**Syntax**

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



* The **selector points** to the HTML element/tag 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 this example below <p> tag will be center-aligned, with a green text color:

p {

color: green;

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. There are many kinds of selectors available in CSS such as:

1. Element Selector
2. ID Selector
3. Class Selector
4. Grouping Selectors

**Element Selector:**

The element selector selects elements based on the element name. For example, the <p> elements in below example has been styled to be center-aligned, with a green text color:

p {

text-align: center;

color: green;

}

**ID Selector:**

The id selector uses the id attribute of an HTML element to select a specific element. The element should be unique within a page. To select an element with a specific id, write a hash (#) character, followed by the id of the element. For example, to apply style to paragraph with id = “para”:

<p id="para">Test Example</p>

The following style rules be applied:

#para {

text-align: center;

color: red;

}

**Class Selector:**

The class selector uses the class attribute of an HTML element to select a specific class element. The element have no conditions to be unique. To select an element with a specific class, write a hash (.) character, followed by the class name of the elements. For example, to apply style to paragraph with class = “para”:

<p class="para">Test Example</p>

The following style rules be applied:

.para {

text-align: center;

color: red;

}

The rule will get applied to all the elements that has the class “para”.

### Units

CSS has several different units for expressing a length. Many CSS properties take “length” values, such as width, margin, padding, font-size, etc. Length is a number followed by a length unit, such as 10px, 2em, etc.

There are two types of length units: absolute and relative. But we will only be covering absolute length here.

##### Absolute Lengths

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

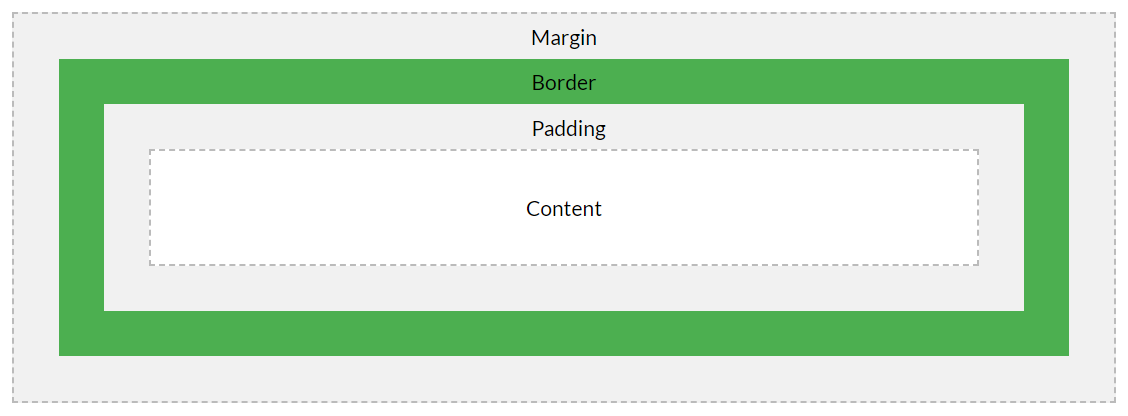
Example of Absolute length include:  
cm : centimeters  
mm : millimeters  
in : inches (1in = 96px = 2.54cm)  
px : pixels (1px = 1/96th of 1in)  
pt : points (1pt = 1/72 of 1in)  
pc : picas (1pc = 12 pt)

The most common one you will encounter all the time will be pixel (px). They are used specially when applying width, margin or padding to the elements.

**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

### Colors

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values. They can be applied to different kind of elements and exist in many forms such as background color, text color or border colors.

Example of each case is give below:

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

<h3 style="border:2px solid red;">Hello World</h3>

<p style="color:green;">Lorem ipsum...</p>

### Spacing

There are 2 important concepts when it comes to CSS spacing i.e Margin and Padding. You can use CSS code to reduce or increase the spacing of elements by using the padding and margin CSS commands.

##### 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).

**Example:** Set different margin for all four sides of a <div> element:

div {

margin-top: 100px;

margin-bottom: 100px;

margin-right: 150px;

margin-left: 80px;

}

**Note:**If you wish to apply same margin across all the sides then simply using margin rather direction (top,bottom,left,right).

**Example:** Set same margin for all four sides of a <div> element:

div {

margin: 100px;

}

##### 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).

**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;

}

**Note:**If you wish to apply same padding across all the sides then simply using padding rather direction (top,bottom,left,right).

**Example:** Set same margin for all four sides of a <div> element:

div {

margin: 100px;

}

### Sizing

There are 2 important concepts when it comes to CSS sizing i.e Height and Width. They both are useful properties when it comes to defining the size of the boxes/divs/paragraphs.

##### Setting height and width

The height and width properties are used to set the height and width of an element.

The height and width can be set to auto (this is default. Means that the browser calculates the height and width), or be specified in length values, like px, cm, etc., or in percent (%) of the containing block.

Example: Set the height and width of a <div> element

div {

height: 200px;

width: 50%;

background-color: powderblue;

}

**Fonts**

The CSS font properties define the font family, boldness, size, and the style of a text.

**Font Family**

The font family of a text is set with the font-family property. Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

**Note**: If the name of a font family is more than one word, it must be in quotation marks, like: “Times New Roman”.

More than one font family is specified in a comma-separated list, example give below:

p {

font-family: "Times New Roman", Times, serif;

}

**Font Style**

The font-style property is mostly used to specify italic text.

This property has three values:

* normal – The text is shown normally
* italic – The text is shown in italics
* oblique – The text is “leaning” (oblique is very similar to italic, but less supported)

.normal {

  font-style: normal;

}

.italic {

  font-style: italic;

}

.oblique {

  font-style: oblique;

}

**Font Size**

The font-size property sets the size of the text. Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs. Always use the proper HTML tags, like <h1> – <h6> for headings and <p> for paragraphs.

**Example:** Setting the text size with pixels gives you full control over the text size:

h1 {

    font-size: 40px;

}

h2 {

    font-size: 30px;

}

p {

    font-size: 14px;

}

**Font Weight**

The font-weight property specifies the weight of a font. It could either be bold or normal.

p.normal {

font-weight: normal;

}

p.bold {

font-weight: bold;

}

**Decorations**

Some of the most common decoration properties for text include:

* Text Color
* Text Alignment
* Text Decoration
* Text Transformation
* Text Indentation
* Letter Spacing
* Line Height

Lets look into each property in detail with example.

**Text Color**

The color property is used to set the color of the text. The color is specified by:

* a color name – like “red”
* a HEX value – like “#ff0000”
* an RGB value – like “rgb(255,0,0)”

The example below showcases the use of text color:

body {

  color: blue;

}

h1 {

  color: green;

}

**Text Alignment**

The text-align property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified. The following example shows justify, center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left):

div {

text-align: justify;

}

h1 {

text-align: center;

}

h2 {

text-align: left;

}

h3 {

text-align: right;

}

**Text Decoration**

The text-decoration property is used to set or remove decorations from text. The value text-decoration: none; is often used to remove underlines from links. The other text-decoration values are used to decorate text such as overline, underline or line-through. The following example demonstrates how to use text decoration property:

a {

text-decoration: none;

}

h1 {

text-decoration: overline;

}

h2 {

text-decoration: line-through;

}

h3 {

text-decoration: underline;

}

**Text Transformation**

The text-transform property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word. The following example demonstrates how to use text-transform property for a paragraph.

p.uppercase {

  text-transform: uppercase;

}

p.lowercase {

  text-transform: lowercase;

}

p.capitalize {

  text-transform: capitalize;

}

**Text Indentation**

The text-indent property is used to specify the indentation of the first line of a text. The following example demonstrates how to define a indentation of the first line of a text.

p {

text-indent: 50px;

}

**Letter Spacing**

The letter-spacing property is used to specify the space between the characters in a text. The following example demonstrates how to increase or decrease the space between characters:

h1 {

  letter-spacing: 3px;

}

h2 {

  letter-spacing: -3px;

}

**Line Height**

The line-height property is used to specify the space between lines. The following example demonstrates how to define a line height for a paragraph:

p.small {

  line-height: 0.8;

}

p.big {

  line-height: 1.8;

}

**Positioning**

The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky). There are five different position values:

* static
* relative
* fixed
* absolute
* sticky

We will only be looking at the relative position.

**position: relative;**

An element with position: relative; is positioned relative to its normal position. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

Here is the CSS that is used:

div.relative {

position: relative;

left: 30px;

border: 3px solid #73AD21;

}