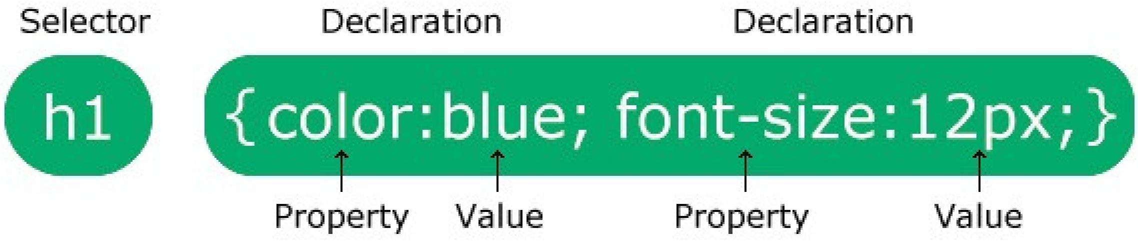
CSS Notes

CSS Syntax



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

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


<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>


</body>
```

```
</html>
```

Output :

Hello World!

These paragraphs are styled with CSS.

Example Explained

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

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

CSS Selectors

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)

This page will explain the most basic CSS selectors.

The CSS element Selector

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

```
<!DOCTYPE  html>

<html>

<head>

<style>
```

```
p {
text-align: center;
color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Output :

Every paragraph will be affected by the style.

Me too!

And me!

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

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

```
<!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>
```

Output :

```
                              Hello World!

This paragraph is not affected by the style.
```

The CSS class Selector

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

To select elements with a specific class, write a period (.) character, followed by the class name.

```
<!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>
```

Output :

```
                Red and center-aligned heading

                 Red and center-aligned paragraph.
```

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

```
<!DOCTYPE   html>

<html>

<head>

<style>

p.center {

text-align:    center;

color: red;

}

</style>

</head>
```

```
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>


</body>
</html>
```

Output:

This heading will not be affected

This paragraph will be red and center-aligned.

HTML elements can also refer to more than one class.

```
<!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>
```

Output :

This heading will not be affected

This paragraph will be red and center-aligned.

This paragraph will be red, center-aligned, and in a large font-size.

The CSS Universal Selector

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

```
<!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>
```

Output :

```
Hello world!

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

Me too!

And me!
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

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

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

```
<!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>
```

Output:

Hello World!

Smaller heading!

This is a paragraph.

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 <div> 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

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

```
<!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>
```

Output :

This is a heading

This is a paragraph.

An external style sheet can be written in any text editor, and must be saved with a .css extension.

The external .css file should not contain any HTML tags.

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

"mystyle.css"

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

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

Incorrect (space): margin-left: 20 px;
Correct (nospace): 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>
```

Output :

```
      This is a heading
This is a paragraph.
```

Inline CSS

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<!DOCTYPE html>

<html>

<body>


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

<p style="color:red;">This is a paragraph.</p>


</body>

</html>
```

Output :

This is a heading

This is a paragraph.

Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

Assume that an **external style sheet** has the following style for the <h1> element:

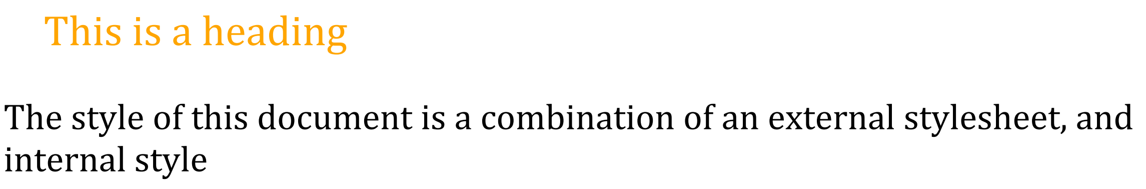
```
h1 {
color: navy;
}
```

Then, assume that an **internal style sheet** also has the following style for the <h1> element:

```
h1 {
color: orange;
}
```

```
<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="mystyle.css">

<style>

h1 {

color: orange;

}

</style>

</head>

<body>


<h1>This is a heading</h1>

<p>The style of this document is a combination of an external stylesheet, and internal
style</p>


</body>

</html>
```

Output :



This is a heading

The style of this document is a combination of an external stylesheet, and internal style

Example

If the internal style is defined **after** the link to the external style sheet, the <h1> elements will be "orange":

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
color: orange;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet, and internal style</p>

</body>
</html>
```

Output :

This is a heading

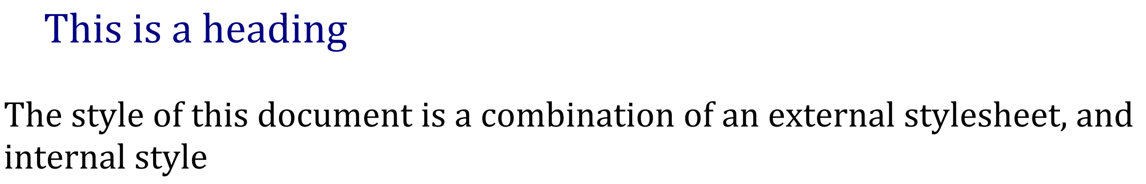
The style of this document is a combination of an external stylesheet, and internal style

Example

However, if the internal style is defined **before** the link to the external style sheet, the <h1> elements will be "navy":

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet, and internal style</p>

</body>
</html>
```

Output :



This is a heading

The style of this document is a combination of an external stylesheet, and internal style

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

Multiple Styles Will Cascade into One

Here, the background color of the page is set with inline CSS, and also with an internal CSS, and also with an external CSS.

Try experimenting by removing styles to see how the cascading stylesheets work (try removing the inline CSS first, then the internal, then the external).

CSS Comments

CSS comments are not displayed in the browser, but they can help document your source code.

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment is placed inside the <style> element, and starts/* and ends with */:

```
with <!DOCTYPE html>
<html>
<head>
<style>
/* This is a single-line comment */
p {
color: red;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>

</body>
</html>
```

Output :

```
Hello World!
```

This paragraph is styled with CSS.

CSS comments are not shown in the output.

You can add comments wherever you want in the code:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
color: red; /* Set text color to red */
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>

</body>
</html>
```

Comments can also span multiple lines:

Example

```
/* This is
a   multi-line
comment */

p {
color: red; }
```

HTML and CSS Comments

From the HTML tutorial, you learned that you can add comments to your HTML source by using the <!--...--> syntax.

In the following example, we use a combination of HTML and CSS comments:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
color: red; /* Set text color to red */
}
</style>
</head>
<body>

<h2>My Heading</h2>

<!-- These paragraphs will be red -->
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>

</body>
</html>
```

CSS Colors

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

CSS Color Names

In CSS, a color can be specified by using a predefined color name:

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

<h1 style="background-color:Tomato;">Tomato</h1>
<h1 style="background-color:Orange;">Orange</h1>
<h1 style="background-color:DodgerBlue;">DodgerBlue</h1>
<h1  style="background-color:MediumSeaGreen;">MediumSeaGreen</h1>
<h1 style="background-color:Gray;">Gray</h1>
<h1 style="background-color:SlateBlue;">SlateBlue</h1>
<h1 style="background-color:Violet;">Violet</h1>
<h1 style="background-color:LightGray;">LightGray</h1>

</body>
</html>
```

Tomato

Orange

DodgerBlue

MediumSeaGreen

Gray

SlateBlue

Violet

LightGray

CSS Background Color

You can set the background color for HTML elements:

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

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

```
<!DOCTYPE html>

<html>

<body>
```

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

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

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
tincidunt ut laoreet dolore magna aliquam erat volutpat.
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut
aliquip ex ea commodo consequat.
</p>


</body>

</html>
```

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

CSS Text Color

You can set the color of text:

Hello World

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

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

```
<!DOCTYPE html>

<html>

<body>



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



<p style="color:DodgerBlue;">Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>


<p style="color:MediumSeaGreen;">Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>


</body>

</html>
```

Hello World

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

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

CSS Border Color

You can set the color of borders:

Hello World

Hello World

Hello World

```
<!DOCTYPE html>

<html>

<body>

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

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

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

</body>

</html>
```

CSS Color Values

In CSS, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Same as color name "Tomato":

```
<!DOCTYPE html>

<html>

<body>



<p>Same as color name "Tomato":</p>



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

```
<h1 style="background-color:#ff6347;">#ff6347</h1>

<h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1>


<p>Same as color name "Tomato", but 50% transparent:</p>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h1>


<p>In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.</p>


</body>

</html>
```

CSS Backgrounds

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.

```
<!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>
```

Hello World!

This page has a light blue background color!

With CSS, a color is most often specified by:

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

Other Elements

You can set the background color for any HTML elements:

Example

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

```
h1 {
background-color: green;
}

div {
 background-color:  lightblue;
}

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

Opacity / Transparency

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

```
<!DOCTYPE html>

<html>

<head>

<style>

div {

background-color: green; }



div.first {

opacity: 0.1;

}



div.second {

opacity: 0.3;

}



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

<h1>Transparent Boxes</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">
<h1>opacity 0.1</h1>
</div>

<div class="second">
<h1>opacity 0.3</h1>
</div>

<div class="third">
<h1>opacity 0.6</h1>
</div>
```

```
<div>

<h1>opacity  1  (default)</h1>

</div>



</body>

</html>
```

Transparency using RGBA

If you do not want to apply opacity to child elements, like in our example above, use **RGBA** color values. The following example sets the opacity for the background color and not the text:

You learned from our CSS Colors Chapter, that you can use RGB as a color value. In addition to RGB, you can use an RGB color value with an **alpha** channel (RGBA) - which specifies the opacity for a color.

An RGBA color value is specified with: rgba(red, green, blue, *alpha*). The *alpha* parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

Tip: You will learn more about RGBA Colors in our CSS Colors Chapter.

Example

```
div

{
  background: rgba(0, 128, 0, 0.3) /* Green background with 30% opacity */
}
```

CSS background-image

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

Example

Set the background image for a page:

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

```
<!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>
```

Hello World!

This page has an image as the background!

CSS background-repeat

By default, the`background-image` property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

```
<!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>
```

Hello World!

Strange background image...

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

Example

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

CSS background-repeat: no-repeat

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

Example

Show the background image only once:

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

CSS background-position

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

Example

Position the background image in the top-right corner:

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

```
<!DOCTYPE html>

<html>

<head>

<style>

body {

  background-image: url("img_tree.png");

background-repeat: no-repeat;

background-position: right top;

margin-right: 200px;

}

</style>
```

```
</head>

<body>


<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>
```

Hello World!

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

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

CSS background-attachment

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

Example

Specify that the background image should be fixed:

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

```
<!DOCTYPE html>
```

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

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

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

<p><strong>Tip:</strong> If you do not see any scrollbars, try to resize
the browser window.</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>

</body>
</html>
```

Example

Specify that the background image should scroll with the rest of the page:

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

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

I have borders on all sides.

I have a red bottom border.

I have rounded borders.

I have a blue left border.

CSS Border Style

The border-style property specifies what kind of border to display.

The following values are allowed:

- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- groove - Defines a 3D grooved border. The effect depends on the border-color value
- ridge - Defines a 3D ridged border. The effect depends on the border-color value
- inset - Defines a 3D inset border. The effect depends on the border-color value
- outset - Defines a 3D outset border. The effect depends on the border-color value
- none - Defines no border
- hidden - Defines a hidden border

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

Example

Demonstration of the different border styles:

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

```
<!DOCTYPE html>

<html>

<head>

<style>

p.dotted {border-style: dotted;}

p.dashed {border-style: dashed;}

p.solid {border-style: solid;}

p.double {border-style: double;}

p.groove {border-style: groove;}

p.ridge {border-style: ridge;}

p.inset {border-style: inset;}

p.outset {border-style: outset;}

p.none {border-style: none;}

p.hidden {border-style: hidden;}

p.mix {border-style: dotted dashed solid double;}

</style>

</head>
```

```
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>

</body>
</html>
```

CSS Border Width

The border-width property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick:

Example

Demonstration of the different border widths:

```
p.one {
  border-style: solid;
  border-width: 5px;
}

p.two {
border-style: solid;
  border-width: medium;
}

p.three {
  border-style: dotted;
border-width: 2px;
}

p.four {
  border-style: dotted;
border-width: thick;
}
```

Result:

5px border-width

medium border-width

2px border-width

thick border-width

Specific Side Widths

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

Example

```
p.one {
border-style: solid;
  border-width: 5px 20px; /* 5px top and bottom, 20px on the sides */
}

p.two {
border-style: solid;
```

```
 border-width: 20px 5px; /* 20px top and bottom, 5px on the sides */
}

p.three {
border-style: solid;
  border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom
and 35px left */
}
```

CSS Border Color

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

The color can be set by:

□ name - specify a color name, like "red"
□ HEX - specify a HEX value, like "#ff0000"
□ RGB - specify a RGB value, like "rgb(255,0,0)"
□ HSL - specify a HSL value, like "hsl(0, 100%, 50%)"
□ transparent

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

Example

Demonstration of the different border colors:

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

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

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

Result:

Red border

Green border

Blue border

Specific Side Colors

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

Example

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

CSS Border - Individual Sides

From the examples on the previous pages, you have seen that it is possible to specify a different border for each side.

In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left):

Example

```
p {
border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
border-left-style: solid;
}
```

Result:

Different Border Styles

The example above gives the same result as this:

Example

```
p {
  border-style: dotted solid;
}
```

So, here is how it works:

If the border-style property has four values:

□ **border-style: dotted solid double dashed;**

o top border is dotted
o right border is solid
o bottom border is double
o left border is dashed

If the border-style property has three values:

□ **border-style: dotted solid double;**

o top border is dotted
o right and left borders are solid
o bottom border is double

If the border-style property has two values:

□ **border-style: dotted solid;**

o top and bottom borders are dotted
o right and left borders are solid

If the border-style property has one value:

□ **border-style: dotted;**

o all four borders are dotted

Example

```
/* Four values */
p {
  border-style: dotted solid double dashed;
}

/* Three values */
p {
  border-style: dotted solid double;
}

/* Two values */
p {
border-style: dotted solid;
}

/* One value */
p {
border-style: dotted;
}
```

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

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

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

ADVERTISEMENT

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`

So, here is how it works:

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

Example

Use the margin shorthand property with four values:

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

If the margin property has three values:

 margin: 25px 50px 75px;

o top margin is 25px
o right and left margins are 50px
o bottom margin is 75px

Example

Use the margin shorthand property with three values:

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

If the margin property has two values:

 margin: 25px 50px;

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

Example

Use the margin shorthand property with two values:

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

If the `margin` property has one value:

□margin: 25px;

- o all four margins are 25px

Example

Use the margin shorthand property with one value:

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

CSS Padding

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

This element has a padding of 70px.

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).

Padding - Individual Sides

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

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

All the padding properties can have the following values:

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

Note: Negative values are not allowed.

Example

Set different padding for all four sides of a <div> element:

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

ADVERTISEMENT

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

So, here is how it works:

If the padding property has four values:

□ padding: 25px 50px 75px 100px;

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

Example

Use the padding shorthand property with four values:

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

If the padding property has three values:

□ padding: 25px 50px 75px;

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

Example

Use the padding shorthand property with three values:

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

If the padding property has two values:

□ padding: 25px 50px;

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

Example

Use the padding shorthand property with two values:

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

If the padding property has one value:

- padding: 25px;
    - all four paddings are 25px

Example

Use the padding shorthand property with one value:

```
div {
  padding: 25px;
}
```

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...

Example

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

ADVERTISEMENT

Example - float: left;

The following example specifies that an image should float to the **left** 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...

Example

```
img {
  float: left;
}
```

Example - No float

In the following example the image will be displayed just where it occurs in the text (float: none;):

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...

Example

```
img {
  float: none;
}
```

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.
Note: In OS X Lion (on Mac), scrollbars are hidden by default and only shown when being used (even though "overflow:scroll" is set).

overflow: visible

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

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.

Example

```
div {
width: 200px;
height: 65px;
  background-color: coral;
overflow: visible;
}
```

overflow: hidden

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

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.

Example

```
div {
  overflow: hidden;
}
```

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):

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.

Example

```
div {
  overflow: scroll;
}
```

overflow: auto

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

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.

Example

```
div {
  overflow: auto;
}
```

overflow-x and overflow-y

The overflow-x and overflow-y properties specifies whether to change the overflow of content just horizontally or vertically (or both):

overflow-x specifies what to do with the left/right edges of the content.

overflow-y specifies what to do with the top/bottom edges of the content.

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.

Example

```
div {
  overflow-x: hidden; /* Hide horizontal scrollbar */
overflow-y: scroll; /* Add vertical scrollbar */
}
```