

# Unit3 :CSS

# What is CSS?

- CSS stands for Cascading Style Sheets
- 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

# 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.

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

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

```
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

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

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>

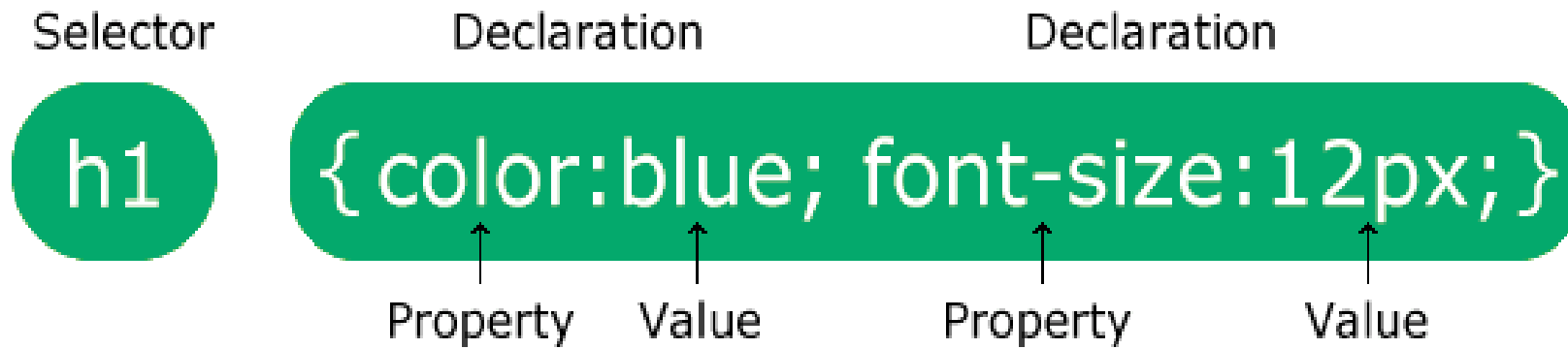
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

# My First CSS Example

This is a paragraph.

# CSS Syntax



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

- In this example all <p> elements will be center-aligned, with a red text color:

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

- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

# Group selectors

- To group selectors, separate each selector with a comma.

```
<!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>
```

**Hello World!**

**Smaller heading!**

This is a paragraph.



# Three Ways to Insert CSS

- There are three ways of inserting a style sheet:
  - External CSS
  - Internal CSS
  - Inline CSS

# External CSS

- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each HTML page must include a reference to the external style sheet file inside the `<link>` element, inside the head section.

- An external style sheet can be written in any text editor, and must be saved with a .css extension.
- The external .css file should not contain any HTML tags.

```
<!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>
```

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

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

---

# This is a heading

This is a paragraph.

# 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: powderblue;
}

h1
{
color: blue;
}
p
{
color: red;
}
</style>
</head>
<body>

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

</body>
</html>
```

# This is a heading

This is a paragraph.

# Inline CSS

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

```
<!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>
```

**This is a heading**

This is a paragraph.



## Cascading Order

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. Internal and External 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.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
  body {background-color: red;}
</style>
</head>

  <body style="background-color: green">

<h1>Multiple Styles Will Cascade into One</h1>
<p>Here, the background color of the page is set with inline CSS, and also with an internal CSS,
and also with an external CSS.</p>
<p>Try experimenting by removing styles to see how the cascading stylesheets work (try removing the
inline CSS first, then the internal, the the external).</p>

</body>
</html>
```

## Multiple Styles Will Cascade into One

Here, the background color of the page is set with inline CSS, and also with an internal CSS, and also with an external CSS.

Try experimenting by removing styles to see how the cascading stylesheets work (try removing the inline CSS first, then the internal, the the external).

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
  body {background-color: red;}
</style>
</head>

<!--<body style="background-color: green"> -->

<h1>Multiple Styles Will Cascade into One</h1>
<p>Here, the background color of the page is set with inline CSS, and also with an internal CSS,
and also with an external CSS.</p>
<p>Try experimenting by removing styles to see how the cascading stylesheets work (try removing the
inline CSS first, then the internal, the the external).</p>

</body>
</html>
```

## Multiple Styles Will Cascade into One

Here, the background color of the page is set with inline CSS, and also with an internal CSS, and also with an external CSS.

Try experimenting by removing styles to see how the cascading stylesheets work (try removing the inline CSS first, then the internal, the the external).

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
  <!--<style>
    body {background-color: red;}
  </style> -->
</head>

  <!--<body style="background-color: green"> -->

<h1>Multiple Styles Will Cascade into One</h1>
<p>Here, the background color of the page is set with inline CSS, and also with an internal CSS,
and also with an external CSS.</p>
<p>Try experimenting by removing styles to see how the cascading stylesheets work (try removing the
inline CSS first, then the internal, the the external).</p>

</body>
</html>
```

## Multiple Styles Will Cascade into One

Here, the background color of the page is set with inline CSS, and also with an internal CSS, and also with an external CSS.

Try experimenting by removing styles to see how the cascading stylesheets work (try removing the inline CSS first, then the internal, the the external).

# The CSS element Selector

- The element selector selects HTML elements based on the element name.
- Example
- Here, all <p> elements on the page will be center-aligned, with a red text color:
- ```
p {  
  text-align: center;  
  color: red;  
}
```

- <html>
- <head>
- <style>
- div {
- border: 5px inset gold;
- width: 300px;
- text-align: center;
- }
- p {
- color: green;
- }
- h1 {
- text-decoration-line: underline;
- }
- </style>
- </head>
- <body>
- <div>
- <h1>Type selector</h1>
- <p>div with border and text-aligned to center</p>
- <p>paragraph with green color</p>
- <p>h1 with an underline</p>
- </div>
- </body>
- </html>

# Type selector

div with border and text-aligned to center

paragraph with green color

h1 with an underline

# The CSS id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- Example
- The CSS rule below will be applied to the HTML element with id="para1":
- ```
#para1 {  
  text-align: center;  
  color: red;  
}
```

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

Hello World!

This paragraph is not affected by the style.



```
<!DOCTYPE html>
<html>
<head>
  <title>ID Selector Example</title>
  <style>
    /* CSS styles using ID selector */
    #myElement {
      background-color: lightblue;
      color: white;
      padding: 10px;
      font-size: 20px;
      font-family: Arial, sans-serif;
    }
  </style>
</head>
<body>
  <!-- HTML element with the ID "myElement" -->
  <div id="myElement">
    This is a div with the ID "myElement".
    It will have a light blue background, white text, and other specified styles.
  </div>
  <p>
    This is another paragraph on the page that is not affected by the ID selector.
  </p>
</body>
</html>
```

**Output:**

This is a div with the ID "myElement". It will have a light blue background, white text, and other specified styles.

This is another paragraph on the page, which is not affected by the ID selector.

# Styling Elements with CSS Classes

```
<!DOCTYPE html>
<html lang="en">

<head>
  <style>
    .para {
      font-size: larger;
      margin-bottom: 35px;
      background-color: lightgreen;
    }

    .second_para {
      color: red;
    }
  </style>
</head>

<body>
  <p class="para">
    Hello there.
  </p>

  <p class="para second_para">
    Welcome to Our portal.
  </p>
</body>

</html>
```

Hello there.

Welcome to Our portal.

# CSS Image ID

We can use the ID selector on various HTML elements, such as headings, images, buttons, etc.

- For instance, let's style a particular image. We wish to use different shapes, borders, and opacity for each image. In such a scenario, we can accomplish this using the ID selector.

- E.g

Here, we have three images.

We will assign unique IDs to each of the images.

- <!DOCTYPE html>
- <html lang="en">
- <head>
- <meta charset="UTF-8">
- <meta http-equiv="X-UA-Compatible" content="IE=edge">
- <meta name="viewport" content="width=device-width, initial-scale=1.0">
- <title>CSS Image ID</title>
- <link rel="stylesheet" href="styles.css">
- </head>
- <body>
- <h2>CSS Image ID</h2>
- <!-- Feel free to insert images of your choice.   -->
- 
- 
- 
- </body>
- </html>

## CSS

- We will make the first image rounded using the id selector.
- We will add a border around the second image.
- We will add opacity to the last one.

```
h2 {  
    text-align: center;  
    font-size: 2rem;  
}
```

```
img {  
    width: 400px;  
    height: 300px;  
    margin: 2.5rem;  
}
```

```
/* Styling the image 1 */
```

```
#img1 {  
    border-radius: 100%;  
}
```

```
/* Styling the image 2 */
```

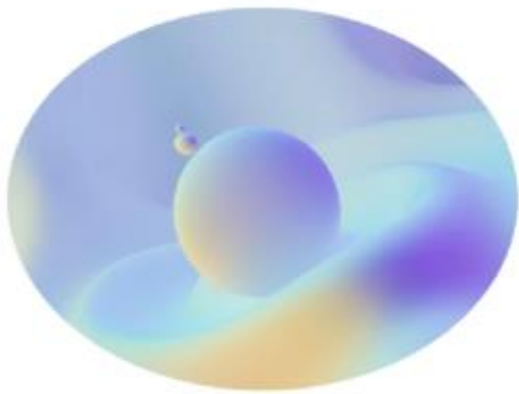
```
#img2 {  
    border-radius: 15px;  
    border: 10px double black;  
    padding: 0.5rem;  
}
```

```
/* Styling the image 3 */
```

```
#img3 {  
    box-shadow: rgb(0 0 0 / 25%) 0px 5px 10px 2px;  
    opacity: 0.8;  
}
```

Output:

**CSS Image ID**



# The CSS class Selector

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.



- <!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>

```
<!DOCTYPE html>
<html>
<head>
<style>
.myDiv {
  border: 5px outset red;
  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 some text in a div element.</p>
</div>

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

</body>
</html>
```

## The div element

### This is a heading in a div element

This is some text in a div element.

This is some text outside the div element.

## **When to use Classes**

We should use classes when our style needs to be applied multiple times on the same page. For example, we can use a class selector to style paragraphs, links, buttons, input boxes, etc.

## **When to use IDs**

- We should use IDs for single elements that appear on the page only once, or we need a unique identifier for that element.
- For example, we can use an ID to style a header, footer, menu, etc.

# The CSS Universal Selector

- The universal selector (\*) selects all HTML elements on the page.

# Example

```
<!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>
```

**Hello world!**

Every element on the page will be affected by the style.

Me too!

And me!

# 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):

# 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>
```

**Hello World!**

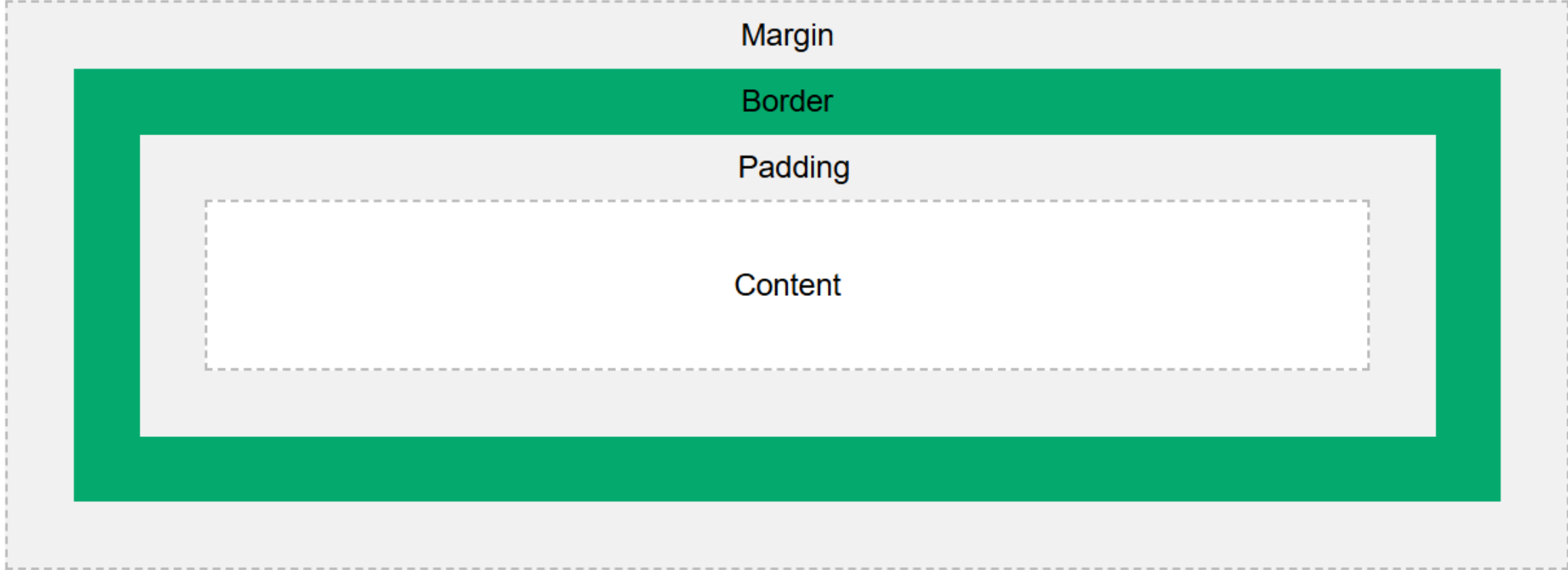
**Smaller heading!**

This is a paragraph.

# 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: content, padding, borders and margins.





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

```
<!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>
```

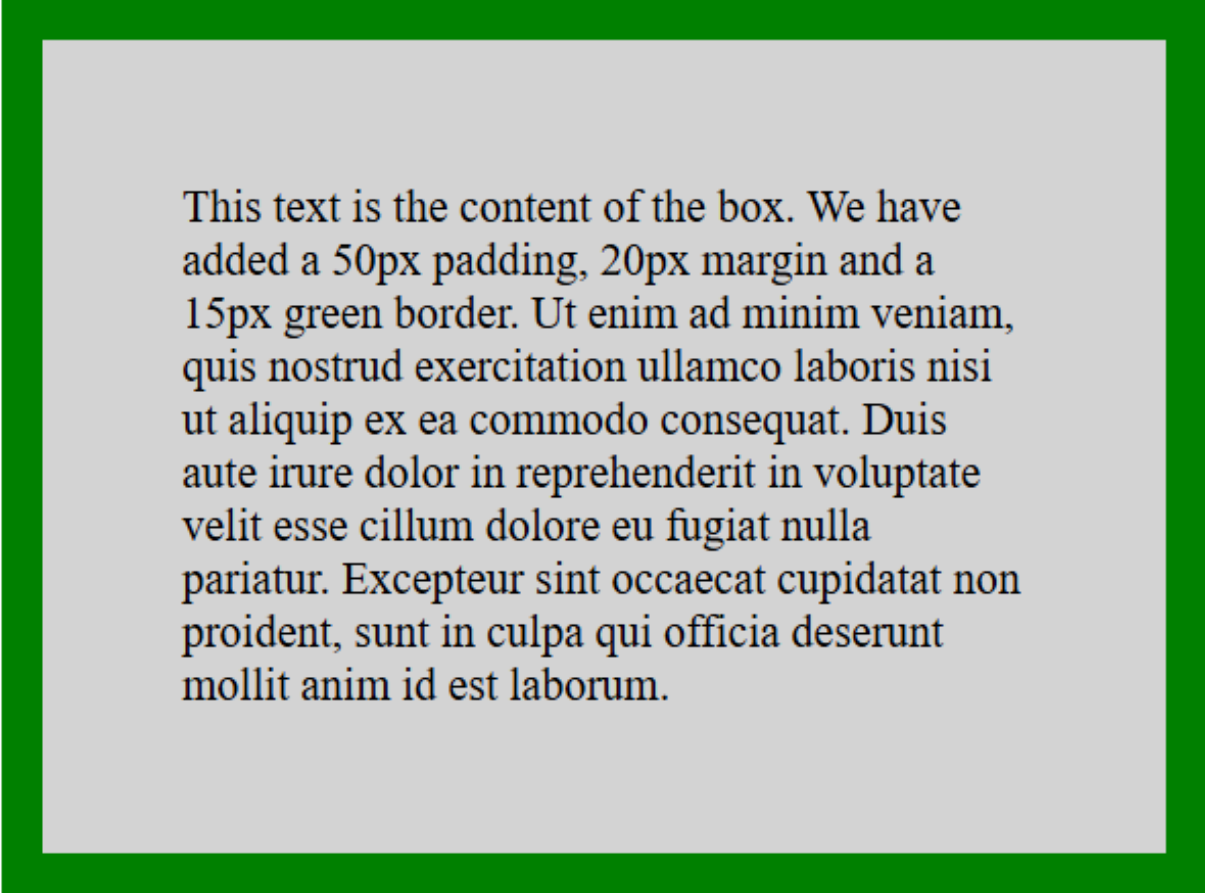
```
<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px
green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex
ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
eu 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.



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.

# Width and Height of an Element

- In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.
- When you set the width and height properties of an element with CSS, you just set the width and height of the **content area**. To calculate the total width and height of an element, you must also include the padding and borders.

## Example

- This `<div>` element will have a total width of 350px and a total height of 80px:

- ```
div {  
  width: 320px;  
  height: 50px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 0;  
}
```

Here is the calculation:

```
320px (width of content area)  
+ 20px (left padding + right padding)  
+ 10px (left border + right border)  
= 350px (total width)  
  
50px (height of content area)  
+ 20px (top padding + bottom padding)  
+ 10px (top border + bottom border)  
= 80px (total height)
```

