

Cascading Style Sheet (CSS)

Introduction

- **CSS** stands for **C**ascading **S**tyl**S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- It can control the **layout** of multiple web pages all at once
- External stylesheets are stored in **CSS files**

Why use CSS?

- CSS is used to define **styles** for your web pages including the **design, layout and variations** in display for **different devices and screen sizes**
- CSS removed the style formatting from the HTML page
- change the look of an entire website by changing just one file

Why use CSS?

This is a bold text

**This is another bold text **

Change all the bold text to underline???

Change all the text to bold, font-style Verdana
and color red???

Why use CSS?

Solution:

```
<p class="myStyle">My CSS styled text</p>
```

```
<style type="text/css">
```

```
  .myStyle {
```

```
    font-family: Verdana, Arial, Helvetica, sans-serif;
```

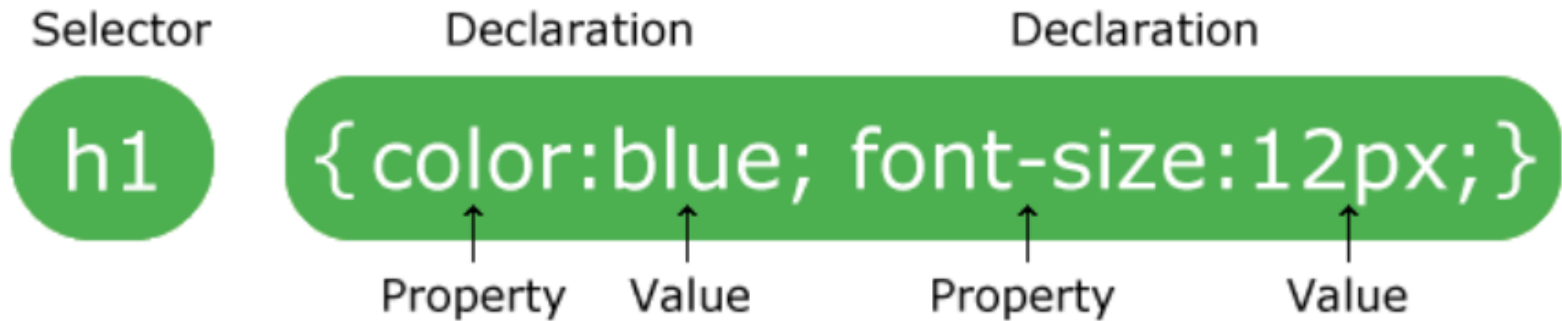
```
    font-weight: bold;
```

```
    color: #FF0000;
```

```
  }
```

```
</style>
```

CSS Syntax



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

CSS Syntax (example.htm)

```
<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.

Selecting multiple elements

```
p, li, h1 {  
    color: red;  
}
```

Different types of selectors

Selector Name	Example
Element Selector	<code>p</code> Selects <code><p></code>
ID Selector	<code>#my-id</code> Selects <code><p id="my-id"></code> or <code></code>
Class Selector	<code>.my-class</code> Selects <code><p class="my-class"></code> and <code></code>

Different types of selectors

Selector Name	Example
Attribute selector	<code>img[src]</code> Selects <code></code> but not <code></code>
Pseudo-class selector	<code>a:hover</code> Selects <code><a></code> , but only when the mouse pointer is hovering over the link.

CSS Comments

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

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

Three Ways to Insert CSS

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

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

Internal Style Sheet

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

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

Inline Styles

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

This is a heading

This is a paragraph.

Multiple Style Sheets

What 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.

`Mystyle.css`

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

Multiple Style Sheets (Ex.)

```
<html>
  <head>
    <link rel="stylesheet" href="mystyle.css">
    <style>
      h1 {
        color: orange;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
  </body>
</html>
```

Multiple Style Sheets (Output)

This is a heading

Cascading Order

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

- 1) **Browser default** (No styles specified)
- 2) **External style sheet and internal styles** (**as per ordering**)
- 3) **Inline style** (inside an HTML element) (**Highest priority**)

CSS Colors

Colors are specified using

- predefined color names
- RGB
- HEX
- HSL
- RGBA
- HSLA values

Predefined color names

`<h1 style="background-color:Tomato">Tomato</h1>`

`<h1 style="background-color:Orange">Orange</h1>`

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

Tomato

Orange

DodgerBlue

RGB, Hex, HSL, RGBA, HSLA

rgb(255, 99, 71)

rgb(255, 99, 71)

#ff6347

#ff6347

hsl(9, 100%, 64%)

hsl(9, 100%, 64%)

rgba(255, 99, 71, 0.5)

rgba(255, 99, 71, 0.5)

hsla(9, 100%, 64%, 0.5)

hsla(9, 100%, 64%, 0.5)

RGB, Hex, HSL, RGBA, HSLA

The **alpha** parameter is a number between

0.0 (fully transparent) and

1.0 (fully opaque)

Opacity / Transparency

- The `opacity` property specifies the opacity/transparency of an element
- It can take on value from 0 to 1
- The lower value, more transparent

Opacity / Transparency

<pre>div { background-color: green; }</pre>	<pre><div class="first"> <h1>opacity 0.1</h1> </div></pre>
<pre>div.first { opacity: 0.1; }</pre>	<pre><div class="second"> <h1>opacity 0.3</h1> </div></pre>
<pre>div.second { opacity: 0.3; }</pre>	<pre><div class="third"> <h1>opacity 0.6</h1> </div></pre>
<pre>div.third { opacity: 0.6; }</pre>	

Opacity / Transparency

opacity 0.1

opacity 0.3

opacity 0.6

opacity 1 (default)

HTML Block and Inline Elements

- Every HTML element has a default display value, depending on what type of element it is.
- There are two display values: block and inline.

Block-level Elements

- A block-level element always starts on a new line.
- A block-level element always takes up the full width available (stretches out to the left and right as far as it can).
- A block level element has a top and a bottom margin, whereas an inline element does not.

Block-level Elements (Ex.)

<address>	<article>	<aside>	<blockquote>
<canvas>	<dd>	<div>	<dl>
<dt>	<fieldset>	<figcaption>	<figure>
<footer>	<form>	<h1>-<h6>	<header>
<hr>		<main>	<nav>
<noscript>		<p>	<pre>
<section>	<table>	<tfoot>	
<video>			

Inline Elements

- An inline element does not start on a new line.
- An inline element only takes up as much width as necessary.

Inline Elements (Ex.)

<a>

<abbr>

<acronym>

<bdo>

<big>

**
**

<button>

<cite>

<code>

<dfn>

<i>

<input>

<kbd>

<label>

<map>

<object>

<output>

<q>

<samp>

<script>

<select>

<small>

<sub>

<sup>

<textarea>

<time>

<tt>

<var>

Inheritance

- In CSS, **inheritance** controls what happens when no value is specified for a property on an element.
- CSS properties can be categorized in two types:
 - **inherited properties**, which by default are set to the computed value of the parent element
 - **non-inherited properties**, which by default are set to initial value of the property

Inherited Properties

- When no value for an inherited property has been specified on an element, the element gets the computed value of that property on its parent element.
- Only the root element of the document gets the initial value given in the property's summary.



```
p {  
  color: green;  
}
```

```
<p>This paragraph has <em>emphasized text</em>in it.</p>
```

This paragraph has *emphasized text*in it.

Non-inherited Properties

- When no value for a **non-inherited property** has been specified on an element, the element gets the **initial value** of that property
- the words "emphasized text" will not have a border

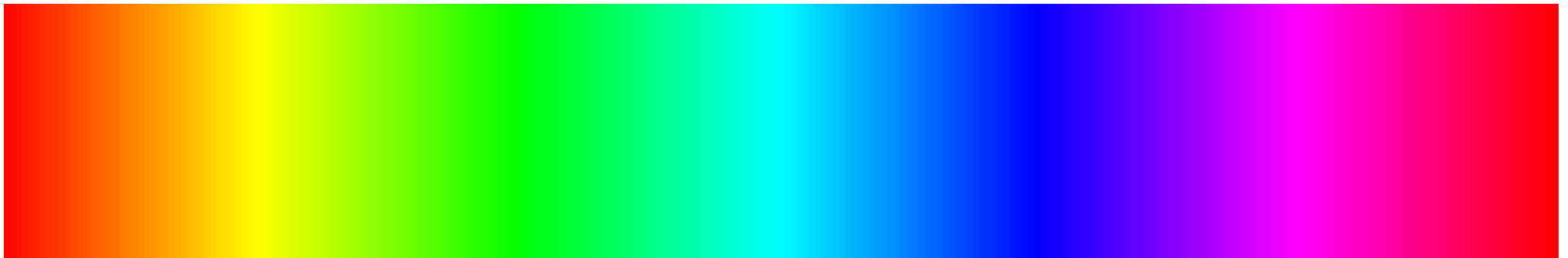
```
p {  
  color:  green;  
  border: 2px  black solid;  
  width: 400px;  
}
```

```
<p>This paragraph has <em>emphasized text</em>in it.</p>
```

This paragraph has *emphasized text*in it.

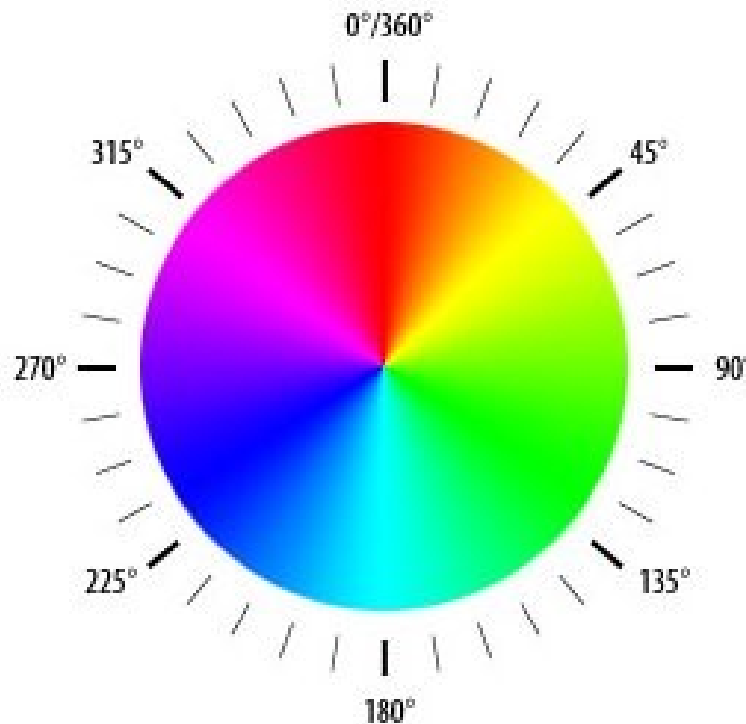
HSL Color Model

- **Hue** indicates the color sensation of the light
- i.e. if the color is red, yellow, green, cyan, blue, magenta



HSL Color Model

- **Hue** works circular, so it can be represented on a circle instead.
- A hue of 360° looks the same again as a hue of 0° .



HSL Color Model

- **Saturation** indicates the degree to which the hue differs from a neutral gray
- The values run from 0%, which is no color, to 100% full saturation

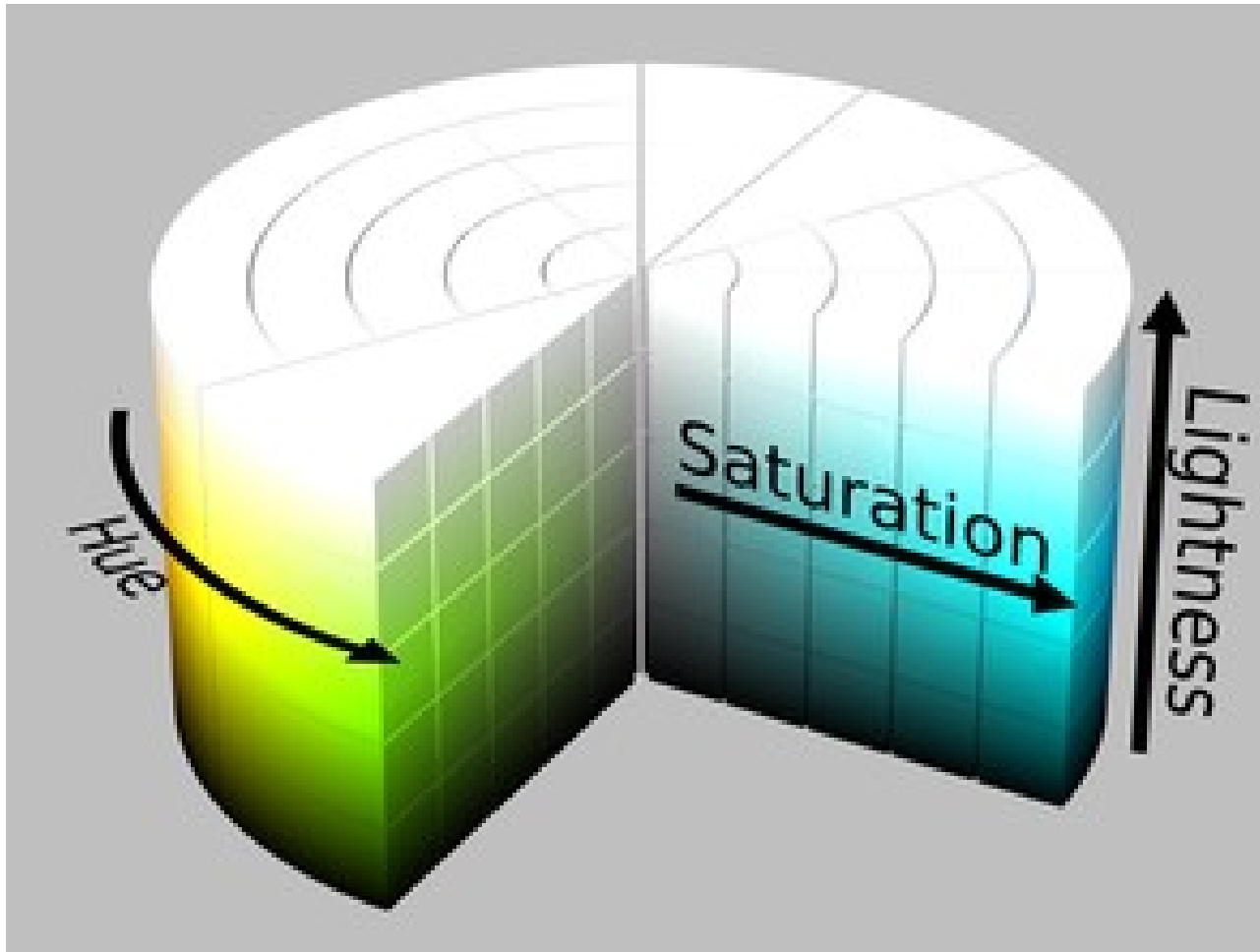


HSL Color Model

- **Lightness** indicates the illumination of the color
- At 0% the color is completely black
- At 50% the color is pure, and
- At 100% it becomes white



HSL Color Model



CSS Backgrounds

background-color

background-image [url("image.png")]

background-repeat (repeat-x, repeat-y, no-repeat)

background-position (right top)

background-attachment (scroll, fixed)

CSS Backgrounds (Ex.)

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

Background – Shorthand Property

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

the order of the property values is:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

It does not matter if one of the property values is missing, as long as the other ones are in this order.

CSS Borders

- border-style
- border-width
- border-color
- border-radius
- Border individual sides
 - border-top-style
 - border-right-style
 - border-bottom-style
 - border-left-style

CSS Borders (border-style)

border-style (dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden, mix)

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

```
<p class="dotted">    A dotted border.    </p>
<p class="dashed">    A dashed border.    </p>
<p class="solid">      A solid border.      </p>
```

.....
 :A dotted border.
 :.....

A dashed border.

A solid border.

CSS Borders (border-width)

- specific size (in px, pt, cm, em, etc)
- thin, medium, or thick
- one to four values
 - for the **top** border, **right** border, **bottom** border, and the **left** border

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

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

```
p.three {  
    border-style: solid;  
    border-width: 2px 10px 4px 20px;  
}
```

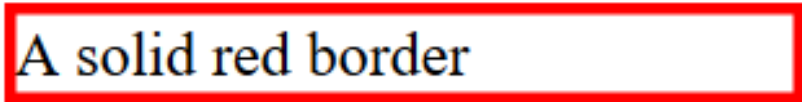
CSS Borders (border-color)

- Name
- Hex
- RGB
- Transperent

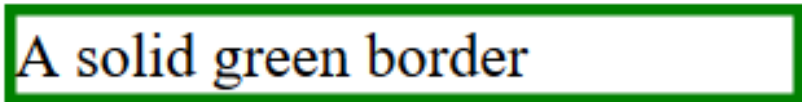
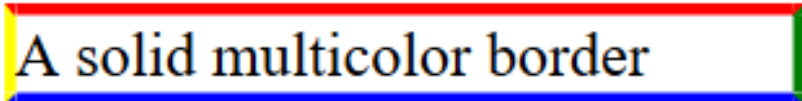
CSS Borders (border-color)

| | | |
|--------------------------------------|---------------------------|-------------------------|
| <code><p class="one"></code> | A solid red border | <code></p></code> |
| <code><p class="two"></code> | A solid green border | <code></p></code> |
| <code><p class="three"></code> | A solid multicolor border | <code></p></code> |

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

A rectangular box with a solid red border and the text "A solid red border" inside.

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

A rectangular box with a solid green border and the text "A solid green border" inside.A rectangular box with a multicolored border (red on top, green on the right, blue on the bottom, and yellow on the left) and the text "A solid multicolor border" inside.

```
p.three {  
  border-style: solid;  
  border-color: red green blue yellow;  
}
```


CSS Borders (border-color)

```
p.round1 {  
  border: 2px solid red;  
  border-radius: 5px;  
  width: 150px;  
}
```

Round border

```
p.round2 {  
  border: 2px solid red;  
  border-radius: 8px;  
  width: 150px;  
}
```

Rounder border

```
p.round3 {  
  border: 2px solid red;  
  border-radius: 12px;  
  width: 150px;  
}
```

Roudest border

```
<p class="round1">Round border</p>  
<p class="round2">Rounder border</p>  
<p class="round3">Roudest border</p>
```

Border - Shorthand Property

- border-width
- border-style (required)
- border-color

```
<html>
<head>
<style>
p {
    border: 5px solid red;
    width: 100px;
}
</style>
</head>
<body>

<h2>The border Property</h2>

<p>Some text.</p>

</body>
</html>
```

The border Property

Some text.

CSS Margins

- margin-top
- margin-right
- margin-bottom
- Margin-left

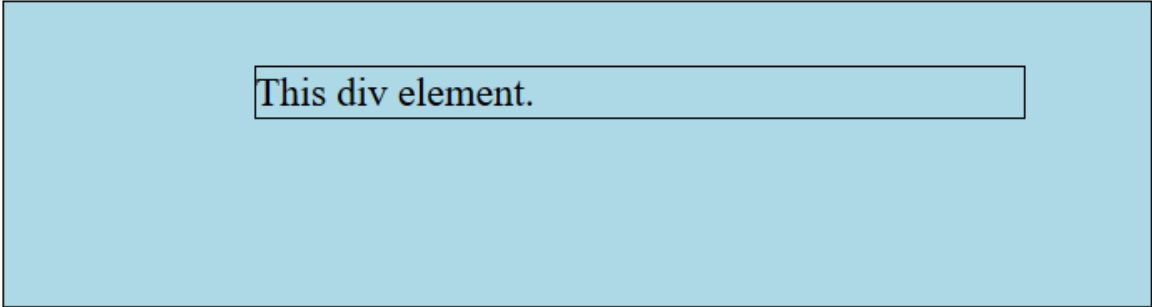
margin properties can have the following values:

- 1) **auto** - the browser calculates the margin
- 2) **length** - specifies a margin in px, pt, cm, etc.
- 3) **%** - specifies a margin in % of the width of the containing element
- 4) **inherit** - margin should be inherited from the parent element

CSS Margins (ex.)

```
<html>
<head>
  <style>
    div {
      border: 1px solid black;
      margin-top: 25px;
      margin-right: 50px;
      margin-bottom: 75px;
      margin-left: 100px;
      background-color: lightblue;
    }
  </style>
</head>
<body>
  <div>
    <div>This div element.</div>
  </div>
</body>
</html>
```

CSS Margins (output)



This diagram illustrates CSS margins. It features a large, light blue rectangular container. Inside this container, there is a smaller, white rectangular box with a thin black border. The text "This div element." is centered within the white box. The space between the white box and the light blue container represents the margin.

This div element.

Margin - Shorthand Property

`margin: 25px 50px 75px 100px;` top, right, bottom, left

`margin: 25px 50px 75px;` top, left-right, bottom

`margin: 25px 50px;` top-bottom, right-left

`margin: 25px;` all

`margin: auto;`

`auto` to horizontally center the element within its container.

The paragraph.

Margin Collapse

Top and bottom margins of elements are collapsed into a single margin

- that is equal to the **largest of the two**

```
h1 {  
    margin: 0 0 50px 0;  
}  
h2 {  
    margin: 20px 0 0 0;  
}
```

```
<h1>Heading 1</h1>  
<h2>Heading 2</h2>
```

Heading 1

Heading 2

Padding - Shorthand Property

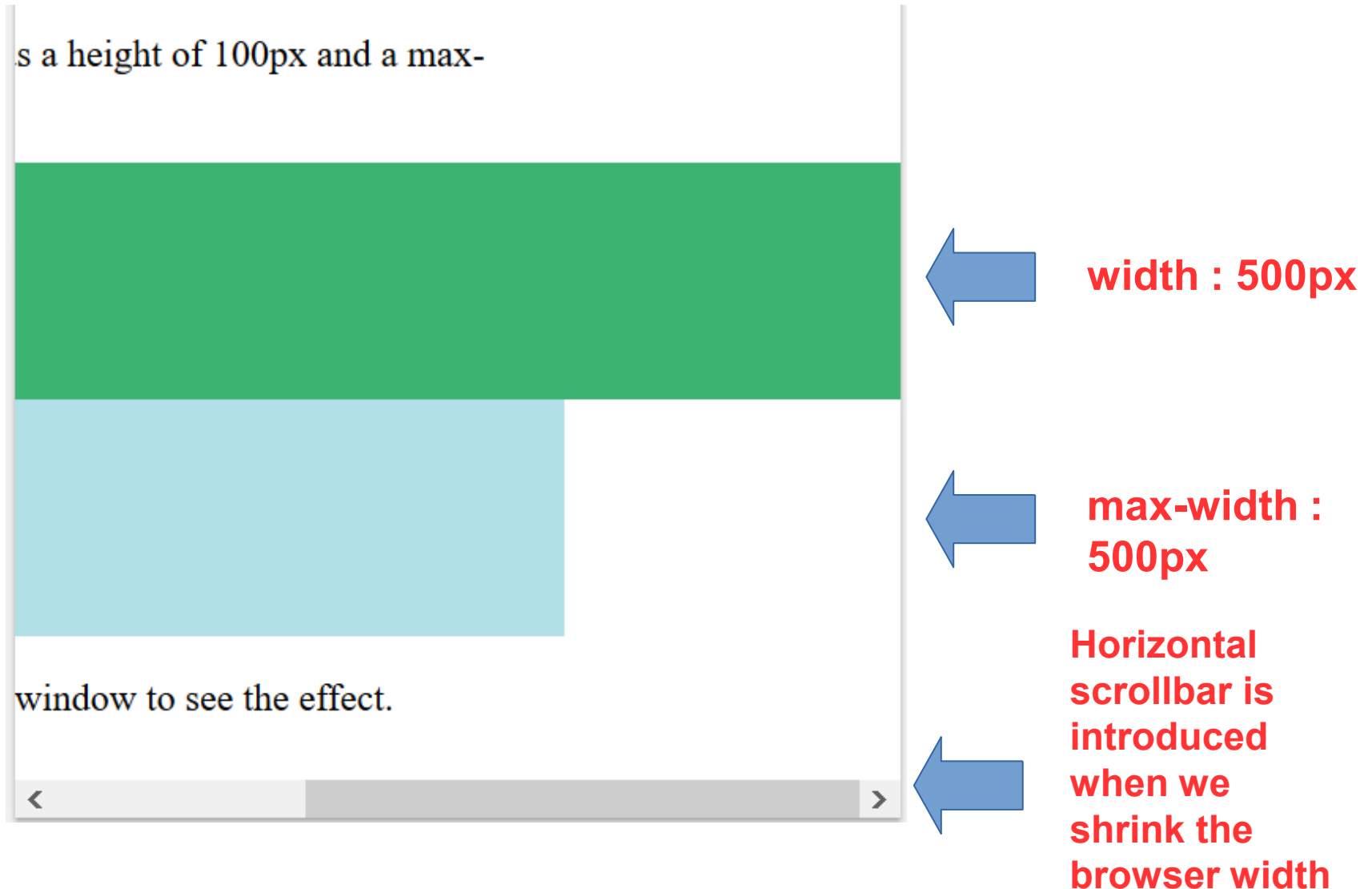
`padding: 25px 50px 75px 100px;` top, right, bottom, left

`padding: 25px 50px 100px;` top, left-right, bottom

`padding: 25px 50px;` top-bottom, right-left

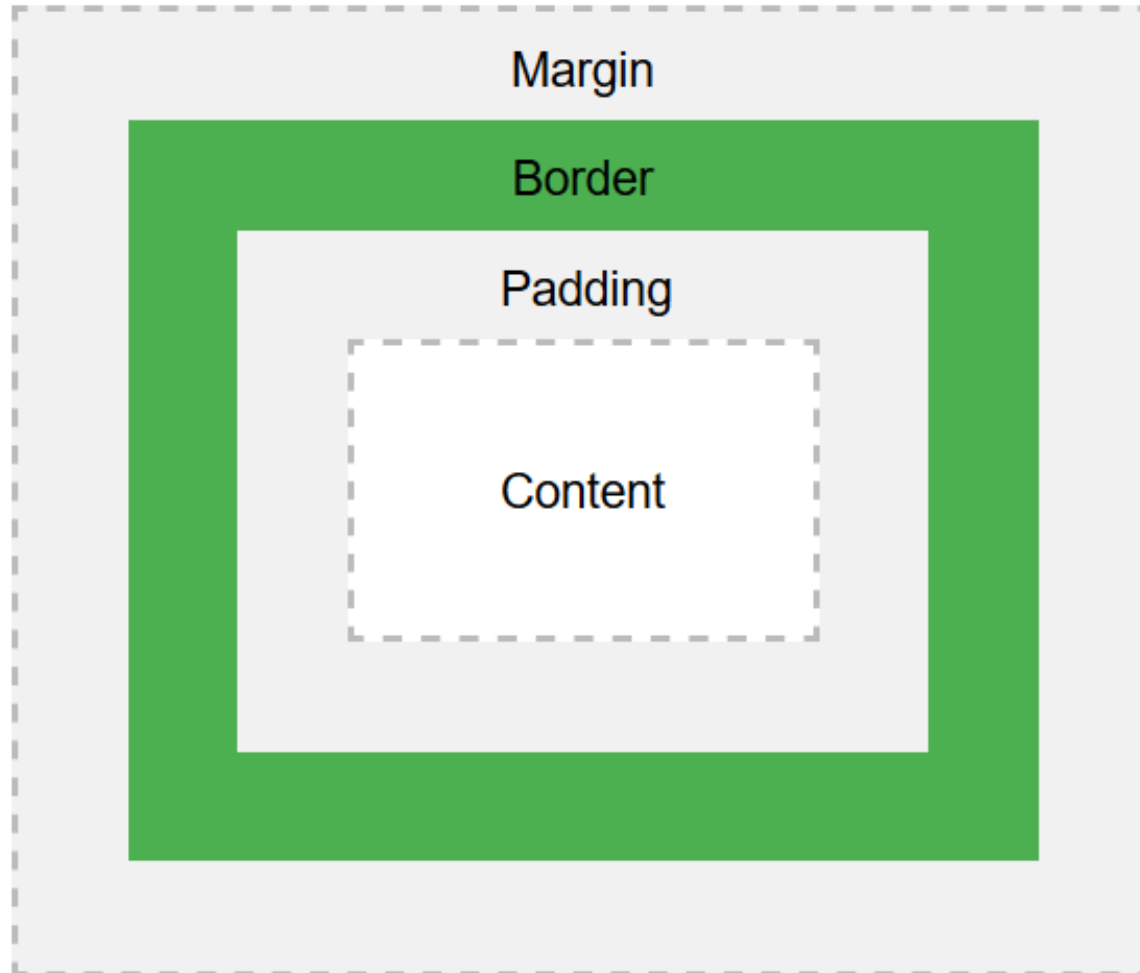
`padding: 25px;` all

Width vs. Max-width (output)



CSS Box Model

All HTML elements can be considered as boxes.



CSS Box Model

- **Content** - 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)

CSS Box Model (ex.)

```
div {  
    background-color: lightgrey;  
    width: 300px;  
    border: 25px solid green;  
    padding: 25px;  
    margin: 25px;  
}
```

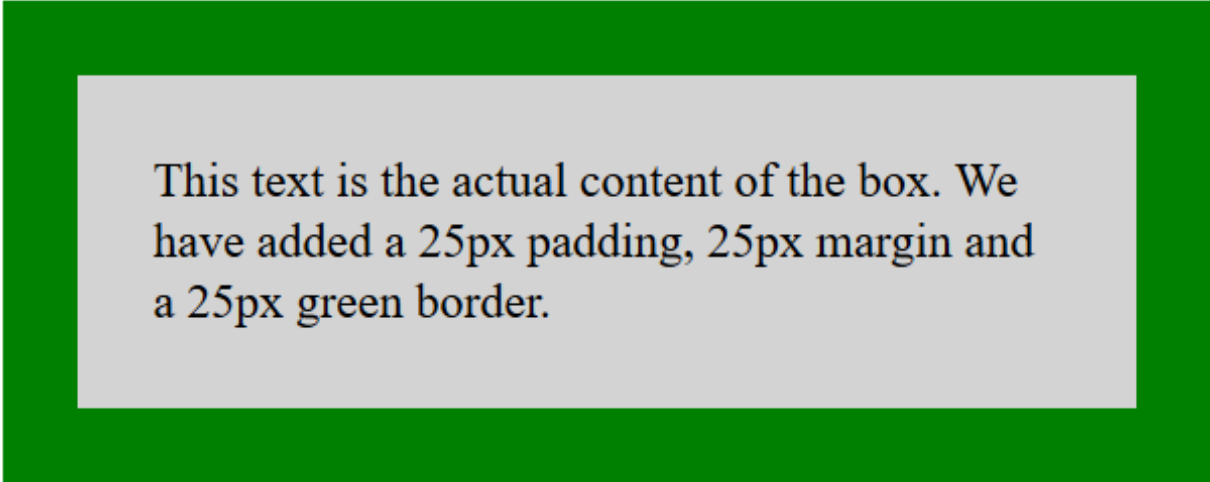
<h2>Demonstrating the Box Model</h2>

<div>This text is the actual content of the box. We have added a 25px padding, 25px margin and a 25px green border. </div>

CSS Box Model (output)

<code>width: 300px;</code>	300px (width)
<code>padding: 25px;</code>	+ 50px (left + right padding)
<code>border: 25px solid green;</code>	+ 50px (left + right border)
<code>margin: 25px;</code>	+ 50px (left + right margin)
	= 450px

Demonstrating the Box Model



This text is the actual content of the box. We have added a 25px padding, 25px margin and a 25px green border.

References

- <https://www.w3schools.com/Css/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>