Unit-2

Introduction to CSS

Content

- ✓ What is CSS,
- ✓ CSS Syntax,
- ✓ Location of Styles,
- ✓ Selectors,
- ✓ The Cascade: How Styles Interact,
- ✓ The Box Model,
- ✓ CSS Text Styling.
- ✓ Advanced CSS:
  - ✓ Layout,
  - ✓ Normal Flow,
  - ✓ Positioning Elements,
  - ✓ Floating Elements,
  - ✓ Constructing Multicolumn Layouts,
  - ✓ Approaches to CSS Layout,
  - ✓ Responsive Design,
  - ✓ CSS Frameworks.

What is CSS?

- CSS stands for Cascading Style Sheets
- Used to override the default styles of the elements of HTML documents.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files
- Example:
  - The <h2> tag has the font-size property, for which a browser would have the default value of a particular size.
  - Font-size property for <h2> be set to a larger size

CSS Syntax

- The general form of Style Specification Formats

```
style = "property_1 : value_1; property_2 : value_2; . . . ;
         property_n : value_n;"
```

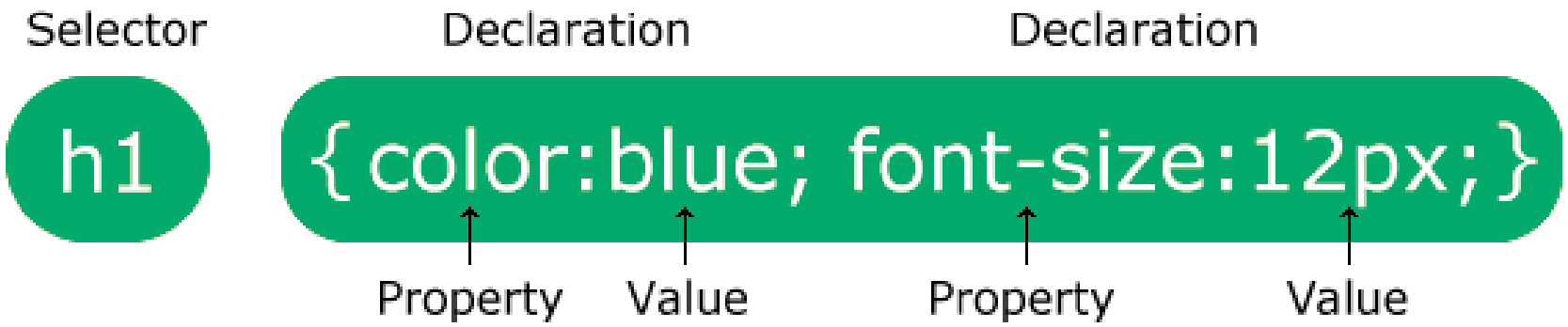
- The general form of the content of a style element is as follows:

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

- The form of a style rule is as follows

```
selector {property_1 : value_1; property_2 : value_2; . . . ;
          property_n : value_n; }
```

Example - CSS Syntax




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

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  font-family: verdana; color:red;
  font-size: 40px;
}
</style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

Three Ways to Insert CSS

- There are three ways of inserting a style sheet:
  1. External CSS
  2. Internal CSS
  3. Inline CSS

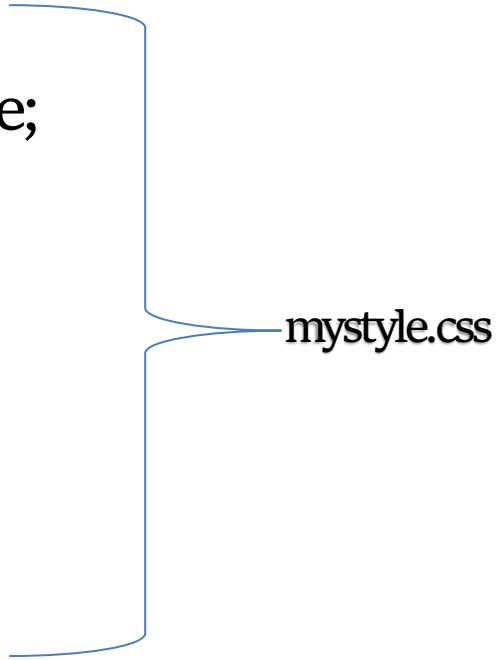
External CSS

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

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

</body>
</html>
```

External style sheet with file extension as **.css**

Cont'd

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

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



Internal CSS

- An internal style sheet may be used if one single HTML page has a unique style.
- The internal style is defined inside the <style> element, inside the head section.

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

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

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

</body>
</html>
```

Inline CSS

- An inline style may be used to apply a unique style for a single/specific element of the HTML document.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.
- Example:

```
<!DOCTYPE html>
<html>
<body>
        <h1 style="color: blue; text-align:center;">This is a heading</h1>
        <p style="color: red;">This is a paragraph.</p>
</body>
</html>
```

Multiple Style Sheets

Assume that an **external style sheet** has the following style for the <h1> element:

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

Then, assume that an **internal style sheet** also has the following style for the <h1> element:

```
h1 {
  color: orange;
}
```

Example

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

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

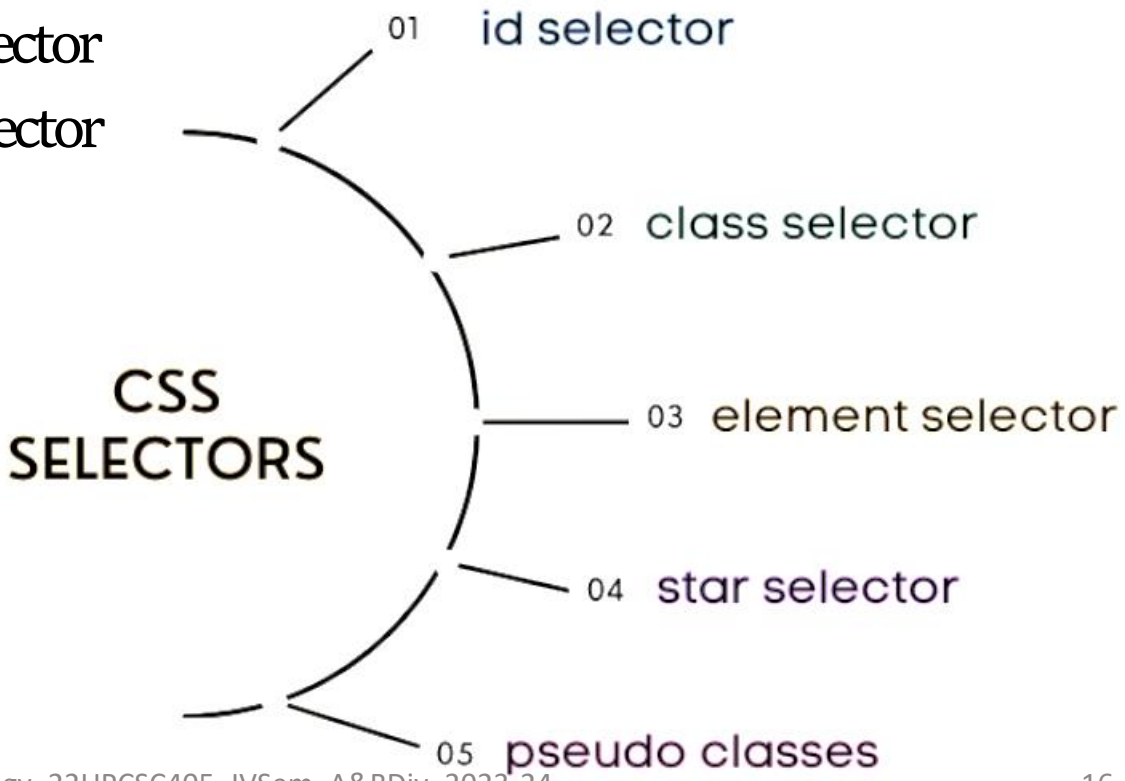
Example

However, if the internal style is defined **before** the link to the external style sheet, the <h1> elements will be "navy":

```
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Selector

- The CSS element Selector
- The CSS id Selector
- The CSS class Selector
- The CSS Universal Selector
- The CSS Grouping Selector



The CSS element Selector

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

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Example – id selector

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
#para2 {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello World!</p>
<p id="para2">This paragraph is not affected by the style.</p>
</body>
</html>
```

The CSS class Selector

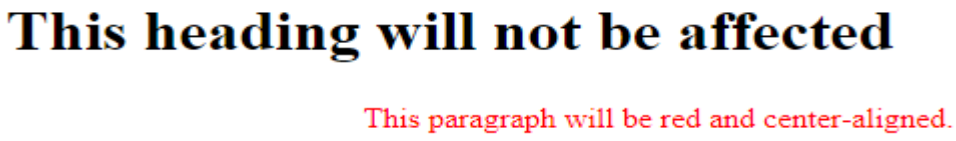
```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
p.large {
  font-size: 300%;
}
</style>
</head>
<body>
<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>
</body>
</html>
```

In this example only <p> elements with class="center" will be red and center-aligned:

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

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>

</body>
</html>
```

OUTPUT



The CSS Universal Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<h1>Hello world!</h1>

<p>Every element on the page will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

The CSS rule below will affect every HTML element on the page:

The CSS Grouping Selector

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

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

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

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

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
        <h1>Hello World!</h1>
        <h2>Smaller heading!</h2>
        <p>This is a paragraph.</p>
</body>
</html>
```

Cascading Order: style priority

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

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority and will override external and internal styles and browser defaults.

Some Font Examples

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	Brush Script MT Lucida Handwriting

The CSS font-family Property

Example

Specify some different fonts for three paragraphs:

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

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

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

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.p1 {
  font-family: "Times New Roman", Times, serif;
}
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
</style>
</head>
<body>
<h1>CSS font-family</h1>
<p class="p1">This is a paragraph, shown in the Times New Roman font.</p>
<p class="p2">This is a paragraph, shown in the Arial font.</p>
<p class="p3">This is a paragraph, shown in the Lucida Console font.</p>
</body>
</html>
```

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment is placed inside the `<style>` element, and starts with `/*` and ends with `*/`:

Example

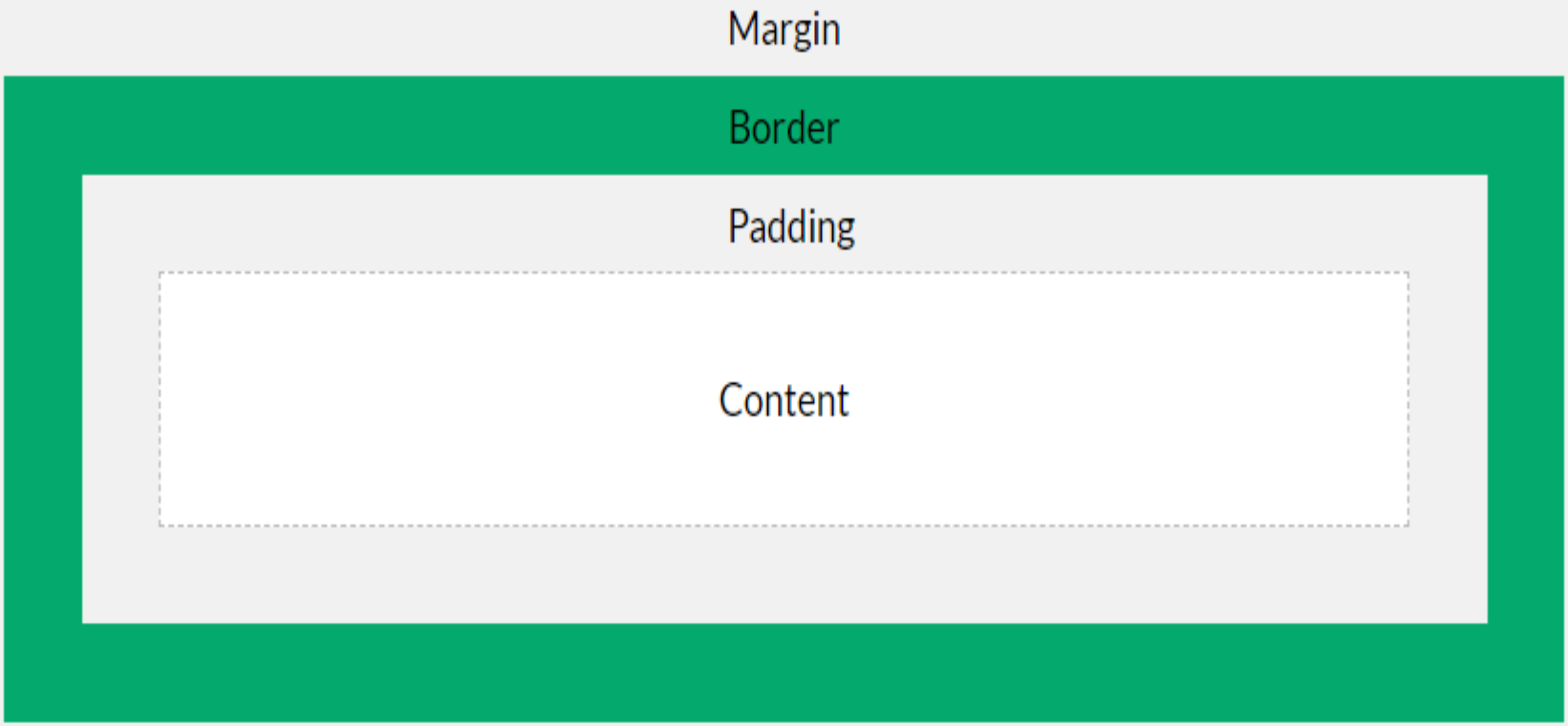
```
/* This is a single-line comment */
p {
  color: red;
}
```

```
<html>
<head>
<style>
/* This is
a multi-line
comment */
p {
  color: red;
}
</style>
</head>
<body>
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>
</body>
</html>
```

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 image below illustrates the box model:

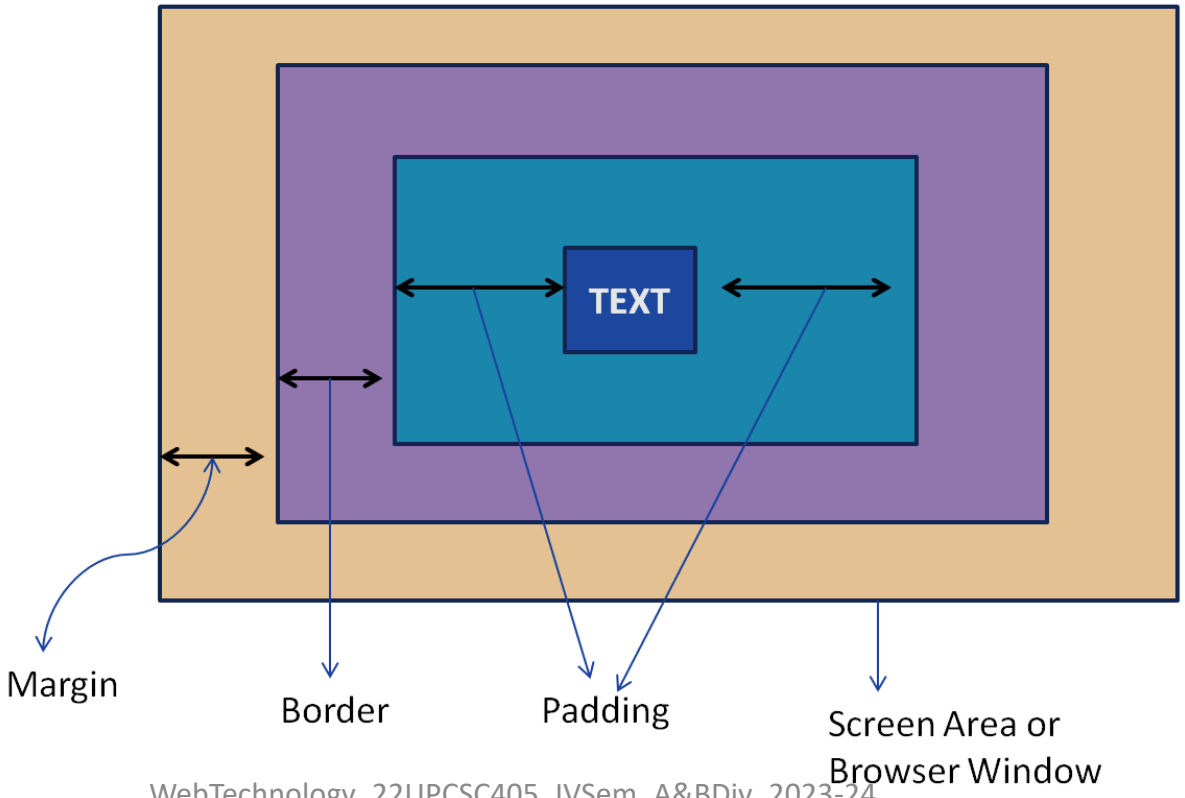


Web page box model

Box Modal:
MBPC – Margin, Border, Padding and Content

Width = L.Margin + L.Border + L.Padding + Width of Content +
R.Margin + R.Border + R.Padding + Width of Content

Height = T.Margin + T.Border + T.Padding + Height of Content +
B.Margin + B.Border + B.Padding + Height of Content



Contd…

Explanation of the different parts:

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

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

Contd...

- Margin and padding are both used to control the space around elements in CSS

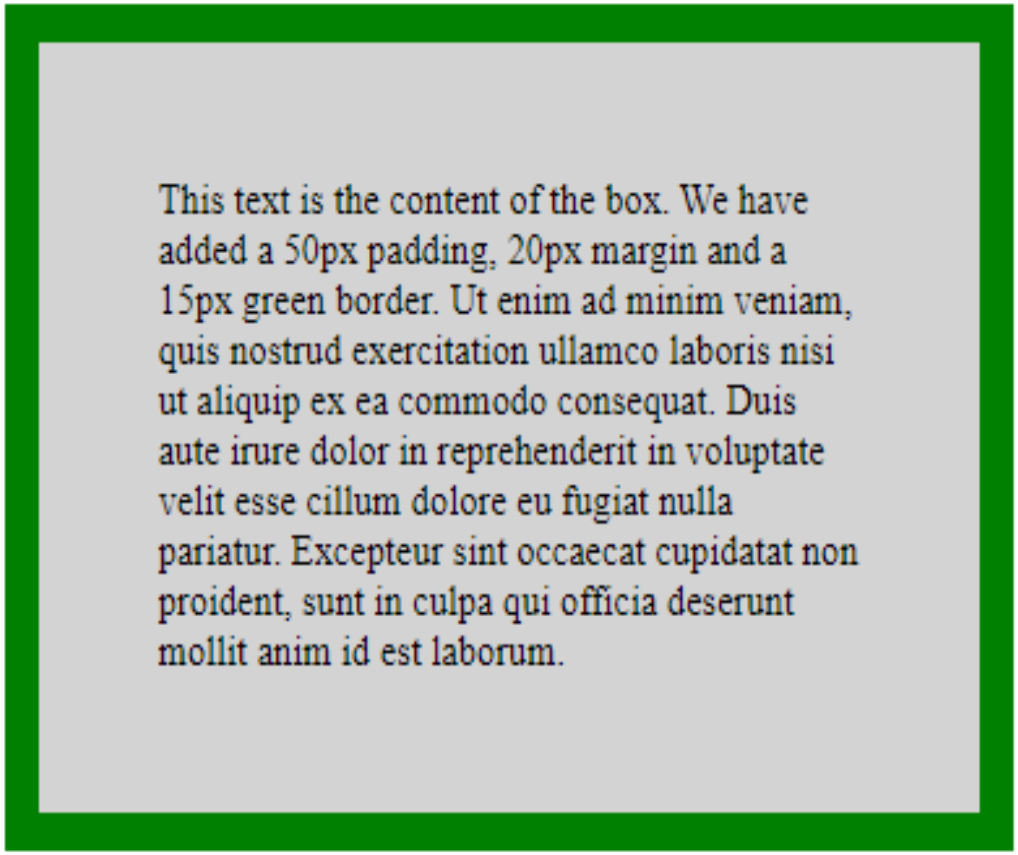
Margin	Padding
Margin refers to the space outside the element's border.	Padding refers to the space between the element's content and its border.
It creates a gap between the element and its surroundings	It creates space inside the element, increasing its overall size.
control the margin for each side of an element (top, right, bottom, left) independently.	control padding for each side of an element.

Example Demonstration of the box model:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2>Demonstrating the Box Model</h2>

<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: 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. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>

</body>
</html>
```

Demonstrating the 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.



Contd..

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

Contd..

- The total width of an element should be calculated like this:
- 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 like this:
- Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

TEXT FORMATTING

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)"

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.

Example

```
body {
  color: blue;
}

h1 {
  color: green;
}
```

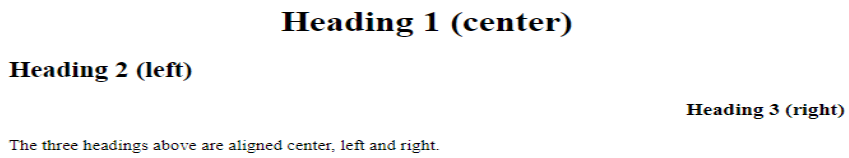
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightgrey;
  color: blue;
}

h1 {
  background-color: black;
  color: white;
}

div {
  background-color: blue;
  color: white;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This page has a grey background color and a blue text.</p>
<div>This is a div.</div>

</body>
</html>
```

TEXT ALIGNMENT

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-align: center;
}
h2 {
  text-align: left;
}
h3 {
  text-align: right;
}
</style>
</head>
<body>
<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>
<p>The three headings above are aligned center, left and right.</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-align: center;
}

h2 {
  text-align: left;
}

h3 {
  text-align: right;
}
</style>
</head>
<body>

<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>

<p>The three headings above are aligned center, left and right.</p>

</body>
</html>
```

OUTPUT



```html
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {
  direction: rtl;
  unicode-bidi: bidi-override;
}
</style>
</head>
<body>

<p>This is the default text direction.</p>

<p class="ex1">This is right-to-left text direction.</p>

</body>
</html>
```

OUTPUT

This is the default text direction.

.noitcerid txet tfel-ot-thgir si sihT

TEXT DECORATION

```
<!DOCTYPE html>
<html>
<head>
<style>
h2 {  text-decoration: overline; }
h3 {  text-decoration: line-through; }
h4 {   text-decoration: underline; }
</style>
</head>
<body>
<h1>Some different text decorations</h1>
<h2>Overline text decoration</h2>
<h3>Line-through text decoration</h3>
<h4>Underline text decoration</h4>
<p><strong>Note:</strong> It is not recommended to underline text that is not a link, as this often confuses
the reader.</p>
</body>
</html>
```

CSS Website Layout

‹ Previous Next ›

Website Layout

A website is often divided into headers, menus, content and a footer:

Advanced Layout

CSS Flow Layout

Normal Flow, or Flow Layout, is the way that Block and Inline elements are displayed on a page before any changes are made to their layout. The flow is essentially a set of things that are all working together and know about each other in your layout. Once something is taken *out of flow* it works independently.

In normal flow, **inline** elements display in the inline direction, that is in the direction words are displayed in a sentence according to the Writing Mode of the document. **Block** elements display one after the other, as paragraphs do in the Writing Mode of that document. In English therefore, inline elements display one after the other, starting on the left, and block elements start at the top and move down the page.

Basic Example

The following example demonstrates Block and Inline Level boxes. The two paragraph elements with a green border are Block Level, displaying one under the other.

The first sentence also includes a span element with a blue background. This is inline level and therefore displays in place in the sentence.

Layout : POSITIONING

The position Property

The position property specifies the type of positioning method used for an element.

There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

Positioning: static example

```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
  position: static;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: static;</h2>

<p>An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:</p>

<div class="static">
This div element has position: static;
</div>
</body>
</html>
```

Float property

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: right;
}
</style>
</head>
<body>
<h1>The float Property</h1>

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

<p><img src="pineapple.jpg" alt="Pineapple" style="width:170px;height:170px;margin-left:15px;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.</p>

</body>
</html>
```

The float Property

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

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

CSS Multiple Columns

CSS Multi-column Layout

The CSS multi-column layout allows easy definition of multiple columns of text - just like in newspapers:

Daily Ping

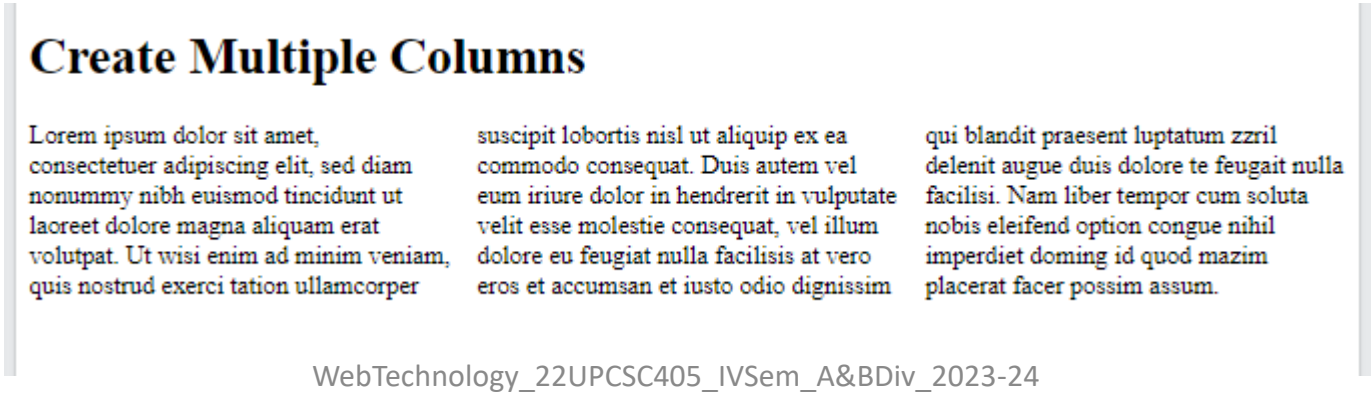
Lorem ipsum
dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper

suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui

blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

Example : column

```
<!DOCTYPE html>
<html>
<head>
<style>
.newspaper {
  column-count: 3;
}
</style>
</head>
<body>

<h1>Create Multiple Columns</h1>

<div class="newspaper">
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
</div>

</body>
</html>
```



RESPONSIVE

Responsive Web Design - Introduction

What is Responsive Web Design?

Responsive web design makes your web page look good on all devices.

Responsive web design uses only HTML and CSS.

Responsive web design is not a program or a JavaScript.

Designing For The Best Experience For All Users

Web pages can be viewed using many different devices: desktops, tablets, and phones. Your web page should look good, and be easy to use, regardless of the device.

Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:

Desktop

Tablet

Phone

It is called responsive web design when you use CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

Don't worry if you don't understand the example below, we will break down the code, step-by-step, in the next chapters:

GRID VIEW

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}

.header {
  border: 1px solid red;
  padding: 15px;
}

.row::after {
  content: "";
  clear: both;
  display: table;
}

[class*="col-"] {
  float: left;
  padding: 15px;
  border: 1px solid red;
}

.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
</style>
</head>
<body>

<div class="header">
  <h1>Chania</h1>
</div>

<div class="row">

<div class="col-3">
  <ul>
    <li>The Flight</li>
    <li>The City</li>
    <li>The Island</li>
    <li>The Food</li>
  </ul>
</div>

<div class="col-9">
  <h1>The City</h1>
  <p>Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.</p>
  <p>Resize the browser window to see how the content respond to the resizing.</p>
</div>
</div>
</body>
</html>
```

OUT PUT

The Bharat

- The Flight
- The City
- The Island
- The Food

The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

Resize the browser window to see how the content respond to the resizing.