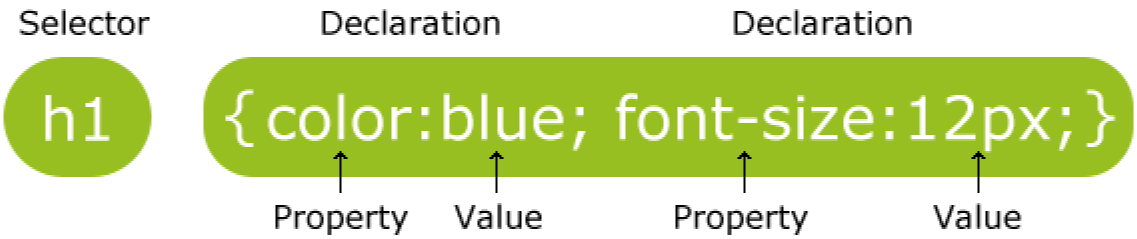
What is CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets
- Styles define **how to display** HTML elements
- Styles were added to HTML 4.0 **to solve a problem**
- **External Style Sheets** can save a lot of work
- External Style Sheets are stored in **CSS files**

CSS Syntax

A CSS rule set consists of a selector and a declaration block:



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

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

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

CSS Comments

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

A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

Example

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

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

CSS Selectors

CSS selectors allow you to select and manipulate HTML element(s).

CSS selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more.

The element Selector

The element selector selects elements based on the element name.

You can select all <p> elements on a page like this: (all <p> elements will be center-aligned, with a red text color)

Example

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

The id Selector

The id selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the element.

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

Example

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

The class Selector

The class selector finds elements with the specific class.

The class selector uses the HTML class attribute.

To find elements with a specific class, write a period character, followed by the name of the class:

In the example below, all HTML elements with class="center" will be center-aligned:

Example

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

Grouping Selectors

In style sheets there are often elements with the same style:

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

h2 {
    text-align: center;
    color: red;
}

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

To minimize the code, you can group selectors.

To group selectors, separate each selector with a comma.

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

Example

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

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- **External style sheet**
- **Internal style sheet**
- **Inline style**

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing just one file.

Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:

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

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file is shown below:

"myStyle.css":

```
hr {color: sienna;}
p {margin-left: 20px;}
body {background-image: url("images/background.gif");}
```

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the <style> tag, like this:

```
<head>
<style>
hr {color: sienna;}
p {margin-left: 20px;}
body {background-image: url("images/background.gif");}
</style>
</head>
```

Inline Styles

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

To use inline styles, add the style attribute to the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

```
<p style="color:sienna;margin-left:20px;">This is a paragraph.</p>
```

All CSS Background Properties

Property	Description
background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated

All CSS Text Properties

Property	Description
color	Sets the color of text
direction	Specifies the text direction/writing direction
letter-spacing	Increases or decreases the space between characters in a text
line-height	Sets the line height
text-align	Specifies the horizontal alignment of text
text-decoration	Specifies the decoration added to text
text-indent	Specifies the indentation of the first line in a text-block
text-shadow	Specifies the shadow effect added to text
text-transform	Controls the capitalization of text
unicode-bidi	Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document

vertical-align	Sets the vertical alignment of an element
white-space	Specifies how white-space inside an element is handled
word-spacing	Increases or decreases the space between words in a text

All CSS Font Properties

Property	Description
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font

Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

In addition, links can be styled differently depending on what **state** they are in.

The four links states are:

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

Example

```
/* unvisited link */
a:link {
    color: #FF0000;
}

/* visited link */
a:visited {
    color: #00FF00;
}

/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
```

You can also put all the other style elements Like Text-Decoration ,Font Style Etc.

List

In HTML, there are two types of lists:

- unordered lists - the list items are marked with bullets
- ordered lists - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

All CSS List Properties

Property	Description
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies if the list-item markers should appear inside or outside the content flow
list-style-type	Specifies the type of list-item marker

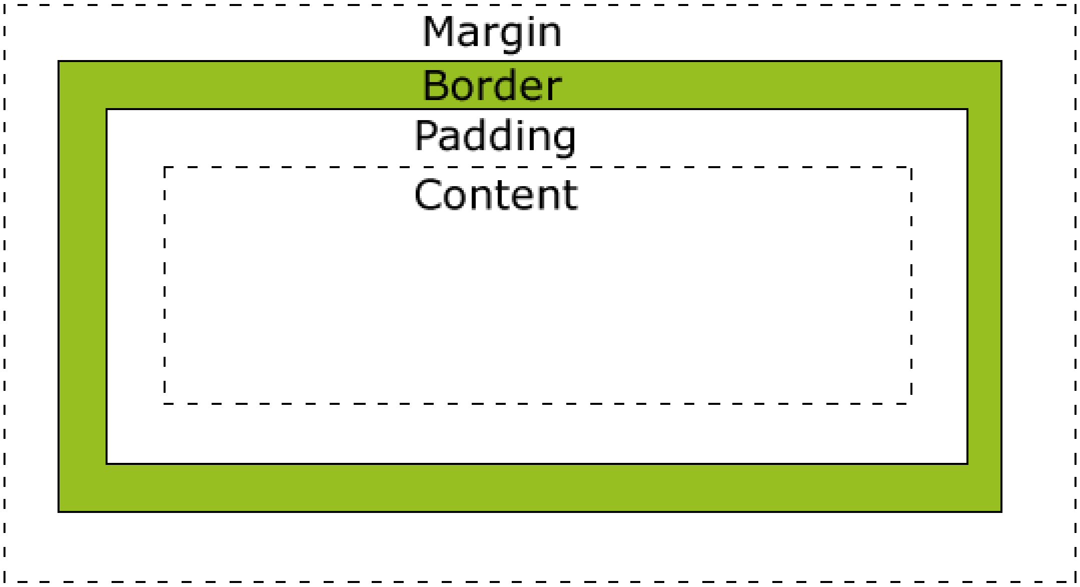
The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to place a border around elements and space elements in relation to other elements.

The image below illustrates the box model:



Explanation of the different parts:

- **Margin** - Clears an area around the border. The margin does not have a background color, it is completely transparent
- **Border** - A border that goes around the padding and content. The border is inherited from the color property of the box
- **Padding** - Clears an area around the content. The padding is affected by the background color of the box
- **Content** - The content of the box, where text and images appear

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

All CSS Dimension Properties

Property	Description	Values
height	Sets the height of an element	auto *length* *%* inherit
max-height	Sets the maximum height of an element	none *length* *%* inherit
max-width	Sets the maximum width of an element	none *length* *%* inherit

min-height	Sets the minimum height of an element	*length* *%* inherit
min-width	Sets the minimum width of an element	*length* *%* inherit
width	Sets the width of an element	auto *length* *%* inherit

CSS Pseudo-elements

CSS pseudo-elements are used to add special effects to some selectors.

The ::first-line Pseudo-element

The ::first-line pseudo-element is used to add a special style to the first line of a text.

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

Example

Format the first line of the text in p elements:

```
p::first-line {
    color: #ff0000;
    font-variant: small-caps;
}
```

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

The ::first-letter Pseudo-element

The ::first-letter pseudo-element is used to add a special style to the first letter of a text.

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

Example

Format the first letter of the text in p elements:

```
p::first-letter {
    color: #ff0000;
    font-size: xx-large;
}
```

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 each <h1> element:

Example

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

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 each <h1> element:

Example

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

All CSS Pseudo Classes/Elements

Selector	Example	Example description
:link	a:link	Selects all unvisited links
:visited	a:visited	Selects all visited links
:active	a:active	Selects the active link
:hover	a:hover	Selects links on mouse over
:focus	input:focus	Selects the input element which has focus
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
::before	p::before	Insert content before every <p> element
::after	p::after	Insert content after every <p> element
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"

CSS Float

Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.
A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.
The elements after the floating element will flow around it.
The elements before the floating element will not be affected.
If an image is floated to the right, a following text flows around it, to the left:

```
<!DOCTYPE html>
```

Example

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

<p>In the paragraph below, we have added an image with style <b>float:right</b>. The result is that the image will float to the right in the paragraph.</p>
<p><img src="anyimg.jpg" width="100" height="140">
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.
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.
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.
This is some text. This is some text. This is some text.
</p>

</body>
</html>
```

CSS Image Opacity / Transparency

Creating a Transparent Image

The CSS3 property for transparency is **opacity**.

First we will show you how to create a transparent image with CSS.

Regular image:

The same image with transparency:

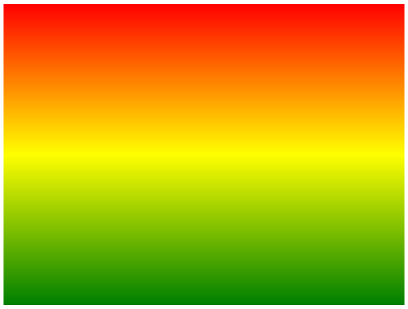
Look at the following CSS:

```
img {
    opacity: 0.4;
    filter: alpha(opacity=40); /* For IE8 and earlier */
}
```

CSS3 Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

Example of Linear Gradient:



Syntax

```
background: linear-gradient(direction, color-stop1, color-stop2, ...);
```

Linear Gradient - Top to Bottom (this is default)

The following example shows a linear gradient that starts at the top. It starts red, transitioning to blue:

Example

A linear gradient from top to bottom:

```
#grad {
  background: -webkit-linear-gradient(red, blue); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(red, blue); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(red, blue); /* For Firefox 3.6 to 15 */
  background: linear-gradient(red, blue); /* Standard syntax */
}
```

Linear Gradient - Left to Right

The following example shows a linear gradient that starts from the left. It starts red, transitioning to blue:

Example

A linear gradient from left to right:

```
#grad {
  background: -webkit-linear-gradient(left, red , blue); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(right, red, blue); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(right, red, blue); /* For Firefox 3.6 to 15 */
  background: linear-gradient(to right, red , blue); /* Standard syntax */
}
```

Linear Gradient - Diagonal

You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.

The following example shows a linear gradient that starts at top left (and goes to bottom right). It starts red, transitioning to blue:

Example

A linear gradient that starts at top left (and goes to bottom right):

```
#grad {
  background: -webkit-linear-gradient(left top, red , blue); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(bottom right, red, blue); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(bottom right, red, blue); /* For Firefox 3.6 to 15 */
  background: linear-gradient(to bottom right, red , blue); /* Standard syntax */
}
```

Using Transparency

CSS3 gradients also support transparency, which can be used to create fading effects.

To add transparency, we use the rgba() function to define the color stops. The last parameter in the rgba() function can be a value from 0 to 1, and it defines the transparency of the color: 0 indicates full transparency, 1 indicates full color (no transparency).

The following example shows a linear gradient that starts from the left. It starts fully transparent, transitioning to full color red:

Example

A linear gradient from left to right, with transparency:

```
#grad {
  background: -webkit-linear-gradient(left,rgba(255,0,0,0),rgba(255,0,0,1)); /*Safari 5.1-6*/
  background: -o-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /*Opera 11.1-12*/
  background: -moz-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /*Fx 3.6-15*/
  background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1)); /*Standard*/
}
```

rowser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property	e				O
column-count	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1
column-gap	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1
column-rule	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS3 Multiple Columns

CSS3 Multiple Columns

With CSS3, you can create multiple columns for laying out text - like in newspapers!

In this chapter you will learn about the following multiple column properties:

- column-count
- column-gap
- column-rule

Example

Specify the width, style and color of the rule between columns:

```
<!DOCTYPE html>
<html>
<head>
<style>
.newspaper {
   -webkit-column-count: 3; /* Chrome, Safari, Opera */
   -moz-column-count: 3; /* Firefox */
   column-count: 3;
   -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
   -moz-column-gap: 40px; /* Firefox */
   column-gap: 40px;
   -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
   -moz-column-rule: 4px outset #ff00ff; /* Firefox */
   column-rule: 4px outset #ff00ff;
}
</style>
</head>
<body>

<p><b>Note:</b> Internet Explorer 9, and earlier versions, does not support the column-rule property.</p>
```

```
<div class="newspaper">
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.
Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.
</div>

</body>
</html>
```

CSS3 Multiple Columns Properties

The following table lists all the multiple columns properties:

Property	Description	CSS
column-count	Specifies the number of columns an element should be divided into	3
column-fill	Specifies how to fill columns	3
column-gap	Specifies the gap between the columns	3
column-rule	A shorthand property for setting all the column-rule-* properties	3
column-rule-color	Specifies the color of the rule between columns	3
column-rule-style	Specifies the style of the rule between columns	3
column-rule-width	Specifies the width of the rule between columns	3
column-span	Specifies how many columns an element should span across	3
column-width	Specifies the width of the columns	3
columns	A shorthand property for setting column-width and column-count	3

What Are CSS3 Transitions?

CSS3 transitions are effects that let an element gradually change from one style to another.

To do this, you must specify two things:

- the CSS property you want to add an effect to
- the duration of the effect

Note: If the duration part is not specified, the transition will have no effect, because the default value is 0.

The transition effect will start when the specified CSS property changes value. A typical CSS property change would be when a user mouse-over an element:

<style>

UNDER CSS

```
div {
   width: 100px;
   height: 100px;
   background: red;
   -webkit-transition: width 2s; /* For Safari 3.1 to 6.0 */
   transition: width 2s;
}

div:hover {
   width: 300px;
}
```

UNDER HTML BODY

```
<p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>

<div></div>

<p>Hover over the div element above, to see the transition effect.</p>
```

Multiple Changes

To add transition effects for more than one CSS property, separate the properties with a comma:

UNDER CSS

```
<style>
```

```
div {
   width: 100px;
   height: 100px;
   background: red;
   -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /* For Safari 3.1 to 6.0 */
   transition: width 2s, height 2s, transform 2s;
}

div:hover {
   width: 200px;
   height: 200px;
   -webkit-transform: rotate(180deg); /* Chrome, Safari, Opera */
   transform: rotate(180deg);
}
</style>
```

UNDER HTML BODY

```
<p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>

<div>Hover over me to see the transition effect!</div>
```