

# CSS CRASH COURSE

FOR ABSOLUTE BEGINNERS



BY SHAHAN KHAN

(Css Coder)

## **Course Table Of Content:**

### **Module One: INTRODUCTION TO CSS**

- 1) Welcome to Css**
- 2) Inline, Embedded, External Css**
- 3) Css Rules And Selectors**
- 4) Css Comments**

### **Module Two: CSS COLORS**

- 1) Css background-color**
- 2) Css text-color**
- 3) Css border-color**
- 4) Css color values**
- 5) Css rgb value**
- 6) Css hex value**
- 7) Css hsl value**
- 8) Css rgba value**
- 9) Css hsla value**

### **Module Three: CSS BORDERS**

- 1) border-style**
- 2) border-color**
- 3) border-width**
- 4) border shorthand property**
- 5) rounded borders**

## **Module Four: CSS MARGIN**

- 1) margin - individual sides**
- 2) margin - inherit value**
- 3) margin - shorthand property**
- 4) margin - auto value**

## **Module Five: CSS PADDING**

- 1) padding – individual sides**
- 2) padding shorthand property**
- 3) padding and element width**

## **Module Six: CSS HEIGHT AND WIDTH**

- 1) Css Height / Width values**
- 2) max-height and max-width**
- 3) min-height and min-width**

## **Module Seven: CSS TEXT**

- 1) text-color**
- 2) text-decoration**
- 3) text-transformation**
- 4) text-alignment**
- 5) letter-spacing**
- 6) line-height**
- 7) text-direction**
- 8) word-spacing**
- 9) text-shadow**
- 10) text-indentation**

## **Module Eight: CSS BOX MODEL**

**1) Css box-model**

## **Module Nine: CSS FONT**

- 1) font-family**
- 2) font-style**
- 3) font-size**
- 4) font-weight**
- 5) font-variant**

## **Module Ten: CSS OUTLINE**

- 1) outline-style**
- 2) outline-color**
- 3) outline-width**
- 4) outline-offset**
- 5) outline shorthand property**

## **Module Eleven: CSS LISTS**

- 1) different list item markers**
- 2) image as list item marker**
- 3) list-marker position**
- 4) list shorthand property**
- 5) remove default list settings**
- 6) styling lists**

## **Module Twelve: CSS TABLES**

- 1) table borders**
- 2) border collapse**
- 3) border spacing**
- 4) table width and height**
- 5) caption side**
- 6) table padding**
- 7) table layout**

## **Module Thirteen: CSS DISPLAY**

- 1) block level elements**
- 2) inline elements**
- 3) inline-block elements**
- 4) display none**

## **Module Fourteen: CSS POSITIONING ELEMENTS**

- 1) position fixed**
- 2) position static**
- 3) position relative**
- 4) position absolute**
- 5 position sticky**

## **Module Fifteen: CSS FLOAT**

- 1) float left**
- 2) float right**
- 3) float none**
- 4) float inherit**
- 5) navigation bar using float**

## **Module Sixteen: CSS ALIGN**

- 1) center align elements**
- 2) center align text**
- 3) center image**
- 4) left and right align – using position**
- 5) center vertically using padding**

## **Module Seventeen: CSS COMBINATORS**

- 1) general sibling selector**
- 2) adjacent sibling selector**
- 3) child selector**
- 4) descedent selctor**

## **Module Eighteen: CSS PSEUDO CLASSES**

- 1) hover pseudo class**
- 2) active pseudo class**
- 3) visited pseudo class**
- 4) focus pseuo class**

## **Module Ninteen: CSS PSEUDO ELEMENTS**

- 1) before pseudo element**
- 2) after pseudo element**
- 3) first leter pseudo element**
- 4) first line pseudo element**

## **Summary Of The Course:**

In this CSS crash course we have covered all the basics, necessary to start your journey with CSS language. I hope after going through this crash course you will be aware with lot of css properties, how they work and where to use them with your HTML code. Thanks!

Looking for a positive response. . .

## MODULE ONE

### WELCOME TO CSS!

CSS is a language that describes the style of an HTML document.

CSS describes how HTML elements should be displayed.

This tutorial will teach you CSS from basic to advanced.

#### 1) CSS INTRODUCTION :

##### What is CSS?

- CSS stands for Cascading Style Sheet.
- 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.

## CSS Solved a Big Problem:

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

When tags like `<font>`, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

## QUESTION:

What is the main purpose of CSS?

- 1) to script web pages
- 2) to store the keywords of web pages
- 3) to define styles for web pages (correct answer)
- 4) to make the web page independent

## NOTE:

To understand CSS, you should already have a basic knowledge of HTML.

**TOPIC ONE ENDS HERE**

## 2) INLINE, EMBEDDED, EXTERNAL CSS

### A) Inline CSS:

Using an inline style is the one of the ways to insert a style sheet. With an inline style, a unique style is applied to a single element.

To use an inline-style, add the style attribute to the relevant tag.

The example below shows how to create a paragraph with a skyblue background and black text.

#### Code:

```
<p style="color: black; background-color: skyblue;">
```

Example of Inline styling

```
</p>
```

#### Result:

#### Picture:



Example of Inline styling

#### QUESTION:

Select the attribute that is correct for inline styling:

- 1) class
- 2) id
- 3) style(correct)
- 4) file

**Note:** The style attribute can contain any CSS property.

## 2) Embedded CSS:

Internal styles are defined within the **<style>** element, inside the **head** section of an HTML page.

Lets look at the example below:

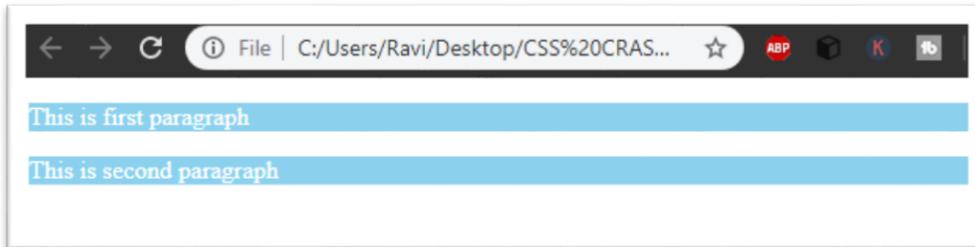
### Code:

```
<html>
  <head>
    <style>
      p{
        color: white;
        background-color: skyblue;
      }
    </style>
  </head>
  <body>
    <p>This is first paragraph</p>
    <p>This is second paragraph</p>
  </body>
</html>
```

### Result:

All paragraphs have a white font and a skyblue background:

## Picture:



## Question:

Where should the style tag be declared to create an internal/embedded CSS?

- 1) body
- 2) external file
- 3) head(Correct)

## Note:

Embedded style is used if you want a specific page to be styled unique otherwise not a good option.

## 3) External CSS:

With external CSS all styling rules are saved in a single text file, saved with `.css` extension.

This CSS file is then linked with HMTL using the `<link>` element which is placed inside the head section.

Here is example:

### Code:

#### HTML:

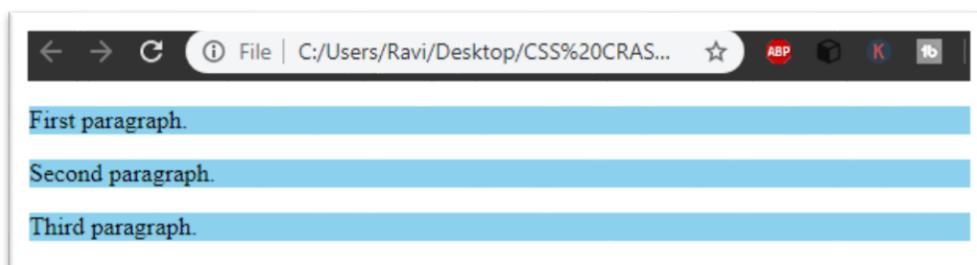
```
<head>
    <link rel="stylesheet" type="text/css" href="example.css">
</head>
<body>
    <p>First paragraph.</p>
    <p>Second paragraph.</p>
    <p>Third paragraph.</p>
</body>
```

#### CSS:

```
p{
    color: black;
    background-color: skyblue;
}
```

#### Result:

#### Picture:



## Question:

How to call an external style sheet in HTML?

- 1) <head>
- 2) <p>
- 3)<link>(Correct)**
- 4)<body>

## Note:

External stylesheet is best option to code for CSS because it helps to read code better if someone reads your code.

**TOPIC TWO ENDS HERE**

## 3) CSS RULES AND SELECTORS

### A) CSS Syntax:

A CSS rule-set consists of a selector, property and a declaration block:

### Picture:



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.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

### Question:

In rule the selector:

- 1) serves as property
- 2) selects element which to style(correct)
- 3) to substitute the selected attribute

### B) Type Selectors:

The most common and easy to understand selectors are type selectors. This selector select element types on the page.

Here is example to target all the paragraphs on the page:

#### Code:

```
p{  
color: blue;  
font-size: 16px;  
}
```

## Question:

Which one is a correct CSS style rule:

- 1){p } color: black;
- 2) } p { color: black;
- 3) p { color: black; } (correct)

## C) Id and class Selectors:

### 1) 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 Below:

The CSS rule below will be applied to the HTML element with id="para":

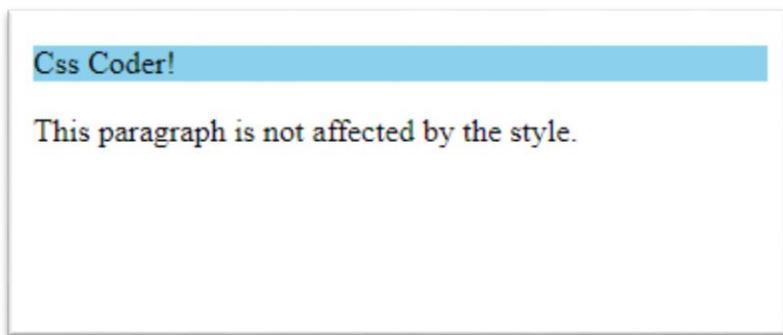
## Code:

```
<html>
<head>
<style>
#para {
    background: skyblue;
    color: black;
}
</style>
</head>
```

```
<body>  
<p id="para">Css Coder!</p>  
<p>This paragraph is not affected by the style.</p>  
</body>  
</html>
```

## Result:

### Picture:



## 2) 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.

Example Below:

### Code:

```
<html>  
<head>  
<style>  
.center {  
background: skyblue;  
color: black;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1 class="center">Black Color wih skyblue background</h1>
```

```
<p class="center">Black Color wih skyblue background</p>
```

```
</body>
```

```
</html>
```

## Result:

### Picture:



## Question:

To select an element with a specific id, write a

- 1) semicolon before the id of the element
- 2) hash before the id of the element (correct)**
- 3) dot before the id of the element

## D) Descendent Selectors:

The descendant selector matches all elements that are descendants of a specified element. You can select as many levels deep as you need to.

**Example Below:**

The following example selects all `<p>` elements inside `<div>` elements:

**Code:**

```
<html>
<head>
<style>
div p {
    background-color: skyblue;
}
</style>
</head>
<body>
<div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
    <section><p>Paragraph 3 in the div.</p></section>
</div>
    <p>Paragraph 4. Not in a div.</p>
    <p>Paragraph 5. Not in a div.</p>
</body>
</html>
```

**Result:**

**Picture:**

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

### Question:

Correct syntax for selecting all descendant p elements inside a div is:

- 1) p > div
- 2) div ~ p
- 3) div p (correct)
- 4) div (p)

### Note:

It matches all elements that are descendants of a specified element.

**TOPIC THREE ENDS HERE**

## 4) 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.

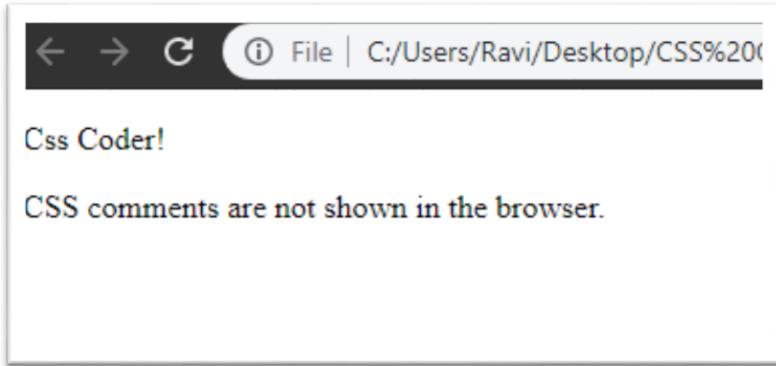
Example:

### Code:

```
<html>
<head>
<style>
p {
    color: black;
    /* This is a single-line comment */
    font-size: 20px;
}
/* This is
   a multi-line
comment */
</style>
</head>
<body>
<p>Css Coder!</p>
<p>CSS comments are not shown in the browser.</p>
</body>
</html>
```

**Result:**

**Picture:**



### Question:

Proper format to write a comment in your code:

- 1) /\* Css Coder \*/ (Correct)
- 2) \*/ Css Coder \*/
- 3) /\* Css Coder \*/

### Note:

Comments can span multiple lines.

**TOPIC FOUR ENDS HERE**

**END OF MODULE ONE**

## MODULE TWO

### CSS COLORS

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

#### 1) CSS background-color:

You can set the background color for HTML elements:

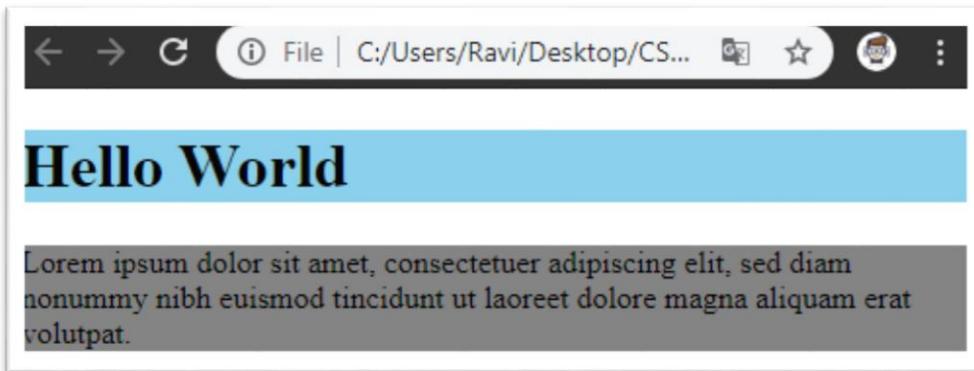
Example:

#### Code:

```
<html>
<body>
<h1 style="background-color: skyblue;">Hello World</h1>
<p style="background-color: gray;">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy
    nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
</p>
</body>
</html>
```

#### Result:

#### Picture:



### Question:

Proper syntax for defining background-color?

- 1) bg-color: skyblue;
- 2) Background-color: skyblue;
- 3) background-color: skyblue; (Correct)

### Note:

Background-color is useful to give colourful background to HTML elements.

**TOPIC ONE ENDS HERE**

### 2) CSS text-color:

You can set the color of text.

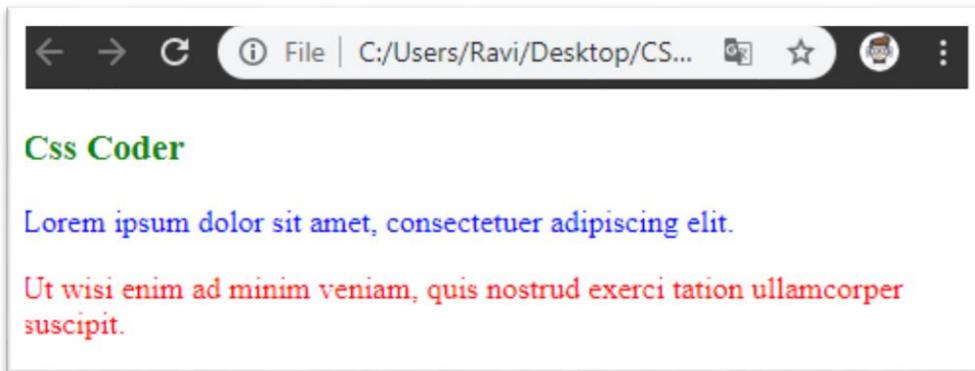
Example:

**Code:**

```
<html>
<body>
<h3 style="color: green;">Css Coder</h3>
<p style="color: blue;">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<p style="color: red;">Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit.</p>
</body>
</html>
```

**Result:**

**Picture:**



**Question:**

Proper syntax for defining text-color?

- 1) color: blue; (correct)
- 2) Color: Blue;
- 3) text-color: red;

**TOPIC TWO ENDS HERE**

### 3) CSS border-color:

You can set the color of borders.

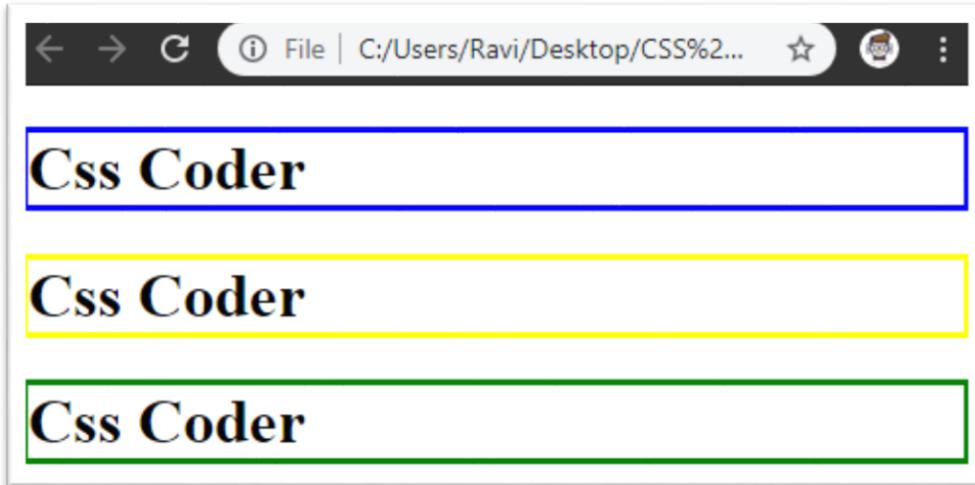
Example:

#### Code:

```
<html>
<body>
<h1 style="border: 3px solid blue;">Css Coder</h1>
<h1 style="border: 3px solid yellow;">Css Coder</h1>
<h1 style="border: 3px solid green;">Css Coder</h1>
</body>
</html>
```

#### Result:

#### Picture:



### Question:

Proper syntax for defining border-color?

- 1) border-color: blue; (Correct)
- 2) Border-Color: blue;
- 3) border-Color: blue;

**TOPIC THREE ENDS HERE**

### 4) CSS Color Values:

In CSS, colors can also be applied using RGB values, HEX values, HSL values, RGBA values, and HSLA values.

Example:

### Code:

```
<html>
<body>
<p>Same as color name "Tomato":</p>
<h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1>
<h1 style="background-color:#ff6347;">#ff6347</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1>
<p>Same as color name "Tomato", but 50% transparent:</p>
<h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71,
0.5)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%,
0.5)</h1>
<p>In addition to the predefined color names, colors can be specified using
RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color
values.</p>
</body>
</html>
```

## Result:

## Picture:



Same as color name "Tomato":

**rgb(255, 99, 71)**

**#ff6347**

**hsl(9, 100%, 64%)**

Same as color name "Tomato", but 50% transparent:

**rgba(255, 99, 71, 0.5)**

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

In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.

## Question:

The correct format for rgba value is?

- 1) **rgba(100, 100, 150, 0.5) (correct)**
- 2) **rgba(100; 100; 150)**
- 3) **Rgba(100; 100, 150., 0.5)**

## Note:

The most used color scheme is HEX values.

**TOPIC FOUR ENDS HERE**

## 5) CSS RGB Value:

In CSS, a color can be specified as an RGB value, using this formula:

**rgb(*red*, *green*, *blue*).**

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

For example, `rgb(255, 0, 0)` is displayed as red, because red is set to its highest value (255) and the others are set to 0.

To display the color black, all color parameters must be set to 0, like this: `rgb(0, 0, 0)`.

To display the color white, all color parameters must be set to 255, like this: `rgb(255, 255, 255)`.

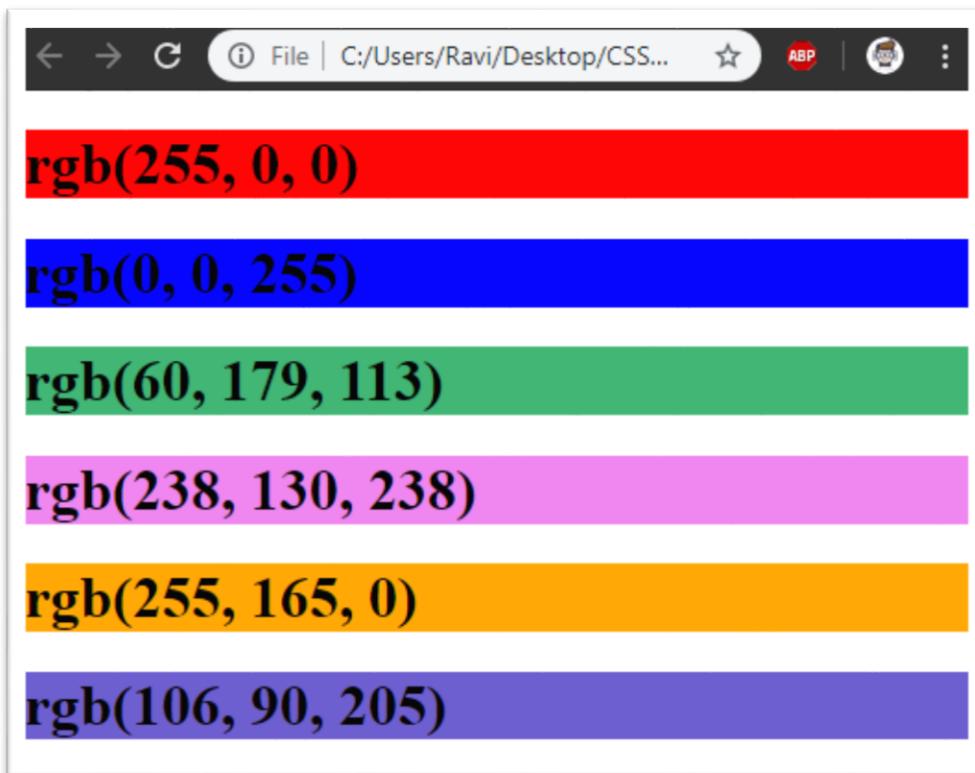
Example:

### Code:

```
<html>
  <body>
    <h1 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h1>
    <h1 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255)</h1>
    <h1 style="background-color:rgb(60, 179, 113);">rgb(60, 179, 113)</h1>
    <h1 style="background-color:rgb(238, 130, 238);">rgb(238, 130, 238)</h1>
    <h1 style="background-color:rgb(255, 165, 0);">rgb(255, 165, 0)</h1>
    <h1 style="background-color:rgb(106, 90, 205);">rgb(106, 90, 205)</h1>
  </body>
</html>
```

**Result:**

**Picture:**



**Note:**

In HTML, you can specify colors using RGB values.

**Question:**

To display the color black all values should be?

- 1) `rgb(255, 255, 255)`
- 2) `rgb(100, 100, 100)`
- 3) `rgb(0, 0, 0)` (Correct)**
- 4) `rgb(150, 100, 150)`

**TOPIC FIVE ENDS HERE**

## 6) CSS HEX Value:

In CSS, a color can be applied using a hexadecimal value: **#f155a5**

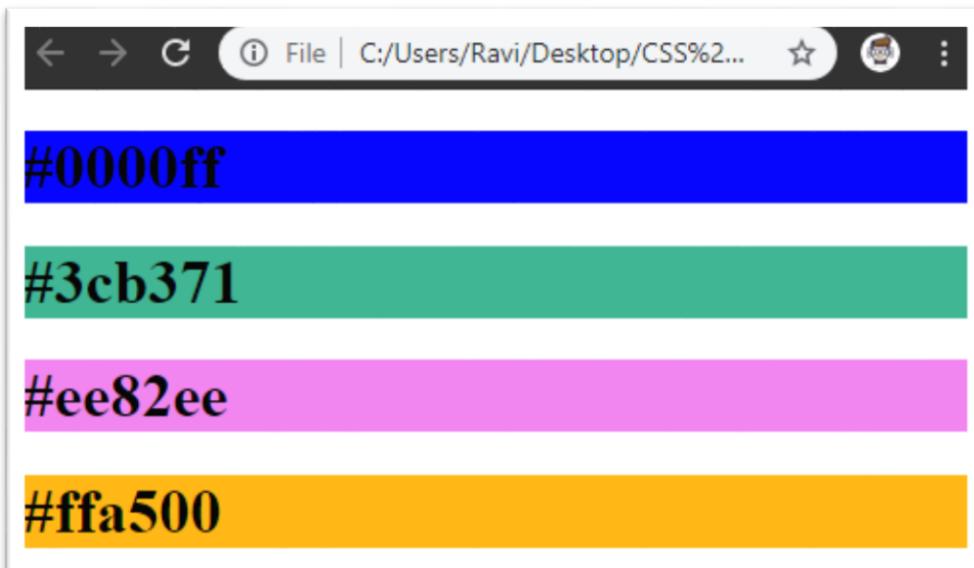
Example:

### Code:

```
<html>
<body>
<h1 style="background-color:#0000ff;">#0000ff</h1>
<h1 style="background-color:#3cb371;">#3cb371</h1>
<h1 style="background-color:#ee82ee;">#ee82ee</h1>
<h1 style="background-color:#ffb500;">#ffa500</h1>
</body>
</html>
```

### Result:

### Picture:



### Question:

Correct syntax for HEX Value is?

- 1) #ff0000 (Correct)
- 2) ff#857
- 3) ff0000#
- 4) ff0000

### Note:

In HTML, you can specify colors using Hex values.

**TOPIC SIX ENDS HERE**

### 7) CSS HSL Value:

In CSS, a color can be applied using hue, saturation, and lightness (HSL) in the form:

***hsl(hue, saturation, lightness).***

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white

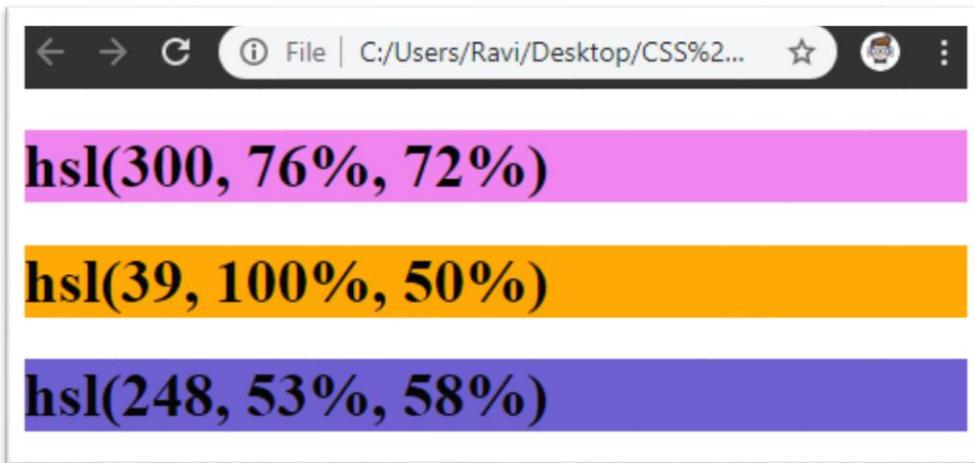
Example:

**Code:**

```
<html>
<body>
<h1 style="background-color:hsl(300, 76%, 72%);>hsl(300, 76%, 72%)</h1>
<h1 style="background-color:hsl(39, 100%, 50%);>hsl(39, 100%, 50%)</h1>
<h1 style="background-color:hsl(248, 53%, 58%);>hsl(248, 53%, 58%)</h1>
</body>
</html>
```

**Result:**

**Picture:**



### Question:

HSL consist of ?

- 1) hsl(saturation, lightness, hue)
- 2) hsl(hue, lightness, saturation)
- 3) hsl(hue, saturation, lightness) (Correct)

### Note:

In HTML, you can apply colors using HSL values.

**TOPIC SEVEN ENDS HERE**

### 8) CSS RGBA Value:

RGBA color values are an extension of RGB color values with an alpha channel - which adds the opacity for a color.

An RGBA color value is consist of:

***rgba(red, green, blue, alpha)***

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all).

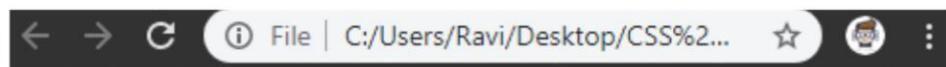
Example:

**Code:**

```
<html>
<body>
<h1 style="background-color:rgba(255, 109, 81, 0);">rgba(255, 99, 71, 0)</h1>
<h1 style="background-color:rgba(255, 109, 81, 0.2);">rgba(255, 99, 71,
0.2)</h1>
<h1 style="background-color:rgba(255, 109, 81, 0.4);">rgba(255, 99, 71,
0.4)</h1>
<h1 style="background-color:rgba(255, 109, 81, 0.6);">rgba(255, 99, 71,
0.6)</h1>
<h1 style="background-color:rgba(255, 109, 81, 0.8);">rgba(255, 99, 71,
0.8)</h1>
<h1 style="background-color:rgba(255, 109, 81, 1);">rgba(255, 99, 71, 1)</h1>
</body>
</html>
```

**Result:**

**Picture:**



**rgba(255, 99, 71, 0)**

**rgba(255, 99, 71, 0.2)**

**rgba(255, 99, 71, 0.4)**

**rgba(255, 99, 71, 0.6)**

**rgba(255, 99, 71, 0.8)**

**rgba(255, 99, 71, 1)**

### Question:

RGBA stands for?

- 1) red, blue, green, alpha
- 2) red, green, blue, alpha (Correct)**
- 3) green, red, blue, alpha

### Note:

You can make transparent colors by using the RGBA color value.

**TOPIC EIGHT ENDS HERE**

### 9) CSS HSLA Value:

HSLA color values are an extension of HSL color values with an alpha channel, which adds the opacity for a color.

The alpha parameter is a number between 0.0 (full transparent) and 1.0 (not transparent).

Example:

**Code:**

```
<html>
<body>
<h1 style="background-color:hsla(9, 100%, 64%, 0);>hsla(9, 100%, 64%, 0)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.2);>hsla(9, 100%, 64%, 0.2)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.4);>hsla(9, 100%, 64%, 0.4)</h1>

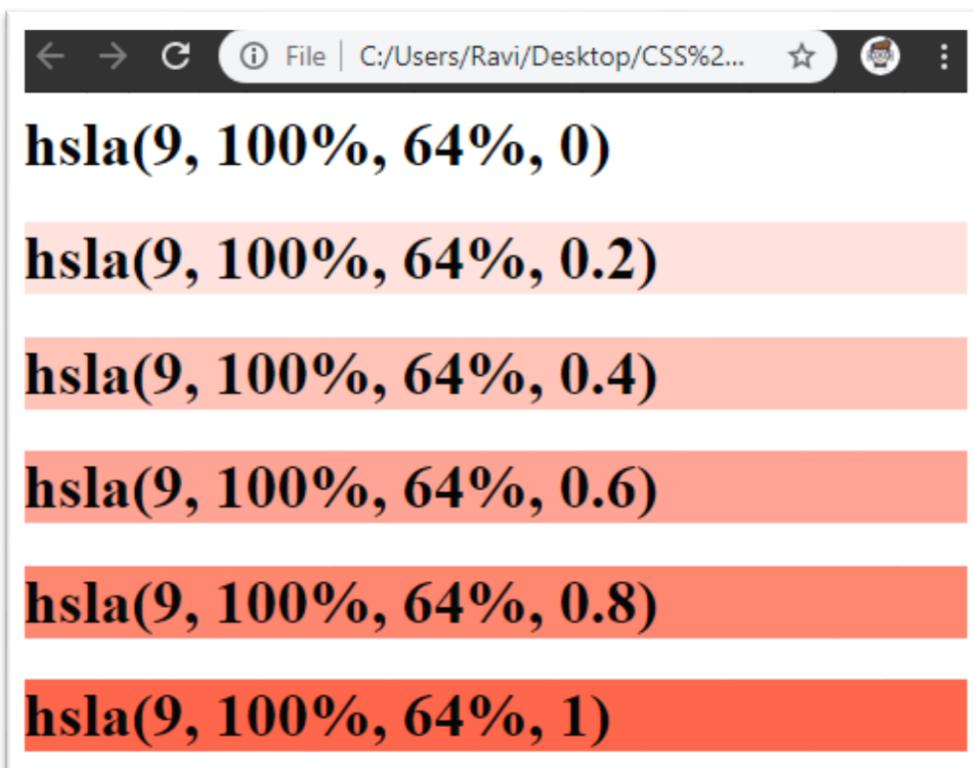
<h1 style="background-color:hsla(9, 100%, 64%, 0.6);>hsla(9, 100%, 64%, 0.6)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.8);>hsla(9, 100%, 64%, 0.8)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 1);>hsla(9, 100%, 64%, 1)</h1>
</body>
</html>
```

**Result:**

**Picture:**



### Question:

HSLA is an extension of?

- 1) rgb
- 2) hex
- 3) primary
- 4) hsl (Correct)

### Note:

By hsla you can make transparent colors.

**TOPIC NINE ENDS HERE**

**END OF MODULE TWO**

## MODULE THREE

### CSS BORDERS

The CSS `border` properties allow you to specify the style, width, and color of an element's border.

#### 1) CSS Border Style:

The `border-style` property specifies what kind of border to display.

The following values are allowed:

- `dotted` - Defines a dotted border

- **dashed** - Defines a dashed border
- **solid** - Defines a solid border
- **double** - Defines a double border
- **groove** - Defines a 3D grooved border. The effect depends on the border-color value
- **ridge** - Defines a 3D ridged border. The effect depends on the border-color value
- **inset** - Defines a 3D inset border. The effect depends on the border-color value
- **outset** - Defines a 3D outset border. The effect depends on the border-color value
- **none** - Defines no border
- **hidden** - Defines a hidden border

Example:

### Code:

```
<html>
<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;}  
</style>  
</head>  
<body>  
<h2>The border-style Property</h2>  
<p>This property specifies what kind of border to display:</p>  
<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>  
</body>  
</html>
```

**Result:**

**Picture:**

This screenshot shows a web browser window displaying various examples of CSS border styles. The browser interface includes a toolbar with icons for back, forward, file, and other functions. The main content area has a title "The border-style Property" and a descriptive text: "This property specifies what kind of border to display:". Below this, there are ten examples, each consisting of a label followed by a small rectangular box demonstrating the border style:

- 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.

The "A mixed border." example is highlighted with a dotted border around its text and box.

## Question:

The CSS border properties allow you to specify the style, width and?

- 1) height
- 2) background
- 3) color (Correct)
- 4) shadow

## Note:

None of the other CSS border properties described below will have any effect unless the `border-style` property is set!

The `border-style` property can have from one to four values (for the top border, right border, bottom border, and the left border).

## TOPIC ONE ENDS HERE

### 2) CSS Border Color:

The `border-color` property is used to set the color of the four borders.

The `border-color` property can have from one to four values (for the top border, right border, bottom border, and the left border).

The color can be set by:

- name - specify a color name, like "red"
- Hex - specify a hex value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- Transparent

Example:

#### Code:

```
<html>
<head>
<style>
p.one {
    border-style: solid;
    border-color: blue;
}
p.two {
    border-style: solid;
    border-color: yellow;
```

```

}

p.three {
    border-style: solid;
    border-color: yellow black green skyblue;
}

</style>

</head>

<body>

<h2>The border-color Property</h2>

<p>This property specifies the color of the four borders:</p>

<p class="one">A solid blue border</p>

<p class="two">A solid yellow border</p>

<p class="three">A solid multicolor border</p>

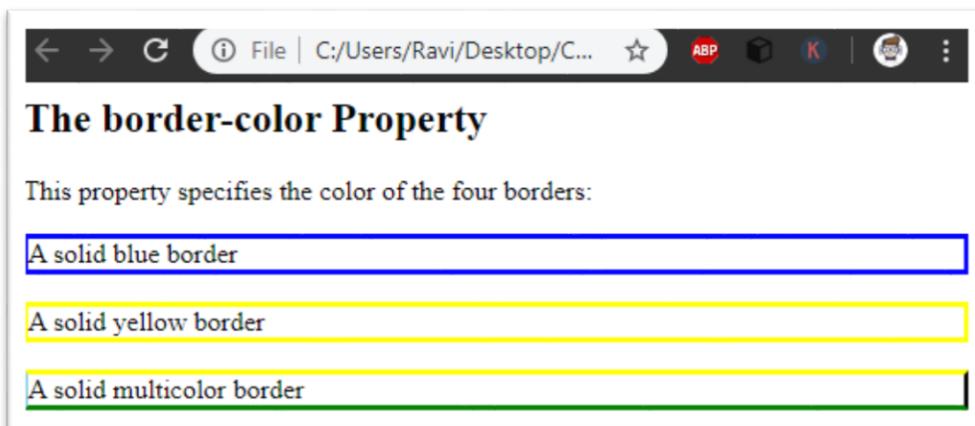
</body>

</html>

```

## Result:

### Picture:



## Question:

Border-color property is used to set?

- 1) width
- 2) color (Correct)**
- 3) background-color
- 4) text-color

## Note:

The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.

## TOPIC TWO ENDS HERE

### 2) CSS Border Width:

The **border-width** property is used for the width of the four borders.

The width can be set to a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

The **border-width** property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example:

#### Code:

```
<html>
<head>
<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;  
}  
  
p.seven {  
    border-style: solid;  
    border-width: 2px 10px 4px 20px;  
}
```

```
</style>

</head>

<body>

<h2>The border-width Property</h2>

<p>This property specifies the width of the four borders:</p>

<p class="one">Some text.</p>

<p class="two">Some text.</p>

<p class="three">Some text.</p>

<p class="four">Some text.</p>

<p class="five">Some text.</p>

<p class="six">Some text.</p>

<p class="seven">Some text.</p>

</body>

</html>
```

**Result:**

**Picture:**

The screenshot shows a web browser window with the title "The border-width Property". Below the title, a text block states: "This property specifies the width of the four borders:". Below this text are eight examples of different border widths:

- A solid black border with a width of 2px.
- A solid black border with a width of 1px.
- A dotted border with a width of 1px.
- A dotted border with a width of 2px.
- A solid black border with a width of 4px.
- A solid black border with a width of 3px.
- A solid black border with a width of 1px.
- A solid black border with a width of 1px.

## Question:

Border width property can have values?

- 1) two
- 2) one
- 3) four (Correct)
- 4) five

## Note:

Always specify the "border-style" property to set the borders first.

**TOPIC THREE ENDS HERE**

## 4) CSS Border Shorthand Property:

As you can see from the examples above, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property.

The `border` property is a shorthand property for the following individual border properties:

- `border-width`
- `border-style`
- `border-color`

Example:

**Code:**

```
<html>
  <head>
    <style>
      p {
        border: 8px solid green;
      }
    </style>
  </head>
  <body>
    <h2>The border Property</h2>
    <p>This property is a shorthand property for border-width, border-style, and border-color.</p>
  </body>
</html>
```

**Result:**

## Picture:



## Question:

For what border shorthand property is used?

- 1) To give clear example
- 2) To shorten the code(Correct)
- 3) To solve a problem

## Note:

You can also specify all the individual border properties for just one side.

**TOPIC FOUR ENDS HERE**

## 4) CSS Rounded Borders:

The `border-radius` property is used to add rounded borders to an element.

Example:

## Code:

```
<html>
<head>
<style>
p.normal {
    border: 2px solid green;
}

p.border1 {
    border: 2px solid blue;
    border-radius: 8px;
}

p.border2 {
    border: 2px solid pink;
    border-radius: 10px;
}

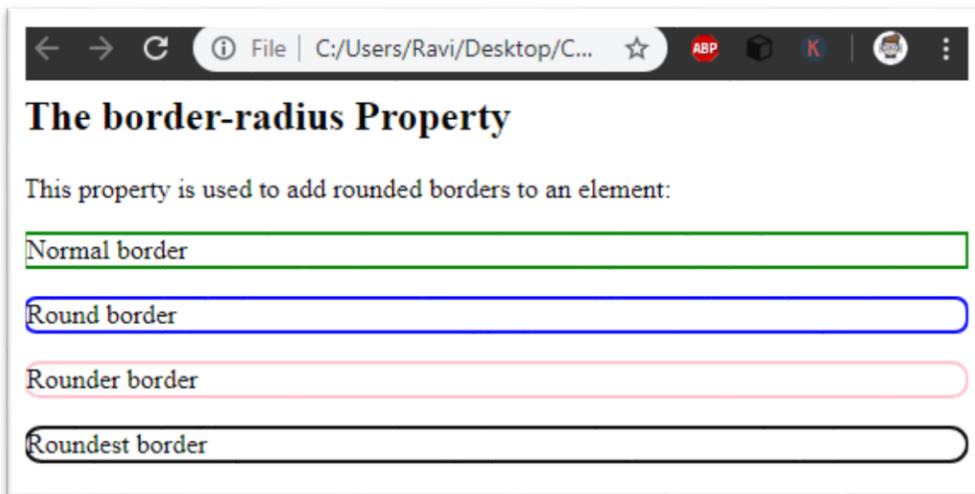
p.border3 {
    border: 2px solid black;
    border-radius: 15px;
}

</style>
</head>
<body>
<h2>The border-radius Property</h2>
<p>This property is used to add rounded borders to an element:</p>
<p class="normal">Normal border</p>
<p class="border1">Round border</p>
```

```
<p class="border2">Rounder border</p>  
<p class="border3">Roundest border</p>  
</body>  
</html>
```

## Result:

## Picture:



## Question:

Border radius property is used to add \_\_\_\_\_ to an element?

- 1) width
- 2) style
- 3) color
- 4) roundness (Correct)

## Note:

The **border-radius** property is not supported in IE8 and earlier versions.

**TOPIC FIVE ENDS HERE**

**END OF MODULE THREE**

## MODULE FOUR

### CSS MARGIN

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

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

#### 1) Margin - Individual Sides:

CSS has properties for applying margin to each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

All the margin properties have the following values:

- `auto` - the browser calculates the margin
- `length` - applies a margin in px, pt, cm, etc.
- `%` - applies a margin in % of the width of the containing element
- `inherit` - applies that the margin should be inherited from the parent element

Example:

**Code:**

```
<html>
<head>
<style>
div {
    border: 1px solid black;
    margin-top: 50px;
    margin-bottom: 70px;
    margin-right: 50px;
    margin-left: 90px;
    background-color: pink;
}
</style>
</head>
<body>
<h1>Using individual margin properties</h1>
<div>This div element has a top margin of 50px, a right margin of 70px, a bottom margin of 50px, and a left margin of 90px</div>
</body>
</html>
```

**Result:**

**Picture:**



## Question:

How many properties for setting the margin around an element?

- 1) three
- 2) two
- 3) four (correct)
- 4) one

## Note:

Negative values are also allowed.

**TOPIC ONE ENDS HERE**

## 2) Margin Inherit Value:

The inherit value takes the margin from the parent element.

Example:

**Code:**

```
<html>
<head>
<style>
div {
    border: 1px solid red;
    margin-right: 150px;
}
p.inherit {
    margin-right: inherit;
}
</style>
</head>
<body>
<h2>Use of the inherit value</h2>
<p>Let the right margin be inherited from the parent element:</p>
<div>
<p class="inherit">This paragraph has an inherited right margin (from the div
element).</p>
</div>
</body>
</html>
```

**Result:**

**Picture:**



## Question:

Inherit margin means?

- 1) take value from parent (correct)
- 2) take value from sibling
- 2) take value from descendant

## TOPIC TWO ENDS HERE

### 3) Margin Shorthand Property:

To shorten the code, it's possible to apply all the margin properties in one property.

The `margin` property is a shorthand property for the following individual margin properties:

- `margin-top`
  - `margin-right`
  - `margin-bottom`
  - `margin-left`
- 
- If the `margin` property has four values:
  - `margin: 20px 70px 85px 150px;`

- top margin is 20px
- right margin is 70px
- bottom margin is 85px
- left margin is 150px

Example:

**Code:**

```
<html>
<head>
<style>
div {
    margin: 20px 70px 85px 150px;
    background-color: pink;
    border: 3px solid green;
}
</style>
</head>
<body>
<h1>The margin shorthand property - 4 values</h1>
<div>This div element has a top margin of 20px, a right margin of 70px, a
bottom margin of 85px, and a left margin of 150px</div>
</body>
</html>
```

**Result:**

**Picture:**

This div element has a top margin of 20px, a right margin of 70px, a bottom margin of 85px, and a left margin of 150px.

### Question:

Margin Shorthand property can have \_\_\_\_\_ values?

- 1) two
- 2) three
- 3) four (correct)

### Note:

- If the `margin` property has one value:
- `margin: 10px;`
- all four margins are 10px.

**TOPIC THREE ENDS HERE**

### 4) Margin Auto Value:

By setting the margin property to `auto` you horizontally center the element within its container.

The element will take up the applied width, and the remaining space will be split equally between the left and right margins.

Example:

**Code:**

```
<html>
<head>
<style>
div {
    width: 200px;
    border: 3px solid blue;
    margin: auto;
}
</style>
</head>
<body>
<h1>Use of margin:auto</h1>
<div>
This div will be horizontally centered because we have set margin to auto.
</div>
</body>
</html>
```

**Result:**

**Picture:**



### Note:

Its best practice to use margin auto to center text absolutely inside a div.

**TOPIC FOUR ENDS HERE**

**END OF MODULE FOUR**

## MODULE FIVE

### CSS PADDING

The CSS **padding** properties are used to add space around an element's content, inside of any defined borders.

With CSS, you have control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

#### 1) Padding on Individual sides:

CSS has properties for applying the padding for each side of an element:

- padding-top
- padding-right

- padding-bottom
- padding-left

All the padding properties have the following values:

- *Length*, applies padding in px, pt, cm, etc.
- *%*, applies padding in % of the width of the containing element
- Inherit, applies that the padding should be inherited from the parent element

Example:

### Code:

```
<html>
<head>
<style>
div {
    border: 4px solid yellow;
    background-color: pink;
    padding-top: 30px;
    padding-right: 50px;
    padding-bottom: 30px;
    padding-left: 100px;
}
</style>
</head>
<body>
<h2>Using individual padding properties</h2>
<div>This div element has a top padding of 30px, a right padding of 50px, a bottom padding of 30px, and a left padding of 100px.</div>
</body>
```

```
</html>
```

**Result:**

**Picture:**



**Question:**

Padding-top means?

- 1) An element will have padding from left
- 2) An element will have padding from top (correct)
- 3) An element will have padding from bottom
- 4) An element will have padding from right

**Note:**

Negative values are not allowed in setting padding.

**TOPIC ONE ENDS HERE**

**2) Padding – shorthand property:**

The padding CSS property sets the padding area on all four sides of an element. Its a shorthand for padding-top, padding-bottom, padding-right, paddind-left.

Example:

**Code:**

```
<html>
<head>
<style>
div {
    border: 3px solid blue;
    padding: 20px 30px 60px 80px;
    background-color: pink;
}
</style>
</head>
<body>
<h2>The padding shorthand property - 4 values</h2>
<div>This div element has a top padding of 20px, a right padding of 30px, a bottom padding of 60px, and a left padding of 80px.</div>
</body>
</html>
```

**Result:**

**Picture:**

The padding shorthand property - 4 values

This div element has a top padding of 20px, a right padding of 30px, a bottom padding of 60px, and a left padding of 80px.

### Question:

An element's padding area is the space between its \_\_\_\_\_?

- 1) content and margin
- 2) content and border (correct)
- 3) padding and margin area

### Note:

Padding creates extra space within an element. In contrast, margin creates extra space *around* an element.

## TOPIC TWO ENDS HERE

### 3) Padding And Element Width:

The CSS width property applies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element (the box model).

So, if an element has a applied width, the padding added to that element will be added to the total width of the element. This is often an undesirable result.

Example:

Here, the <div> element is given a width of 500px. However, the actual width of the <div> element will be 600px (500px + 50px of left padding + 50px of right padding):

### Code:

```
<html>
<head>
<style>
div.no-padding {
    width: 500px;
    background-color: pink;
}

div.padding {
    width: 500px;
    padding: 50px;
    background-color: skyblue;
}

</style>
</head>
<body>
<h2>Padding and element width</h2>
<div class="no-padding">This div is 500px wide.</div>
<br>
<div class="padding">The width of this div is 600px, even though it is defined
as 500px in the CSS.</div>
</body>
</html>
```

## Result:

### Picture:



### Note:

To keep the width at 500px, no matter the amount of padding, you can use the `box-sizing` property. This causes the element to maintain its width; if you increase the padding, the available content space will decrease.

**TOPIC THREE ENDS HERE**

**END OF MODULE FIVE**

**MODULE SIX**

**CSS HEIGHT And WIDTH**

CSS includes height and width properties to help you specify the size of your elements.

height and width Properties

Applies to all HTML elements except non-replaced inline elements, table columns and column groups.

You can use a fixed height (i.e. pixels) or a percentage height.

## 1) CSS height/width Values:

The `height` and `width` properties may have the following values:

- `auto` - This is default. The browser calculates the height and width
- `length` - Defines the height/width in px, cm etc.
- `%` - Defines the height/width in percent of the containing block
- `initial` - Sets the height/width to its default value
- `inherit` - The height/width will be inherited from its parent value

Example:

### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
div {
    height: 150px;
    width: 75px;
    background-color: skyblue;
}
</style>
<div>
    This div has height and width applied.
</div>
```

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

## Result:

## Picture:



## Question:

The height and width properties do not include the area of?

- 1) padding
- 2) padding, margin
- 3) padding, margin, border (correct)

## Note:

Remember that the **height** and **width** properties do not include padding, borders, or margins! They set the height/width of the area inside the padding, border, and margin of the element!

**TOPIC ONE ENDS HERE**

## 2) Max-height and Max-width:

Enables you to constrain the height and/or width of an element to a maximum value.

Example:

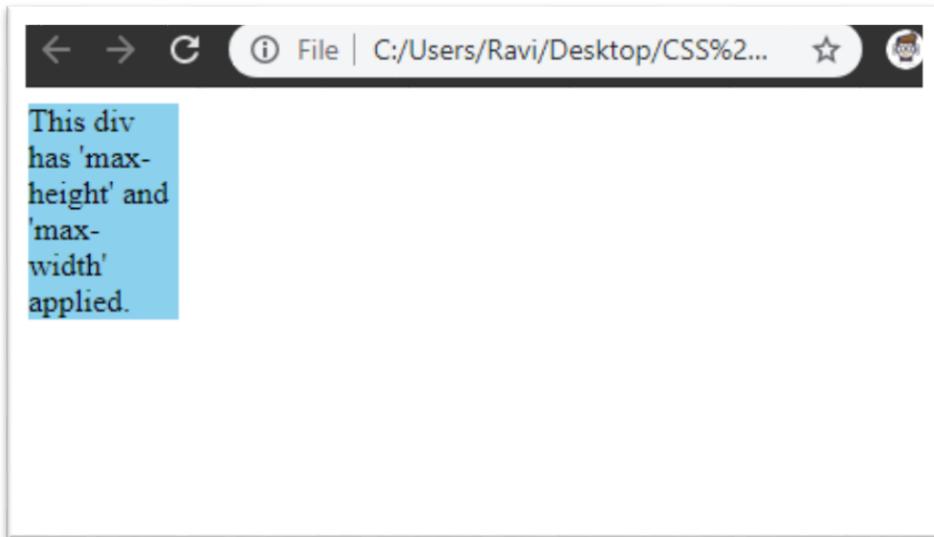
### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
div {
    max-height: 400px;
    max-width: 75px;
    background-color: skyblue;
}
</style>
</head>
<body>
<div>
    This div has 'max-height' and 'max-width' applied.
</div>
</body>
</html>
```

### Result:

### Picture:



### Question:

The max-height and max-width property is used to set \_\_\_\_\_ of an element?

- 1) maximum height and width (correct)
- 2) minimum width
- 3) minimum width and height

### Note:

The value of the `max-width` property overrides `width`.

**TOPIC TWO ENDS HERE**

### 3) Min-height and Min-width:

Enables you to set the height and/or width of an element to a minimum value.

Example:

### Code:

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
div {
    min-height: 150px;
    min-width: 75px;
    background-color: skyblue;
}
</style>
</head>
<body>
<div>
    <h1>This div has 'min-height' and 'min-width' applied.</h1>
</div>
</body>
</html>
```

## Result:

### Picture:



## **Question:**

Minimum height and Minimum width will be applied when?

- 1) content is larger than the minimum height and width
- 2) content is smaller than just minimum height
- 3) content is smaller than just minimum width
- 4) content is smaller than the minimum height and width (correct)

## **Note:**

This prevents the value of the `height` and `width` from becoming smaller than `min-height`.

**TOPIC THREE ENDS HERE**

**END OF MODULE SIX**

**MODULE SEVEN**

**CSS Text**

## **1) Text Color:**

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

- a color name - like "blue"
- a HEX value - like "#ff7856"
- an RGB value - like "rgb(255,255,255)"

Example:

**Code:**

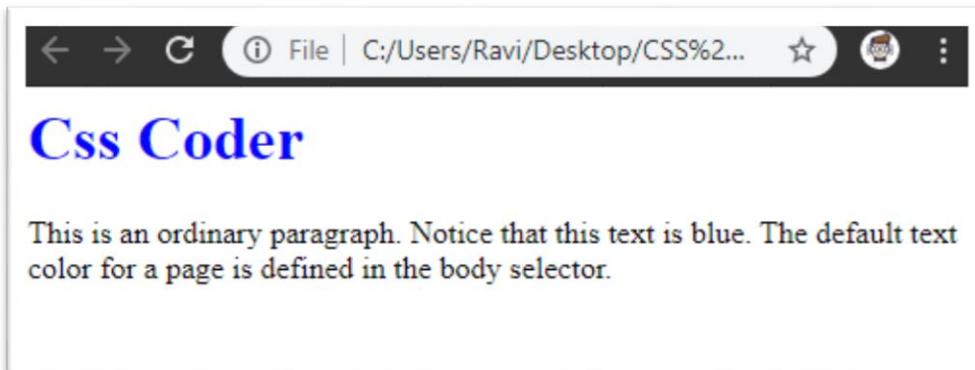
```
<!DOCTYPE html>

<html>
  <head>
    <style>
      body {
        color: black;
      }

      h1 {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Css Coder</h1>
    <p>This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.</p>
  </body>
</html>
```

**Result:**

## Picture:



## Note:

The default text color for a page is defined in the body selector.

## TOPIC ONE ENDS HERE

## 2) Text Decoration:

The `text-decoration` property is used to set or remove decorations from text.

### Example:

### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
a {
    text-decoration: none;
}
h1 {
    text-decoration: overline;
```

```
}

h2 {
    text-decoration: line-through;
}

h3 {
    text-decoration: underline;
}

</style>

</head>

<body>

<p>A link with no underline: <a
    href="https://www.instagram.com/css_coder/">Css_coder</a></p>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

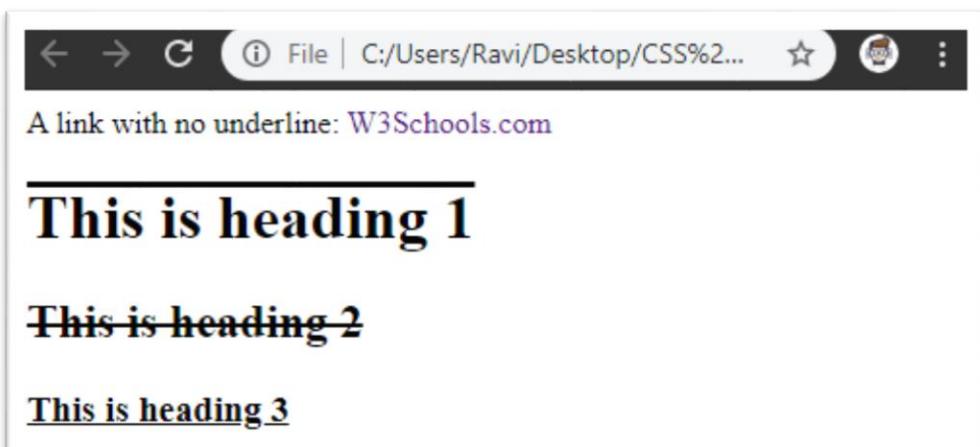
<h3>This is heading 3</h3>

</body>

</html>
```

**Result:**

**Picture:**



### Note:

The value `text-decoration: none;` is often used to remove underlines from links.

It is not recommended to underline text that is not a link, as this often confuses the reader.

## TOPIC TWO ENDS HERE

### 3) Text Transformation:

The `text-transform` property is used to apply uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Example:

#### Code:

```
<html>
<head>
<style>
h1.uppercase {
    text-transform: uppercase;
```

```
}

h1.lowercase {
    text-transform: lowercase;
}

h1.capitalize {
    text-transform: capitalize;
}

</style>

</head>

<body>

<h1 class="uppercase">Instagram/css_coder.</h1>

<h1 class="lowercase">Youtube/Css Coder.</h1>

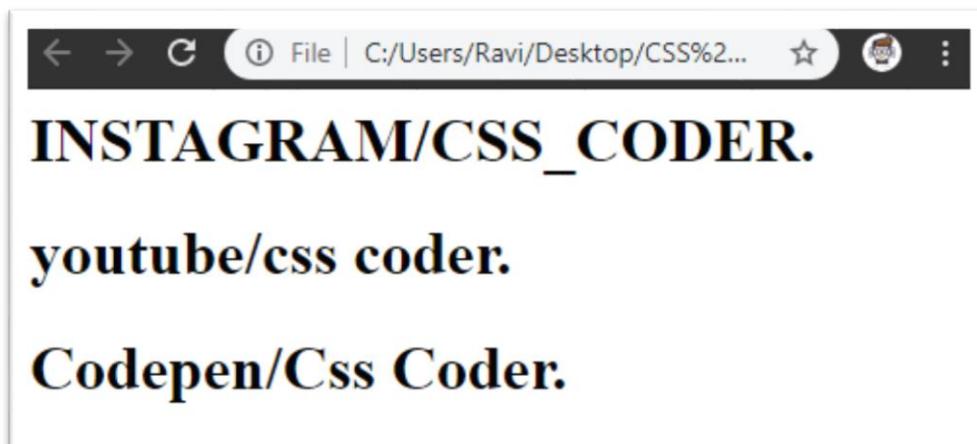
<h1 class="capitalize">Codepen/Css Coder.</h1>

</body>

</html>
```

## Result:

## Picture:



## Question:

Difference between text-decoration and text-transformation?

- 1) text transform removes decoration from text and text decoration is used to add color to text
- 2) text transform removes decoration from text and text decoration makes text uppercase or lowercase
- 3) text transform makes text uppercase or lowercase and text decoration removes decoration from text (correct)

## Note:

Capitalize is used to change the first letter of each word to uppercase.

**TOPIC THREE ENDS HERE**

## 4) Text Alignment:

Text alignment property is used to set the horizontal alignment of the text.

The text can be set to left, right, centered and justified alignment.

In justified alignment, line is stretched such that left and right margins are straight.

Example:

### Code:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h1
```

```
{  
color:red;  
text-align:center;  
}  
  
h2  
{  
color:green;  
text-align:left;  
}  
  
</style>  
</head>  
<body>  
<h1>  
CSS CODER  
</h1>  
<h2>  
TEXT FORMATTING  
</h2>  
</body>  
</html>
```

**Result:**

**Picture:**



### Question:

Text-alignment is used for?

- 1) remove decorations from text
- 2) horizontal alignment of text (correct)
- 3) indentation of first line of text

### Note:

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

**TOPIC FOUR ENDS HERE**

### 5) Letter Spacing:

This property is used to specify the space between the characters of the text.

Example:

### Code:

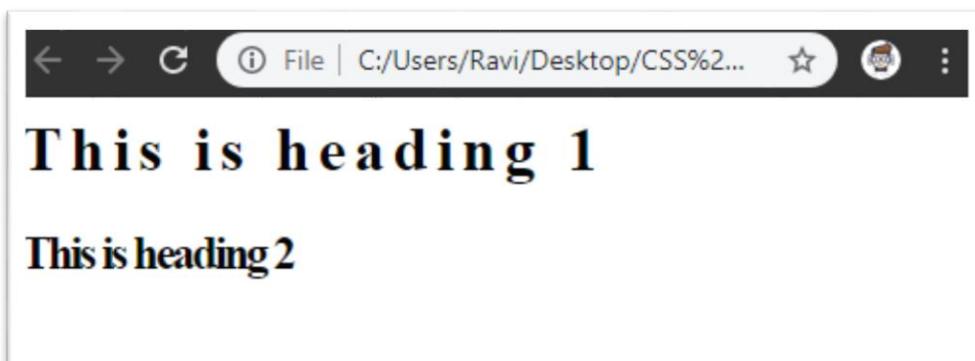
```
<!DOCTYPE html>

<html>
<head>
<style>
h1 {
    letter-spacing: 4px;
}

h2 {
    letter-spacing: -2px;
}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
</body>
</html>
```

**Result:**

**Picture:**



## Question:

Letter spacing is used for?

- 1) space between the characters in a text (correct)
- 2) space between the words in a text
- 3) alignment of text

## Note:

The size can be given in px for letter spacing.

## TOPIC FIVE ENDS HERE

## 6) Line Height:

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

Example:

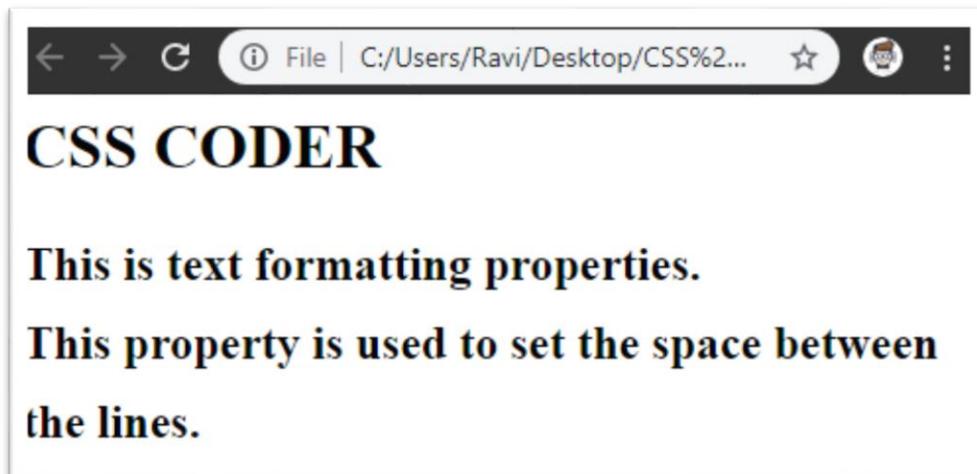
### Code:

```
<html>
<head>
<style>
h2
{
line-height:40px;
}
</style>
</head>
```

```
<body>
<h1>
CSS CODER
</h1>
<h2>
This is text formatting properties.<br>
This property is used to set the space between the lines.
</h2>
</body>
</html>
```

## Result:

### Picture:



## Question:

Line Height property is used for?

- 1) change the direction of an element
- 2) adds shadow to text
- 3) specifiy space between the lines (correct)

## TOPIC SIX ENDS HERE

### 7) Text Direction:

The `direction` property is used to change the text direction of an element.

Example:

Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
h2
{
    direction: rtl;
    text-align:center;
}
</style>
</head>
<body>
<h1>
CSS CODER
</h1>
<h2><bdo dir="rtl">
This is text formating properties.
</bdo>
```

```
</h2>  
</body>  
</html>
```

**Result:**

**Picture:**



**Note:**

The direction can be set by using rtl : right to left .

**TOPIC SEVEN ENDS HERE**

**8) Word Spacing :**

Word spacing is used to specify the space between the words of the line.

Example:

**Code:**

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

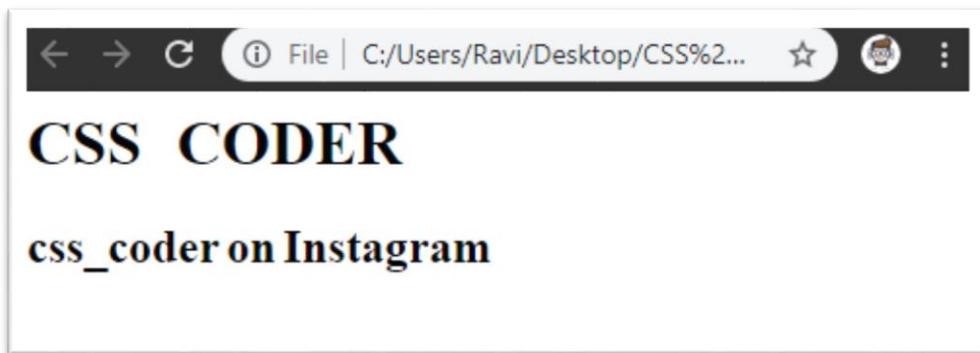
```
<style>
h1 {
    word-spacing: 10px;
}

h2 {
    word-spacing: -2px;
}

</style>
</head>
<body>
<h1>CSS CODER</h1>
<h2>css_coder on Instagram</h2>
</body>
</html>
```

## Result:

## Picture:



## Note:

The size can be given in px.

## TOPIC EIGHT ENDS HERE

### 9) Text Shadow:

The `text-shadow` property adds shadow to text.

Example:

#### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
h1
{
text-shadow: 3px 1px blue;
}
</style>
</head>
<body>
<h1>
CSS CODER
</h1>
<h2>
This is text formating properties.
</h2>
</body>
</html>
```

**Result:**

**Picture:**



**Note:**

You can specify the horizontal size, vertical size and shadow color for the text.

## TOPIC NINE ENDS HERE

**10) Text Indentation:**

Text indentation property is used to indent the first line of the paragraph.

**Example:**

**Code:**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

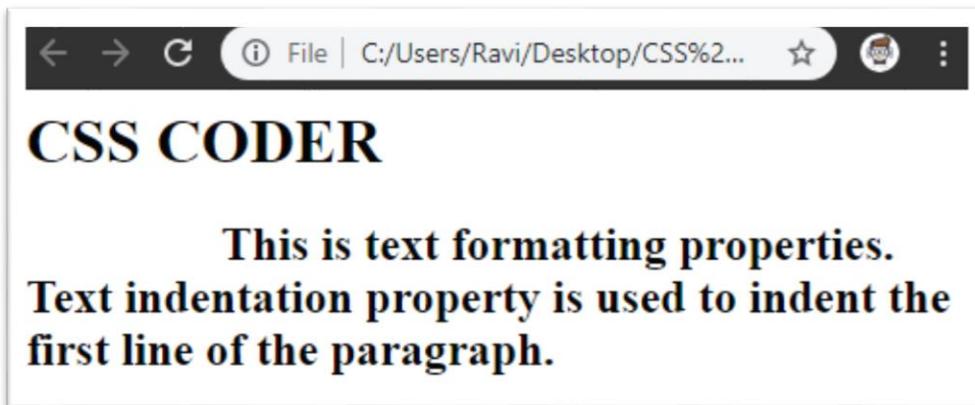
```
<style>
```

```
h2
```

```
{  
text-indent: 100px;  
}  
</style>  
</head>  
<body>  
<h1>  
CSS CODER  
</h1>  
<h2>  
This is text formating properties.<br>  
Text indentation property is used to indent the first line of the paragraph.  
</h2>  
</body>  
</html>
```

**Result:**

**Picture:**



**Note:**

The size can be in px, cm, pt.

**TOPIC TEN ENDS HERE**

**END OF MODULE SEVEN**

## MODULE EIGHT

### CSS BOX MODEL

#### 1) Css Box Model:

CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.

Box-Model has multiple properties in CSS. Some of them are given below:

- borders
- margins
- padding
- Content

#### Picture:



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

Example:

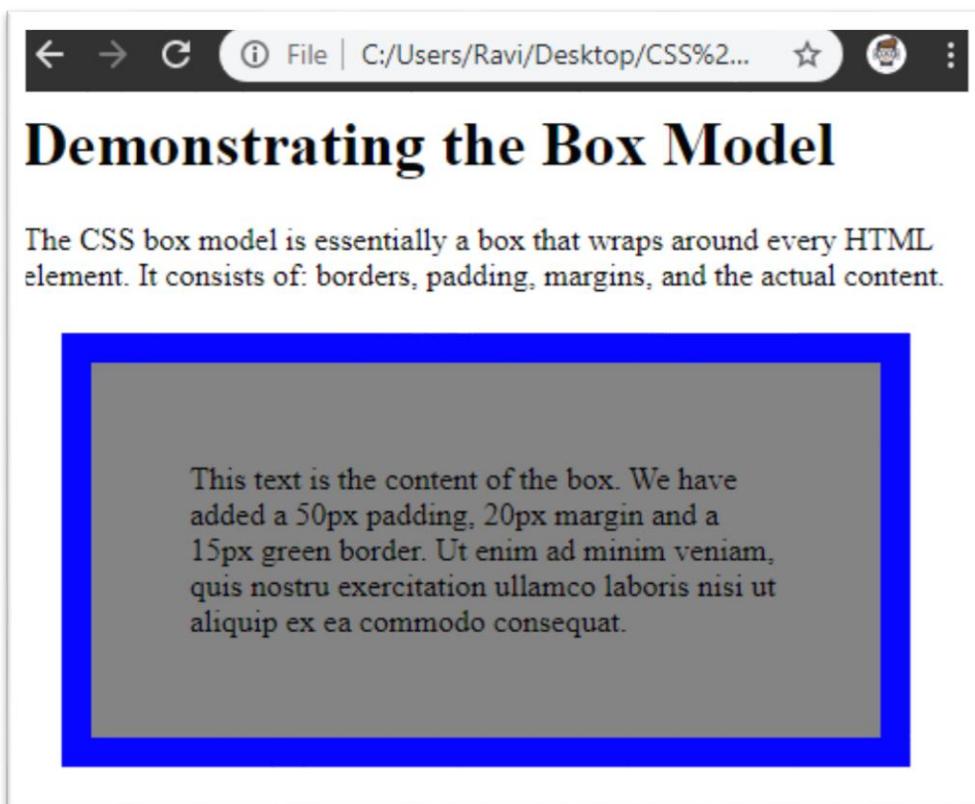
**Code:**

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      div {
        background-color: grey;
        width: 300px;
        border: 15px solid blue;
        padding: 50px;
        margin: 20px;
      }
    </style>
  </head>
  <body>
    <h1>Demonstrating the Box Model</h1>
    <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.</div>
  </body>
</html>
```

## Result:

### Picture:



## Question:

The top placed property for css box model is?

- 1) padding
- 2) border
- 3) margin (correct)
- 4) content

## Note:

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

## END OF MODULE EIGHT

# MODULE NINE

## CSS FONTS

### Css Fonts:

The CSS font property is used to set the fonts content of HTML element. There are many font property in CSS which are discussed below:

- font-family
- font-style
- font-weight
- font-variant
- font-size

### 1) Font Family:

The font family of a text is set with the `font-family` property.

It is used to set the font type of an HTML element. It holds several font names as a fallback system.

Example:

### Code:

```
<!DOCTYPE html>
<html>
<head>
<style>
.css-coder {
```

```

font-family:"Times New Roman";
font-weight:bold;
font-size:40px;
color:#090;
text-align:center;

}

.student {
font-family:"Comic Sans MS", cursive, sans-serif;
text-align:center;
}

</style>

</head>

<body>

<div class = "css-coder">CSS CODER</div>

<div class = "student">A Computer Science Student</div>

</body>

</html>

```

**Result:**

**Picture:**



## Note:

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

## TOPIC ONE ENDS HERE

### 2) Font Style:

The `font-style` property is mostly used to specify italic text.

This property has three values:

- `normal` - The text is shown normally
- `italic` - The text is shown in italics
- `oblique` - The text is "leaning" (oblique is very similar to italic, but less supported)

Example:

### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>

h1.normal {
    font-style: normal;
}

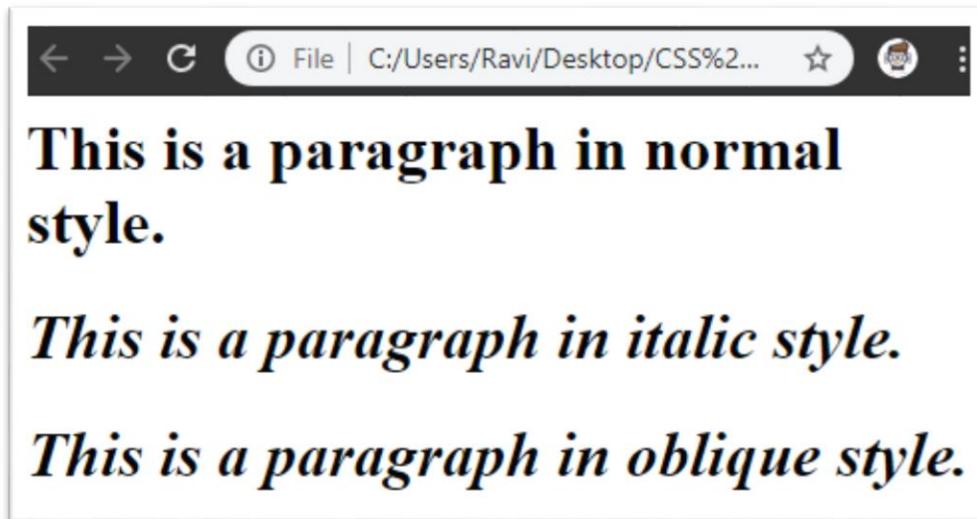
h1.italic {
    font-style: italic;
}

h1.oblique {
```

```
font-style: oblique;  
}  
</style>  
</head>  
<body>  
<h1 class="normal">This is a paragraph in normal style.</h1>  
<h1 class="italic">This is a paragraph in italic style.</h1>  
<h1 class="oblique">This is a paragraph in oblique style.</h1>  
</body>  
</html>
```

## Result:

### Picture:



## Note:

### Syntax:

font-style: style name;

## TOPIC TWO ENDS HERE

### 3) Font Size:

It is used to set the font size of an HTML element. The font-size can be set in different ways like in “pixels, percentage, em or we can set values like small, large” etc.

#### A) Font-size using pixels:

Example:

#### Code:

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      h1 {
        font-size: 50px;
      }

      h2 {
        font-size: 40px;
      }

      p {
        font-size: 24px;
      }
    </style>
  </head>
```

```
<body>  
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
<p>This is a paragraph.</p>  
<p>This is another paragraph.</p>  
</body>  
</html>
```

## Result:

### Picture:



## Note:

If you do not specify a font size, the default size for normal text, like paragraphs, is 16px (16px=1em).

## B) Font-size using Em:

To allow users to resize the text (in the browser menu), many developers use em instead of pixels.

1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.

Example:

**Code:**

```
<!DOCTYPE html>

<html>
<head>
<style>
h1 {
    font-size: 2.5em; /* 40px/16=2.5em */
}
h2 {
    font-size: 1.875em; /* 30px/16=1.875em */
}
p {
    font-size: 0.875em; /* 14px/16=0.875em */
}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
```

```
<p>Specifying the font-size in em allows all major browsers to resize the text.
```

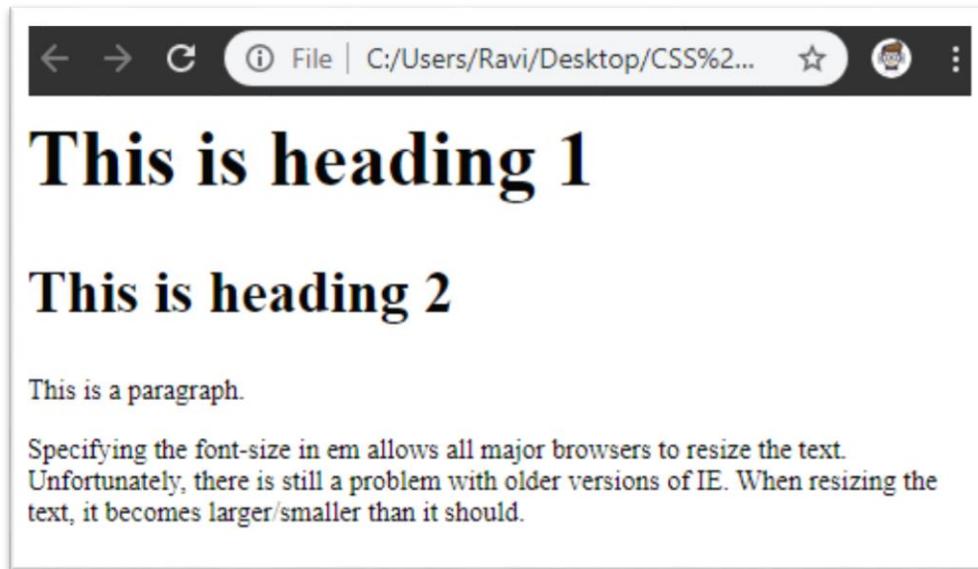
```
Unfortunately, there is still a problem with older versions of IE. When resizing  
the text, it becomes larger/smaller than it should.</p>
```

```
</body>
```

```
</html>
```

## Result:

### Picture:



## Question:

1em is equal to?

- 1) 18px
- 2) 14px
- 3) 16px (correct)**
- 4) 20px

## Note:

The size can be calculated from pixels to em using this formula: *pixels/16=em*

## TOPIC THREE ENDS HERE

### 4) Font Weight:

It is used to set the boldness of the font. Its value can be “normal, bold, lighter, bolder”.

Example:

#### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>

h2.normal {
    font-weight: normal;
}

h2.light {
    font-weight: lighter;
}

h2.thick {
    font-weight: bold;
}

h2.thicker {
    font-weight: 900;
}
```

```
</style>

</head>

<body>

<h2 class="normal">Heading with Normal Font Weight.</h2>

<h2 class="light">Heading with lighter Font Weight..</h2>

<h2 class="thick">Heading with bold Font Weight.</h2>

<h2 class="thicker">Heading with 900 Font Weight.</h2>

</body>

</html>
```

## Result:

## Picture:



## Question:

Font weight is used for?

- 1) length
- 2) size
- 3) width

4) weight (correct)

## TOPIC FOUR ENDS HERE

### 5) Font Variant:

It is used to create the small-caps effect. It can be “normal or small-caps”.

Example:

#### Code:

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      h1.normal {
        font-variant: normal;
      }
      h1.small {
        font-variant: small-caps;
      }
    </style>
  </head>
  <body>
    <h1 class="normal">Hello Its me Css Coder</h1>
    <h1 class="small">A Computer Science Student And Freelancer And
Youtuber</h1>
  </body>
```

```
</html>
```

**Result:**

**Picture:**



**Note:**

In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

**TOPIC FIVE ENDS HERE**

**END OF MODULE NINE**

# MODULE TEN

## CSS OUTLINE

CSS outline property allows us to draw a line around the element, outside the border.

CSS has following outline properties:

1. Outline-style
2. Outline-color
3. Outline-width
4. Outline-offset

### 1) OUTLINE STYLE:

The outline-style property tells us the style or type of outline.

Any other outline property cannot be accessed without setting the outline-style.

Types of outline-style:

1. Dotted
2. Dashed
3. Solid
4. Double
5. Groove
6. Ridge
7. Inset
8. Outset

Example:

### Code:

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

```
p {outline-color: blue;}  
p.dotted {outline-style: dotted;}  
p.dashed {outline-style: dashed;}  
p.solid {outline-style: solid;}  
p.double {outline-style: double;}  
p.groove {outline-style: groove;}  
p.ridge {outline-style: ridge;}  
p.inset {outline-style: inset;}  
p.outset {outline-style: outset;}  
  
p{  
font-size: 24px;  
}  
  
</style>  
</head>  
  
<body>  
  
<h2>The outline-style Property</h2>  
  
<p class="dotted">A dotted outline</p>  
  
<p class="dashed">A dashed outline</p>  
  
<p class="solid">A solid outline</p>  
  
<p class="double">A double outline</p>  
  
<p class="groove">A groove outline. The effect depends on the outline-color  
value.</p>  
  
<p class="ridge">A ridge outline. The effect depends on the outline-color  
value.</p>  
  
<p class="inset">An inset outline. The effect depends on the outline-color  
value.</p>
```

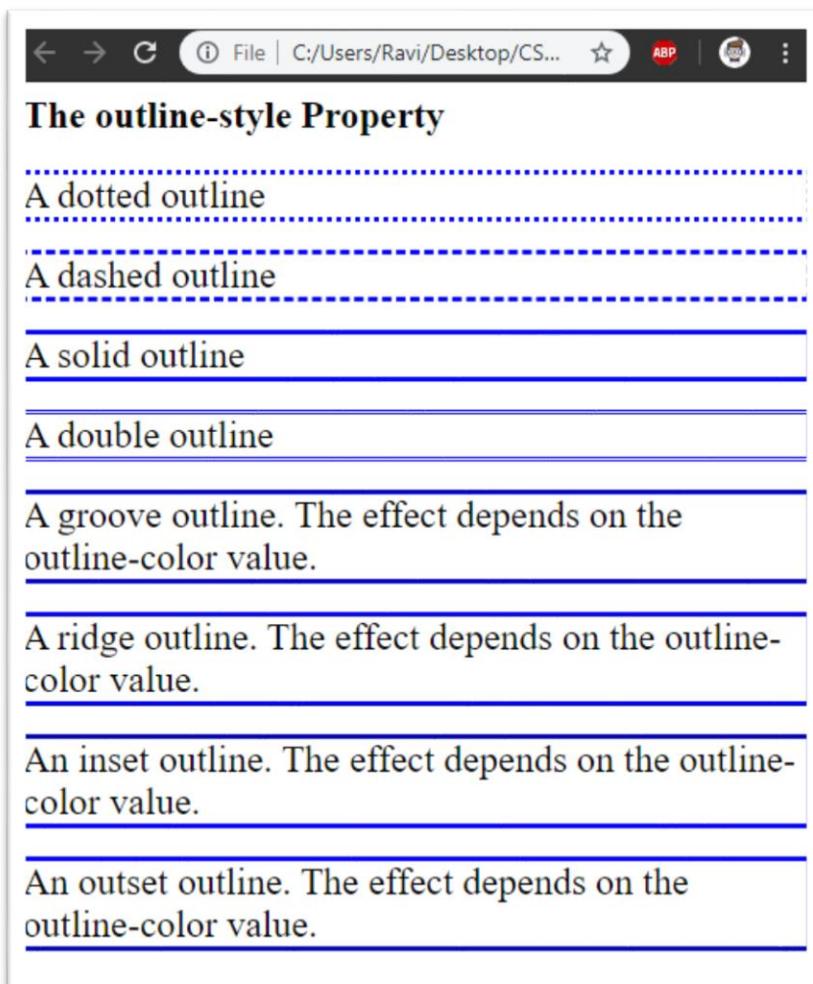
```
<p class="outset">An outset outline. The effect depends on the outline-color value.</p>
```

```
</body>
```

```
</html>
```

## Result:

### Picture:



### Question:

Outline style specifies the \_\_\_\_\_ of ouline?

- 1) color
- 2) width

3) style (correct)

4) font-style

### Note:

Outline differs from borders! Unlike border, the outline is drawn outside the element's border, and may overlap other content. Also, the outline is not a part of the element's dimensions; the element's total width and height is not affected by the width of the outline.

## TOPIC ONE ENDS HERE

### 2) OUTLINE COLOR:

Outline color property specifies the color of the outline.  
The color can be set by its name, RGB value or hex value.

Example:

#### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
h3
{
border: solid red 4px;
outline-style: solid;
outline-color: yellow;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>CSS CODER</h1>
```

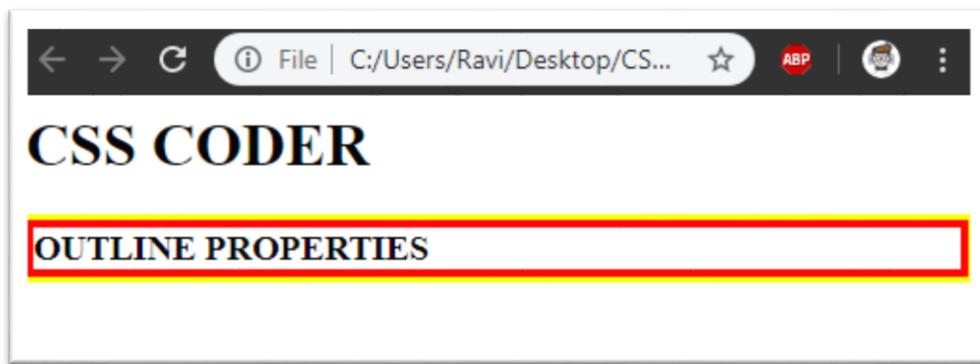
```
<h3>OUTLINE PROPERTIES</h3>
```

```
</body>
```

```
</html>
```

## Result:

## Picture:



## Question:

Outline-color specifies?

- 1) width
- 2) style
- 3) solid outline
- 4) color (correct)

**TOPIC TWO ENDS HERE**

### 3) OUTLINE WIDTH:

Outline width property is used to set the width of the outline. Width of the outline can be set by specifying the size of the width in px, cm, pt, etc or by using terms like thin, thick, medium.

Example:

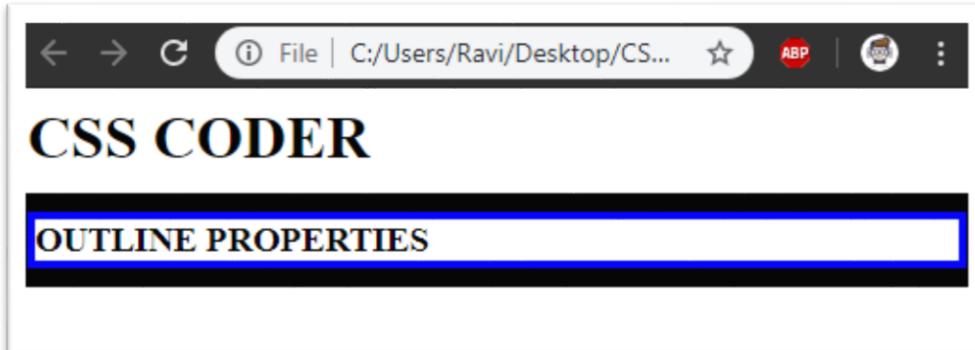
#### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
h3
{
border:solid blue 4px;
outline-style:solid;
outline-width:8px;
}
</style>
</head>
<body>
<h1> CSS CODER</h1>
<h3>OUTLINE PROPERTIES</h3>
</body>
</html>
```

#### Result:

#### Picture:



## TOPIC THREE ENDS HERE

### 4) OUTLINE OFFSET:

Outline offset property is used to specify space between the outline and the border of the element.

Example:

#### Code:

```
<!DOCTYPE html>

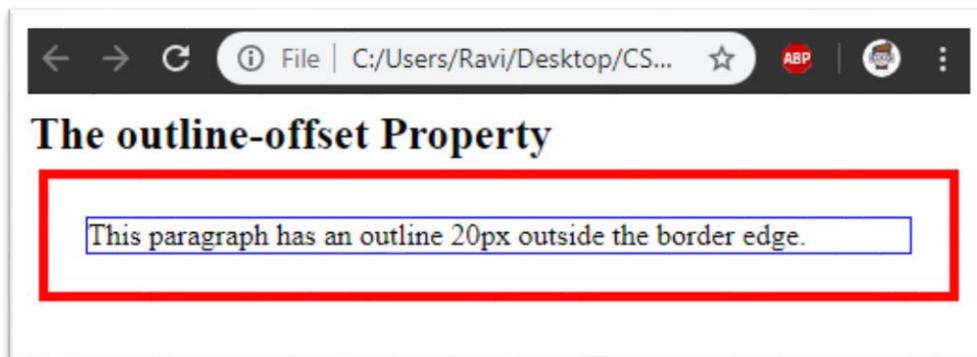
<html>
<head>
<style>

p {
    margin: 30px;
    border: 1px solid blue;
    outline: 1px solid red;
    outline-offset: 20px;
    outline-width: 5px;
}
```

```
</style>
</head>
<body>
<h2>The outline-offset Property</h2>
<p>This paragraph has an outline 20px outside the border edge.</p>
</body>
</html>
```

## Result:

### Picture:



**TOPIC FOUR ENDS HERE**

## 5) OUTLINE SHORTHAND PROPERTY:

The **outline** property is a shorthand property for setting the following individual outline properties:

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

The **outline** property is specified as one, two, or three values from the list above. The order of the values does not matter.

Example:

**Code:**

```
<!DOCTYPE html>

<html>
<head>
<style>

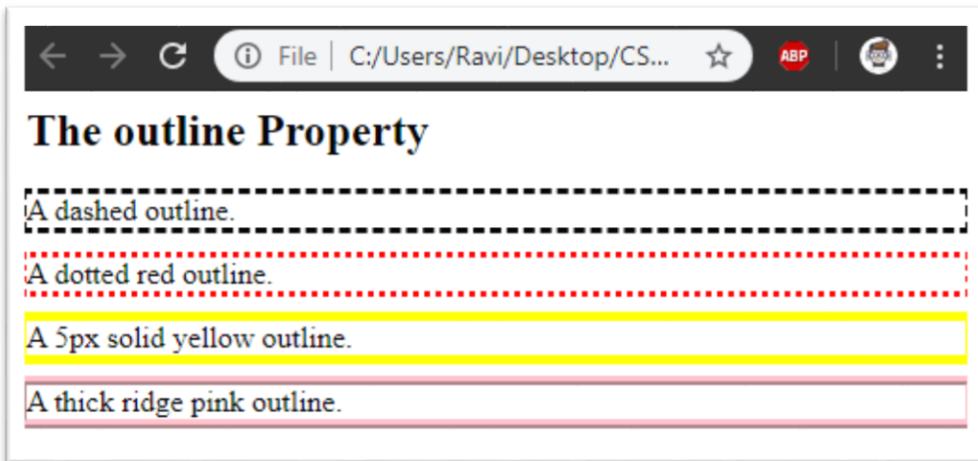
p.ex1 {outline: dashed;}
p.ex2 {outline: dotted red;}
p.ex3 {outline: 5px solid yellow;}
p.ex4 {outline: thick ridge pink;}

</style>
</head>
<body>
<h2>The outline Property</h2>
<p class="ex1">A dashed outline.</p>
<p class="ex2">A dotted red outline.</p>
<p class="ex3">A 5px solid yellow outline.</p>
<p class="ex4">A thick ridge pink outline.</p>

</body>
</html>
```

**Result:**

**Picture:**



### Question:

Outline shorthand property can accept how many values?

- 1) One
- 2) Two
- 3) Three
- 4) four
- 5) All of the above (correct)

**TOPIC FIVE ENDS HERE**

**END OF MODULE TEN**

# MODULE ELEVEN

## CSS LISTS

### HTML Lists and CSS List Properties:

In HTML, there are two main types of lists:

- unordered lists (<ul>) - the list items are marked with bullets
- ordered lists (<ol>) - the list items are marked with numbers or letters

The CSS list properties allow you 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

### 1) Different List Item Markers:

This property specifies the type of item marker i.e. unordered list or ordered.

Now, the value can be following:

- circle
- decimal , eg :1,2,3,etc
- decimal-leading-zeroes , eg :01,02,03,04,etc
- lower-roman
- upper-roman
- lower-alpha, eg : a,b,c,etc
- upper-alpha, eg : A,B,C,etc
- square

Example:

## Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
ul.circle {
    list-style-type: circle;
}
ul.square {
    list-style-type: square;
}
ol.upper-roman {
    list-style-type: upper-roman;
}
ol.lower-alpha {
    list-style-type: lower-alpha;
}
</style>
</head>
<body>
<p>Example of unordered lists:</p>
<ul class="circle">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Coca Cola</li>
</ul>
```

```
<ul class="square">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>

<p>Example of ordered lists:</p>

<ol class="upper-roman">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>

<ol class="lower-alpha">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>

</body>
</html>
```

**Result:**

**Picture:**

The screenshot shows a web browser window with the following content:

Example of unordered lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Example of ordered lists:

- I. Coffee
- II. Tea
- III. Coca Cola

- a. Coffee
- b. Tea
- c. Coca Cola

## Question:

In HTML there are \_\_\_\_ kind of lists?

- 1) three
- 2) one
- 3) two (correct)
- 4) four

## Note:

Some of the values are for unordered lists, and some for ordered lists.

**TOPIC ONE ENDS HERE**

## 2) An Image as the List Item Marker:

This property specifies image as list item marker.

Example:

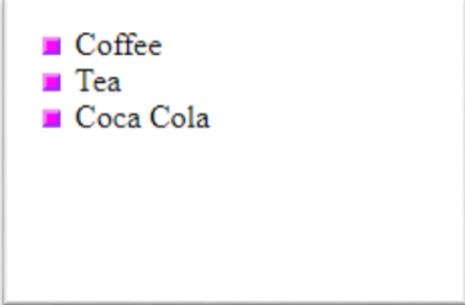
**Code:**

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      ul {
        list-style-image: url('sqpurple.gif');
      }
    </style>
  </head>
  <body>
    <ul>
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ul>
  </body>
</html>
```

**Result:**

**Picture:**

- 
- Coffee
  - Tea
  - Coca Cola

**Question:**

List style image specifies?

- 1) bullet as marker
- 2) circle as marker
- 3) square as marker
- 4) image as marker (correct)

## TOPIC TWO ENDS HERE

### **3) List Marker Position:**

This property specifies the position of the list item marker. There are 2 types of position marker:

#### **A) list-style-position: outside**

In this, the bullet points will be outside the list item. The start of each line of the list will be aligned vertically.

Example:

**Code:**

```
<!DOCTYPE>

<html>

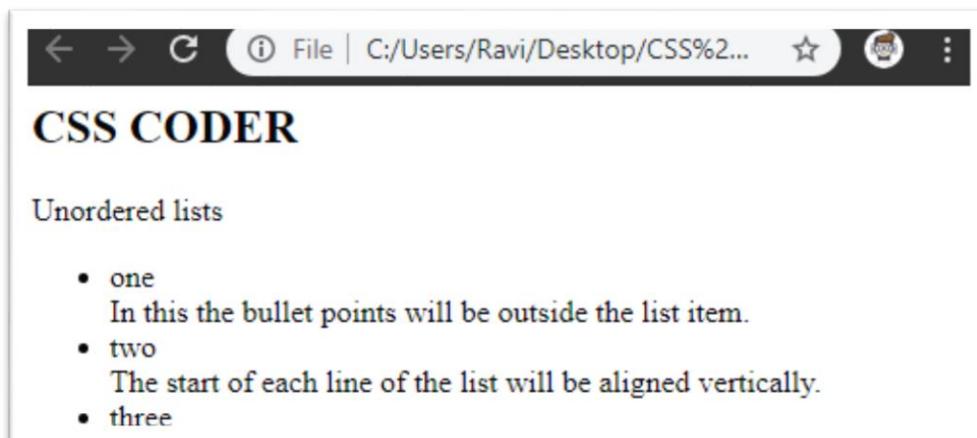
<head>
<style>
    ul.a
    {
        list-style-position:outside;
    }
</style>
</head>

<body>
<h2>
    CSS CODER
</h2>
<p>
    Unordered lists
</p>
<ul class="a">
    <li>one <br>In this the bullet points will be outside the list item.</li>
    <li>two<br>
        The start of each line of the list will be aligned vertically.
    </li>
    <li>three</li>
</ul>
```

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

## Result:

### Picture:



## B) list-style-position: inside

In this the bullet points will be inside the list. The line along with the bullet points will be aligned vertically.

Example:

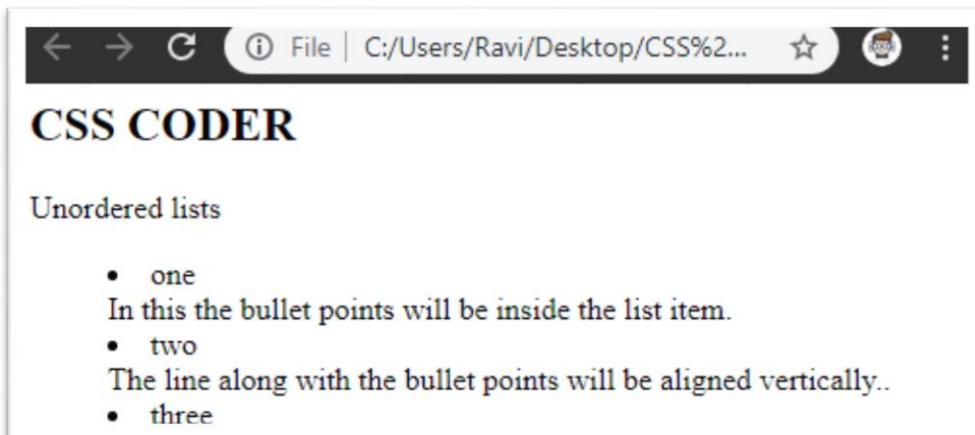
### Code:

```
<!DOCTYPE>  
<html>  
<head>  
  <style>  
    ul.a  
    {
```

```
list-style-position:inside;  
}  
</style>  
</head>  
<body>  
<h2>  
    CSS CODER  
</h2>  
<p>  
    Unordered lists  
</p>  
<ul class="a">  
    <li>one <br>In this the bullet points will be inside the list  
item.</li>  
    <li>two<br>  
        The line along with the bullet points will be aligned vertically..  
    </li>  
    <li>three</li>  
</ul>  
</body>  
</html>
```

**Result:**

**Picture:**



**Question:**

List-style-position: outside means?

- 1) bullet points outside the list item (correct)
- 2) bullet inside the list item
- 3) bullet at the center of list item

**TOPIC THREE ENDS HERE**

#### **4) List - Shorthand property:**

This property allows us to set all the list properties in one command. The order of property is a type, position, and image.

Example:

**Code:**

```
<!DOCTYPE>

<html>

<head>
<style>
    ul.a
    {
        list-style:square inside;
    }
</style>
</head>

<body>
<h2>
    CSS CODER
</h2>
<p>
    Unordered lists
</p>
<ul class="a">
    <li>one</li>
    <li>two</li>
    <li>three</li>
</ul>
</body>
</html>
```

**Result:**

**Picture:**



**Question:**

Shorthand property is used to set all list properties in?

- 1) two declarations
- 2) one declaration (correct)
- 3) three declarations
- 4) four declarations

**TOPIC FOUR ENDS HERE**

## **5) Remove Default List Settings:**

The **list-style-type:none** property can also be used to remove the markers/bullets. Note that the list also has default margin and padding. To remove this, add **margin:0** and **padding:0** to **<ul>** or **<ol>**.

**Example**

**Code:**

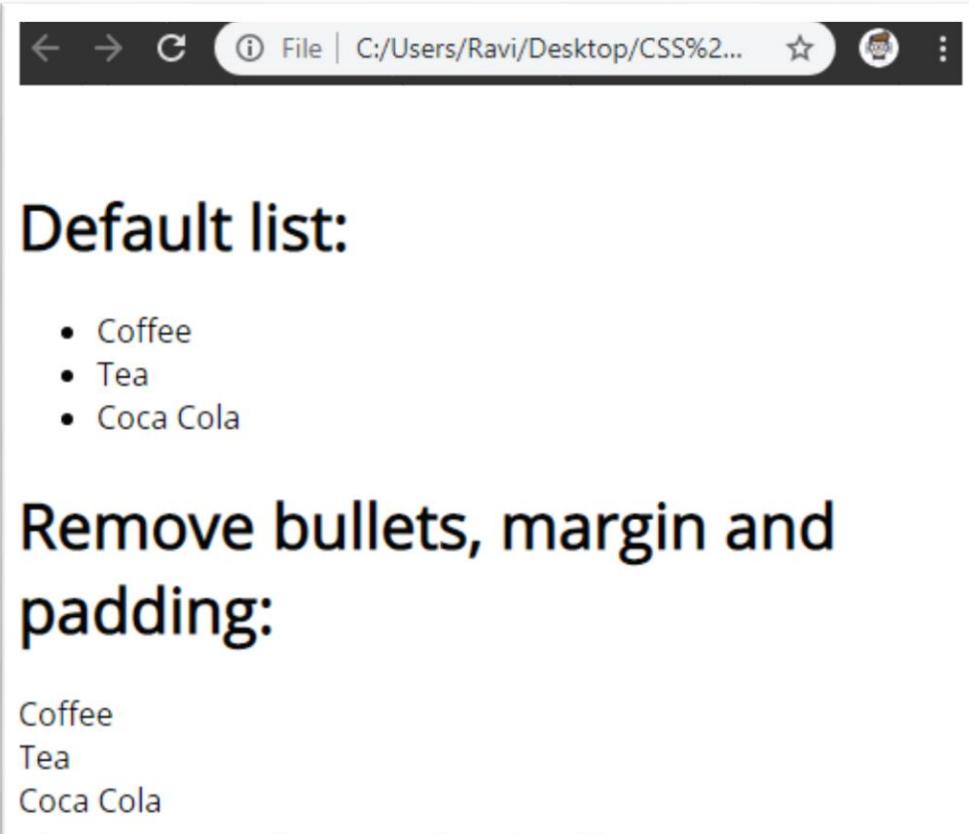
```
<!DOCTYPE html>

<html>
  <head>
    <style>
      ul.css {
        list-style-type: none;
        margin: 0;
        padding: 0;
      }

    </style>
  </head>
  <body>
    <h1>Default list:</h1>
    <ul>
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ul>
    <h1>Remove bullets, margin and padding:</h1>
    <ul class="css">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ul>
  </body>
</html>
```

**Result:**

**Picture:**



The screenshot shows a web browser window with a dark theme. The address bar displays the file path: 'File | C:/Users/Ravi/Desktop/CSS%2...'. The main content area contains the heading 'Default list:' followed by an unordered list with three items: '• Coffee', '• Tea', and '• Coca Cola'. Below this, there is another heading 'Remove bullets, margin and padding:' followed by the same three items, but now they are displayed as plain text without bullet points, margin, or padding.

**Default list:**

- Coffee
- Tea
- Coca Cola

**Remove bullets, margin and padding:**

Coffee  
Tea  
Coca Cola

**Question:**

To remove default styling from lists we would use?

- 1) list-style
- 2) list-style-type (correct)
- 3) list-type

**TOPIC FIVE ENDS HERE**

**5) Styling Lists:**

The list can be formatted in CSS. Different colors, borders, background, and paddings can be set for the lists.

## Example

### Code:

```
<!DOCTYPE>

<html>

<head>

<style>

ul.a

{

list-style:square;

background:hotpink;

padding:20px;

}

</style>

</head>

<body>

<h2>

CSS CODER

</h2>

<p>

Unordered lists

</p>

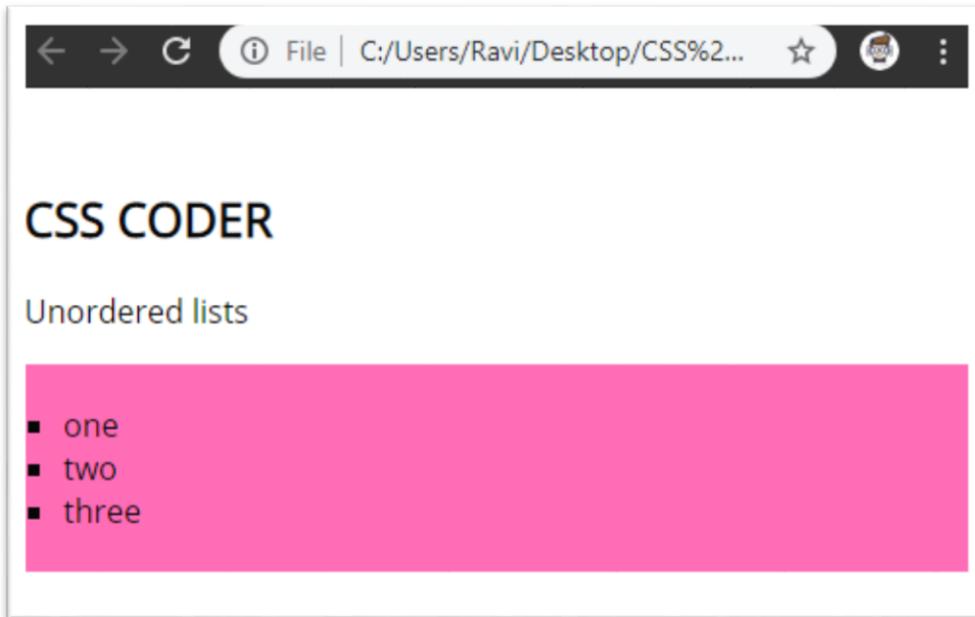
<ul class="a">

<li>one</li>
```

```
<li>two</li>
<li>three</li>
</ul>
</body>
</html>
```

## Result:

### Picture:



### Note:

Anything added to the `<ol>` or `<ul>` tag, affects the entire list, while properties added to the `<li>` tag will affect the individual list items.

**TOPIC SIX ENDS HERE**

**END OF MODULE ELEVEN**

# MODULE TWELVE

## CSS TABLES

CSS is used to style the [HTML tables](#).

### 1) Table Borders:

It is used for specifying borders in the table.

#### Example Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
table, th, td {
    border: 3px solid hotpink;
}
</style>
</head>
<body>
<h2>Add a border to a table:</h2>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
```

```
</tr>

<tr>
    <td>Css</td>
    <td>Coder</td>
</tr>

<tr>
    <td>Css</td>
    <td>Coder</td>
</tr>

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

## Result:

## Picture:



**Note:**

Notice that the table in the example above has double borders. This is because both the table and the <th> and <td> elements have separate borders.

## TOPIC ONE ENDS HERE

### 2) Border Collapse:

The border-collapse property tells us whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.

#### Example

##### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
body {
    text-align: left;
}

h1 {
    color: green;
}

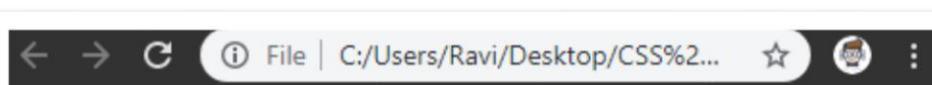
table.one {
    /* Styling border collapse for table one. */
    border-collapse: collapse;
}
```

```
table.two {  
    /* Styling border separate for table two. */  
    border-collapse: separate;  
}  
  
table,  
td,  
th {  
    border: 1.5px solid blue;  
}  
  
</style>  
  
</head>  
  
<body>  
    <h1>CSS CODER</h1>  
    <h2>borders collapsed:</h2>  
    <table class="one">  
        <tr>  
            <th>Roll Number</th>  
            <th>Name</th>  
        </tr>  
        <tr>  
            <td>1</td>  
            <td>A_B_C</td>  
        </tr>  
        <tr>  
            <td>2</td>  
            <td>X_Y_Z</td>  
        </tr>  
    </table>
```

```
</tr>
</table>
<br>
<h2>borders separated:</h2>
<table class="two">
<tr>
<th>Roll Number</th>
<th>Name</th>
</tr>
<tr>
<td>1</td>
<td>A_B_C</td>
</tr>
<tr>
<td>2</td>
<td>X_Y_Z</td>
</tr>
</table>
</body>
</html>
```

**Result:**

**Picture:**



## CSS OCDER

borders collapsed:

Roll Number	Name
1	A_B_C
2	X_Y_Z

borders separated:

Roll Number	Name
1	A_B_C
2	X_Y_Z

### Question:

Border-collapse property sets whether the body should have?

- 1) single border
- 2) separate border
- 3) collapse border (correct)
- 4) double border

### Note:

If you only want a border around the table, only specify the `border` property for `<table>`.

## TOPIC TWO ENDS HERE

### 3) Border Spacing:

This property specifies the space between the borders of the adjacent cells.

#### Example Code:

```
<!DOCTYPE html>

<html>
<head>
<style>

body {
    text-align: left;
}

h1 {
    color: green;
}

table.one {
    border-collapse: separate;

    /* Styling the border-spacing
    between adjacent cells. */
    border-spacing: 10px;
}

table.two {
```

```
border-collapse: separate;  
  
/* Styling the border-spacing  
between adjacent cells. */  
border-spacing: 10px 30px;  
}  
  
table,  
td,  
th {  
    border: 1.5px solid blue;  
}  
  
</style>  
  
</head>  
  
<body>  
    <h1>CSS CODER</h1>  
    <h2>border spacing:</h2>  
    <table class="one">  
        <tr>  
            <th>Roll Number</th>  
            <th>Name</th>  
        </tr>  
        <tr>  
            <td>1</td>  
            <td>A_B_C</td>  
        </tr>  
        <tr>
```

```
<td>2</td>
<td>X_Y_Z</td>
</tr>
</table>
<br>
<br>
<h2>border spacing:</h2>
<table class="two">
<tr>
<th>Roll Number</th>
<th>Name</th>
</tr>
<tr>
<td>1</td>
<td>A_B_C</td>
</tr>
<tr>
<td>2</td>
<td>X_Y_Z</td>
</tr>
</table>
</body>
</html>
```

**Result:**

## Picture:

The screenshot shows a web browser window with a title bar indicating the file path: C:/Users/Ravi/Desktop/CSS%2... . Below the title bar, there are standard browser controls for back, forward, search, and refresh.

The main content area displays the text "CSS CODER" in green. Below it, the text "border spacing:" is shown twice, each followed by a table example.

The first "border spacing:" example shows a table with a single row and two columns. The first column contains the text "Roll Number" and the second column contains "Name". Below these are two rows of data: "1" and "A\_B\_C" in the first column, and "2" and "X\_Y\_Z" in the second column. All elements are enclosed in a single blue border.

The second "border spacing:" example shows a similar table structure, but with horizontal and vertical spacing between the table cells. The first column has "1" and "A\_B\_C", and the second column has "2" and "X\_Y\_Z". The entire table is enclosed in a blue border.

**TOPIC THREE ENDS HERE**

## 4) Table Width And Height:

Width and height of a table are defined by the `width` and `height` properties.

**Example**

**Code:**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
table, td, th {  
    border: 3px solid hotpink;  
}  
  
table {  
    border-collapse: collapse;  
    width: 100%;  
}  
  
th {  
    height: 70px;  
}  
  
</style>  
  
</head>  
  
<body>  


## The width and height Properties



Set the width of the table, and the height of the table header row:



| Firstname | Lastname | Savings |
|-----------|----------|---------|
| Peter     | Griffin  | \$100   |


```

```
<tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$130</td>
</tr>

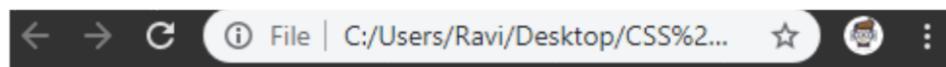
<tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$200</td>
</tr>

<tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
</tr>

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

**Result:**

**Picture:**



## The width and height Properties

Set the width of the table, and the height of the table header row:

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$130
Joe	Swanson	\$200
Cleveland	Brown	\$250

## TOPIC FOUR ENDS HERE

### 5) Caption Side:

Caption side property is used for controlling the placement of caption in the table.

#### Example

#### Code:

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

```
text-align: left;  
}  
  
h1 {  
    color: green;  
}  
  
table.one {  
    border-collapse: separate;  
    border-spacing: 10px;  
  
/* Controlling the placement of caption. */  
    caption-side: top;  
}  
  
table.two {  
    border-collapse: separate;  
    border-spacing: 10px;  
  
/* Controlling the placement of caption. */  
    caption-side: bottom;  
}  
  
table,  
td,  
th {  
    border: 1.5px solid blue;  
}  
  
</style>  
</head>
```

```
<body>

<h1>CSS CODER</h1>

<h2>Caption on top:</h2>

<table class="one">

    <caption>Caption at the top of the table.</caption>

    <tr>

        <th>Roll Number</th>

        <th>Name</th>

    </tr>

    <tr>

        <td>1</td>

        <td>A_B_C</td>

    </tr>

    <tr>

        <td>2</td>

        <td>X_Y_Z</td>

    </tr>

</table>

<br>

<br>

<h2>Caption at bottom:</h2>

<table class="two">

    <caption>

        Caption at the bottom of the table

    </caption>

    <tr>
```

```
<th>Roll Number</th>
<th>Name</th>
</tr>
<tr>
<td>1</td>
<td>A_B_C</td>
</tr>
<tr>
<td>2</td>
<td>X_Y_Z</td>
</tr>
</table>
</body>
</html>
```

**Result:**

**Picture:**

The screenshot shows a web browser window with the title "CSS CODER". Below the title, there are two sections demonstrating the use of the `caption-side` property.

**Caption on top:**

Caption at the top of the table.

Roll Number	Name
1	A_B_C
2	X_Y_Z

**Caption at bottom:**

Caption at the bottom of the table

Roll Number	Name
1	A_B_C
2	X_Y_Z

## Question:

Caption side property is used for?

- 1) controlling position of border
- 2) controlling color of table
- 3) controlling spacing of content
- 4) controlling placement of caption in table (correct)

## Note:

By default, captions are placed above the table.

## TOPIC FIVE ENDS HERE

### 6) Table Padding:

To control the space between the border and the content in a table, use the [padding](#) property on <td> and <th> elements.

#### Example

##### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
table, td, th {
    border: 1px solid #ddd;
    text-align: left;
}
table {
    border-collapse: collapse;
    width: 100%;
}
th, td {
    padding: 15px;
}
</style>
</head>
<body>
<h2>The padding Property</h2>
```

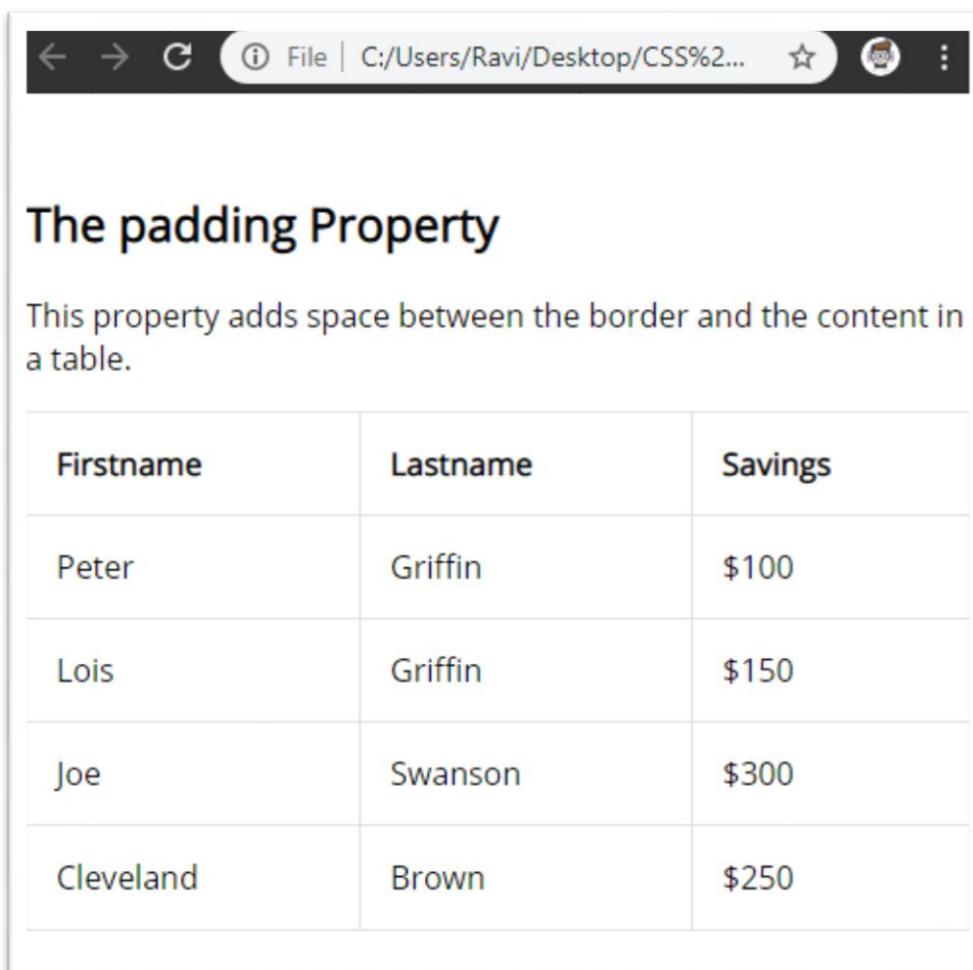
```
<p>This property adds space between the border and the content in a table.</p>
```

```
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
  </tr>
```

```
</tr>  
</table>  
</body>  
</html>
```

**Result:**

**Picture:**



The screenshot shows a web browser window with a dark header bar containing navigation icons, a file path 'C:/Users/Ravi/Desktop/CSS%2...', and other browser controls. Below the header is a large, bold title 'The padding Property'. A descriptive text follows: 'This property adds space between the border and the content in a table.' Below this is a table with five rows and three columns. The columns are labeled 'Firstname', 'Lastname', and 'Savings'. The data in the table is as follows:

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

**TOPIC SIX ENDS HERE**

## 7) Table Layout:

The table layout property is used to set up the layout algorithm used for the table.

### Example

#### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
body {
    text-align: left;
}

h1 {
    color: green;
}

table.one {
    width: 80px;
    border-collapse: separate;
    border-spacing: 10px;
}

/* Layout of table is auto. */
```

```
table-layout: auto;  
}  
  
  
table.two {  
width: 80px border-collapse: separate;  
border-spacing: 10px;  
  
/* Layout of table is fixed. */  
table-layout: fixed;  
}  
  
  
table,  
td,  
th {  
border: 1.5px solid green;  
width: 80px;  
}  
  
</style>  
</head>  
<body>  
<h1>CSS CODER</h1>  
<h2>auto table layout:</h2>  
<table class="one">  
<tr>  
<th>Roll Number</th>  
<th>Student Name</th>
```

```
</tr>
<tr>
    <td>1</td>
    <td>A_B_C_D_E_F_G_H_I_J_K_L_M_N_O_P</td>
```

```
</tr>
<tr>
    <td>2</td>
    <td>X_Y_Z</td>
</tr>
</table>
```

```
<br>
<h2>fixed table layout:</h2>
<table class="two">
    <tr>
        <th>Roll Number</th>
        <th>Student Name</th>
    </tr>
    <tr>
        <td>1</td>
        <td>A_B_C_D_E_F_G_H_I_J_K_L_M_N_O_P</td>
    </tr>
    <tr>
        <td>2</td>
        <td>X_Y_Z</td>
```

```
</tr>  
</table>  
</body>  
</html>
```

**Result:**

**Picture:**

The screenshot shows a web browser window with the address bar containing "E:/CSS%20CRASH%20CO...". The main content area displays two tables side-by-side.

**Table 1 (Left):** This is a "fixed table layout". It has a green border and contains four rows. The first row has two columns: "Roll Number" and "Student Name". The second row has two columns: "1" and "A\_B\_C\_D\_E\_F\_G\_H\_I\_J\_K\_L\_M\_N\_O\_P". The third row has two columns: "2" and "X\_Y\_Z".

Roll Number	Student Name
1	A_B_C_D_E_F_G_H_I_J_K_L_M_N_O_P
2	X_Y_Z

**Table 2 (Right):** This is an "auto table layout". It also has a green border and contains four rows. The first row has two columns: "Roll Number" and "Student Name". The second row has two columns: "1" and "A\_B\_C\_D\_E\_F\_G\_H\_I\_J\_K\_L\_M\_N\_O\_P". The third row has two columns: "2" and "X\_Y\_Z".

Roll Number	Student Name
1	A_B_C_D_E_F_G_H_I_J_K_L_M_N_O_P
2	X_Y_Z

### **Question:**

In fixed table layout the browsers ignores?

- 1) the content and uses table's width (correct)
- 2) padding and margin
- 3) border layout
- 4) content box

### **Note:**

**Supported Browsers:** The browsers supported by *Tables*

- Google Chrome
- Edge
- Mozilla Firefox
- Opera And Safari

**TOPIC SEVEN ENDS HERE**

**END OF MODULE TWELVE**

# MODULE THIRTEEN

## CSS Display

The [display](#) property is the most important CSS property for controlling layout.

The [display](#) property specifies if/how an element is displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is [block](#) or [inline](#).

### 1) Block Level Elements:

This property is used as the default property of div. This property places the div one after another vertically. Height and width of the div can be changed using the block property if the width is not mentioned, then div under block property will take up the width of the container.

Example:

#### Code:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS | Display property</title>
<style>
#coder1{
    height: 100px;
    width: 200px;
    background: hotpink;
```

```
display: block;  
}  
  
#coder2{  
height: 100px;  
width: 200px;  
background: skyblue;  
display: block;  
}  
  
#coder3{  
height: 100px;  
width: 200px;  
background: lightgreen;  
display: block;  
}  
  
.css {  
margin-left:20px;  
font-size:42px;  
font-weight:bold;  
color:#009900;  
}  
  
.coder {  
font-size:25px;  
margin-left:30px;  
}  
  
.main {  
margin:50px;
```

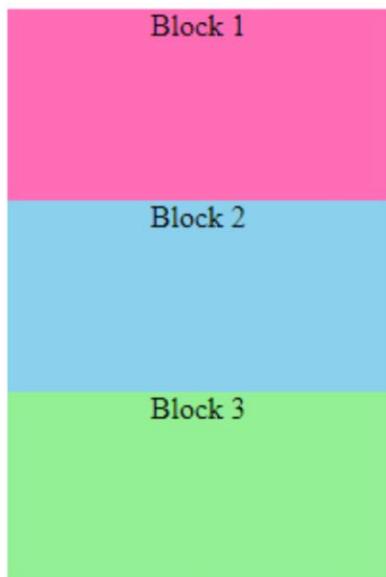
```
text-align:center;  
}  
</style>  
</head>  
  
<body>  
  <div class = "css">CSS OCDER</div>  
  <div class = "coder">display: block; property</div>  
  <div class = "main">  
    <div id="coder1">Block 1 </div>  
    <div id="coder2">Block 2</div>  
    <div id="coder3">Block 3</div>  
  </div>  
</body>  
</html>
```

**Result:**

**Picture:**

# CSS ORDER

display: block; property



## Question:

Block level element starts on?

- 1) previous line
- 2) existing line
- 3) new line (correct)

## Note:

Block level element takes up the full width available (stretches out to the left and right as far as it can).

**TOPIC ONE ENDS HERE**

## 2) Inline Elements:

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

Examples of inline elements:

- <span>
- <a>
- <img>

Example:

Code:

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      li {
        display: inline;
        padding: 10px;
      }
      a{
        text-decoration: none;
        color: red;
      }
    </style>
  </head>
  <body>
```

```
<p>Display a list of links as a horizontal menu:</p>

<ul>

<li><a href="#" target="_blank">HTML</a></li>

<li><a href="#" target="_blank">CSS</a></li>

<li><a href="#" target="_blank">JavaScript</a></li>

</ul>

</body>

</html>
```

## Result:

### Picture:



## Question:

Inline is used for?

- 1) placing elements in vertical manner
- 2) placing the elements in box manner
- 3) placing the elements in horizontal manner (correct)
- 4) placing the elements in content manner

## Note:

The inline display property ignores the height and the width set by the user.

## TOPIC TWO ENDS HERE

### 3) Inline-block Elements:

This features uses the both properties mentioned above, block and inline. So, this property aligns the div inline but the difference is it can edit the height and the width of block. Basically, this will align the div both in block and inline fashion.

Example:

#### Code:

```
<!DOCTYPE html>

<html>
  <head>
    <title>CSS | Display property</title>
    <style>
      #main{
        height: 100px;
        width: 200px;
        background: hotpink;
        display: inline-block;
      }
      #main1{
        height: 100px;
        width: 200px;
        background: skyblue;
        display: inline-block;
      }
    </style>
  </head>
  <body>
    <div id="main"></div>
    <div id="main1"></div>
  </body>
</html>
```

```
}

#main2{
    height: 100px;
    width: 200px;
    background: lightgreen;
    display: inline-block;
}

.css-coder {
    margin-left:200px;
    font-size:42px;
    font-weight:bold;
    color:#009900;
}

.coder {
    font-size:25px;
    margin-left:210px;
}

.main {
    margin:50px;
}

</style>

</head>

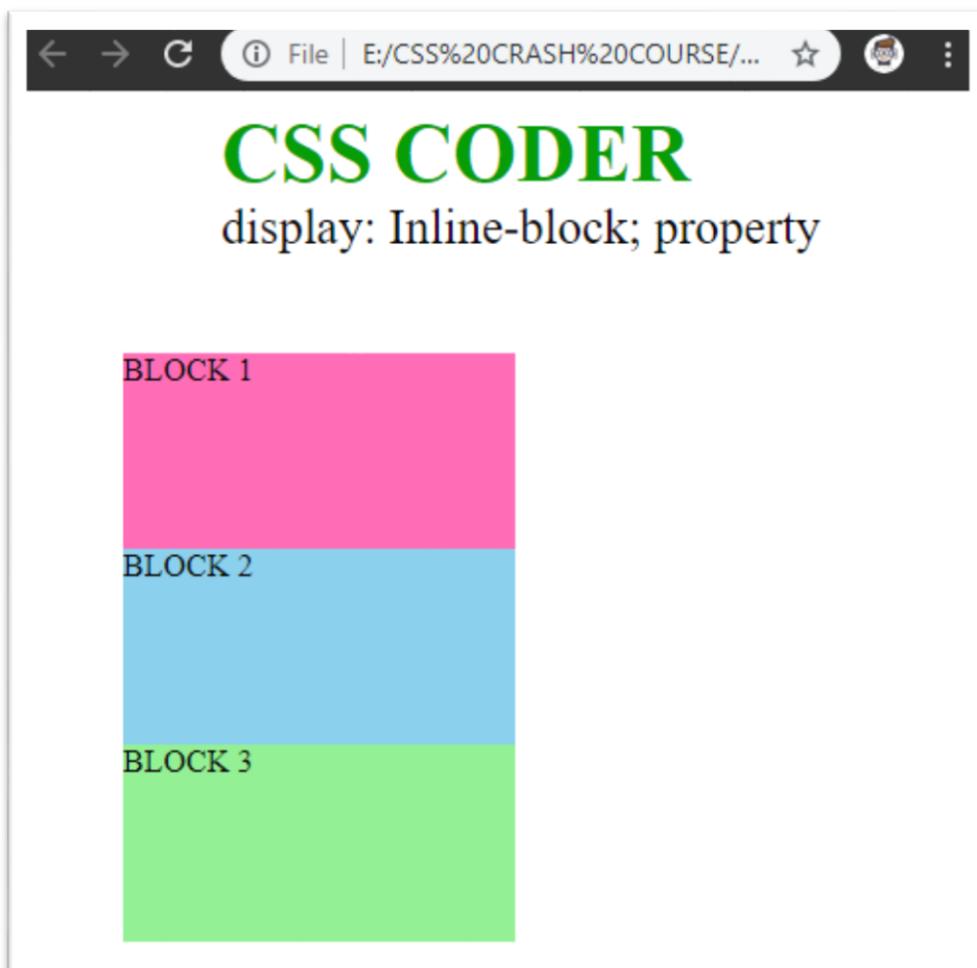
<body>

<div class = "css-coder">CSS CODER</div>
<div class = "coder">display: inline-block; property</div>
```

```
<div class = "main">  
    <div id="main"> BLOCK 1 </div>  
    <div id="main1"> BLOCK 2</div>  
    <div id="main2">BLOCK 3 </div>  
</div>  
</body>  
</html>
```

## Result:

## Picture:



## Question:

In inline-block we can edit what which we can't do in inline?

- 1) font-size
- 2) padding and margin
- 3) width and height (correct)
- 4) border

## TOPIC THREE ENDS HERE

### 4) Display: none:

This property hides the div or the container which use this property. Using it on one of the div below it will make working clear.

Example:

#### Code:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS | Display property</title>
<style>
#main{
    height: 100px;
    width: 200px;
    background: hotpink;
    display: none;
}

```

```
#main1{  
    height: 100px;  
    width: 200px;  
    background: lightblue;  
    display: block;  
  
}  
  
#main2{  
    height: 100px;  
    width: 200px;  
    background: lightgreen;  
    display: block;  
  
}  
  
.css-coder {  
    margin-left:20px;  
    font-size:42px;  
    font-weight:bold;  
    color:#009900;  
  
}  
  
.coder {  
    font-size:25px;  
    margin-left:20px;  
  
}  
  
.main {  
    margin:50px;  
}
```

```
</style>

</head>

<body>

<div class = "css-coder">CSS CODER</div>

<div class = "coder">display: none; property, We will hide the first div see
below the first one is not showing!</div>

<div class = "main">
    <div id="main"> BLOCK 1 </div>
    <div id="main1"> BLOCK 2</div>
    <div id="main2">BLOCK 3 </div>
</div>

</body>

</html>
```

**Result:**

**Picture:**

**CSS CODER**

display: none; property, We will hide the first div see below the first one is not showing!

BLOCK 2

BLOCK 3

### Question:

What's the difference between display: none and visibility: hidden?

- 1) both removes the element from the document
- 2) display none removes element and visibility none hide it (correct)
- 3) display none hides the element and visibility removes it

### Note:

**Supported Browsers:** The browsers supported by **Display property** are listed below:

- Google Chrome 4.0
- Internet Explorer 8.0
- Firefox 3.0
- Opera 7.0
- Safari 3.1

## TOPIC FOUR ENDS HERE

## END OF MODULE THIRTEEN

## MODULE FOURTEEN

### **CSS Positioning Elements**

The *position* property in CSS tells about the method of positioning for an element or an HTML entity. There are five different types of position property available in CSS:

1. Fixed
2. Static
3. Relative
4. Absolute
5. Sticky

The positioning of an element can be done using the *top*, *right*, *bottom* and *left* property. These specify the distance of an HTML element from the edge of the viewport. To set the position by these four properties, we have to declare the positioning method.

Let's talk about each of these position methods in details.

#### **1) Position: fixed:**

An element with **position: fixed;** is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.

Example:

#### **Code:**

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

```
<style>
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    font-size: 30px;
    border: 5px solid green;
}

</style>
</head>
<body>
<h2>position: fixed;</h2>
<p>An element with position: fixed; is positioned relative to the viewport,  
which means it always stays in the same place even if the page is scrolled:</p>
<div class="fixed">
This div element has position: fixed;
</div>
</body>
</html>
```

**Result:**

**Picture:**

A screenshot of a web browser window. The address bar shows 'File | E:/CSS%20CRASH%20COURSE/...'. The main content area contains the text 'position: fixed;'. Below this, a green-bordered box contains the text 'This div element has position: fixed;'. The browser interface includes standard navigation buttons (back, forward, search) and a toolbar with icons for file operations.

**position: fixed;**

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:

This div element has  
position: fixed;

### Question:

Position: fixed; element always stays?

- 1) moving
- 2) to one position (correct)
- 3) to border
- 4) to viewport

## TOPIC ONE ENDS HERE

### 2) Position: static:

This method of positioning is set by default. If we don't mention the method of positioning for any element, the element has the **position:static** method by default. By defining Static, the top, right, bottom and left will not have any control over the element. The element will be positioned with the normal flow of the page.

Example:

**Code:**

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      div.static {
        position: static;
        border: 5px solid red;
      }
    </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>
```

**Result:**

**Picture:**

The screenshot shows a browser window with the following content:

**position: static;**

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

This div element has position: static;

### Question:

HTML elements are positioned \_\_\_\_\_ by default?

- 1) fixed
- 2) relative
- 3) static (correct)
- 4) absolute

## TOPIC TWO ENDS HERE

### 3) Position: relative:

An element with **position: relative;** is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

Example:

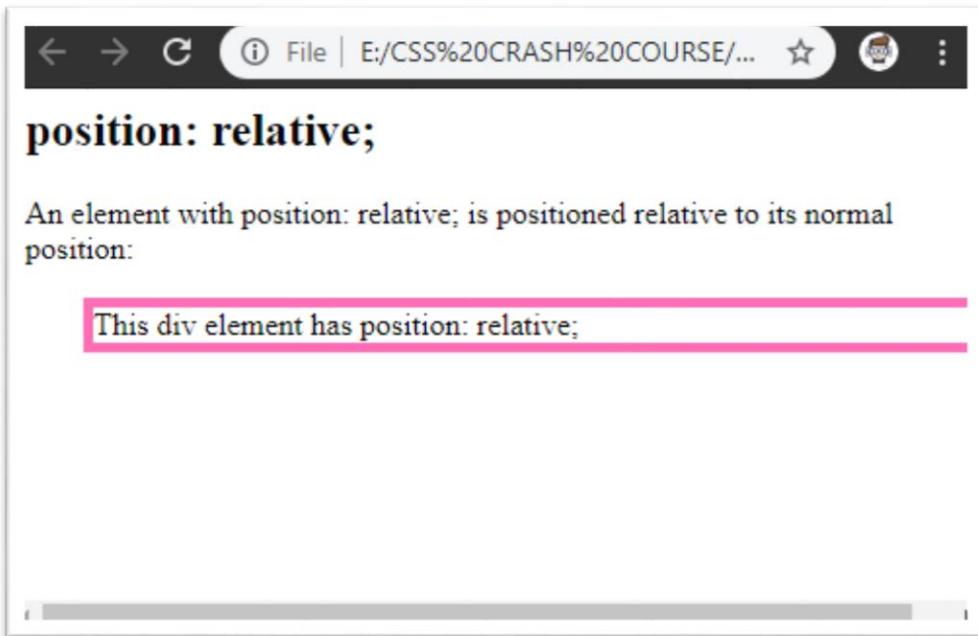
#### Code:

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
div.relative {
    position: relative;
    left: 30px;
    border: 5px solid hotpink;
}
</style>
</head>
<body>
<h2>position: relative;</h2>
<p>An element with position: relative; is positioned relative to its normal
position:</p>
<div class="relative">
This div element has position: relative;
</div>
</body>
</html>
```

## Result:

## Picture:



### Question:

Which position property is used to change normal position of an element?

- 1) static
- 2) absolute
- 3) fixed
- 4) relative (correct)

**TOPIC THREE ENDS HERE**

### 4) Position: absolute:

An element with **position: absolute** will be positioned with respect to its parent. Positioning of this element does not depend upon its siblings or the elements which are at same level.

Example:

Code:

```
<!DOCTYPE html>

<html>
<head>
<style>

div.relative {
    position: relative;
    width: 500px;
    height: 300px;
    border: 5px solid hotpink;
}

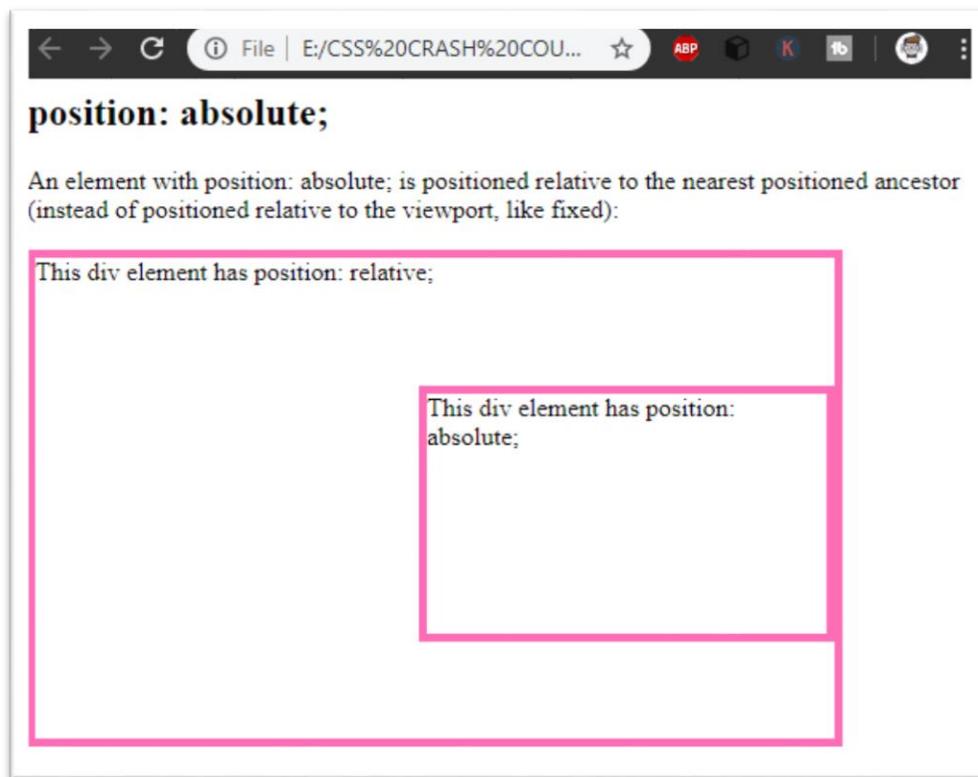
div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 250px;
    height: 150px;
    border: 5px solid hotpink;
}

</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>
```

## Result:

### Picture:



## Question:

A positioned element is one whose position is anything except?

- 1) fixed
- 2) absolute
- 3) relative
- 4) static (correct)

**Note:**

If an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

## TOPIC FOUR ENDS HERE

### 5) Position: sticky:

Element with `position: sticky` and `top: 0` played a role between `fixed` & `relative` based on the position where it is placed. If the element is placed at the middle of the document then when user scrolls the document, the sticky element starts scrolling until it touches the top. When it touches the top, it will be fixed at that place inspite of further scrolling.

Example:

**Code:**

```
<!DOCTYPE html>

<html>
<head>
<style>
div.sticky {
    position: sticky;
    top: 0;
```

```
padding: 5px;  
background-color: #cae8ca;  
border: 4px solid hotpink;  
}  
</style>  
</head>  
<body>  
<p>Try to <b>scroll</b> inside this frame to understand how sticky positioning works.</p>  
<div class="sticky">I am sticky!</div>  
<div style="padding-bottom:2000px">  
    <h2>In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.</h2>  
    <p>Scroll back up to remove the stickyness.</p>  
    <p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum  
    definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo.  
    Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus  
    repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae  
    voluptatibus.</p>  
</div>  
</body>  
</html>
```

## Result:

## Picture:

The screenshot shows a web browser window with a title bar containing file navigation icons and the URL 'File | E:/CSS%20CRASH%20COURSE/...'. Below the title bar is a message: 'Try to scroll inside this frame to understand how sticky positioning works.' A green rectangular box with a pink border contains the text 'I am sticky!'. Below this box is a large bold text block: 'In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.' At the bottom of the frame, there is some placeholder Latin text: 'Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudianda nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.'

### Note:

We can stick the element at bottom, with the **bottom** property.

IE/Edge 15 and earlier versions do not support sticky position.

**TOPIC FIVE ENDS HERE**

**END OF MODULE FOURTEEN**

**MODULE FIFTEEN**

## CSS Float

Float is a CSS property written in CSS file or directly in the style of an element. The float property defines the flow of content. Below are the types of floating properties.

float: left	Element floats on left side of the container
float: right	Element floats on right side of container
float: inherit	Element inherits floating property of it's parent (div, table etc...)
float: none	Element is displayed as it is (Default).

### 1) float left:

Element floats on left side of the container.

Example:

#### Code:

```
<!DOCTYPE html>

<html>
<head>
<title>Float</title>
</head>
<body>
<div class = "css-coder"
style = "font-size: 50px;
color: hotpink;
float: left;">
CSS CODER
</div>
</body>
```

```
</html>
```

## Result:

## Picture:



## Question:

How Float: left property works?

- 1) move the content to center
- 2) move the content to right
- 3) move the content to left (correct)**
- 4) move it as inherit

## Note:

Using float the element is removed from the normal flow of the page, though still remaining a part of the flow (in contrast to absolute positioning).

**TOPIC ONE ENDS HERE**

## 2) float right:

Element floats on right side of container.

Example:

### Code:

```
<!DOCTYPE html>

<html>
<head>
<title>Float</title>
</head>
<body>
<div class = "css-coder"
      style = "font-size: 40px;
                color: hotpink;
                float: right;">
    CSS CODER
</div>
</body>
</html>
```

**Result:**

**Picture:**



**Question:**

How float: right property works?

- 1) move content to left side
- 2) move content to right side (correct)
- 3) move content to center
- 4) move content to bottom

**Note:**

A *floating element* is one where the computed value of float is not none.

**TOPIC TWO ENDS HERE**

### 3) float none:

Element is displayed as it is (Default).

Example:

#### Code:

```
<!DOCTYPE html>

<html>
<head>
<title>Float</title>
</head>
<body>
<div
style = "font-size: 50px;
color: hotpink;
float: none;">
CSS CODER
</div>
</body>
</html>
```

#### Result:

#### Picture:



### Question:

Which of the following is default value for float?

- 1) float: none (correct)
- 2) float: left
- 3) float: inherit
- 4) float: right

## TOPIC THREE ENDS HERE

### 4) float inherit:

Element inherits floating property of it's parent (div, table etc...)

Example:

### Code:

```
<!DOCTYPE html>

<html>
<head>
    <title>Float</title>
</head>
<body>
```

```
<div style="float:right">  
    <div  
        style = "font-size: 50px;  
            color: hotpink;  
            float: inherit;">  
            CSS CODER  
        </div>  
    </div>  
    </body>  
</html>
```

## Result:

### Picture:



## Question:

Float: inherit applies the styling from?

- 1) sibling
- 2) present element
- 3) general sibling
- 4) parent (correct)

## TOPIC FOUR ENDS HERE

### 5) Navigation Bar using float:

Use **float** with a list of hyperlinks to create a horizontal menu.

Example:

Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: hotpink;
}

li {
    float: left;
}

li a {
    display: inline-block;
```

```
color: #000;  
text-align: center;  
padding: 14px 16px;  
text-decoration: none;  
font-size: 30px;  
}
```

```
.active {  
background-color: lightgreen;  
}  
</style>  
</head>  
<body>
```

```
<ul>  
<li><a href="#home" class="active">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>
```

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

**Result:**

**Picture:**



**Note:**

It is also common to do entire web layouts using the `float` property.

**TOPIC FIVE ENDS HERE**

**END OF MODULE FIFTEEN**

**MODULE SIXTEEN**

## CSS Align

In [CSS](#) we align items [horizontally](#) and [vertically](#). Various methods and techniques are used to center them, take care of the left and the right margin, etc.

### 1) Center Align Elements :

To horizontally center a block element (like <div>), use [margin: auto;](#)

Setting the width of the element will prevent it from stretching out to the edges of its container.

The element will then take up the specified width, and the remaining space will be split equally between the two margins:

Example:

**Code:**

```
<!DOCTYPE html>

<html>
<head>
<style>
.center {
    margin: auto;
    width: 50%;
    border: 5px solid hotpink;
    padding: 12px;
}
</style>
</head>
<body>
<h1>Center Align Elements</h1>
```

```
<p>To horizontally center a block element (like div), use margin: auto;</p>

<div class="center">

<p><b>Note:</b> Using margin:auto will not work in IE8, unless a !DOCTYPE is declared.</p>

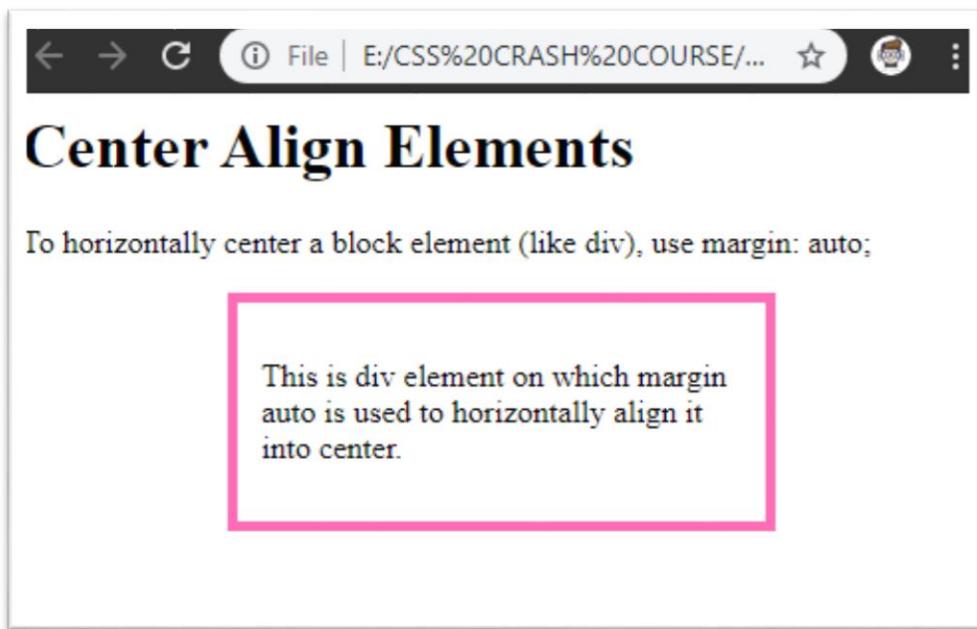
</div>

</body>

</html>
```

## Result:

### Picture:



## Question:

Margin: auto; is used for?

- 1) align a block element to left
- 2) align a block element to right
- 3) align a block element to center (correct)
- 4) align a block element to bottom

**Note:**

Using margin:auto will not work in IE8, unless a !DOCTYPE is declared.

## TOPIC ONE ENDS HERE

### 2) Center Align Text:

To just center the text inside an element, use `text-align: center;`

Example:

**Code:**

```
<!DOCTYPE html>

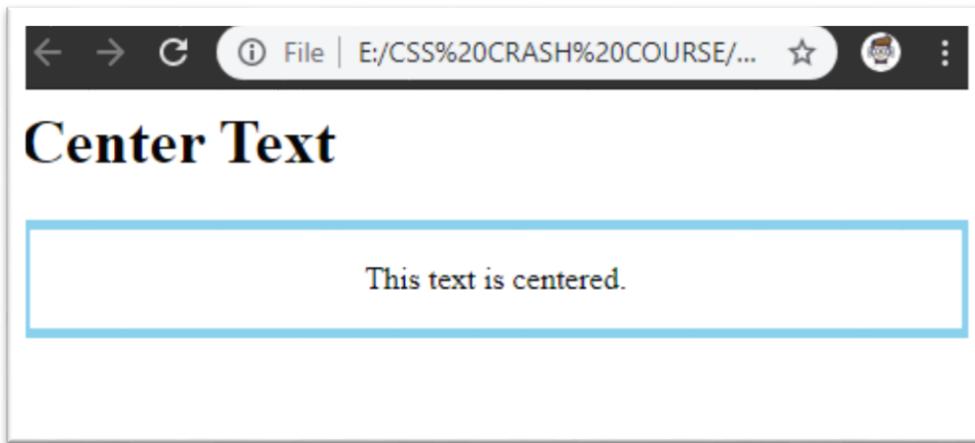
<html>
<head>
<style>
.center {
    text-align: center;
    border: 5px solid skyblue;
}
</style>
</head>
<body>
<h1>Center Text</h1>
<div class="center">
    <p>This text is centered.</p>
</div>
```

```
</body>
```

```
</html>
```

## Result:

### Picture:



### Question:

To center text inside an element we will use?

- 1) text-align: center; (correct)
- 2) text-align: right;
- 3) text: center;
- 4) text-align: left;

**TOPIC TWO ENDS HERE**

### 3) Center an Image:

To center an image, set left and right margin to `auto` and make it into a `block` element:

**Example:**

**Code:**

```
<!DOCTYPE html>

<html>
<head>
<style>
img {
    display: block;
    margin-left: auto;
    margin-right: auto;
}
</style>
</head>
<body>
<h2>Center an Image</h2>
<p>To center an image, set left and right margin to auto, and make it into a block element.</p>

</body>
</html>
```

**Result:**

**Picture:**



## Center an Image

To center an image, set left and right margin to auto, and make it into a block element.



### Question:

To center an image we will set as?

- 1) inline-block
- 2) inline
- 3) block-inline
- 4) block (correct)

## TOPIC THREE ENDS HERE

### 4) Left and Right Align - Using position:

We can align the items using this property.

Example:

#### Code:

```
<!DOCTYPE html>

<html>
<head>
<style>
```

```
.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    border: 5px solid hotpink;  
    padding: 12px;  
}  
</style>  
</head>  
<body>  
<h2>Right Align</h2>  
<p>An example of how to right align elements with the position property:</p>  
<div class="right">  
<p> Absolute positioned elements can overlap other elements.</p>  
</div>  
</body>  
</html>
```

## Result:

## Picture:

A screenshot of a web browser window. The address bar shows 'File | E:/CSS%20CRASH%20COURSE/...'. The main content area has a pink border and contains the text 'Absolute positioned elements can overlap other elements.' in black font.

### Note:

Absolute positioned elements are removed from the normal flow, and can overlap elements.

## TOPIC FOUR ENDS HERE

### 5) Center Vertically using Padding:

There are many ways to center an element vertically in CSS. A simple solution is to use top and bottom padding.

Example:

#### Code:

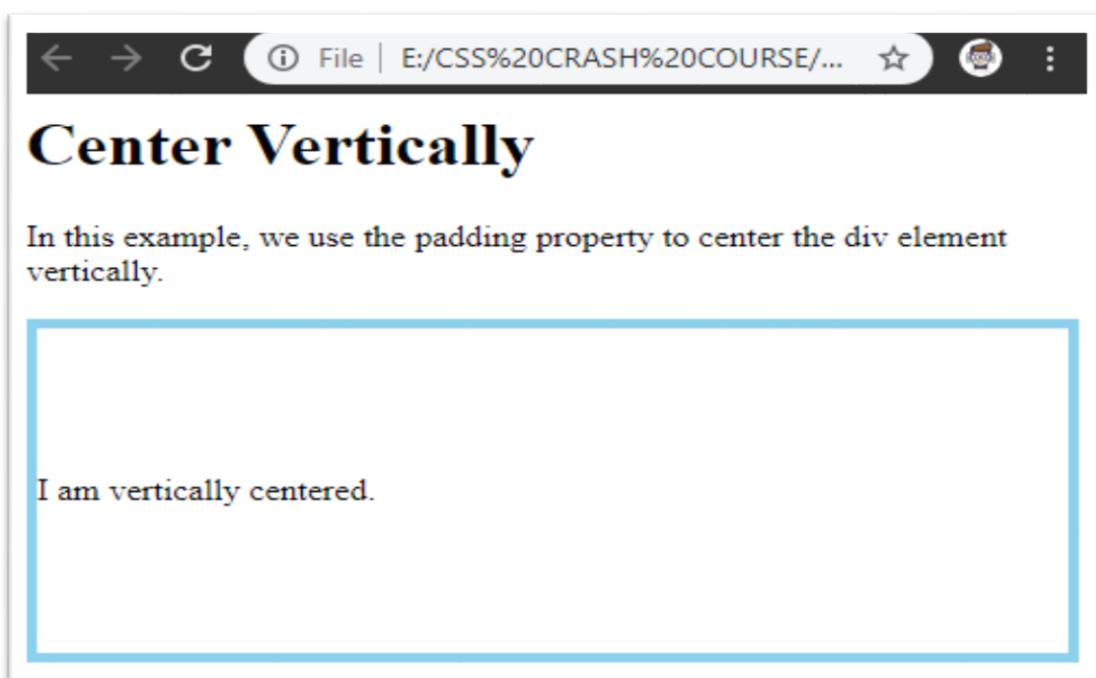
```
<!DOCTYPE html>

<html>
<head>
<style>
.center {
    padding: 60px 0;
```

```
border: 5px solid skyblue;  
}  
</style>  
</head>  
<body>  
<h1>Center Vertically</h1>  
<p>In this example, we use the padding property to center the div element vertically.</p>  
<div class="center">  
<p>I am vertically centered.</p>  
</div>  
</body>  
</html>
```

## Result:

## Picture:



### **Question:**

To center vertically using padding we will apply?

- 1) padding from left
- 2) padding from left and right
- 3) padding from top and bottom (correct)**
- 4) padding from top

### **Note:**

- 1) To center both vertically and horizontally, use `padding` and `text-align: center`.
- 2) You can also center Vertically - Using `line-height`
- 3) Also using `position absolute` and `transform` property you can vertically center element.

**TOPIC FIVE ENDS HERE**

**END OF MODULE SIXTEEN**

**MODULE SEVENTEEN**

### **CSS Combinators**

CSS combinators are explaining the relationship between two selectors. CSS selectors are the patterns used to select the elements for style purpose. A CSS selector can be a simple selector or a complex selector consisting of more than one selector connected using combinators.

There are four types of combinators available in CSS which are discussed below:

- General Sibling selector (~)
- Adjecant Sibling selector (+)
- Child selector (>)
- Descendant selector (space)

## 1) General Sibling Selector:

The general sibling selector selects all elements that are siblings of a specified element.

Example:

### Code:

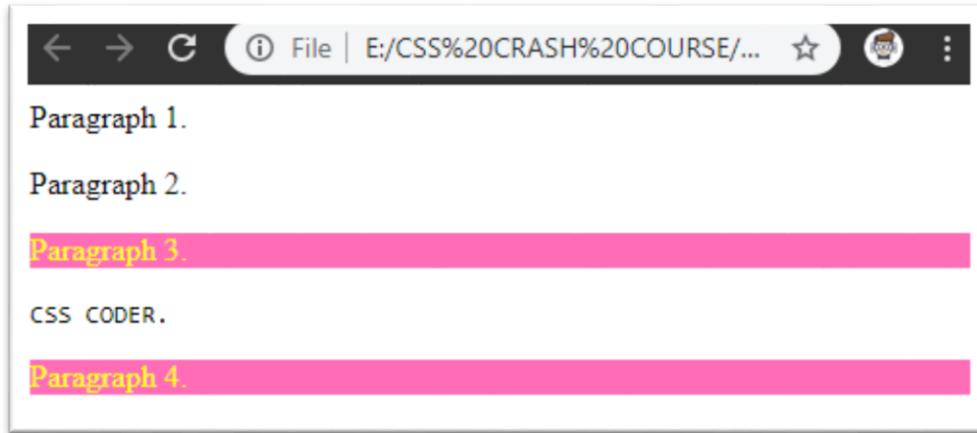
```
<!DOCTYPE html>

<html>
  <head>
    <style>
      div ~ p {
        background-color: hotpink;
        color: yellow;
      }
    </style>
  </head>
  <body>
    <p>Paragraph 1.</p>
    <div>
      <p>Paragraph 2.</p>
    </div>
```

```
<p>Paragraph 3.</p>  
<code>CSS CODER.</code>  
<p>Paragraph 4.</p>  
</body>  
</html>
```

## Result:

### Picture:



## Question:

General sibling selector selects all elements that are?

- 1) child of specified element
- 2) adjacent siblings of specified element
- 3) siblings of specified element (correct)**
- 4) descendent of specified element

## Note:

It can be used to select a group of elements that share the same parent element.

## TOPIC ONE ENDS HERE

### 2) Adjacent Sibling Selector:

The Adjacent sibling selector is used to select the element that is adjacent or the element that is the next to the specified selector tag. This combinator selects only one tag that is just next to the specified tag.

Example:

Code:

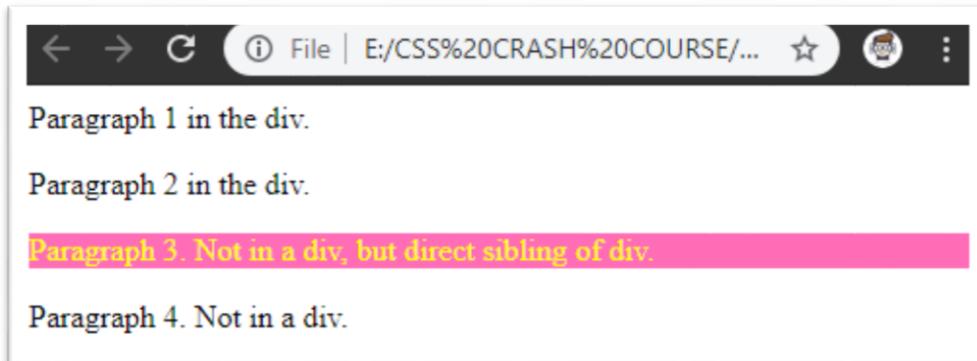
```
<!DOCTYPE html>

<html>
<head>
<style>
div + p {
    background-color: hotpink;
    color: yellow;
}
</style>
</head>
<body>
<div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
</div>
<p>Paragraph 3. Not in a div, but direct sibling of div.</p>
<p>Paragraph 4. Not in a div.</p>
</body>
```

```
</html>
```

## Result:

## Picture:



## Question:

The sign for adjacent sibling selector is?

- 1) ~
- 2) >
- 3) =
- 4) + (correct)

## Note:

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

**TOPIC TWO ENDS HERE**

### 3) Child Selector:

The child selector selects all elements that are the children of a specified element.

Example:

Code:

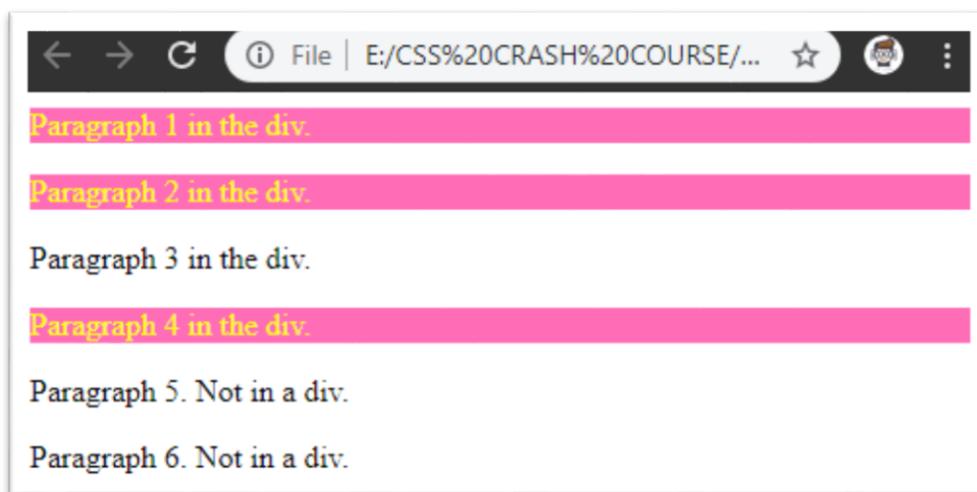
```
<!DOCTYPE html>

<html>
  <head>
    <style>
      div > p {
        background-color: hotpink;
        color: yellow;
      }
    </style>
  </head>
  <body>
    <div>
      <p>Paragraph 1 in the div.</p>
      <p>Paragraph 2 in the div.</p>
      <section><p>Paragraph 3 in the div.</p></section> <!-- not Child but
      Descendant -->
      <p>Paragraph 4 in the div.</p>
    </div>
```

```
<p>Paragraph 5. Not in a div.</p>  
<p>Paragraph 6. Not in a div.</p>  
</body>  
</html>
```

## Result:

## Picture:



## Question:

Correct sign for child selector is?

- 1) > (correct)
- 2) +
- 3) ~
- 4) -

## Note:

This combinator is stricter than the descendant selector because it selects only the second selector if it has the first selector element as its parent.

## TOPIC THREE ENDS HERE

### 4) Descendent Selector:

The descendant selector matches all elements that are descendants of a specified element.

Example:

#### Code:

```
<!DOCTYPE html>

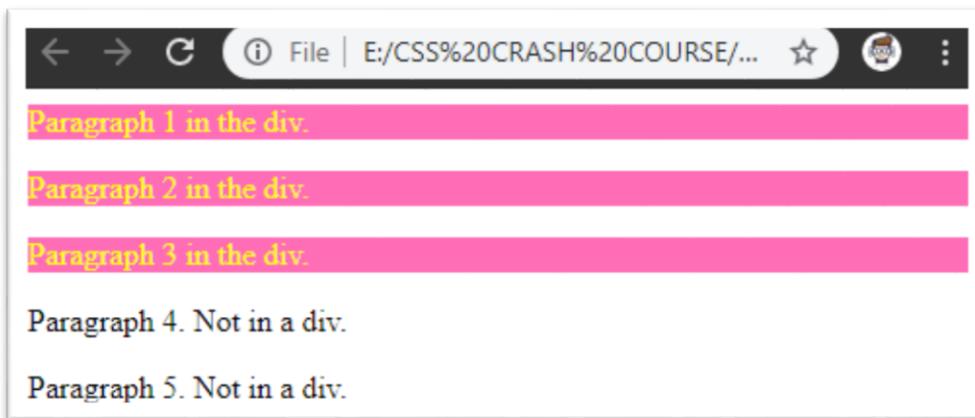
<html>
  <head>
    <style>
      div p {
        background-color: hotpink;
        color: yellow;
      }
    </style>
  </head>
  <body>
    <div>
      <p>Paragraph 1 in the div.</p>
      <p>Paragraph 2 in the div.</p>
      <section><p>Paragraph 3 in the div.</p></section>
    </div>
    <p>Paragraph 4. Not in a div.</p>
    <p>Paragraph 5. Not in a div.</p>
```

```
</body>
```

```
</html>
```

## Result:

### Picture:



### Note:

This combinator combines the two selectors such that selected elements have an ancestor same as the first selector element.

**TOPIC FOUR ENDS HERE**

**END OF MODULE SEVENTEEN**

## MODULE EIGHTEEN

### CSS Pseudo-classes

A Pseudo class in CSS is used to define the special state of an element. It can be combined with a CSS selector to add an effect to existing elements based on their states. For Example, changing the style of an element when the user hovers over it, or when a link is visited. All of these can be done using Pseudo Classes in CSS.

There are many Pseudo classes in CSS but the ones which are most commonly used are as follows:

#### 1) :hover pseudo-class:

This pseudo-class is used to add special effect to an element when our mouse pointer is over it. The below example demonstrates that when your mouse enters the box area, its background color changes from pink to green.

Example:

#### Code:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS :hover-class property</title>
```

```
<style>

.box{
background-color: pink;
width: 300px;
height: 200px;
margin: auto;
font-size: 40px;
text-align: center;

}
```

```
.box:hover{
background-color: green;
}
```

```
h1, h2{
color: green;
text-align: center;

}
```

```
</style>
```

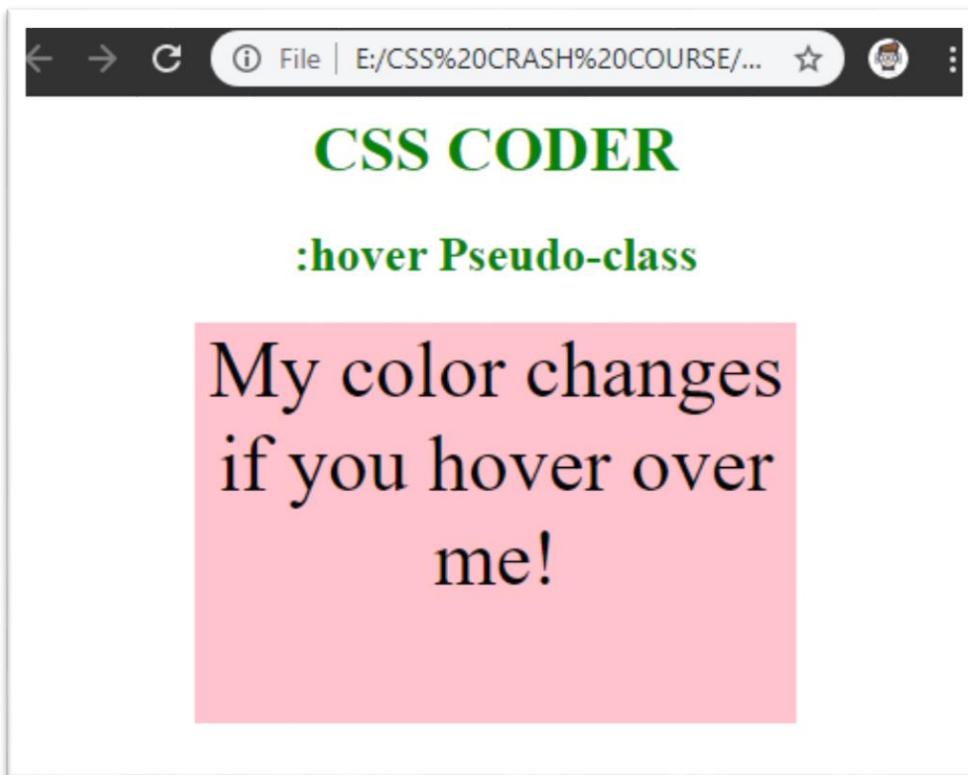
```
</head>
```

```
<body>
<h1>CSS CODER</h1>
<h2>:hover Pseudo-class</h2>
<div class="box">
My color changes if you hover over me!
```

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

**Result:**

**Picture:**



**Question:**

How pseudo-class works?

- 1) by focus on elemnt
- 2) by mouse over on element (correct)**
- 3) by clicking an element
- 4) by selecting an element

## TOPIC ONE ENDS HERE

### 2) :active pseudo-class:

This pseudo-class is used to select an element which is activated when the user clicks on it. The following example demonstrates that when you click on the box, its background color changes for a moment.

Example:

Code:

```
<!DOCTYPE html>

<html>
<head>
<title>CSS :active-class property</title>
<style>
.box{
background-color: hotpink;
width: 400px;
height: 250px;
margin: auto;
font-size: 50px;
text-align: center;
}
.box:active{
background-color: skyblue;
}
h1, h2{
```

```
color: hotpink;  
text-align: center;  
}  
</style>  
</head>  
<body>  
<h1>CSS CODER</h1>  
<h2>:active Pseudo-class</h2>  
<div class="box">  
    My color changes for a moment if you click me!  
</div>  
</body>  
</html>
```

**Result:**

**Picture:**



## CSS CODER

:active Pseudo-class

My color changes  
for a moment if you  
click me!

### Question:

Correct syntax for active class is?

- 1) div.active
- 2) div"active"
- 3) div:active (correct)
- 4) div=active

### Note:

The **:active** pseudo-class is a dynamic **class** which applies when an element is being clicked by the user.

**TOPIC TWO ENDS HERE**

### 3) :visited pseudo-class:

This pseudo-class is used to select the links which have been already visited by the user. In the following example, the color of the link changes once it is visited.

Example:

#### Code:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS :visited pseudo-class property</title>
<style>
body{
    text-align: center;
}
h1, h2{
    color: hotpink;
}
a:visited{
    color: red;
}
</style>
</head>
<body>
<h1>CSS CODER</h1>
```

```
<h2>:visited Pseudo-class</h2>  
  
<p>  
  <a href="#" target="_blank">  
    My color changes once you vist this link  
  </a>  
</p>  
</body>  
</html>
```

### Result:

### Picture:



### Question:

Visit psuedo-class is used for?

- 1) style unvisited link
- 2) style links on hover
- 3) style visited links (correct)**
- 4) stle active links

## Note:

Use the `:link` selector to style links to unvisited pages, the `:hover` selector to style links when you mouse over them, and the `:active` selector to style links when you click on them.

## TOPIC THREE ENDS HERE

### 4) :focus pseudo-class:

This pseudo-class is used to select an element which is currently focussed by the user. It works on user input elements used in forms and is triggered as soon as the user clicks on it. In the following example, the background color of the input field which is currently focussed changes.

Example:

#### Code:

```
<!DOCTYPE html>

<html>
<head>
<title>CSS :focus pseudo-class property</title>
<style>
form{
    width: 300px;
    height: 200px;
    margin: 0 auto;
    text-align: center;
    line-height: 32px;
}
```

```
label{  
    width: 30%;  
}  
  
input{  
    background-color: default;  
    float: right;  
}  
  
input:focus{  
    background-color: pink;  
}  
  
h1, h2{  
    color: hotpink;  
    text-align: center;  
}  
</style>  
</head>  
  
<body>  
    <h1>CSS CODER</h1>  
    <h2>:focus Pseudo-class</h2>  
    <form>  
        <label for="username">Username:</label>  
        <input type="text" name="username"
```

```

placeholder="Enter your username" />

<label for="emailid">Email-Id:</label>
<input type="email" name="emailid"
placeholder="Enter your email-id" />

<label for="Password">Password:</label>
<input type="password" name="Password"
placeholder="Enter your password" />
</form>
</body>
</html>

```

## Result:

### Picture:



## Question:

Which of the following pseudo class is use for input fields?

- 1) visited
- 2) active
- 3) focus (correct)**
- 4) hover

**Note:**

Browsers limits the styles that can be set for a:visited links, due to security issues.

**TOPIC FOUR ENDS HERE**

**END OF MODULE EIGHTEEN**

# MODULE NINETEEN

## CSS Pseudo-elements

Pseudo-element in CSS is used to add style in specified parts of an element.  
Example: Using style before or after an element.

### Use of Pseudo-Element:

Below are some examples to describe the use of pseudo-element.

#### 1) ::before Pseudo-element:

The `::before` pseudo-element can be used to insert some content before the content of an element.

The following example inserts some text before the content of `<h1>` element.

Example:

#### Code:

```
<html>
<head>
    <title>before</title>
    <style>
        h1::before {
            content: "Before element - ";
        }
    </style>
</head>
<body>
```

```
<h1>CSS CODER</h1>  
<p>I help nerds to master their CSS coding skills.</p>  
<p>You are welcome.</p>  
</body>  
</html>
```

## Result:

### Picture:



## Question:

::before pseudo element is used to insert content?

- 1) after the element
- 2) right of the element
- 3) before the element (correct)**
- 4) in the center of element

## Note:

Notice the double colon notation - `::before` versus `:before`

The double colon replaced the single-colon notation for pseudo-elements in CSS3. This was an attempt from W3C to distinguish between pseudo-classes and pseudo-elements.

## TOPIC ONE ENDS HERE

### 2) `::after` Pseudo-element:

The `::after` pseudo-element can be used to insert some content after the content of an element.

The following example inserts some text after the content of `<h1>` element.

Example:

#### Code:

```
<html>
  <head>
    <title>after</title>
    <style>
      h1::after {
        content: " - after element";
      }
    </style>
  </head>
  <body>
    <h1>CSS CODER</h1>
    <p>I help nerds to master their CSS coding skills.</p>
    <p>You are welcome.</p>
  </body>
```

```
</html>
```

## Result:

## Picture:



## Question:

::after is a pseudo?

- 1) element (correct)
- 2) class
- 3) attribute
- 4) property

## Note:

The single-colon syntax was used for both pseudo-classes and pseudo-elements in CSS2 and CSS1.

For backward compatibility, the single-colon syntax is acceptable for CSS2 and CSS1 pseudo-elements.

## TOPIC TWO ENDS HERE

### 3) ::first-letter Pseudo-element:

It is used to make changes to the first letter of an element.

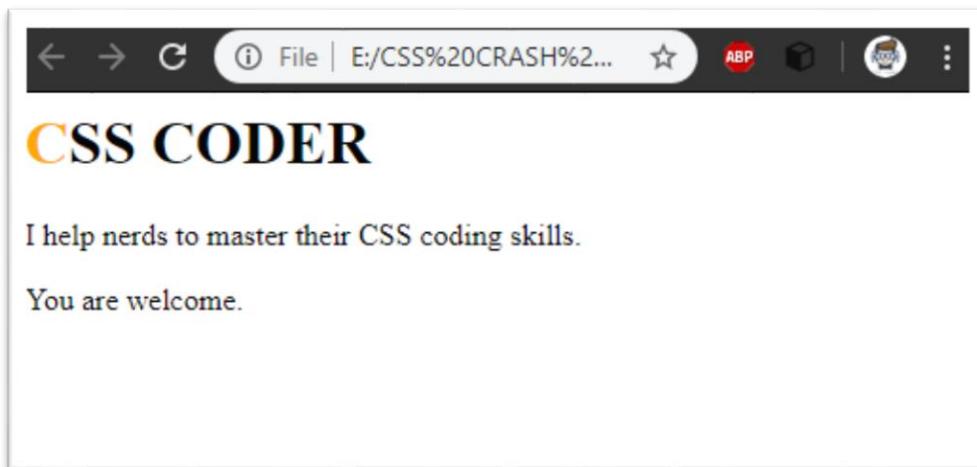
Example:

Code:

```
<html>
<head>
<title>after</title>
<style>
h1::first-letter {
    color: orange;
}
</style>
</head>
<body>
<h1>CSS CODER</h1>
<p>I help nerds to master their CSS coding skills.</p>
<p>You are welcome.</p>
</body>
</html>
```

**Result:**

**Picture:**



**Question:**

In Css ::first-letter is a?

- 1) class
- 2) heading
- 3) pseudo-class
- 4) pseudo-element (correct)

**Note:**

The `::first-letter` pseudo-element can only be applied to block-level elements.

**TOPIC THREE ENDS HERE**

#### **4) ::first-line Pseudo-element:**

The `::first-line` pseudo-element is used to add a special style to the first line of a text.

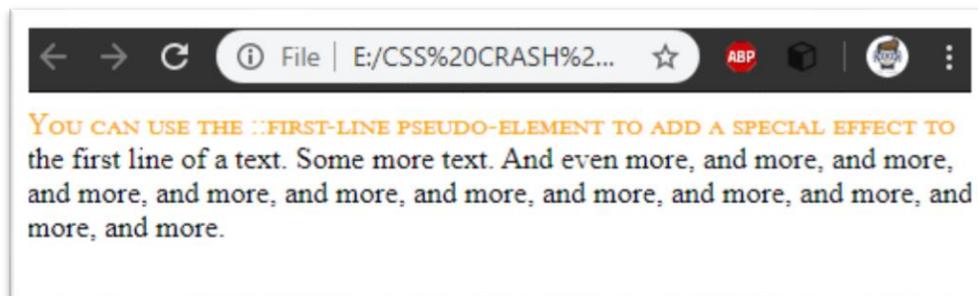
Example:

**Code:**

```
<html>
<head>
<style>
p::first-line {
    color: orange;
    font-variant: small-caps;
}
</style>
</head>
<body>
<p>You can use the ::first-line pseudo-element to add a special effect to the
first line of a text. Some more text. And even more, and more, and more, and
more, and more, and more, and more, and more, and more, and more, and more, and
more, and more.</p>
</body>
</html>
```

**Result:**

**Picture:**



### **Question:**

::first-line pseudo element is used to style?

- 1) headings in a text
- 2) paragraphs in a text
- 3) first line in a text (correct)**
- 4) first letter in a text

### **Note:**

The `::first-line` pseudo-element can only be applied to block-level elements.

**TOPIC FOUR ENDS HERE**

**END OF MODULE NINETEEN**

**END OF THE COURSE**

## **Resources Used For The Creation Of This Ebook:**

- 1) Google
- 2) Codepen
- 3) Github
- 4) Sololearn
- 5) W3Schools
- 6) GeekForGeeks
- 7) Online Tutorials
- 8) Traversy Media
- 9) Css Tricks
- 10) Mdn Docs