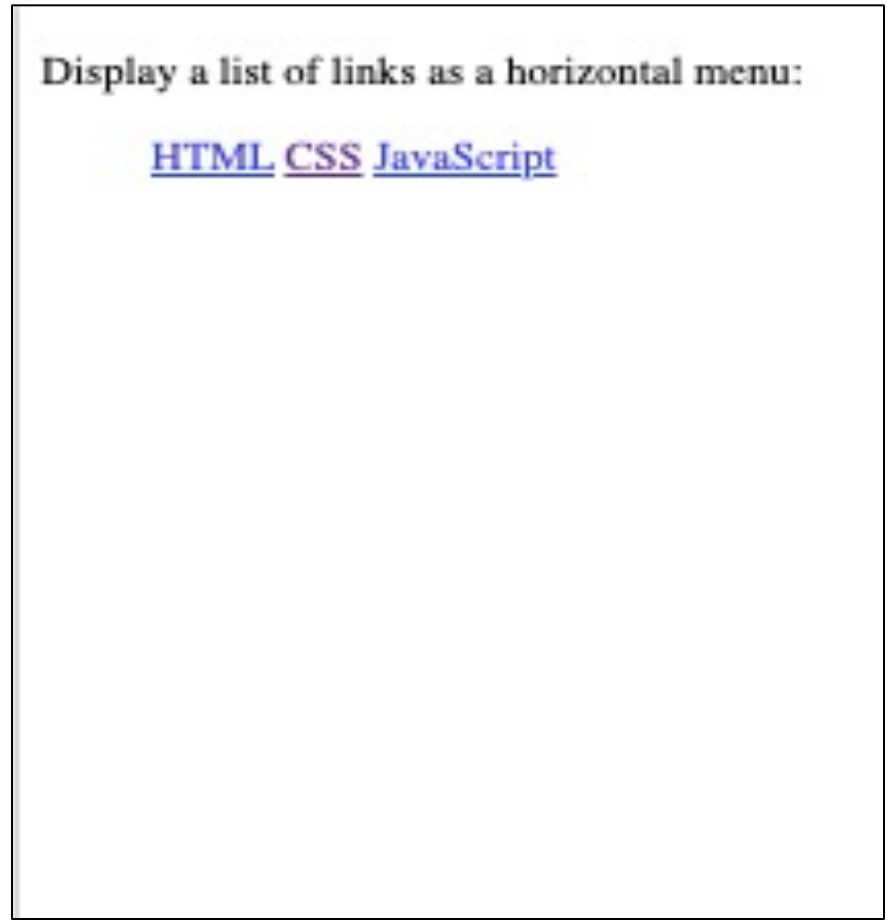
- Display: none;
- display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.
- The <script> element uses display: none; as default.

Override The Default Display Value

- Element has a default display value. However, you can override this.
- Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards.
- A common example is making inline <li> elements for horizontal menus.

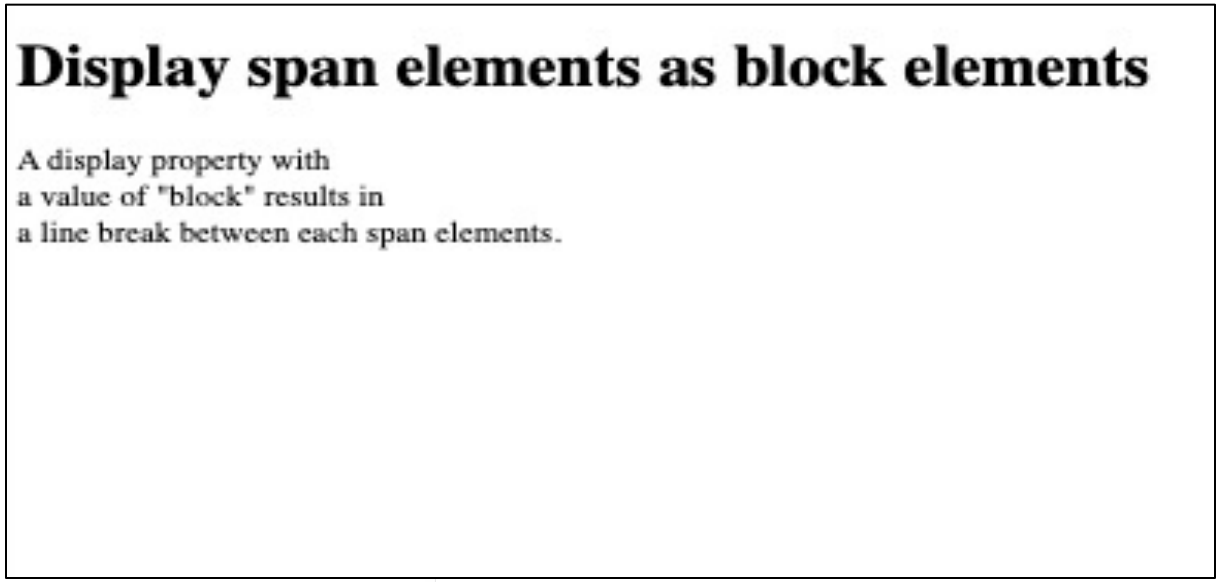
Note: Setting the display property of an element only changes **how the element is displayed**, NOT what kind of element it is. So, an inline element with display: block; is not allowed to have other block elements inside it.

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

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

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

</body>
</html>
```



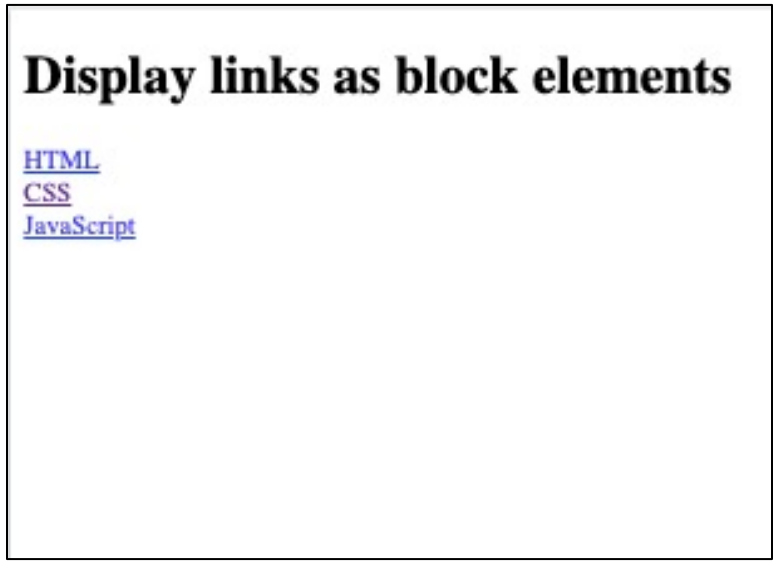
DISPLAY_INLINE.HTML

```html
<!DOCTYPE html>
<html>
<head>
<style>
span {
  display: block;
}
</style>
</head>
<body>

<h1>Display span elements as block elements</h1>

<span>A display property with</span> <span>a value of "block" results in</span> <span>a line break between each span elements.</span>

</body>
</html>
```



Display_span.html

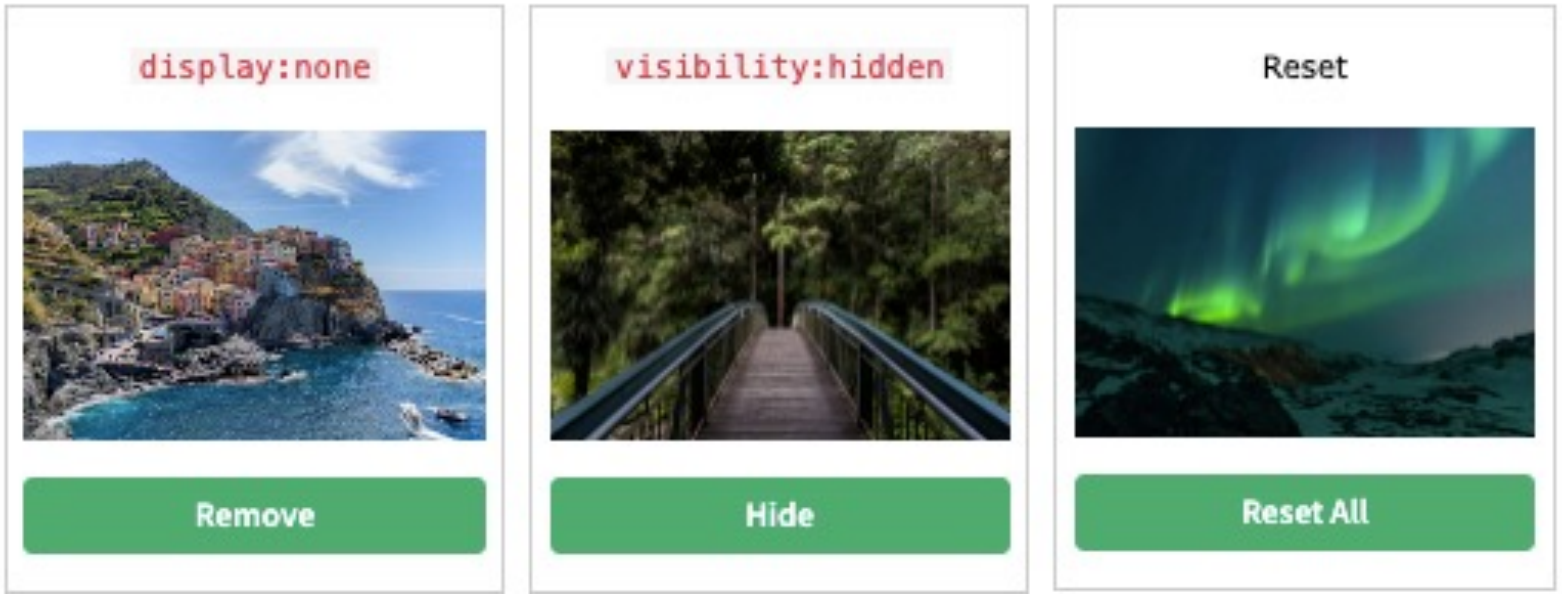
```html
<!DOCTYPE html>
<html>
<head>
<style>
a {
  display: block;
}
</style>
</head>
<body>

<h1>Display links as block elements</h1>

<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>

</body>
</html>
```



Display_block.html

Hide an Element

display:none or visibility:hidden

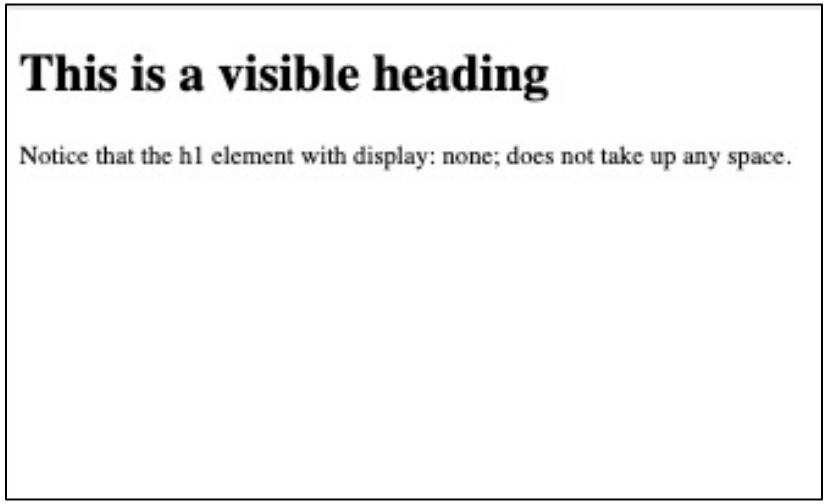


Hiding an element can be done by setting the display property to none.

The element will be hidden, and the page will be displayed as if the element is not there.

Display:none

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  display: none;
}
</style>
</head>
<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>
</html>
```

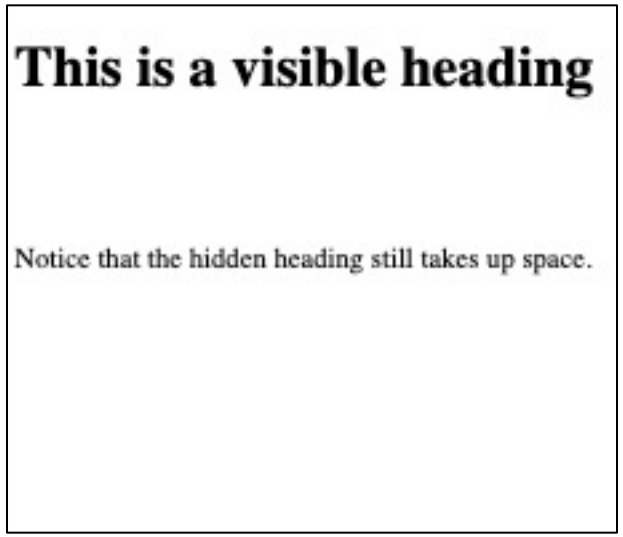


Display_none.html

visibility:hidden

- visibility:hidden; also hides an element.
- However, the element will still take up the same space as before.
- The element will be hidden, but still affect the layout.

```html
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  visibility: hidden;
}
</style>
</head>
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up space.</p>

</body>
</html>
```



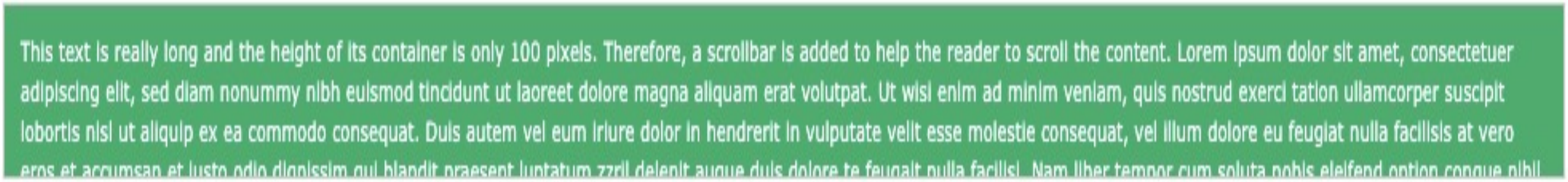
visibility_hidden.html

CSS Display/Visibility Properties

Property	Description
display	Specifies how an element should be displayed
visibility	Specifies whether or not an element should be visible

CSS Layout - Overflow

- The CSS overflow property controls what happens to content that is too big to fit into an area.



CSS Overflow

- The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.
- The overflow property has the following values:

visible - Default. The overflow is not clipped. The content renders outside the element's box

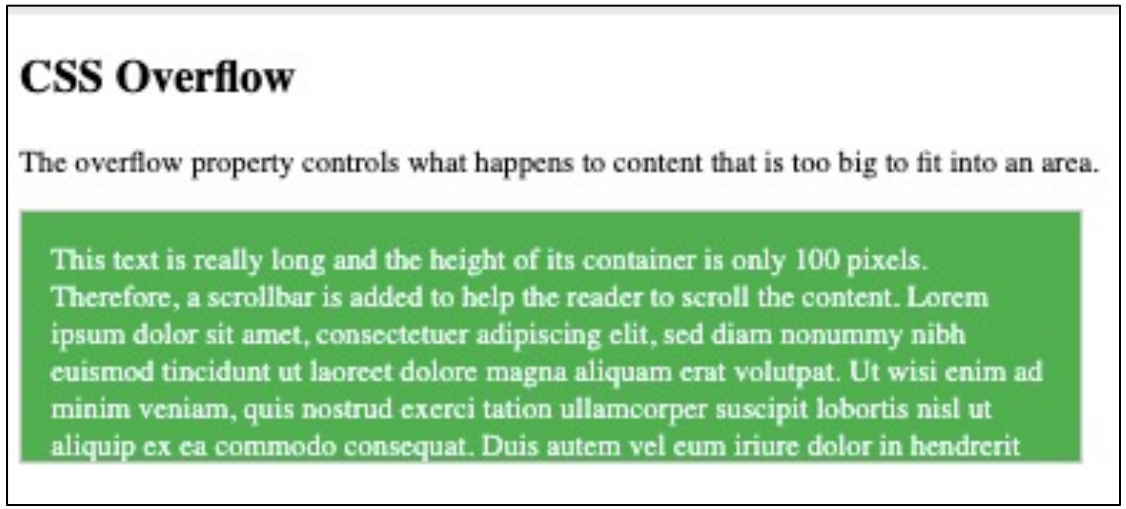
hidden - The overflow is clipped, and the rest of the content will be invisible

scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content

auto - Similar to scroll, but it adds scrollbars only when necessary

Note: The overflow property only works for block elements with a specified height.

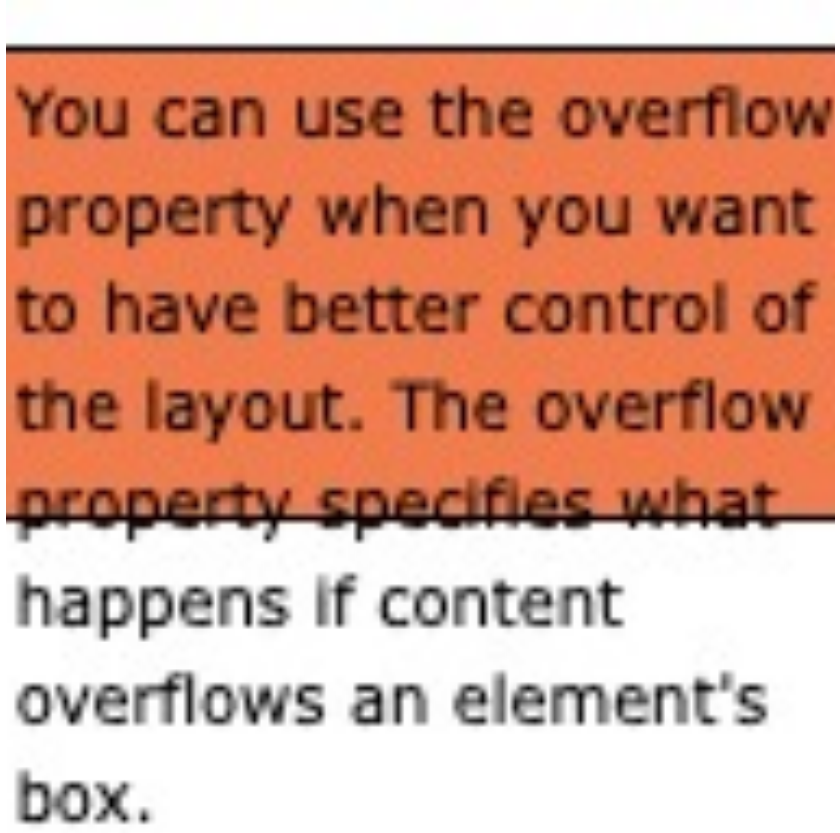
```html
<!DOCTYPE html>
<html>
<head>
<style>
#overflowTest {
  background: #4CAF50;
  color: white;
  padding: 15px;
  width: 50%;
  height: 100px;
  overflow: scroll;
  border: 1px solid #ccc;
}
</style>
</head>
<body>

<h2>CSS Overflow</h2>

<p>The overflow property controls what happens to content that is too big to fit into an area.</p>

<div id="overflowTest">This text is really long and the height of its container is only 100 pixels. Therefore, a scrollbar is added to help the reader to scroll the content. 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. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem.</div>

</body>
</html>
```



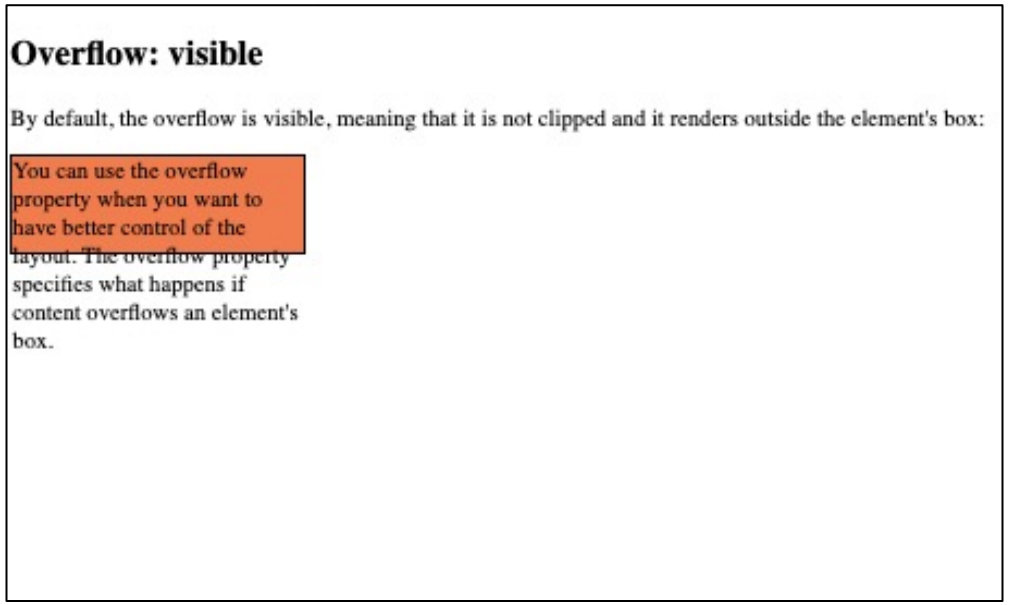
overflow_example.html

overflow: visible

- By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box:



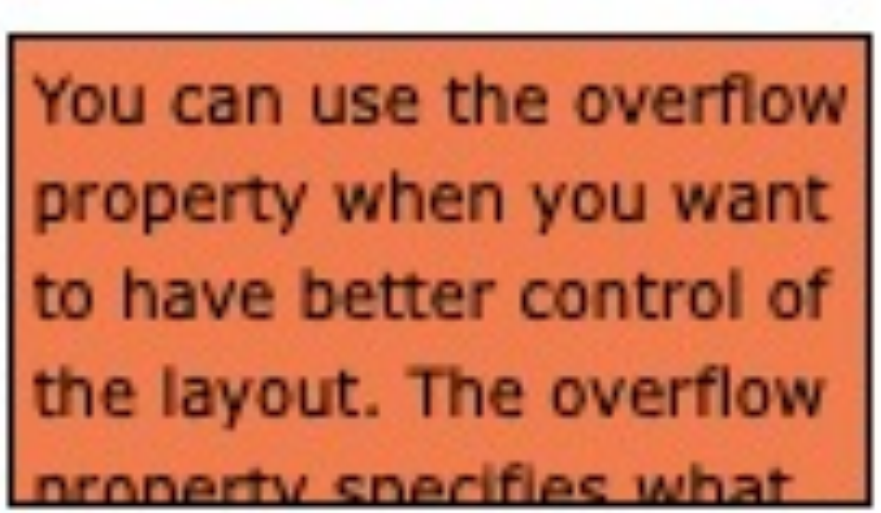
```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: coral;
  width: 200px;
  height: 65px;
  border: 1px solid;
  overflow: visible;
}
</style>
</head>
<body>

<h2>Overflow: visible</h2>

<p>By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box:</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property
specifies what happens if content overflows an element's box.</div>

</body>
</html>
```



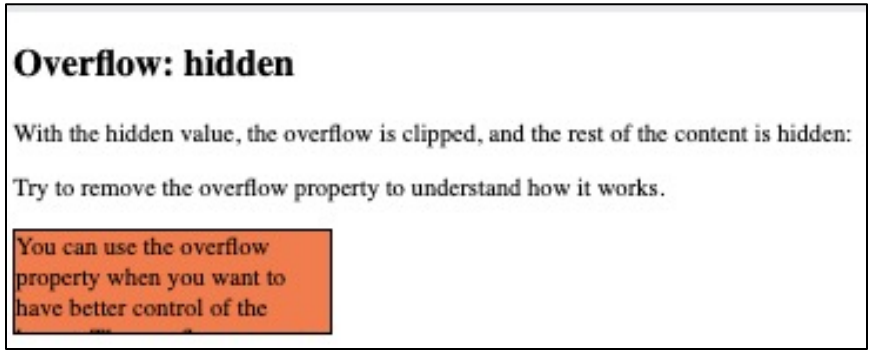
overflow_visible.html

overflow: hidden

- With the hidden value, the overflow is clipped, and the rest of the content is hidden:



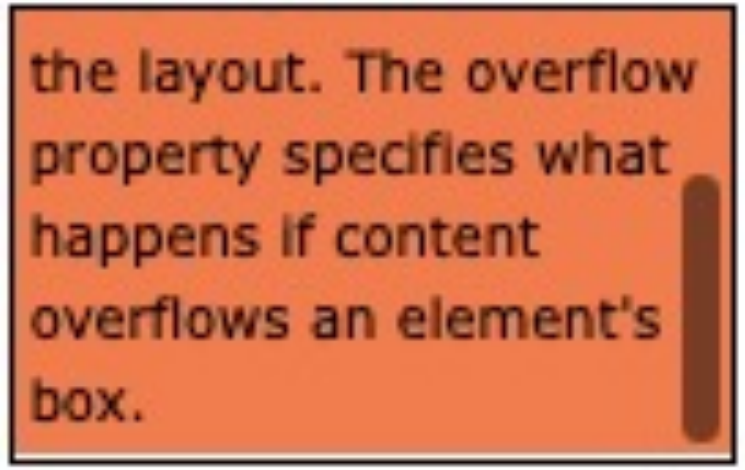
```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: coral;
  width: 200px;
  height: 65px;
  border: 1px solid black;
  overflow: hidden;
}
</style>
</head>
<body>

<h2>Overflow: hidden</h2>

<p>With the hidden value, the overflow is clipped, and the rest of the content is hidden:</p>
<p>Try to remove the overflow property to understand how it works.</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property
specifies what happens if content overflows an element's box.</div>

</body>
</html>
```



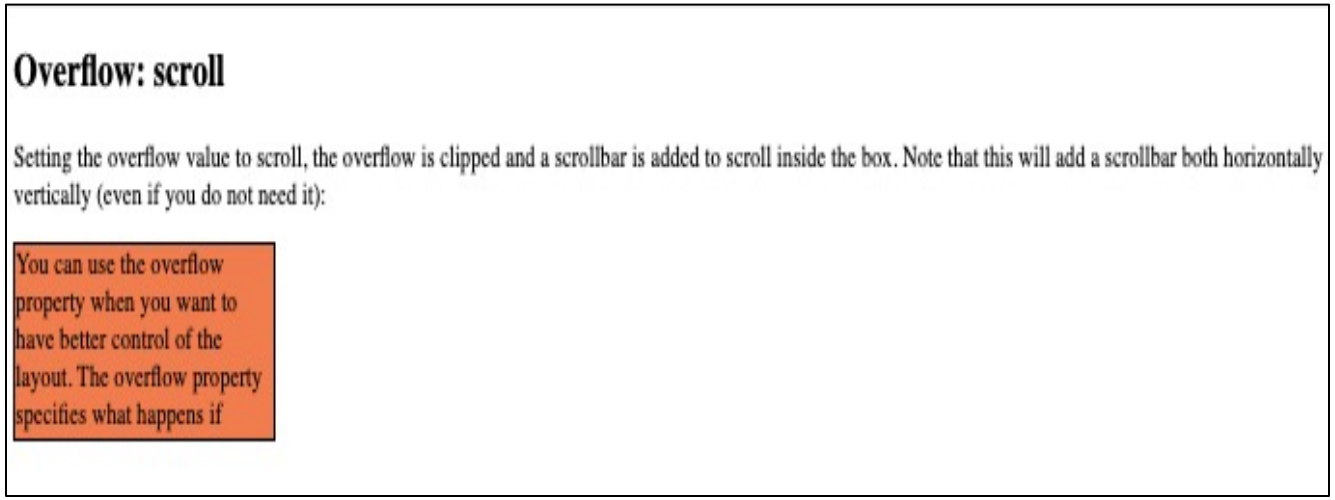
overflow_hidden.html

overflow: scroll

- Setting the value to scroll, the overflow is clipped and a scrollbar is added to scroll inside the box.
- Note that this will add a scrollbar both horizontally and vertically (even if you do not need it).



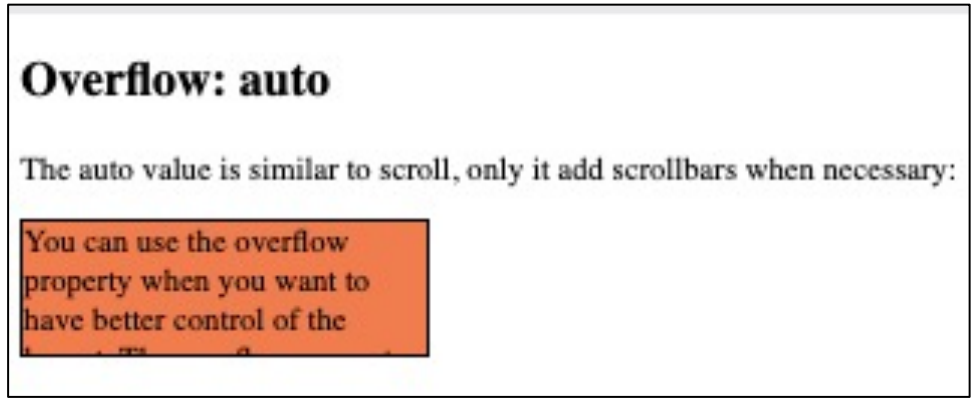
```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: coral;
  width: 200px;
  height: 100px;
  border: 1px solid black;
  overflow: scroll;
}
</style>
</head>
<body>

<h2>Overflow: scroll</h2>

<p>Setting the overflow value to scroll, the overflow is clipped and a scrollbar is added to scroll inside the box.
Note that this will add a scrollbar both horizontally and vertically (even if you do not need it):</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property
specifies what happens if content overflows an element's box.</div>

</body>
</html>
```



overflow_scroll.html

overflow: auto

- The auto value is similar to scroll, but it adds scrollbars only when necessary:




```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: coral;
  width: 200px;
  height: 65px;
  border: 1px solid black;
  overflow: auto;
}
</style>
</head>
<body>

<h2>Overflow: auto</h2>

<p>The auto value is similar to scroll, only it add scrollbars when necessary:</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property
specifies what happens if content overflows an element's box.</div>

</body>
</html>
```

overflow_auto.html

CSS-cursor

- CSS can generate a bunch of different mouse cursors.

CSS Syntax

cursor: *value*;

Example

Run css_cursor.html to see various types of cursors.

CSS white-space Property

- The white-space property specifies how white-space inside an element is handled.

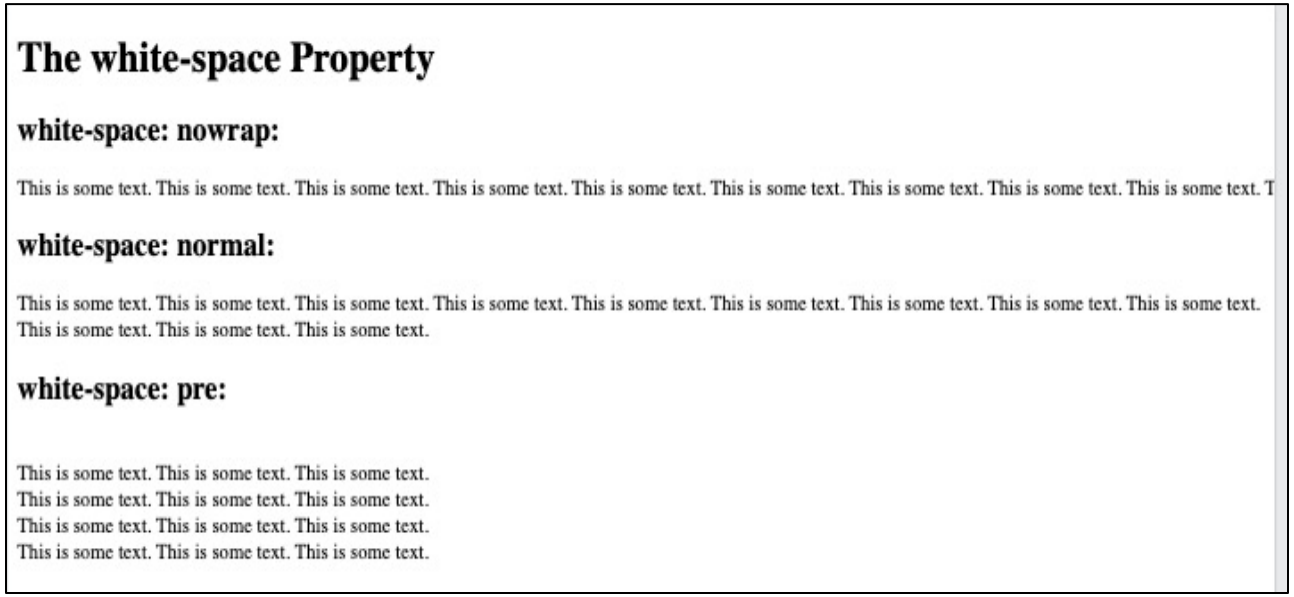
CSS Syntax

white-space: normal|nowrap|pre|pre-line|pre-wrap|initial|inherit;

CSS white-space Property

Value	Description
normal	Sequences of whitespace will collapse into a single whitespace. Text will wrap when necessary. This is default
nowrap	Sequences of whitespace will collapse into a single whitespace. Text will never wrap to the next line. The text continues on the same line until a tag is encountered
pre	Whitespace is preserved by the browser. Text will only wrap on line breaks. Acts like the <pre> tag in HTML
pre-line	Sequences of whitespace will collapse into a single whitespace. Text will wrap when necessary, and on line breaks
pre-wrap	Whitespace is preserved by the browser. Text will wrap when necessary, and on line breaks
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

```html
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
  white-space: nowrap;
}

p.b {
  white-space: normal;
}

p.c {
  white-space: pre;
}
</style>
</head>
<body>

<h1>The white-space Property</h1>

<h2>white-space: nowrap:</h2>
<p class="a">
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>

<h2>white-space: normal:</h2>
<p class="b">
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>

<h2>white-space: pre:</h2>
<p class="c">
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>

</body>
</html>
```



css_whitespace.html