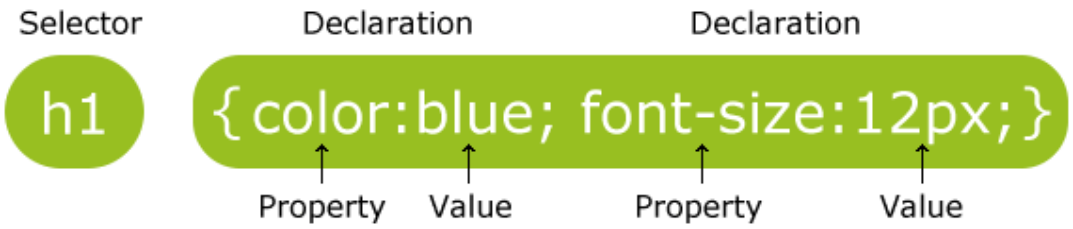
What is CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External 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 CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

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

Example

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

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

Example

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

CSS Selectors

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and 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 (in this case, 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 element to select a specific element.
- An id should be unique within a page, so the id selector is used if you want to select a single, unique element.
- To select 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;
}
```

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>
```

Result:

Hello World!

This paragraph is not affected by the style.

The class Selector

- The class selector selects elements with a specific class attribute.
- To select 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 red and center-aligned:

Example

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

Example:

```
<!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>
```

Result:

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.

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

Example

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

Example:

```
<!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>
```

Result:

This heading will not be affected

This paragraph will be red and center-aligned.

Grouping Selectors:

If you have elements with the same style definitions, like this:

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

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

p {
  text-align: center;
```

```
    color: red;
}
```

- You can group the selectors, to minimize the code.
- 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;
}
```

Example:

```
<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>
```

Result:

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

Inline Styles

- 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.**

The example below shows how to change the color and the left margin of a <h1> element:

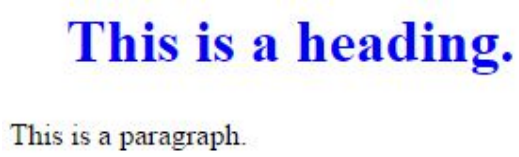
Example

```
<h1 style="color:blue; margin-left:30px;">This is a heading.</h1>
```

Example:

```
<html>
<body>

<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Result;



This is a paragraph.

Internal Style Sheet

- An internal style sheet may be used if one single page has a unique style.
- Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

Example

```
<head>
<style>
body {
  background-color: linen;
}

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

Example:

```
<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>
```

Result:

External Style Sheet

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

Example

```
<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.

Here is how the "**myStyle.css**" looks:

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

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

Example:

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

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

</body>
</html>
```

Result:

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.

Example

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;
}
```

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

Cascading Order

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

Generally speaking we can say that all the styles 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 (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value**.**

CSS Margin Properties

- The CSS margin properties are used to generate space around elements.
- The margin properties set the size of the white space OUTSIDE the border.

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.

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

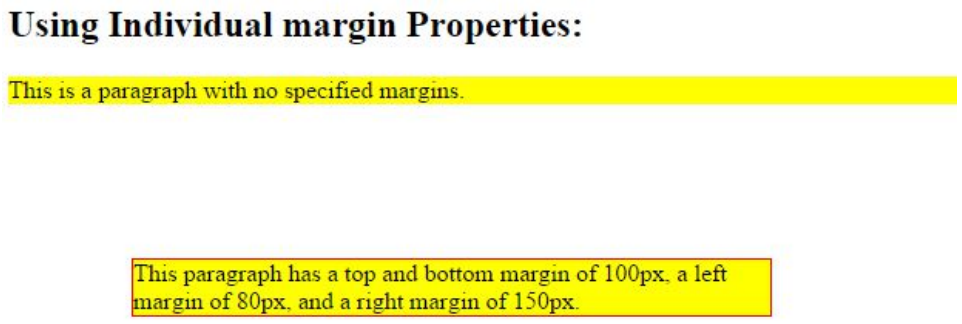
Example

```
p {
    margin: 100px 150px 100px 80px;
}
```

Example on Margin Properties:

```
<html>
<head>
<style>
p {
    background-color: yellow;
}

p.ex {
    border:1px solid red;
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
}
</style>
</head>
<body>

<h2>Using Individual margin Properties:</h2>

<p>This is a paragraph with no specified margins.</p>
<p class="ex">This paragraph has a top and bottom margin of 100px, a left
margin of 80px, and a right margin of 150px.</p>
</body>
</html>
```

Result:



CSS Padding Properties

- The CSS padding properties are used to generate space around content.
- The padding properties set the size of the white space between the element content and the element border.

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

The following example sets different padding for all four sides of a <p> element:

```
<html>
<head>
<style>
p.one {
    border: 1px solid red;
    background-color: yellow;
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
</style>
</head>
<body>

<h2>Using Individual padding Properties:</h2>

<p>This is a paragraph with no specified padding.</p>
<p class="one">This paragraph has a top and bottom padding of 50px, a left
padding of 80px, and a right padding of 30px.</p>

</body>
</html>
```

Result:

Using Individual padding Properties:

This is a paragraph with no specified padding.

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

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

Example

```
p {
    padding: 50px 30px 50px 80px;
}
```

Background Properties:

Property	Description	Possible Values	Examples
background-attachment	Declares the attachment of a background image (to scroll with the page content or be in a fixed position).	*fixed* *scroll*	*div { background-attachment:fixed; }* *div { background-attachment:scroll; }*
background-color	Declares the background color.	Valid color names, RGB values, hexidecimal notation.	*div { background-color:green; }* *div { color:#00FF00; }*
background-image	Declares the background image of an element.	URL values.	*div { background-image:url(images/img.jpg); }* *body { background-image:url(img.jpg); }*
background-position	Declares the position of a background image.	Lengths or percentages for the x and y positions, or one of the predefined values: *top left* *top center* *top right* *center left* *center center* *center right* *bottom left* *bottom center* *bottom right*	*div { background-position:10px 50px; }* *div { background-position:bottom right; }*
background-repeat	Declares how and/or if a background image repeats.	*repeat* *repeat-x* *repeat-y* *no-repeat*	*div { background-repeat:repeat-x; }* *div { background-repeat:no-repeat; }*

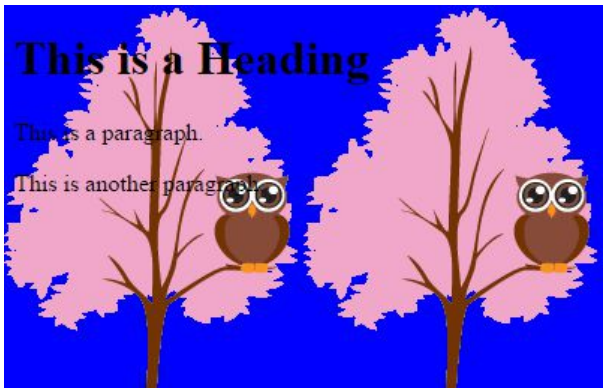
background	Used as a shorthand property to set all the background properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): *background-color* *background-image* *background-repeat* *background-attachment* *background-position*	*div { background:green url(image.jpg) no-repeat fixed center center; }* *div { background:url(image.jpg) fixed; }*

CSS Background Properties Example:

```
<html>
<head>
<style>
body {
    background-image: url("img_tree.png");
    background-color: blue;
    background-repeat: repeat-x;
    background-position: fixed;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
```

</html>

Result:



Border Properties:

Property	Description	Possible Values	Examples

border-top-color	Declares the color of the top border.	Valid color names, RGB values, hexidecimal notation, or the predefined value **`transparent`**.	`div { border-top-color:green; }`
			
			`div { border-top-color:#00FF00; }`

border-top-style	Declares the style of the top border.	`none`
		`hidden`
		`dotted`
		`dashed`
		`solid`
		`double`
		`groove`
		`ridge`
		`inset`
		`outset`
			`div { border-top-style:solid; }`
			
			`div { border-top-style:inset; }`

border-top-width	Declares the width of the top border.	Lengths or the following predefined values:
		
		`thin`
		`medium`
		`thick`
			`div { border-top-width:2px; }`
			
			`div { border-top-width:thin; }`

border-top	Used as a shorthand property to set all the border-top properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values):
		
		`border-top-width`
		`border-top-style`
		`border-top-color`
			`div { border-top:2px solid green; }`
			
			`div { border-top:thick double #00FF00; }`

border-right-color	Declares the color of the right border.	Valid color names, RGB values, hexidecimal notation, or the predefined value transparent.	`div { border-right-color:green; }` `div { border-right-color:#00FF00; }`
border-right-style	Declares the style of the right border.	none hidden dotted dashed solid double groove ridge inset outset	`div { border-right-style:solid; }` `div { border-right-style:inset; }`
border-right-width	Declares the width of the right border.	Lengths or the following predefined values: thin medium thick	`div { border-right-width:2px; }` `div { border-right-`

			width:thin; }
border-right	Used as a shorthand property to set all the border-right properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-right-width border-right-style border-right-color	*div { border-right:2px solid green; }* *div { border-right:thick double #00FF00; }*
border-bottom-color	Declares the color of the bottom border.	Valid color names, RGB values, hexidecimal notation, or the predefined value transparent.	*div { border-bottom-color:green; }* *div { border-bottom-color:#00FF00; }*
border-bottom-style	Declares the style of the bottom border.	none hidden dotted dashed solid double groove ridge inset outset	*div { border-bottom-style:solid; }* *div { border-bottom-style:inset; }*
border-bottom-width	Declares the width of the bottom border.	Lengths or the following predefined values: thin medium thick	*div { border-bottom-width:2px; }* *div { border-bottom-width:thin; }*
border-bottom	Used as a shorthand property to set all the border-bottom properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-bottom-width border-bottom-style border-bottom-color	*div { border-bottom:2px solid green; }* *div { border-bottom:thick double #00FF00; }*
border-left-color	Declares the color of the left border.	Valid color names, RGB values, hexidecimal notation, or the predefined value transparent.	*div { border-left-color:green; }* *div { border-left-color:#00FF00; }*
border-left-style	Declares the style of the left border.	none hidden dotted dashed solid double groove ridge inset outset	*div { border-left-style:solid; }* *div { border-left-style:inset; }*
border-left-width	Declares the width of the left border.	Lengths or the following predefined values: thin	*div { border-left-width:2px; }*

		medium thick	`div { border-left-width:thin; }`
border-left	Used as a shorthand property to set all the border-left properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-left-width border-left-style border-left-color	`div { border-left:2px solid green; }` `div { border-left:thick double #00FF00; }`

border-color	Declares the border color of all four borders at once.	Valid color names, RGB values, hexidecimal notation, or the predefined value transparent.
		Separate the color for each border by a space, declaring the colors for the borders in the following order:
		border-top-color
		border-right-color
		border-bottom-color
		border-left-color
		Undeclared values work as further shorthand notation. If only one color value is declared, all four borders will use that color. If two colors are declared, the top and bottom borders will use the first color while the right and left borders will use the second color. If three colors are declared, the top border will use the first color, the right and left borders will use the second color, and the bottom border will use the third color.
			`div { border-color:green red blue olive; }`
			`div { border-color:green; }`
			`div { border-color:green red; }`
			`div { border-color:green red blue; }`

border-style	Declares the border style of all four borders at once.	none
		hidden
		dotted
		dashed
		solid
		double
		groove
		ridge
		inset
		outset
		Undeclared values work as further shorthand notation. If only one style value is declared, all four borders will use that style. If two styles are declared, the top and bottom borders will use the first style while the right and left borders will use the second style. If three styles are declared, the top border will use the first style, the right and left borders will use the second style, and the bottom border will use the third style.
			`div { border-style:solid dotted dashed double; }`
			`div { border-style:solid; }`
			`div { border-style:solid dotted; }`
			`div { border-style:solid dotted dashed; }`

border-width	Declares the width of all four borders at once.	Lengths or the following predefined values:
		thin
		medium
		thick
		Undeclared values work as further
			`div { border-width:1px 3px 5px 2px; }`
			`div { border-width:thin; }`
			`div { border-`

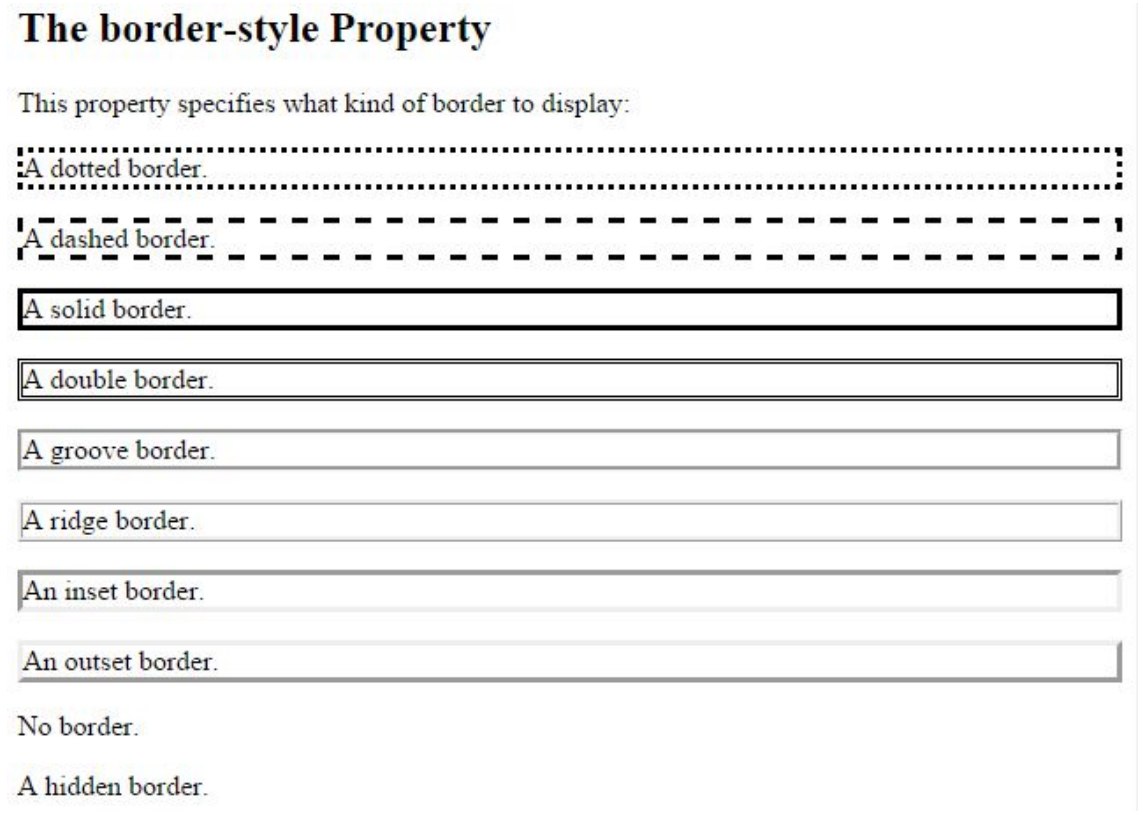
		shorthand notation. If only one width value is declared, all four borders will use that width. If two widths are declared, the top and bottom borders will use the first width while the right and left borders will use the second width. If three widths are declared, the top border will use the first width, the right and left borders will use the second width, and the bottom border will use the third width.	*width:2px 4px; }* *div { border-width:2px 4px 5px; }*
border	Used as a shorthand to declare the border properties when all four borders will have the same appearance.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): border-width border-style border-color	*div { border:1px double green; }* *div { border:thin solid #00FF00; }*

CSS Border Properties Example:

```
<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;    }

</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>

</body> </html>
```

Result:



Dimension Properties:

Property	Description	Possible Values	Examples
height	Declares the height of the element.	Lengths, percentages, and the predefined value **auto**.	*div { height:200px; }* *div { height:50%; }*

max-height	Declares the maximum height of the element.	Lengths, percentages, and the predefined value **auto**.	*div { max-height:200px; }* *div { max-height:50%; }*
min-height	Declares the minimum height of the element.	Lengths, percentages, and the predefined value **auto**.	*div { min-height:200px; }* *div { min-height:50%; }*
width	Declares the width of the element.	Lengths, percentages, and the predefined value **auto**.	*div { width:500px; }* *div { width:75%; }*
max-width	Declares the maximum width of the element.	Lengths, percentages, and the predefined value **auto**.	*div { max-width:500px; }* *div { max-width:75%; }*

Font Properties:

Property	Description	Possible Values	Examples
font-family	Declares the name of the font to be used. Previously set in HTML via the *face* attribute in a <font> tag.	Valid font family names or generic family names, i.e. *Arial*, *Verdana*, *sans-serif*, *"Times New Roman"*, *Times*, *serif*, etc. Font family names can be separated by a comma in the same declaration to allow additional and/or generic family names to be used if the prefereed font is unable to be displayed.	*div { font-family:Arial; }* *div { font-family:Arial, Helvetica, sans-serif; }*
font-size	Declares the size of the font. Previously set in HTML via the *size* attribute in a <font> tag.	Lengths (number and unit type—i.e. *1em*, *12pt*,*10px*, *80%*) or one of the following predefined values: *xx-small* *x-small* *small* *medium* *large* *x-large* *xx-large* *smaller* *larger*	*div { font-size:70%; }* *div { font-size:0.85em; }* *div { font-size:medium; }*
font-size-adjust	**Limited browser support: Was part of CSS 2, but not in CSS 2.1. This property may return in CSS 3.**	Numeric value	*div { font-size-adjust:0.54; }* *div { font-size-adjust:0.46; }*

	Declares the *aspect value*(font size divided by x-height).		
font-stretch	**Limited browser support: Was part of CSS 2, but not in CSS 2.1. This property may return in CSS 3.** Declares the stretch of the font face.	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded	*div { font-stretch:narrower; }* *div { font-stretch:ultra-expanded; }*
font-style	Declares the font style.	*normal* *italic* *oblique*	*div { font-style:italic; }* *div { font-style:oblique; }*
font-variant	Declares the font variant.	*normal* *small-caps*	*div { font-variant:normal; }* *div { font-variant:small-caps; }*
font-weight	Declares the font weight (lightness or boldness)	*normal* *bold* *bolder* *lighter* *100* *200* *300* *400* *500* *600* *700* *800* *900*	*div { font-weight:bolder; }* *div { font-weight:200; }*
Font	Used as a shorthand property to declare all of the font properties at once (except font-size-adjust and font-stretch).	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): *font-style* *font-variant* *font-weight* *font-size* *line-height* *font-family*	*div { font:italic small-caps bold 1em 1.2em Arial }* *div { font:bold 0.8em Verdana }*

List Properties:

Property	Description	Possible Values	Examples
list-style-type	Declares the type of list marker used.	*disc* *circle* *square* *decimal*	*ol { list-style-type:upper-roman; }*

		decimal-leading-zero *lower-roman* *upper-roman* *lower-alpha* *upper-alpha* *lower-greek* *lower-latin* *upper-latin* *hebrew* *armenian* *georgian* *cjk-ideographic* *hiragana* *katakana* *hiragana-iroha* *katakana-iroha*	*ul { list-style-type:square; }*
list-style-position	Declares the position of the list marker.	*inside* *outside*	*ol { list-style-position:inside; }* *ul { list-style-position:outside; }*
list-style-image	Declares an image to be used as the list marker.	URL values.	*ul { list-style-image:url(image.jpg); }*
list-style	Shorthand property to declare three list properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): *list-style-type* *list-style-position* *list-style-image*	*ul { list-style:disc inside url(image.gif); }* *ol { list-style:upper-roman outside; }*
marker-offset	Declares the marker offset for elements with a value of**`marker`** set for the **`display`**property.	Lengths and the predefined value **`auto`**.	*li:before { display:marker; marker-offset:5px; }*

Text Properties:

Property	Description	Possible Values	Examples
color	Declares the color of the text.	Valid color names, RGB values, hexidecimal notation. The predefined color names are: *aqua* *black* *blue* *fuchsia* *gray*	*div { color:green; }* *div {color:rgb(0,255,0);}* *div { color:#00FF00; }*

		green *lime* *maroon* *navy* *olive* *purple* *red* *silver* *teal* *white* *yellow*	
direction	Declares the reading direction of the text.	*ltr* *rtl* ltr = left-to-right rtl = right-to-left	*div { direction:ltr; }* *div { direction:rtl; }*
line-height	Declares the distance between lines.	Numbers, percentages, lengths, and the predefined value of *normal*.	*div { line-height:normal; }* *div { line-height:2em; }* *div { line-height:125%; }*
letter-spacing	Declares the amount of space between text characters.	A length (in addition to the default space) or the predefined value of *normal*.	*div { letter-spacing:normal; }* *div { letter-spacing:5px; }* *div { letter-spacing:-1px; }*
text-align	Declares the horizontal alignment of inline content.	*left* *right* *center* *justify* If used on a set of table cells, this property can be given a string value to which the text of each row of the column will be aligned.	*div { text-align:center; }* *div { text-align:right; }* *td { text-align:"."; }*
text-decoration	Declares the text decoration.	*none* *underline* *overline* *line-through* *blink*	*div { text-decoration:none; }* *div { text-decoration:underline; }*
text-indent	Declares the indentation of the first line of text.	Lengths and percentages.	*div { text-indent:12px; }* *div { text-indent:2%; }*
text-shadow	Declares shadow effects on the text.	A list containg a color followed by numeric values (separated by spaces) that specify:	*div { text-shadow:green 2px 2px 7px; }*

		1. The color for the shadow effect 2. Horizontal distance to the right of the text 3. Vertical distance below the text 4. Blur radius	`div { text-shadow:olive -3px -4px 5px; }`
text-transform	Declares the capitalization effects on the letters in the text.	none capitalize uppercase lowercase	`div { text-transform:uppercase; }` `div { text-transform:lowercase; }`
unicode-bidi	Declares values relating to bidirectional text. May be used in conjunction with the thedirection property.	normal embed bidi-override	`div { unicode-bidi:embed; }` `div { unicode-bidi:bidi-override; }`
white-space	Declares how white space is handled in an element.	normal pre nowrap	`div { white-space:pre; }` `div { white-space:nowrap; }`
word-spacing	Declares the space between words in the text.	A length (in addition to the default space) or the predefined value of normal.	`div { word-spacing:normal; }` `div { word-spacing:1.5em; }`

Table Properties:

Property	Description	Possible Values	Examples

border-collapse	Declares the way borders are displayed.	`collapse`
		`separate`
			`table { border-collapse:collapse; }`
			
			`table { border-collapse:separate; }`

border-spacing	Declares the distance separating borders (if border-collapse is separate).	`Lengths for the horizontal and vertical spacing, separated by a space. If one length is value is declared, that length is used for both the horizontal and vertical spacing. If two lengths are declared, the first one is used for horizontal spacing and the second one is used for vertical spacing.`	`table { border-spacing:5px; }` `table { border-spacing:5px 10px; }`

caption-side	Declares where the table caption is displayed in relation to the	`top`
		`bottom`
		`left`
		`right`
			`caption { caption-side:top; }`
			`caption { caption-side:right; }`

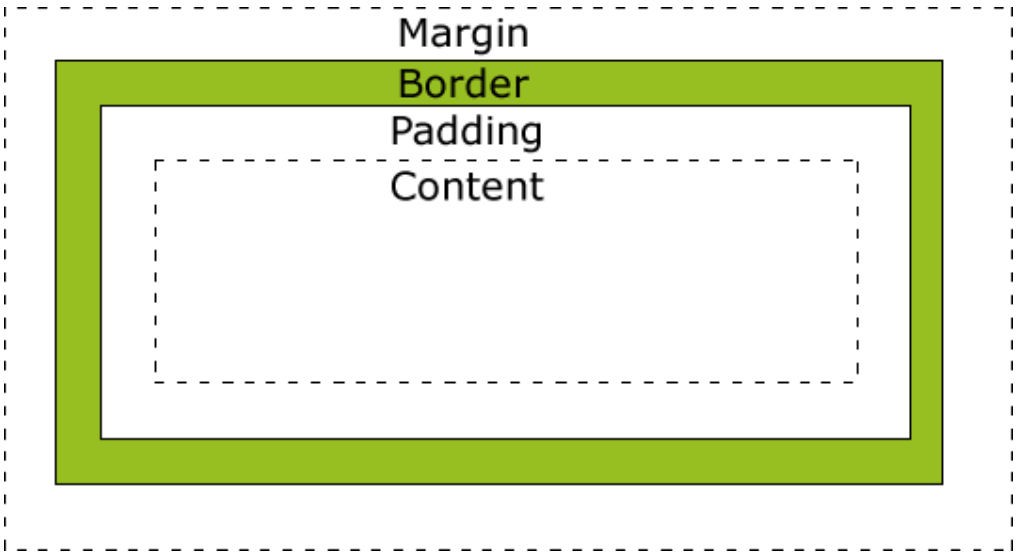
	table.		
empty-cells	Declares the way empty cells are displayed (if border-collapse is separate).	*show* *hide*	*table { empty-cells:show; }* *table { empty-cells:hide; }*
table-layout	Declares the type of table layout.	*auto* *fixed*	*table { table-layout:auto; }* *table { table-layout:fixed; }*

CSS Box Model

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

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

The image below illustrates the box model:



Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

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

```
<html>
<head>
<style>
div {
    background-color: lightgrey;
    width: 300px;
    padding: 25px;
    border: 25px solid navy;
    margin: 25px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML
element. It consists of: margins, borders, padding, and the actual content.
</p>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.</div>

</body>
</html>
```

Result: