



Cascading Style Sheet

Introduction

CSS stands for **Cascading Style Sheets**

Style sheet language used to **describe the presentation of a document** written in HTML or XML

How **HTML elements** are to be displayed on screen, paper, or in other media

Colors, fonts, alignment, borders, backgrounds, spacing, margins, etc...

It can control the layout of **multiple web pages all at once**

Advantage

Reusability

Separate the content and presentation

What is CSS?

3

- A simple mechanism for controlling the style of a Web document without compromising its structure.
- It allows you to **separate the visual design elements** (layout, fonts, colors, margins, and so on) from the contents of a Web page.
- Allows for faster downloads, streamlined site maintenance, and global control of design attributes across multiple pages.

CSS vs. just HTML

What can we do with CSS that we can't do with HTML?

Control of backgrounds.

Set font size to the exact height you want.

Highlight words, entire paragraphs, headings or even individual letters with background colors.

Overlap words and make logo-type headers without making images.

Precise positioning.

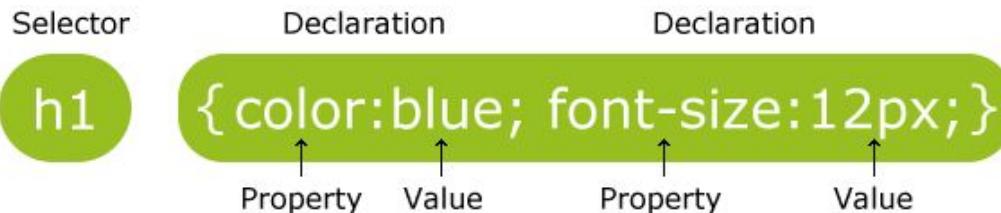
Linked style sheets to control the look of a whole website from one single location.

and more....

How to write CSS?

5

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



Selector

HTML element tags
(examples: p, h2, body, img, table)
class and ID names

Property (examples: color, font-size)

Value (examples: red, 14pt)

`selector (property: value;)`

↑
What HTML tag(s) does the property apply to (e.g. "body")

↑
The property could for example be the background color ("background-color")

↖ The value of the property background color could be red for example ("#FF0000")

Basic Structure of a Style

Each definition contains:

A property

A colon

A value

A semicolon to separate two or more values

Can include one or more values

```
h1 {font-size:12pt; color:red}
```

HTML & CSS Code

To set red color as the background color of a webpage

HTML

```
<body bgcolor="#FF0000">
```

CSS

```
body {background-color: #FF0000;}
```

Example:

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

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

Applying CSS / Types

Three ways u can apply CSS to a HTML Document

Inline

Internal / Embedded

External

Inline - Using the **style attribute** in HTML elements

Embedded (Internal) - Using a `<style>` element in the `<head>` section

External - Using an external CSS file

Types - Inline Style sheet

10

- Styles are placed inside tags

- Specific to a single instance of an html tag on a page.

- Must be used instead of tags to specify font size, color, and typeface and to define margins, etc

- Use to override an external or embedded style specification

- used to apply a unique style for a single element

Types – Internal / Embedded CSS

11

Applicable to an entire document

Styles are defined within the `<style> </style>` tag placed in the **header** of the html file `<head>` and `</head>`

Styles are written in the **same page**

used if one single page has a unique style

```
<html>
<head>
<style>
p {
    color: red;
    text-align: center;
}
</style>
</head>
<body> <p>hai </p> <p> hello</p> </body></html>
```

Types – External CSS

12

Styles are saved in a separate file, with the extension **.css**

This single style sheet can be used to define the look of multiple pages

Call the **.css** file from header of html file

```
<link href=URL rel="relation_type" type="link_type" >
```

- URL is the **file.css**
- Relation_type=**"stylesheet"**
- Link_type=**"text/css"**

Rel – relationship between the current doc and other document

MIME type	Description
text/css	CSS documents
image/png	PNG images
text/javascript	JavaScript markup
text/plain	Plain text
image/jpeg	JPEG image
text/html	HTML markup

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

Types – External CSS - Example

13

```
<html>
<head>
<link href="mystyle.css" rel="stylesheet" type="text/css" >
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

mystyle.css

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

Types – Inline / Embedded / External

Inline CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Inline Style Sheet</title>
  </head>

  <body>
    <p>Text with no style</p>

    <p style = "font-size:30pt;
      font-family: Times New Roman;
      color: green;"> Text with style </p>
  </body>
</html>
```

Text with no style

Text with style

Embedded CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Embedded Style Sheet</title>
    <style type = "text/css">
      body {background-color:pink;}
      h1{font-family:Arial; color:brown;}
      p{font-size:20pt; color:green;}
    </style>
  </head>

  <body>
    <h1>This is heading tag 1</h1>
    <p>This is Paragraph.</p>
    <h2>heading 2</h2>
  </body>
</html>
```

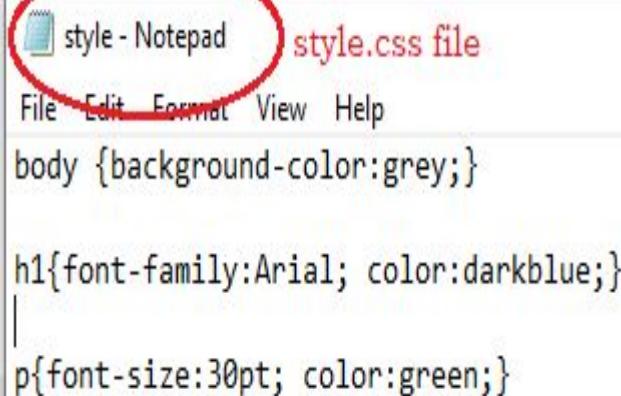
This is heading tag 1

This is Paragraph.

heading 2

Types – Inline / Embedded / External

.CSS file

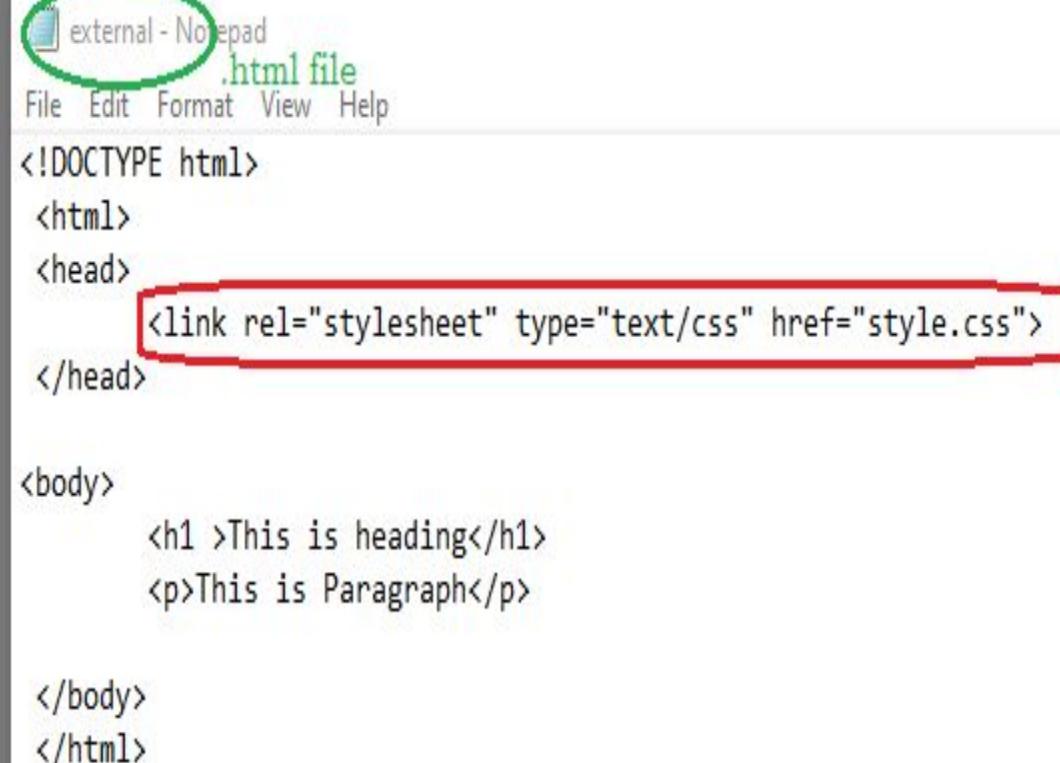


```
style - Notepad style.css file
File Edit Format View Help
body {background-color:grey;}
h1{font-family:Arial; color:darkblue;}
p{font-size:30pt; color:green;}
```

This is heading

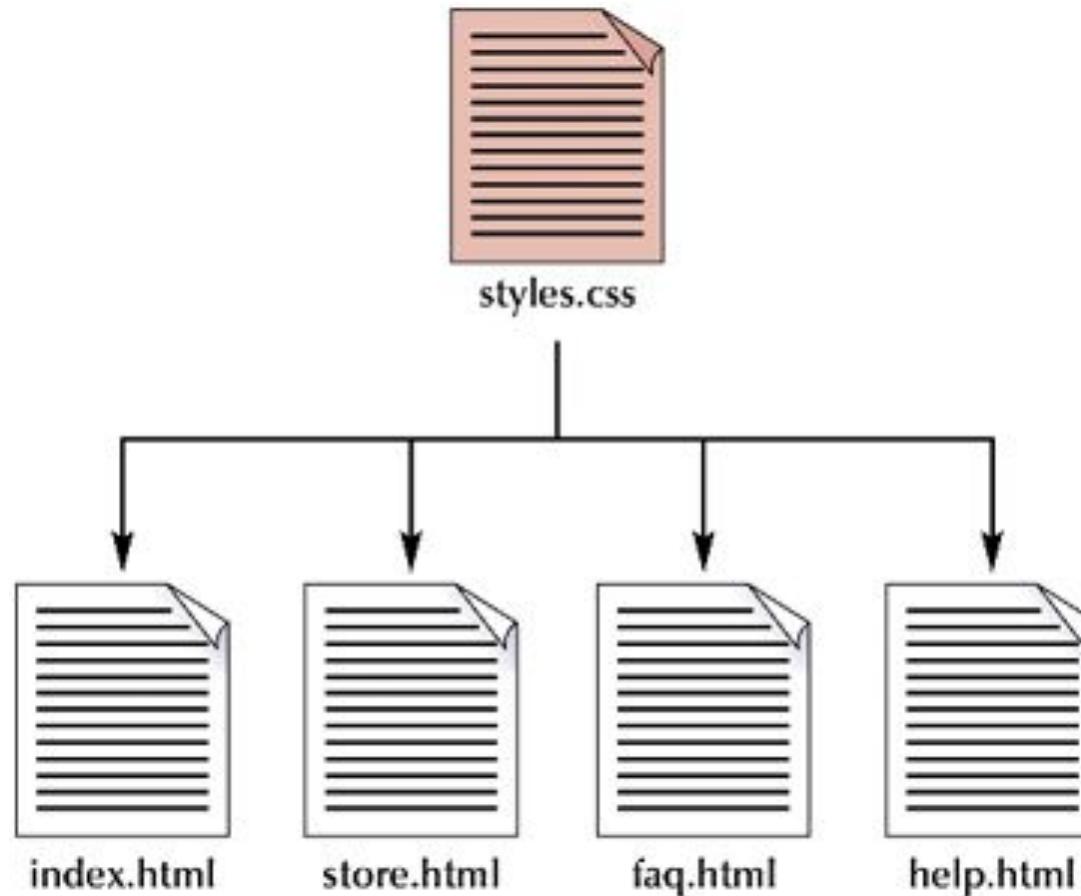
This is Paragraph

External CSS



```
external - Notepad .html file
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<h1>This is heading</h1>
<p>This is Paragraph</p>
</body>
</html>
```

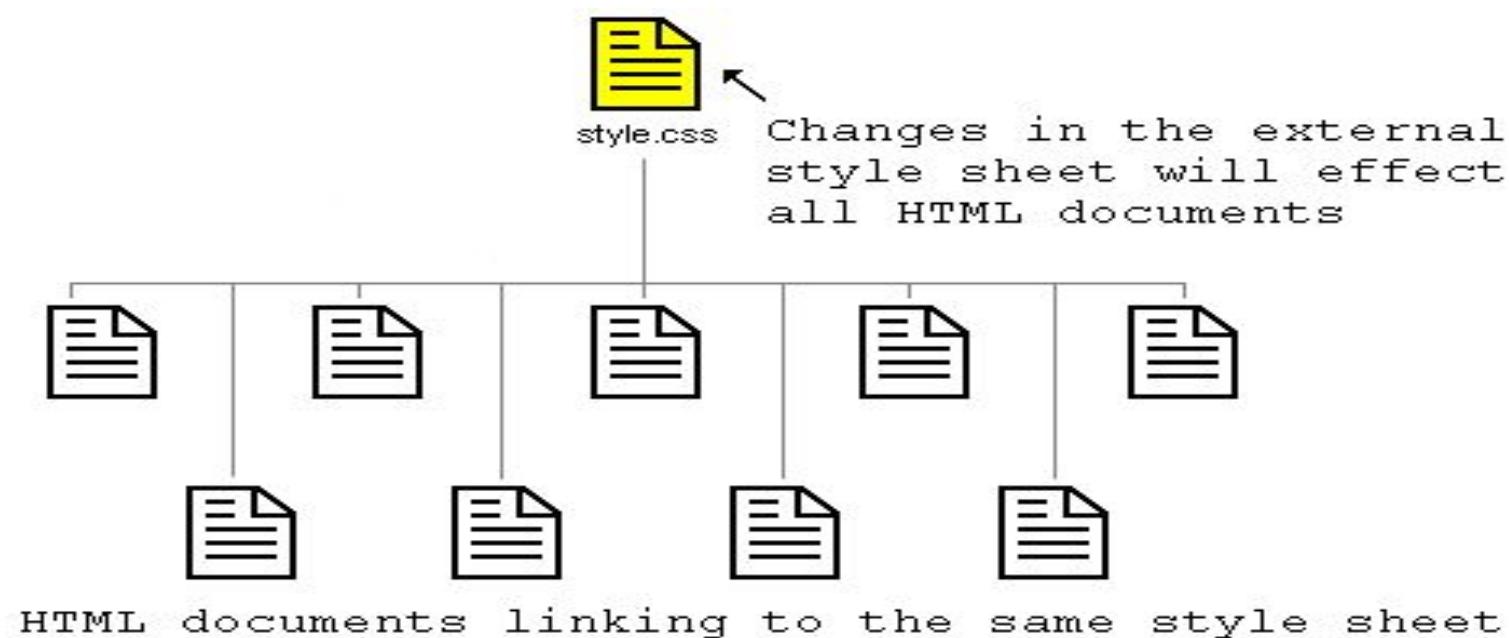
Applying a single style sheet to multiple documents



Advantages of External Style Sheet

Saves Time

can change the look of an entire website by changing just one file



CSS Selectors

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more

Types

Simple selectors (select elements based on name, id, class)

- Element name selector
- Element id selector
- Element class selector
- Universal selector
- Grouping Selectors

Pseudo-class selectors (select elements based on a certain state)

- :hover

CSS Simple Selectors

Selector	Example	Example description
<u>class</u>	.intro	Selects all elements with class="intro"
<u>#id</u>	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element,element,..</u>	div, p	Selects all <div> elements and all <p> elements

CSS Selectors - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p { text-align: center; color: red; }           element selector
#text { text-align: left; color: blue; }         id selector
.styl { text-align: right; color: green; }        class selector
h1, em, b { background-color: lightblue; color: green; } grouped selector
</style>
</head>

<body>
<p>This is an example for simple element selector</p>

<p id="text">This is an example for simple id selector</p>

<p class="styl">This is an example for simple class selector</p>

<h1>This is an example for Grouping selector </h1>           This is an example for simple element selector
<em>This is an example for Grouping selector </em>           This is an example for simple id selector
<b>This is an example for Grouping selector </b>           This is an example for simple class selector
</body>
</html>
```

This is an example for Grouping selector

This is an example for Grouping selector This is an example for Grouping selector

CSS Class Selectors - Example

```
<ul>
  <li class="fruit">Apples</li>
  <li class="vegetable">Carrots</li>
  <li class="fruit">Grapes</li>
  <li class="vegetable">Lettuce</li>
  <li class="fruit">Melons</li>
  <li class="vegetable">Onions</li>
</ul>
```

elements

```
.fruit {color: red}
.vegetable {color: green}
```

styles

• Apples
• Carrots
• Grapes
• Lettuce
• Melons
• Onions

rendered content

CSS Universal Selector - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>

<h1>Hello world!</h1>

<p>This is an example of universal selector </p>
<h1>Hello!</h1>
<i>Good Going!</i>

</body>
</html>
```

Universal selector
- applies to all the
elements in the
page

Hello world!

This is an example of universal selector

Hello!

Good Going!

CSS pseudo classes

A pseudo-class is used to **define a special state of an element**

Example

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Pseudo-class names are not case-sensitive

Types:

- :hover
- :visited
- :active
- :link

To be effective:

- a:hover MUST come after a:link and a:visited in the CSS definition
- a:active MUST come after a:hover in the CSS

CSS pseudo classes - example

```
<!DOCTYPE html>
<html>
<head>
<style>

/* visited link */ changes to green
if visited
a:visited { color: green; }

/* unvisited link */ changes to red if
unvisited
a:link { color: red; }

/* mouse over link */ changes color
when mouse over
a:hover { color: hotpink; }

/* selected link */ applies when the
link is active
a:active { color: blue; }

</style>
</head>
<body>

<p><b><a href="selectors.html" target="_blank">click here</a></b></p>
<a href="image1.jpeg"> click me </a>

<br> <a href = "Internal CSS.html"> click sample </a>

</body>
</html>
```

[click here](#)
[click me](#)
[click sample](#)

[click here](#)
[click me](#)
[click sample](#)

[click here](#)
[click me](#)
[click sample](#)

CSS other pseudo classes - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p:first-child { color: blue; }
.highlight:hover { color: #ff0000; }
a:hover { color: green; }
</style>
</head>

<body>
<p>First use of p tag</p>
<p>second use of p tag</p>

<a class="highlight" href="pseudoclass.html">class hover</a>

<p class="highlight"> Embedding hover within class </p>

<a href=" Inline CSS.html"> Simple hover </a>
</body>
</html>
```

First use of p tag

second use of p tag

class hover

Embedding hover within class

Simple hover

CSS Inheritance: - which style prevails when several are present?

- Inline (local) overrides internal & external styles
- Internal style sheet overrides external styles

Cascading Priority

The way styles will be used when there is more than one style specified for an HTML element:

Browser default

External Style Sheet (Linked) (in an external .css file)

Internal Style Sheet (embedded) (inside the <head> tag)

Inline Style (Local) (inside HTML element)

An inline style (inside an HTML element) has the highest priority, which means that it will override every style declared inside the <head> tag, in an external style sheet, and in the browser (default value).



CSS PROPERTIES

CSS Properties

- ❑ Properties
 - ❑ Color
 - ❑ Background
 - ❑ Border
 - ❑ Font
 - ❑ Margin

Color Property

- ❑ Color - set the font color

Color: value;

- ❑ Examples

- ❑ color:blue;
- ❑ color: "#44ffaa";
- ❑ color:rgb(0,255,0);

Color Property

□ value of color can be given as below:

- a valid color name - like "red"
- an RGB value - like "rgb(255, 0, 0)"
- a HEX value - like "#ff0000"

□ Note:

. hexadecimal values between 00 and FF

. Each parameter `rgb(red, green, blue)` defines the intensity of the color between 0 and 255..

Color Property - Example

```
<h2 style="color: "green">  
    text displayed in green  
</h2>
```

Background property

- The background property is a shorthand property for:
 - background-color
 - background-image
 - background-position
 - background-size
 - background-repeat
 - background-origin
 - background-clip
 - background-attachment

Background property

Value	Description
<u>background-color</u>	Specifies the background color to be used
<u>background-image</u>	Specifies ONE or MORE background images to be used
<u>background-position</u>	Specifies the position of the background images
<u>background-size</u>	Specifies the size of the background images
<u>background-repeat</u>	Specifies how to repeat the background images
<u>background-origin</u>	Specifies the positioning area of the background images
<u>background-clip</u>	Specifies the painting area of the background images
<u>background-attachment</u>	Specifies whether the background images are fixed or scrolls with the rest of the page

Background property

background-color

background-color: red;

background-image

background-image: url("image.jpg");

By default: placed at the top-left corner of an element, and repeated both vertically and horizontally.

Background property

background-position

sets the starting position of a background image

background-position: *value*;

Value	Description
left top	If you only specify one keyword, the other value will be "center"
left center	
left bottom	
right top	
right center	
right bottom	
center top	
center center	
center bottom	
<i>x</i> % <i>y</i> %	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. Default value is: 0% 0%
<i>xpos</i> <i>ypos</i>	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other CSS units . If you only specify one value, the other value will be 50%. You can mix % and positions

Background property

background-size

`background-size: auto;`

Value	Description
auto	Default value. The background image is displayed in its original size
length	Sets the width and height of the background image. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto". Read about length units
percentage	Sets the width and height of the background image in percent of the parent element. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"
cover	Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges
contain	Resize the background image to make sure the image is fully visible

background-repeat

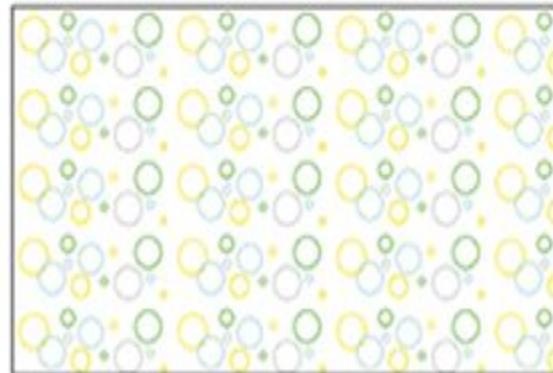
`background-repeat: repeat-y;`

Value	Description
repeat	The background image is repeated both vertically and horizontally. The last image will be clipped if it does not fit. This is default
repeat-x	The background image is repeated only horizontally
repeat-y	The background image is repeated only vertically
no-repeat	The background-image is not repeated. The image will only be shown once
space	The background-image is repeated as much as possible without clipping. The first and last images are pinned to either side of the element, and whitespace is distributed evenly between the images
round	The background-image is repeated and squished or stretched to fill the space (no gaps)

Background repeat examples:



background image



background-image:repeat



background-image:repeat-y



background-image:repeat-x



background-image:no-repeat

Background property

background-origin

background-origin: content-box;

Value	Description
padding-box	Default value. The background image starts from the upper left corner of the padding edge
border-box	The background image starts from the upper left corner of the border
content-box	The background image starts from the upper left corner of the content

background-clip

background-clip: padding-box;

defines how far the background (color or image) should extend within an element.

Value	Description
border-box	Default value. The background extends behind the border
padding-box	The background extends to the inside edge of the border
content-box	The background extends to the edge of the content box

Background property

background-attachment

`background-attachment: fixed;`

sets whether a background image scrolls with the rest of the page, or is fixed

Value	Description
scroll	The background image will scroll with the page. This is default
fixed	The background image will not scroll with the page
local	The background image will scroll with the element's contents

Background-shorthand Property

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

```
<head>
<style>
body {
  background: #ffffff url("tree.jpg") no-repeat right bottom;
  margin-right: 200px;
}
</style>
</head>
```

Background – Example 1

```
<!DOCTYPE html>
<html>
<head>
<title>Background Images</title>
<style type = "text/css">
body { background-image: url("Tulips.jpg");
       background-position: 70px 30px;
       background-repeat: no-repeat;
       background-attachment: fixed;
       background-color: lightgrey;
       background-size: auto;

}

p { font-size: 18pt;
   color: Darkblue;
   text-indent: 1em;
   font-family: arial; }

</style>
</head>
<body>
<pre> This example uses the background-image,
background-position and background-attachment styles to place the
logo in the bottom-right corner of the page.
Notice how the logo stays in the proper position when
you resize the browser window. The background-color fills in where
there is no image.
</pre>
</body>
</html>
```

This example uses the `background-image`,
`background-position` and `background-attachment` styles to place the
logo in the bottom-right corner of the page.
Notice how the logo stays in the proper position when
you resize the browser window. The `background-color` fills in where
there is no image.



Background – Example2

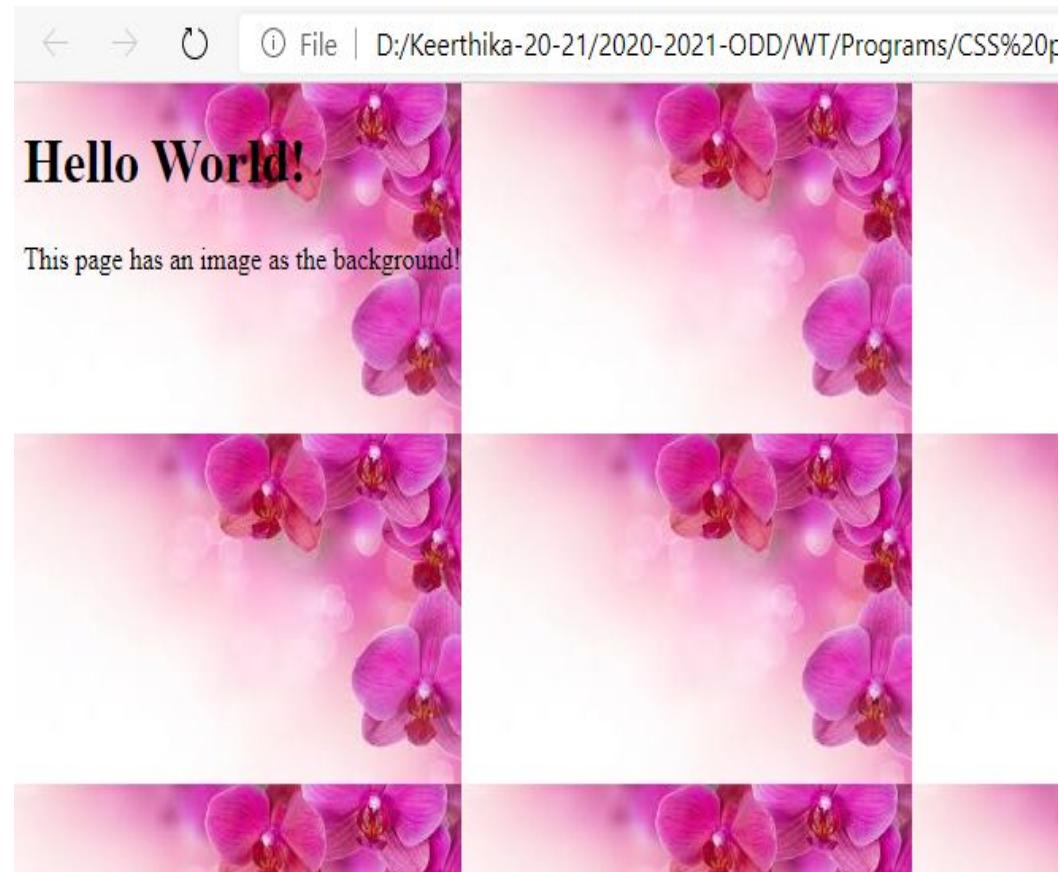
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("cssbgimg.jpg");
    background-repeat: repeat;
}

</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the background!</p>

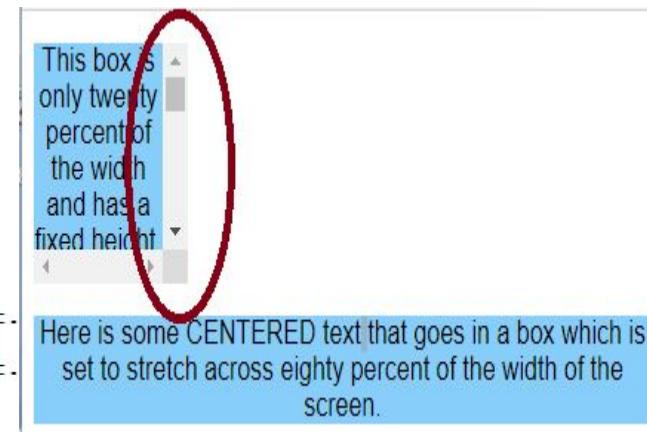
</body>
</html>
```



CSS – Element Dimensions

- In addition to positioning elements, CSS rules can **specify the actual dimensions** of each page element.
- Overflow: Scroll, auto, hidden, visible

```
<!DOCTYPE html>
<html>
<head>
<meta charset = "utf-8">
<title>Box Dimensions</title>
<style type = "text/css">
p { background-color: lightskyblue;
  font-family: arial; }
</style>
</head>
<body>
<p style = "width: 20%; height:120px; text-align: center;
overflow: scroll">
This box is only twenty percent of the width and has a fixed height.
What do we do if it overflows? Set the overflow property to scroll!
This box is only twenty percent of the width and has a fixed height.
What do we do if it overflows? Set the overflow property to scroll!
</p>
<p style = "width: 80%; text-align: center">
Here is some CENTERED text that goes in a box which is set to stretch across eighty
the width of the screen.
</p>
</body>
</html>
```



Width and height

- **height and width** properties are used to set the height and width of an element
- <div> above occurs when the browser window is smaller than the width of the element (500px)
- browser then adds a horizontal scrollbar to the page

CSS - Height and width Example

```
<!DOCTYPE html> <html><head> border </head>
<style>
<style>
p.one {
    height: 200px;
    width: 500px;
    background-color: lightgray;
}
</style> </head>
<body>
<p class="one"> Sample1 <br>
    Sample2 <br> height, width, color, style </p> </body>
</body></html>
```

border
Sample1
Sample2
height, width, color, style

CSS - Height and width Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2>Set the height and width of an element</h2>

<p>This div element has a height of 200px and a width of 50%:</p>

<div></div>
|
</body>
</html>
```

Set the height and width of an element

This div element has a height of 200px and a width of 50%:



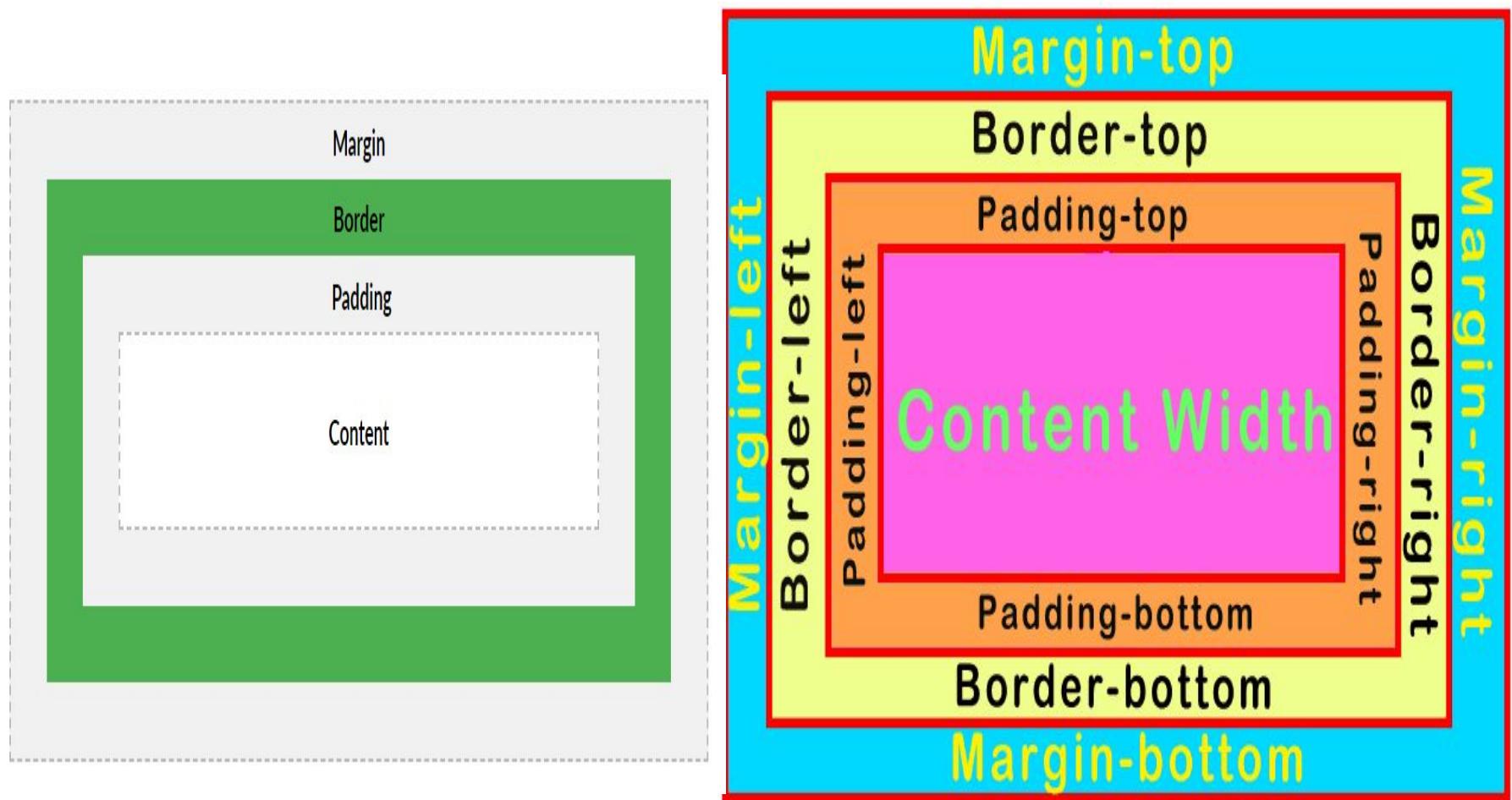
NOTE- width and height

- problem with the `<div>` in the above example
- when the browser window is smaller than the width of the element (500px).
- The browser then adds a horizontal scrollbar to the page.

CSS Box Model

- ❑ All HTML elements can be considered as boxes
- ❑ CSS box model is essentially a **box that wraps around every HTML element.**
- ❑ It consists of
 - ❑ **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

BOX MODEL



BOX MODEL - Example

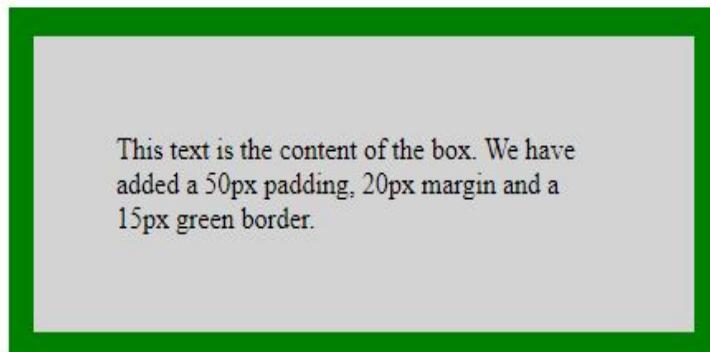
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    background-color: lightgrey;
    width: 300px;
    border: 15px solid green;
    padding: 50px;
    margin: 20px;
}
</style></head>
<body>
<h2>Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.
</p>

<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.</div>

</body>
</html>
```

Box Model

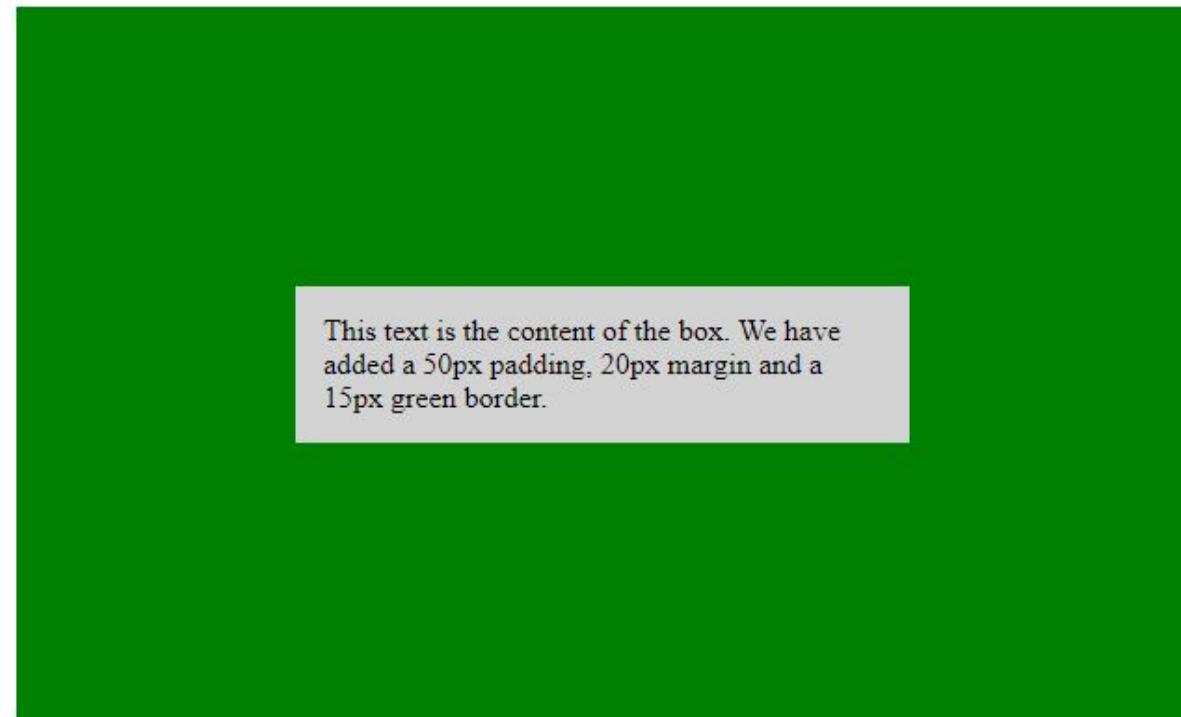
The CSS box model is essentially a box that wraps around every HTML element. It consists of borders, padding, margins, and the actual content.



This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.

BOX MODEL - Example

```
background-color: lightgrey;  
width: 300px;  
border: 15px solid green;  
padding: 15px;  
margin: 20px;
```



This text is the content of the box. We have added a 15px padding, 20px margin and a 15px green border.

BORDER PROPERTY

Shorthand property

border: 2px solid pink

border-left: 2px solid pink

border-right: 2px solid pink

border-bottom: 2px solid pink

border-top: 2px solid pink

border-left

border-left-width

border-left-color

border-left-style

border-right

border-right-width

border-right-color

border-right-style

The **border** property is a shorthand property

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

border-width: 2px

border-style: solid

border-color: pink

border-bottom

border-bottom-width

border-bottom-color

border-bottom-style

border-top

border-top-width

border-top-color

border-top-style

<u>border</u>	Sets all the border properties in one declaration
<u>border-bottom</u>	Sets all the bottom border properties in one declaration
<u>border-bottom-color</u>	Sets the color of the bottom border
<u>border-bottom-style</u>	Sets the style of the bottom border
<u>border-bottom-width</u>	Sets the width of the bottom border
<u>border-color</u>	Sets the color of the four borders
<u>border-left</u>	Sets all the left border properties in one declaration
<u>border-left-color</u>	Sets the color of the left border
<u>border-left-style</u>	Sets the style of the left border
<u>border-left-width</u>	Sets the width of the left border
<u>border-radius</u>	Sets all the four border-* -radius properties for rounded corners
<u>border-right</u>	Sets all the right border properties in one declaration
<u>border-right-color</u>	Sets the color of the right border
<u>border-right-style</u>	Sets the style of the right border
<u>border-right-width</u>	Sets the width of the right border
<u>border-style</u>	Sets the style of the four borders
<u>border-top</u>	Sets all the top border properties in one declaration
<u>border-top-color</u>	Sets the color of the top border
<u>border-top-style</u>	Sets the style of the top border
<u>border-top-width</u>	Sets the width of the top border
<u>border-width</u>	Sets the width of the four borders

CSS Border Style

```
<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;}
p.mix {border-style: dotted dashed solid double;}
h2.test{border:2px dotted red;
color:green; }
</style>
</head>
<body>
<h2 class="test"> The border-style Property
<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>
<p class="mix">A mixed border.</p>
</h2>
</body>
```

The border-style Property

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

CSS Border Style

```
<!DOCTYPE html>
<html>
<head>
<meta charset = "utf-8"> <title>Borders</title> <style type = "text/css">
div { text-align: center; width: 50%; border-width: 6px;
}
.thick { border-width: thick; }
.medium { border-width: medium; }
.thin { border-width: thin; }
.solid { border-style: solid; }
.double { border-style: double; }
.groove { border-style: groove; }
.ridge { border-style: ridge; }
.dotted { border-style: dotted; }
.inset { border-style: inset; }
.outset { border-style: outset; }
.dashed { border-style: dashed; }
.red { border-color: red; }
.blue { border-color: blue; }
</style> </head>
<body>
<p><div class = "solid red">Solid border</div></p>
<p><div class = "double">Double border</div></p>
<p><div class = "groove">Groove border</div></p>
<p><div class = "ridge">Ridge border</div></p>
<p><div class = "dotted">Dotted border</div></p>
<p><div class = "inset">Inset border</div></p>
<p><div class = "outset">Outset border</div></p>
<p><div class = "thick dashed">Thick dashed border</div></p>
<p><div class = "thin red solid">Thin red solid border</div></p>
<p><div class = "medium blue outset">Medium blue outset border</div></p>
</body> </html>
```

combined classes

Solid border

Double border

Groove border

Ridge border

Dotted border

Inset border

Outset border

Thick dashed border

Thin red solid border

Medium blue outset border

CSS Border Width

```
<style>  
p.one { border-style: solid; border-width: 5px; }  
p.two { border-style: solid; border-width: medium; }  
p.three { border-style: dotted; border-width: 2px; }  
p.four { border-style: dotted; border-width: thick; }  
  
p.five { border-style: double; border-width: 15px; }  
p.six { border-style: double; border-width: thick; }  
</style>  


Some text.



Some text.



Some text.



Some text.

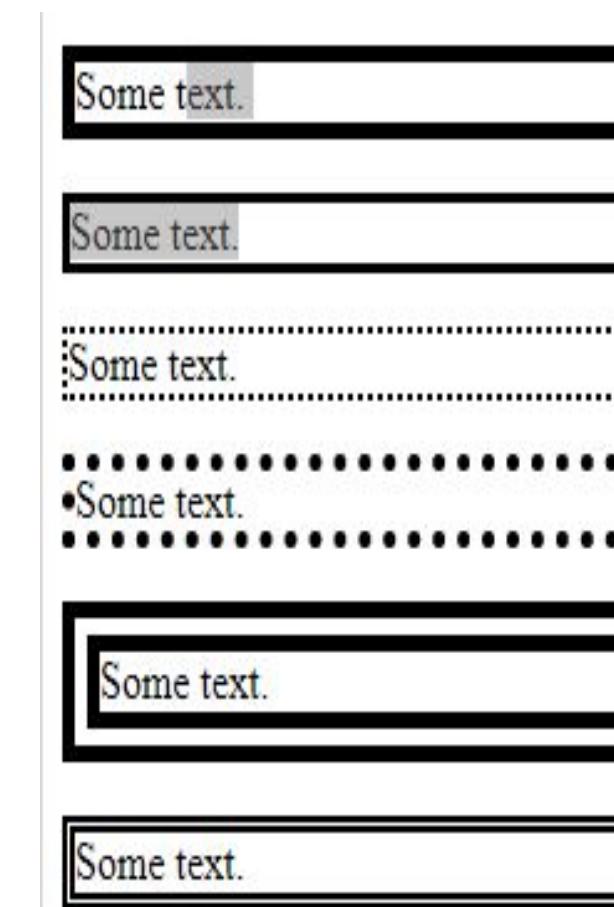


Some text.



Some text.


```



CSS Border-color

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

Red border

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

Green border

```
p.three {  
    border-style: dotted;  
    border-color: blue;  
}
```

Blue border

Style, color, width, top-style, left-style

- Shorthands

```
<!DOCTYPE html> <html><head> border </head>
<style>
p.one { border-style:solid;
         border-color:red;
         border-width:5px;
         border-top-style: dotted; }

p.two { border: 5px solid red;
         border-left-style: dotted;
         border-radius: 15px; }

</style> <body>
<p class="one"> Sample1 </p> <br>
<p class="two"> Sample2 <br> Shorthand Property - width,color, style </p>
</body>
```

border

Sample1

Sample2

Shorthand Property - width,color, style

MARGIN PROPERTY

margin properties are used to generate space around elements[outside border]

CSS `margin` properties are used to create space around elements, outside of any defined borders.

margin:10px

margin-top:5px

margin-right:10px

margin-bottom:10px

margin-left:10px

CSS- Box Margin - Example

```
<!DOCTYPE html> <html><head> border </head>
<style>
p.one { border-style:solid;
         border-color:red;
         border-width:5px;
         border-top-style: dotted;

         margin-top: 50px;
         margin-bottom: 10px;
         margin-right: 150px;
         margin-left: 80px; }

p.two { border: 5px solid red;
         border-left-style: dotted;
         border-radius: 15px;

         margin: 5px 100px 150px 100px; }

        }
```

border

Sample1

Sample2
Shorthand Property - width,color, style

```
<p class="one"> Sample1 </p> <br>
<p class="two"> Sample2 <br> Shorthand Property - width,color, style </p>
</body>
```

PADDING PROPERTY

Padding properties are used to generate space around contents[inside border]

Shorthand property

```
padding: 2px ;  
padding-left::2px ;  
padding-right:2px ;  
padding-bottom:2px;  
padding-top:2px ;
```

CSS- Box-Padding Example

```
<!DOCTYPE html> <html><head> border </head>
<style>
p.one { border-style:solid;
    border-color:red;
    border-width:5px;
    border-top-style: dotted;
    margin-top: 50px;
    margin-bottom: 10px;
    margin-right: 150px;
    margin-left: 80px;

    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
p.two {
    border: 5px solid red;
    border-left-style: dotted;
    border-radius: 15px;
    margin: 5px 100px 150px 100px;

    padding: 25px 50px 75px 100px;
}
</style> </head>
<body>
<p class="one"> Sample1 </p> <br>
<p class="two"> Sample2 <br> Shorthand Property - width,color, style </p>
</body></html>
```

border

Sample1

Sample2
Shorthand Property -
width,color, style

MAX-width, height, Min - width, height properties

<u>max-height</u>	Sets the maximum height of an element
<u>max-width</u>	Sets the maximum width of an element
<u>min-height</u>	Sets the minimum height of an element
<u>min-width</u>	Sets the minimum width of an element

- will improve the browser's handling of small windows

Min-height and max-height

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  max-height: 600px;
  min-height: 400px;
  background-color: powderblue;
}
</style>
</head>
<body>

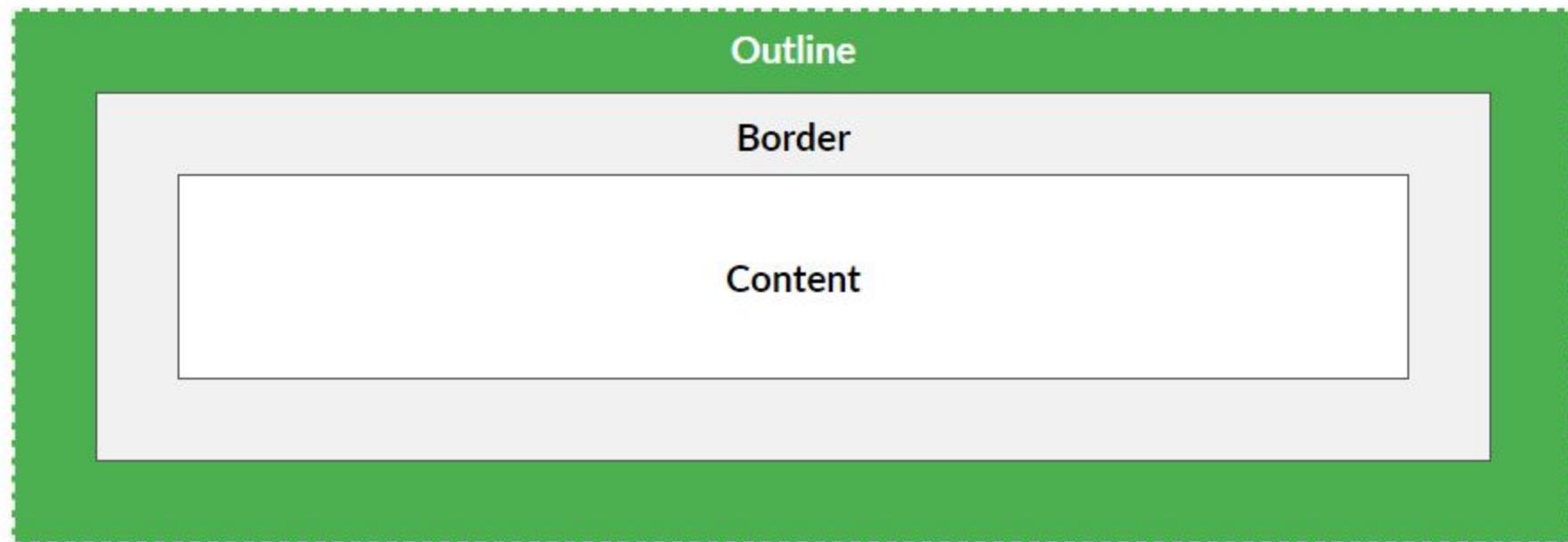
<h2>Set the max-height and min-height of an element</h2>
<p>Resize the browser window to see the effect.</p>

<div>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.</div>

</body>
</html>
```

CSS-OUTLINE Property

Outline is a line that is drawn around elements, OUTSIDE the border.



CSS-OUTLINE Property

<u>outline-color</u>	Specifies the color of the outline
<u>outline-style</u>	Specifies the style of the outline
<u>outline-width</u>	Specifies the width of outline

Shorthand Property

Outline	width color style
---------	-------------------

```
p {  
    border: 10px solid red;  
    outline:40px green solid ;  
    padding:20px;  
    margin:100px      }
```

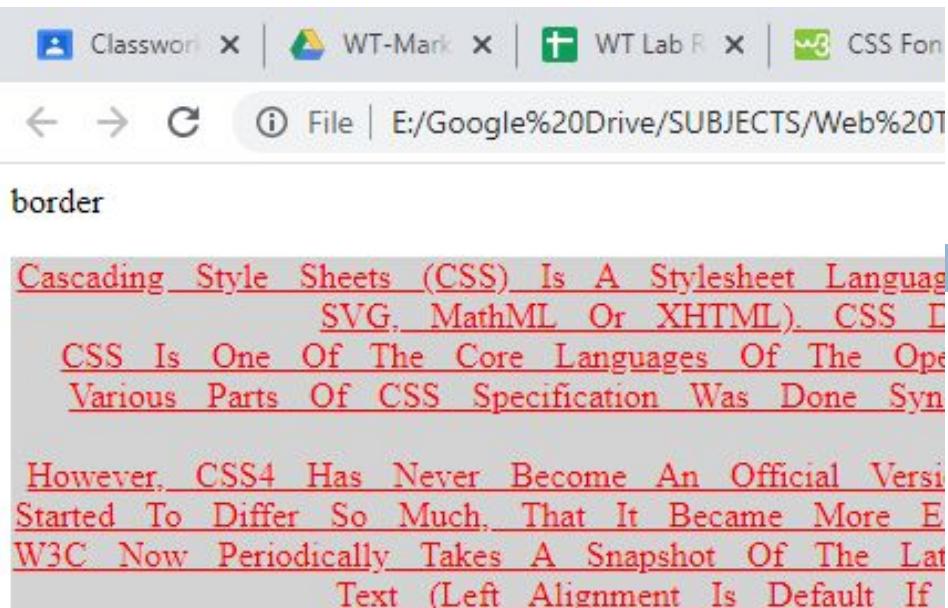
CSS Text Property

- ❑ `color:rgb(10,0,255)`
- ❑ `text-align :center|left|right|justify`
- ❑ `text-transform: capitalize| lowercase|uppercase|none`
- ❑ `text-decoration: none|underline|overline|line-through`
- ❑ `text-indent: length[px ,em]or %`
- ❑ `line-height: normal|length`
- ❑ `letter-spacing: normal|length(px)`
- ❑ `word-spacing: normal|length(px)`
- ❑ `direction: ltr|rtl`
- ❑ `text-shadow : h-shadow v-shadow blur-radius color|none`

https://www.w3schools.com/cssref/tryit.asp?filename=trycss_text-indent

CSS Text

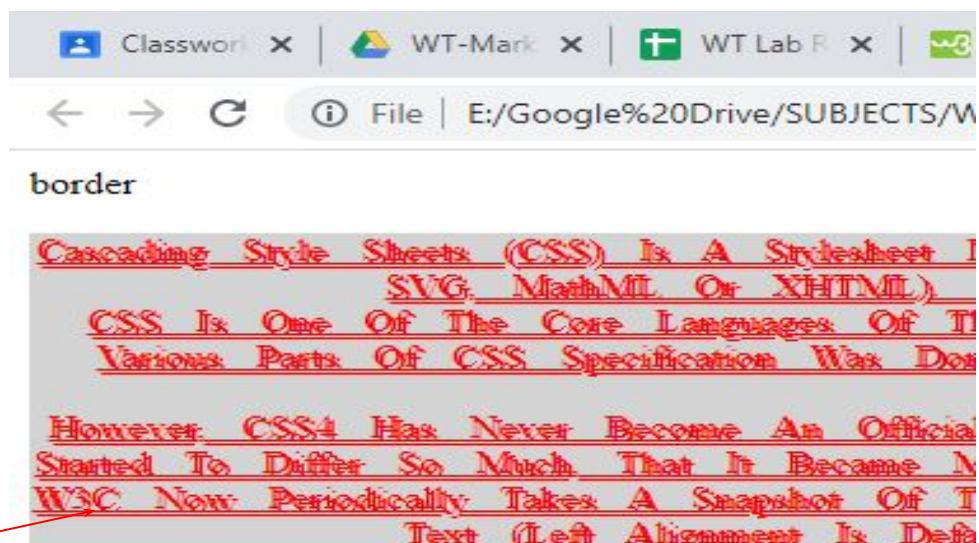
```
p.one {  
color:red;  
background-color: lightgray;  
text-align: center;  
text-decoration: overline;  
text-decoration: line-through;  
text-decoration: underline;  
text-transform: uppercase;  
text-transform: lowercase;  
text-transform: capitalize;  
line-height: 0.8;  
word-spacing: 10px;  
word-spacing: -5px;  
text-shadow: 2px 2px 5px red; }
```



The screenshot shows a browser window with the following tabs: Classwor, WT-Mark, WT Lab R, and CSS Fon. The address bar shows the file path: E:/Google%20Drive/SUBJECTS/Web%201. The page content is a single paragraph with the following styling applied:

- color:red;** The entire paragraph is displayed in red text.
- background-color: lightgray;** The background color of the paragraph is light gray.
- text-align: center;** The text is centered within the paragraph.
- text-decoration: overline;** The text is underlined with a single horizontal line above the text.
- text-decoration: line-through;** The text is crossed out with a single horizontal line through the text.
- text-decoration: underline;** The text is underlined with a single horizontal line below the text.
- text-transform: uppercase;** The text is converted to uppercase.
- text-transform: lowercase;** The text is converted to lowercase.
- text-transform: capitalize;** The text is converted to title case.
- line-height: 0.8;** The line height is set to 0.8 times the font size.
- word-spacing: 10px;** The word spacing is increased to 10 pixels.
- word-spacing: -5px;** The word spacing is decreased to -5 pixels.
- text-shadow: 2px 2px 5px red;** The text has a shadow with a horizontal offset of 2px, a vertical offset of 2px, and a blur radius of 5px, colored red.

The text content of the paragraph is: "Cascading Style Sheets (CSS) Is A Stylesheet Language For Structuring And Presenting Content (e.g. HTML, SVG, MathML Or XHTML). CSS Is One Of The Core Languages Of The Open Web Platform. The Various Parts Of CSS Specification Was Done Separately By W3C. However, CSS4 Has Never Become An Official Version. CSS3 Has Started To Differ So Much That It Became More Easier To Use. W3C Now Periodically Takes A Snapshot Of The Latest CSS Text (Left Alignment Is Default If No Alignment Is Specified)."



The screenshot shows a browser window with the following tabs: Classwor, WT-Mark, WT Lab R, and CSS Fon. The address bar shows the file path: E:/Google%20Drive/SUBJECTS/Web%201. The page content is a single paragraph with the following styling applied:

- color:red;** The entire paragraph is displayed in red text.
- background-color: lightgray;** The background color of the paragraph is light gray.
- text-align: center;** The text is centered within the paragraph.
- text-decoration: overline;** The text is underlined with a single horizontal line above the text.
- text-decoration: line-through;** The text is crossed out with a single horizontal line through the text.
- text-decoration: underline;** The text is underlined with a single horizontal line below the text.
- text-transform: uppercase;** The text is converted to uppercase.
- text-transform: lowercase;** The text is converted to lowercase.
- text-transform: capitalize;** The text is converted to title case.
- line-height: 0.8;** The line height is set to 0.8 times the font size.
- word-spacing: 10px;** The word spacing is increased to 10 pixels.
- word-spacing: -5px;** The word spacing is decreased to -5 pixels.
- text-shadow: 2px 2px 5px red;** The text has a shadow with a horizontal offset of 2px, a vertical offset of 2px, and a blur radius of 5px, colored red.

The text content of the paragraph is: "Cascading Style Sheets (CSS) Is A Stylesheet Language For Structuring And Presenting Content (e.g. HTML, SVG, MathML Or XHTML). CSS Is One Of The Core Languages Of The Open Web Platform. The Various Parts Of CSS Specification Was Done Separately By W3C. However, CSS4 Has Never Become An Official Version. CSS3 Has Started To Differ So Much That It Became More Easier To Use. W3C Now Periodically Takes A Snapshot Of The Latest CSS Text (Left Alignment Is Default If No Alignment Is Specified)."

CSS Text - Example

```
<html>
  <head>
    <style>
      h1{
        text-shadow:5px 5px 10px red;
      }
    </style>
  </head>
  <body>
    <h1>Text-shadow
      effect</h1>
  </body>
</html>
```

LIST properties

- ❑ CSS list properties allows to
 - ❑ Set **different list item markers** for ordered lists
 - ❑ Set different list item markers for unordered lists
 - ❑ Set **an image as the list item marker**
 - ❑ Add **background colors to lists and list items**

List Properties

□ Shorthand Property:

list-style: *list-style-type* *list-style-position* *list-style-image*

list-style-image: none|*url*

list-style-position: inside|outside

List-style:disc|circle|square|none|lower-alpha|upper-alpha|lower-roman|upper-roman

List Properties- Example

```
<html><head>
<style>
ul {
    list-style: circle inside url("cssbgimg.jpg");
}
</style>
</head>
<body>
<ul>
    <li>linux</li>
    <li>windows</li>
    <li>mach</li>
</ul>
</body> </html>
```



linux



windows



CSS Font Properties

Property/Value	Description
<u>font-style</u>	Specifies the font style. Default value is "normal"
<u>font-variant</u>	Specifies the font variant. Default value is "normal"
<u>font-weight</u>	Specifies the font weight. Default value is "normal"
<u>font-size/line-height</u>	Specifies the font size and the line-height. Default value is "normal"
<u>font-family</u>	Specifies the font family. Default value depends on the browser

Property	Description
<u>font-size</u>	medium xx-small x-small small large x-large xx-large smaller larger length
<u>font-style</u>	normal italic oblique
<u>font-weight</u>	normal bold bolder lighter number[100 to 900]

CSS Font Properties

font-family: Times New Roman;

font-style: italic; // normal, oblique

font-size: 30px;

font: 20px Arial, sans-serif;

font: italic bold 12px/30px Georgia, serif;

CSS Font Properties - Example

```
<html><head>
<style>
p.normal {
    font-style: normal;
    font-weight: bold;
}
p.italic {
    font-style: italic;
}
p.oblique {
    font-style: oblique;
}
p.serif {
    font-family: "Times New Roman", Times, serif;
}
p.sansserif {
    font-family: Arial, Helvetica, sans-serif;

</style>
</head>
<body>

<p class="normal serif">This is a paragraph, normal.</p>

<p class="italic sansserif">This is a paragraph, italic.</p>

<p class="oblique">This is a paragraph, oblique.</p>

</body> </html>
```

This is a paragraph, normal.

This is a paragraph, italic.

This is a paragraph, oblique.

CSS – Positioning Elements

Position Property

Static

not affected by the top, bottom, left, and right properties

Relative

positioned relative to its normal position

Fixed

always stays in the same place even if the page is scrolled

Absolute

positioned relative to the nearest positioned ancestor

Sticky

is positioned based on the user's scroll position

It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed)

Z-index – for overlapping elements

Elements with higher Z index will be shown at the first layer, others in the back layer

CSS – Relative Positioning

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">

h2.pos_left {
  position: relative;
  left: -20px;
}

h2.pos_right {
  position: relative;
  left: 20px;
}
.super { position: relative;
  top: -1ex
}

</style>
</head>
<body>
<p> the text <span class= "super"> is in superscript</span>.</p>
<h2>Heading with no position</h2>
<h2 class="pos_left">This heading is moved left according to its normal
position</h2>
<h2 class="pos_right">This heading is moved right according to its normal
position</h2>

<p>Relative positioning moves an element RELATIVE to its original position.
</p>
<p>The style "left:-20px" subtracts 20 pixels from the element's original
left position.</p>
<p>The style "left:20px" adds 20 pixels to the element's original left
position.</p>

</body>
</html>
```

the text ^{is in superscript}

Heading with no position

This heading is moved left according to its normal position

Relative positioning moves an element RELATIVE to its original position.
The style "left:-20px" subtracts 20 pixels from the element's original left position.

The style "left:20px" adds 20 pixels to the element's original left position.

This heading is moved right according to its normal position

CSS – Absolute Positioning

```
<!DOCTYPE html>
<html>
<head>
    <meta charset = "utf-8">
    <title>Absolute Positioning</title>

    <style type = "text/css">
        .background_image { position: absolute; top: 0px;
            left: 0px; z-index: 1; }

        .foreground_image { position: fixed; top: 125px;
            left: 200px; z-index: 3; }

        .text { position: absolute; top: 400px; left: 300px; z-index: 2;
            font-size: 20pt; font-family: arial; }
    </style>
</head>
<body>
    <p><img src = "Tulips.jpg" class = "background_image"
        alt = "image" /></p>

    <p><img src = "image1.jpg" class = "foreground_image"
        alt = "image" /></p>

    <p class = "text">Web technology</p>
</body>
</html>
```



CSS – Positioning Elements

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}

div.absolute {
  position: absolute;
  top: 20px;
  right: 0;
  width: 200px;
  height: 50px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: absolute;</h2>

<p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>

<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>

</body>
</html>
```

position: absolute;

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):

This div element has position: relative;

This div element has position: absolute;

CSS-Float Property

- ❑ used for positioning and formatting content
- ❑ e.g. **let an image float left/right** to the text in a container.
 - ❑ left - The element floats to the left of its container
 - ❑ right - The element floats to the right of its container
 - ❑ none - The element does not float

CSS-Float Property- Example

```
<!DOCTYPE html>
<html>
<head>
<style>
|
img {
  float: right;
}
Content----->
</style>
</head>
<body>
<p>In this example, the image will float to the right in the paragraph, and the text in the
<p>
Content-----> </p>
</body> </html>
```

In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.

Content----->



Float- Right & Left



<Div> & tag

HTML Block (DIV) and Inline (SPAN) Elements

<DIV>

defines a division or a section in an HTML document

used as a **container** for HTML elements - which is then styled with CSS or manipulated with JavaScript

easily styled by using the class or id attribute

an **inline container** used to mark up a part of a text, or a part of a document

easily styled by CSS or manipulated with JavaScript using the class or id attribute

much like the **<div>** element, but **<div>** is a block-level element and **** is an inline element.

<Div> tag

```
<!DOCTYPE html>
<html>
<head>
<style>
.myDiv {
    background-color: lightblue;
    text-align: center;
}
</style>
</head>
<body>
```

```
<h1>The div element</h1>
```

```
<div class="myDiv">
<h2>This is a heading in a div element</h2>
<p>This is a paragraph in a div element.</p>
</div>
```

```
<p>This is paragraph text outside the div element.</p>
```

```
</body>
</html>
```

```
<html>
<body>
<div> defines a division or a section in an HTML document.
It is used as a container for HTML elements - which
is then styled with CSS or manipulated with JavaScript
</div>
</body>
</html>
```

The div element

This is a heading in a div element

This is a paragraph in a div element.

This is paragraph text outside the div element.

defines a division or a section in an HTML document. It is used as a container for HTML elements .

 tag

```
<!DOCTYPE html>
<html>
<body>
    <div style="border: 1px solid black"> This is a block element </div>
    <p>This is an inline span <span style="border: 1px solid black">Hello
    World</span> element inside a paragraph.</p>
    <p>The SPAN element is an inline element, and will not start on a new line and
    only takes up as much width as necessary.</p>
</body>
</html>
```

This is a block element

This is an inline span Hello World element inside a paragraph.

The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.

HTML Block and Inline Elements

Block Elements

<address>

<blockquote>

>

<div>

<fieldset>

<figcaption>

<figure>

<footer>

<form>

<h1>-<h6>

<header>

<hr>

<nav>

<p>

<pre>

<section>

<table>

<tfoot>

<video>

Inline Elements

<a>

<abbr>

<acronym>

<bdo>

<button>

<cite>

<code>

<dfn>

<i>

<sub>

<sup>

<input>

<kbd>

<label>

<map>

<object>

<output>

<q>

<script>

<select>

<small>

<textarea>

<time>

<DIV> tag - Example

```
<style>
.main
{
width:100%;
}
.header{
height:20%;
background-color:lightgreen;
width:100%;
}

.nav{
height:10%;
background-color:pink;
width:100%;
}

.left{
height:400px;
width:25%;
border:1px solid black;
float:left;
}


```

```
.content{
height:400px;
width:25%;
border:1px solid black;
float:left;
}
.right{
height:400px;
width:49.5%;
border:1px solid black;
float:right;
}
.footer{
height:50px;
width:100%;
clear:both;
background-color:green;
}
```

```
<div class="main" >
  <div class="header">
  </div>

  <div class="nav">
  </div>

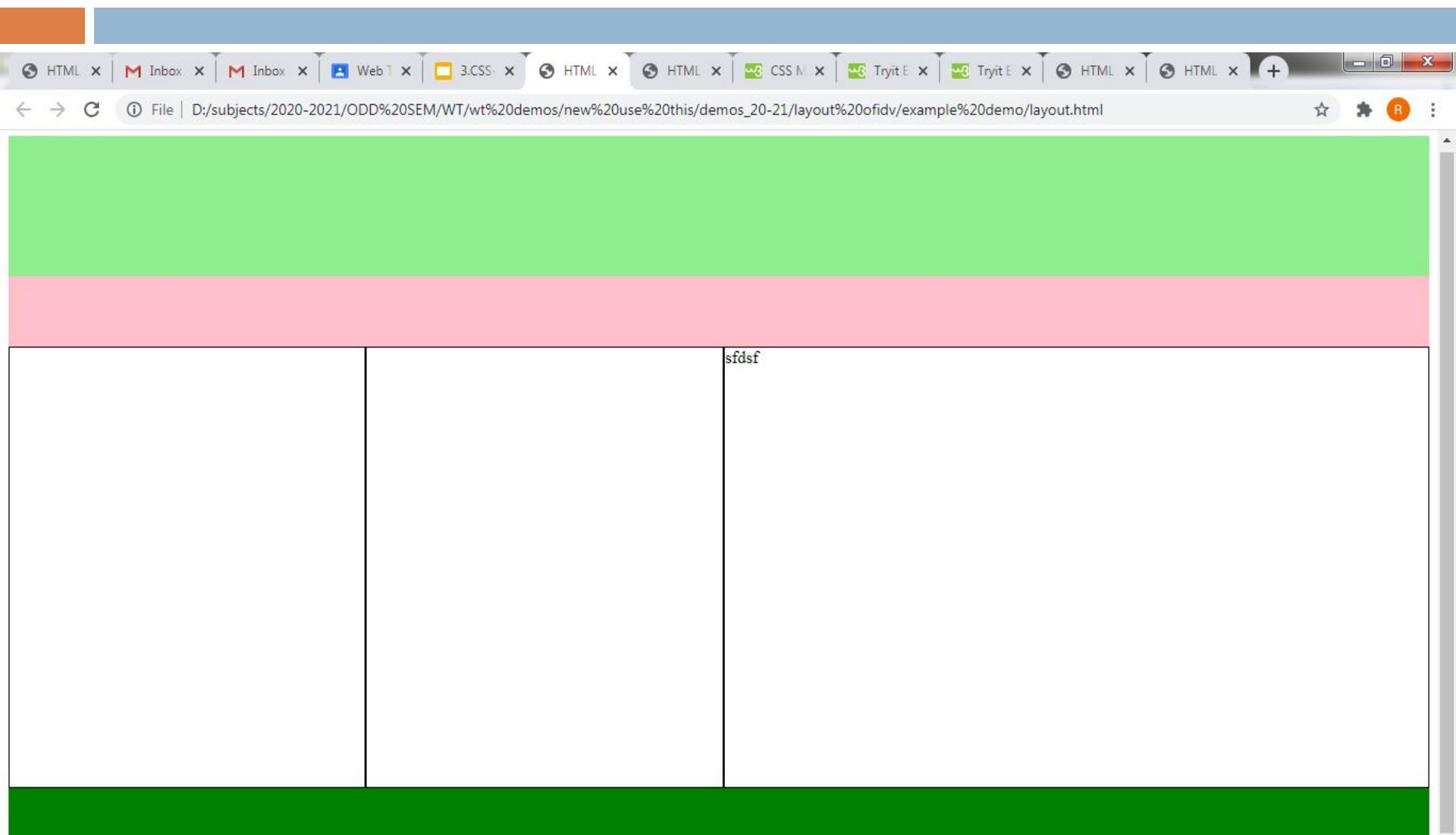
  <div class="left">
  </div>

  <div class="content">
  </div>

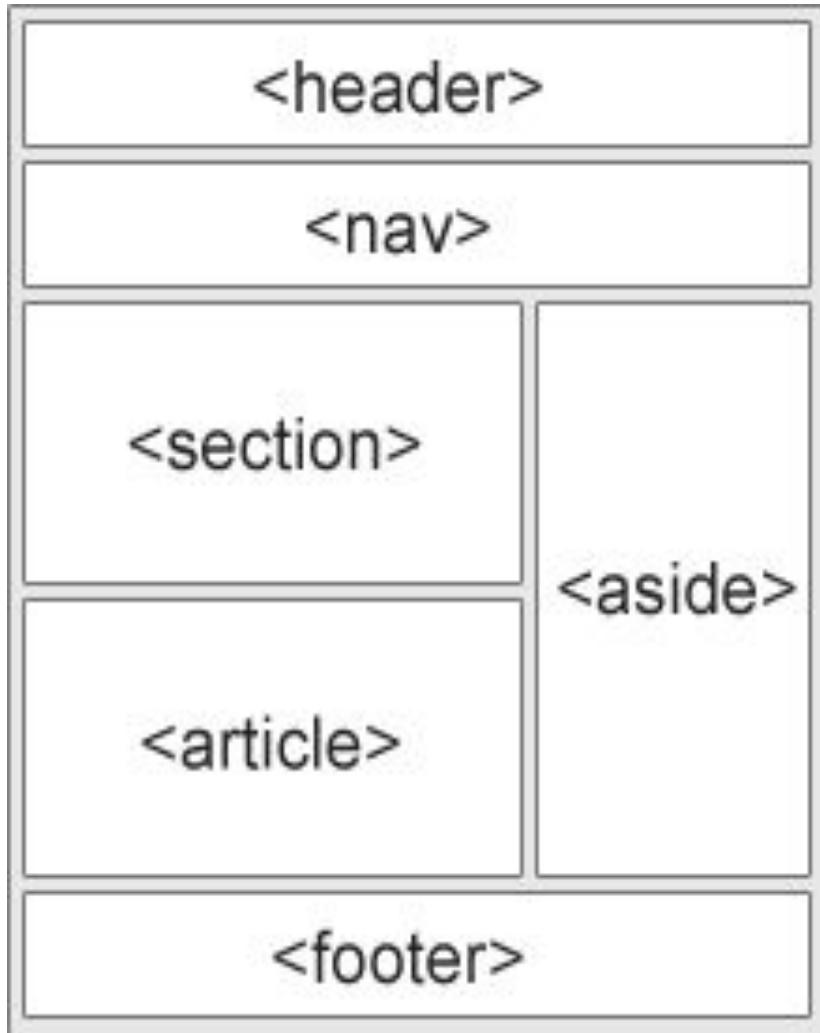
  <div class="right">
    sfdsf
  </div>

  <div class="footer">
  </div>
</div>
```

<DIV> tag - Example



Page Structured Elements



- `<header>` - Defines a header for a document or a section
- `<nav>` - Defines a set of navigation links
- `<section>` - Defines a section in a document
- `<article>` - Defines an independent, self-contained content
- `<aside>` - Defines content aside from the content (like a sidebar)
- `<footer>` - Defines a footer for a document or a section
- `<details>` - Defines additional details that the user can open and close on demand
- `<summary>` - Defines a heading for the `<details>` element

Page Structured Elements

```
<style>
.main
{
width:100%;
}

header
{
height:20%;
background-color:lightgreen;
width:100%;
}

nav{
height:10%;
background-color:pink;
width:100%;
}

.content{
width:60%;
height:400px;
float:left;
}

article{
height:200px;
width:100%;
border:1px solid black;
float:left;
}

section{
height:200px;
width:100%;
border:1px solid black;
float:left;
}

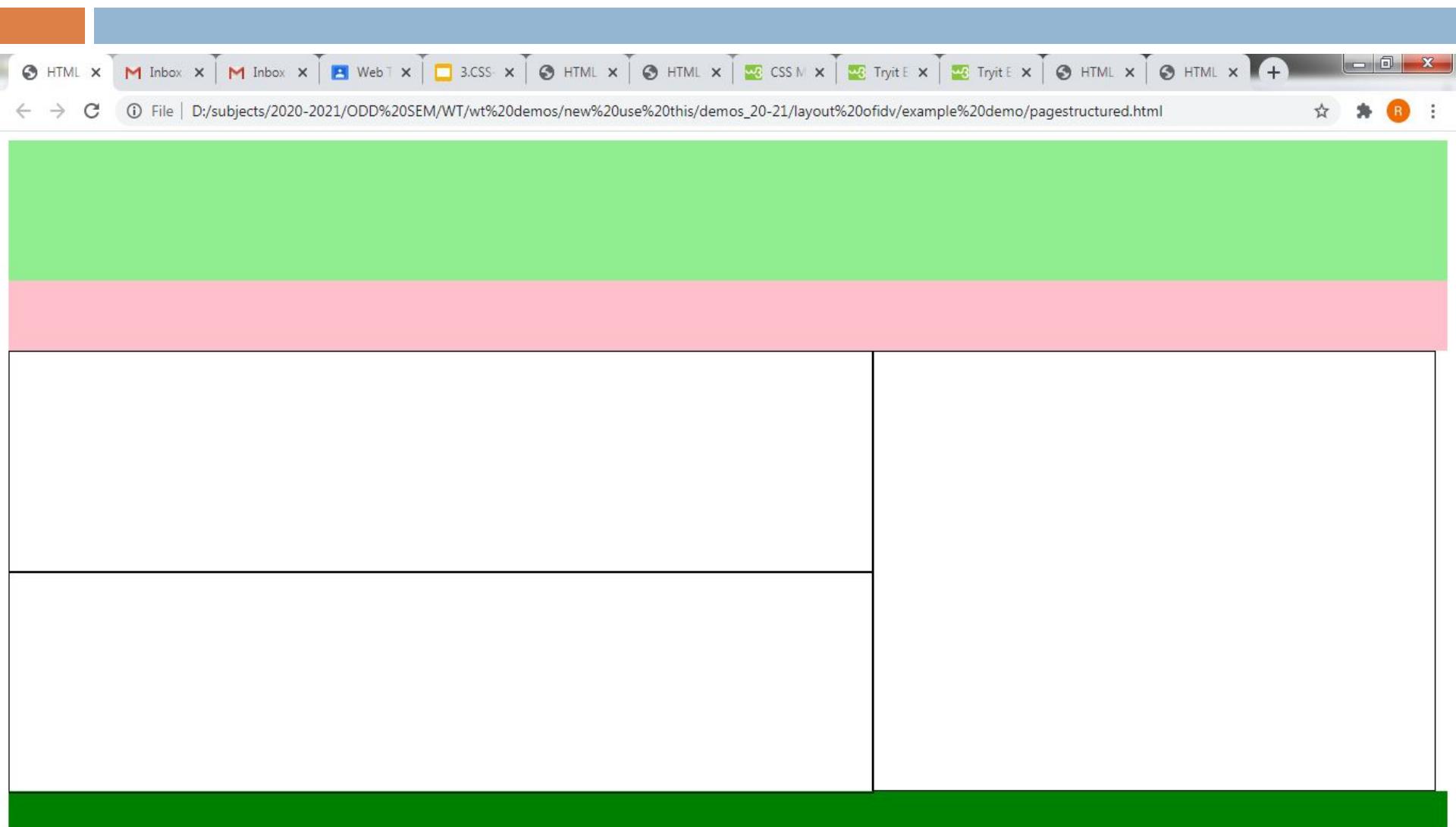
aside{
height:400px;
width:39%;
border:1px solid black;
float:left;
}

footer{
height:40px;
width:100%;
clear:both;
background-color:green;
}

</style>
```

```
<div class="main" >
  <header>
  </header>
  <nav>
  </nav>
  <div class="content">
    <article>
    </article>
    <section>
    </section>
  </div>
  <aside>
  </aside>
</div>
<footer>
</footer>
</div>
```

Page Structured Elements



CSS- NAV Tag

- ❑ Navigation Bar = List of Links
- ❑ using the `` and `` elements makes perfect sense
- ❑ Types
 - ❑ Vertical
 - ❑ Horizontal

CSS-Menu Bar-Vertical

- Menu bars are created using navigation bar
- A navigation bar is basically a list of links

Step 1:

Create the needed menu as list of links

```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
```

- [Home](#)
- [News](#)
- [Contact](#)
- [About](#)

CSS-Menu Bar-Vertical

Step 2:

Remove Bullets and Margins, padding from list

list-style-type: none; - Removes the bullets.

Set **margin: 0;** and **padding: 0;** to remove browser default settings

```
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
</style>
```



Home
News
Contact
About

CSS-Menu Bar-Vertical

Step 3: Vertical navigation bar

To build a vertical navigation bar style the `<a>` elements inside the list

`display: block;`

Makes the whole link area clickable

It allows us to specify the width, padding, margin, height etc

```
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 60px;
}

li a {
  display: block;
}

</style>
```



Home
News
Contact
About

CSS-Menu Bar-Vertical

Step 4: Vertical navigation bar

To create navigation bar with background color

```
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 100px;
    background: #dddddd;
}
li a{
    display: block;
    padding: 8px 16px;
    text-decoration: none;
}
</style>
```



CSS-Menu Bar-Vertical

Step 5: Vertical navigation bar

hover: To changes the background color when user places curser over menu

```
li a:hover {  
  background-color: grey;  
  color: white;  
}
```



CSS-Menu Bar-Horizontal

float: left; - use float to get block elements to slide next to each other

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow:hidden;  
    background-color:#dddddd;  
}  
  
li {float:left;}
```

```
li a{  
    display: block;  
    padding: 8px 16px;  
    text-decoration: none;  
}
```



CSS- Dropdown menu

- Div class is better suited to design drop down menu
- To create a horizontal menu with dropdown, specify div class for menu with links
- For Dropdown menu create a separate div class and add necessary style into that.



CSS- Dropdown menu

Step 1:

- Create a div class with links
- Place button for drop down menu

```
<body>

<div class="navbar">
  <a href="#home">Home</a>
  <a href="#news">News</a>
  <div class="dropdown">
    <button class="mybutton">Dropdown
    </button>

    <div class="dropdown-content">
      <a href="#">Link 1</a>
      <a href="#">Link 2</a>
      <a href="#">Link 3</a>
    </div>
  </div>
</div>
</body>
```

CSS- Dropdown menu

Step 2:

Specify the design for navbar class and link in nav bar

```
<style>
.navbar {
    overflow: hidden;
    background-color: #333;
}
.navbar a {
    float: left;
    font-size: 16px;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}
```



CSS- Dropdown menu

Step 3:

Change the button design and make that as normal menu

```
.dropdown .mybutton {  
    font-size: 16px;  
    border: none;  
    outline: none;  
    color: white;  
    padding: 14px 16px;  
    background-color: inherit;  
    font-family: inherit;  
    margin: 0;  
}
```

Home News Dropdown

Link 1 Link 2 Link 3

CSS- Dropdown menu

Step 4:

- Make the dropdown links invisible for user when page loads

```
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color: #f9f9f9;  
  width: 120px;  
}
```

Home News Dropdown

CSS- Dropdown menu

Step 5:

- On hover the dropdown content should appear
- Specify the design for dropdown content

```
.dropdown-content a {  
  color: black;  
  padding: 12px 16px;  
  text-decoration: none;  
  text-align: left;  
}
```

```
.dropdown:hover .dropdown-content {  
  display: block;  
}
```

Home News Dropdown

Link 1

Link 2

Link 3

CSS- Dropdown menu

Step 6:

- Provide design to show dropdown content below the drop down menu
- On hover color of drop down content may change

```
.dropdown {  
  float: left;  
  overflow: hidden;  
}  
  
.dropdown-content a:hover {  
  background-color: red;  
}
```





SOME MORE TAGS

<Meta> tag

- It defines metadata about an HTML document
- Metadata is data (information) about data inside the <head> element
- used to specify character set, page description, keywords, author of the document, and viewport settings
- Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services

<Meta> tag - Attributes

charset	<ul style="list-style-type: none">character_set	<ul style="list-style-type: none">Specifies the character encoding for the HTML document
content	<ul style="list-style-type: none">text	<ul style="list-style-type: none">Specifies the value associated with the http-equiv or name attribute
name	<ul style="list-style-type: none">application-nameauthorDescriptionkeywords	<ul style="list-style-type: none">Specifies a name for the metadata
http-equiv	<ul style="list-style-type: none">content-security-policycontent-typedefault-stylerefresh	<ul style="list-style-type: none">Provides an HTTP header for the information/value of the content attribute

- Define keywords for search engines:
- `<meta name="keywords" content="HTML, CSS, JavaScript">`
- Define a description of your web page:
- `<meta name="description" content="Free Web tutorials for HTML and CSS">`
- Define the author of a page:
- `<meta name="author" content="John Doe">`

- Refresh document every 30 seconds:
 - <meta http-equiv="refresh" content="30">\
- Setting the viewport to make your website look good on all devices:
 - <meta name="viewport" content="width=device-width, initial-scale=1.0">

viewport meta tag

Without the viewport meta tag



With the viewport meta tag



Example

```
<head>

<meta charset="UTF-8">

<meta name="description" content="Free Web tutorials">

<meta name="keywords" content="HTML,CSS,XML, JavaScript">

<meta name="author" content="John">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="refresh" content="30">

</head>
```

CSS2 Introduced Media Types

The `@media` rule, introduced in CSS2, made it possible to define different style rules for different media types.

Examples: You could have one set of style rules for computer screens, one for printers, one for handheld devices, one for television-type devices, and so on.

Media queries can be used to check many things, such as:

width and height of the viewport

width and height of the device

orientation (is the tablet/phone in landscape or portrait mode?)

resolution

```
@media  
not|only mediatype and (expressions) {  
  CSS-Code;  
}
```

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: pink;
}

@media screen and (min-width: 480px) {
  body {
    background-color: lightgreen;
  }
}
</style>
</head>
```

```
<body>

<h1>Resize the browser window to see
the effect!</h1>
<p>The media query will only apply if the
media type is screen and the viewport is
480px wide or wider.</p>

</body>
</html>
```