# BOX MODEL

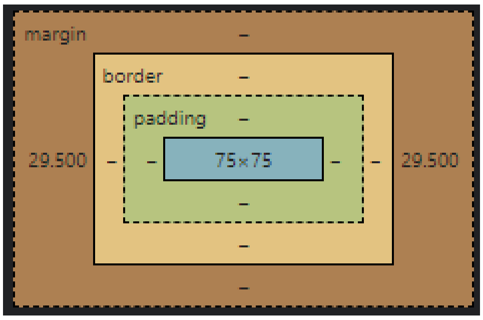
APRIL 2022

YONATAN BENEZRA

## The CSS Box Model

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 below image illustrates the box model:



Explanation of the various 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.

## Width and Height of an Element:

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

**Important:** When you set the width and height properties of an element with CSS, you set the width and height of the content area only. To calculate the full size of an element, you must also add padding, borders, and margins.

### Example:

This <div> element will have a total width of 350px:

```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
```

Here is the calculation:
320px (width)
+ 20px (left + right padding)

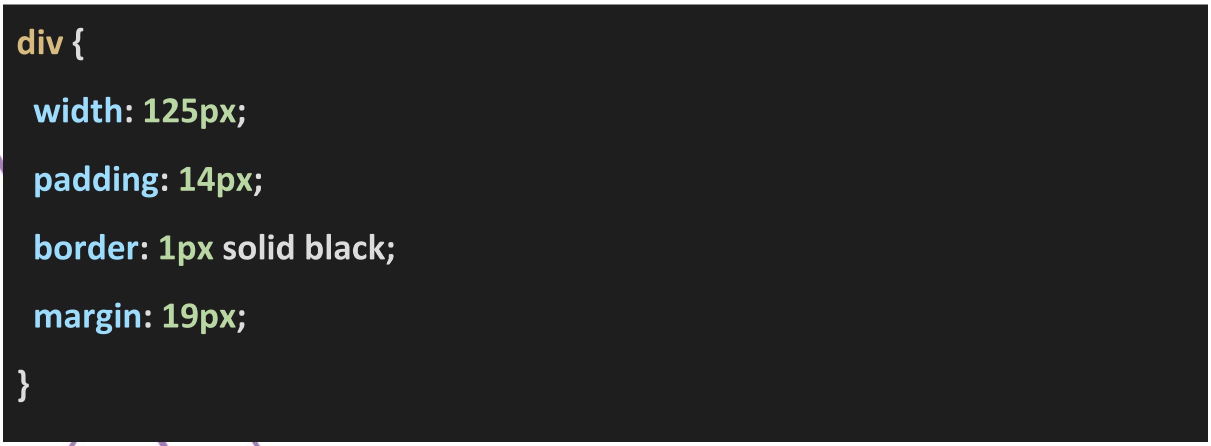
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

**The total width of an element should be calculated as follows:**
Total element width = width + left padding + right padding + left border + right border + left margin + right margin.

The total height of an element should be calculated as follows:
Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin.

**Exercise:**

What will be the width and height of the following div?



```
div {
  width: 125px;
  padding: 14px;
  border: 1px solid black;
  margin: 19px;
}
```

## Text and font properties

Text has many optional formatting properties.
Today, we will learn about the following properties:

- color
- alignment
- transformation
- spacing

## Text-color

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

- A color name – such as "red"
- A HEX value – such as "#ff0000"
- An RGB value – such as "rgb(255,0,0)"

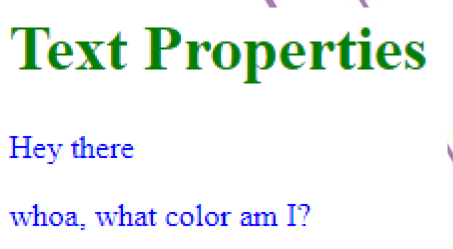
Look at CSS Color Values for a complete list of possible color values.
The default text color for a page is defined in the body selector.

The following code:

```
<h1>Text Properties</h1>

<p>Hey there</p>

<p>whoa, what color am I?</p>
```

```
body {

  color: blue;

 }


 h1 {

  color: green;

 }
```

will bring us the below webpage!



Text Properties

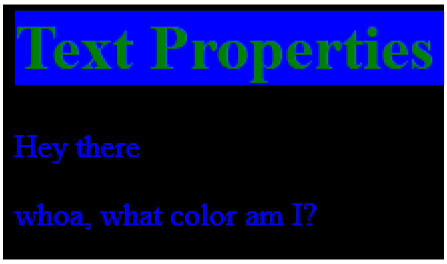
Hey there

whoa, what color am I?

You can change the background color as well, as you have seen in previous lessons.

**Exercise:**

Create a webpage that looks like the following image:



Text Properties

Hey there

whoa, what color am I?

**<u>Text Alignment has several important properties, such as:</u>**

- text-align
- text-align-last
- direction

## 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 center aligned, and left and right aligned text.
(Left alignment is the default if text direction is left-to-right, and right alignment is the default if text direction is right-to-left):

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

h2 {
  text-align: left;
}

h3 {
  text-align: right;
}
```

When the text-align property is set to "justify", each line is stretched so that every line has equal width, making the left and right margins straight (like in magazines and newspapers):

Try it out!

## Text Align Last:

The text-align-last property specifies how to align the last line of a text.

## Text Direction:

The direction property can be used to change the text direction of an element.
We will use this often to change the direction of our web pages from “left-to-right” to “right-to-left”.

We can do so by setting an “rtl” value to the direction property, as follows:

```
.body {
direction: rtl;
}
```

## Text Transformation:

The text-transform property is used to specify uppercase and lowercase letters in a text.
This can be used to turn everything into uppercase or lowercase letters, or to capitalize the first letter of each word.

**For example:**

```
p.uppercase {
  text-transform: uppercase;
}

p.lowercase {
  text-transform: lowercase;
}
```

```
p.capitalize {
  text-transform: capitalize;
}
```

## Text Spacing

### Text Indentation:

The text-indent property is used to specify the indentation of the first line of a text.

### 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.

### Line Height:

The line-height property is used to specify the space between lines.

### Word Spacing:

The word-spacing property is used to specify the space between the words in a text.

### White Space:

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

## CSS Text Spacing Properties:

| Property | Description |
|---|---|
| letter-spacing | Specifies the space between characters in a text |
| line-height | Specifies the line height |
| text-indent | Specifies the indentation of the first line in a text-block |
| white-space | Specifies how to handle white-space inside an element |
| word-spacing | Specifies the space between words in a text |
| letter-spacing | Specifies the space between characters in a text |
| line-height | Specifies the line height |
| text-indent | Specifies the indentation of the first line in a text-block |
| white-space | Specifies how to handle white-space inside an element |
| word-spacing | Specifies the space between words in a text |

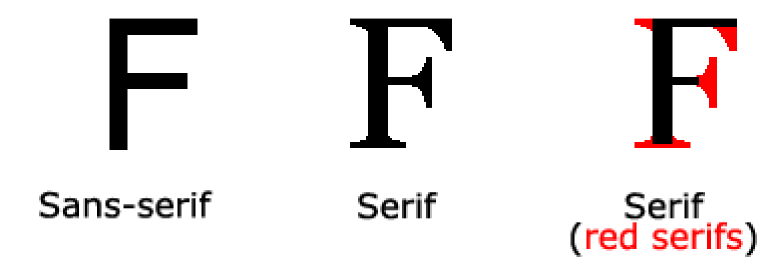
## Generic Font Families

In CSS, there are five generic font families:

- Serif fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
- Sans-serif fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
- Monospace fonts – here, all the letters have the same fixed width. They create a mechanical look.
- Cursive fonts imitate human handwriting.
- Fantasy fonts are decorative/playful fonts.

Each of the different font names belong to one of the generic font families.

## Difference Between Serif and Sans-serif Fonts



***Note:**

On computer screens, sans-serif fonts are considered easier to read than serif fonts.

## The CSS font-family Property:

In CSS, we use the font-family property to specify the font of a text.

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

***Tip:**
The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems.

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). The font names should be separated with commas.

**Example**

```
.p1 {
  font-family: "Times New Roman", Times, serif;
}


.p2 {
  font-family: Arial, Helvetica, sans-serif;
}


.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
```