

## CSS Background,

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

## Border,

### **border-style**

- `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

### **Border-width**

### **Border-color**

### **Border-radius**

### **Border-Sides**

- `border-top-style: dotted;`
- `border-right-style: solid;`
- `border-bottom-style: dotted;`
- `border-left-style: solid;`

## Margin,

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

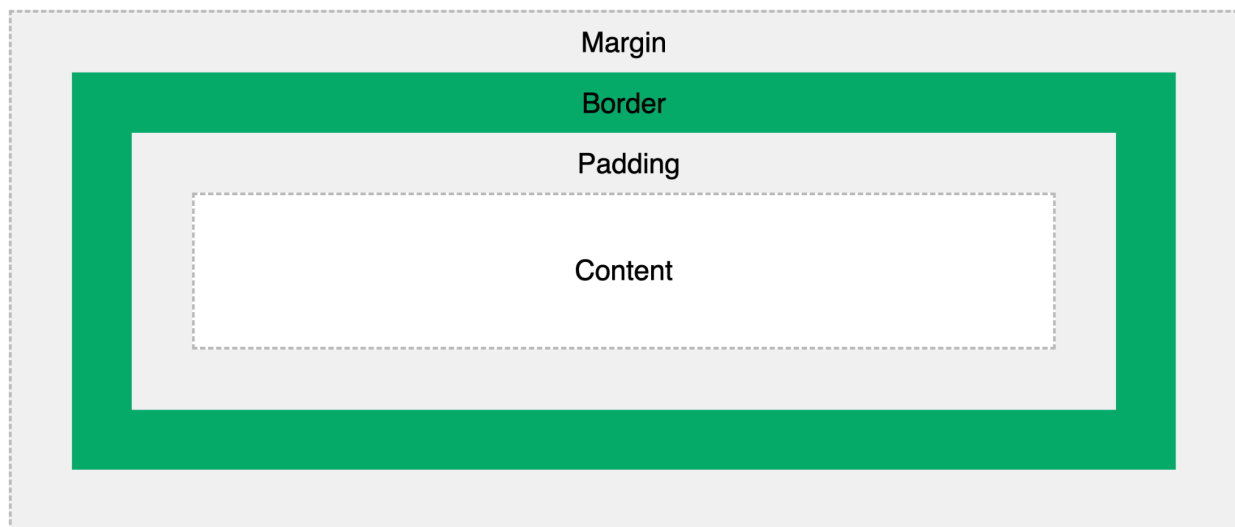
## Padding,

- padding-top
- padding-right
- padding-bottom
- padding-left
- Padding

## Height / width,

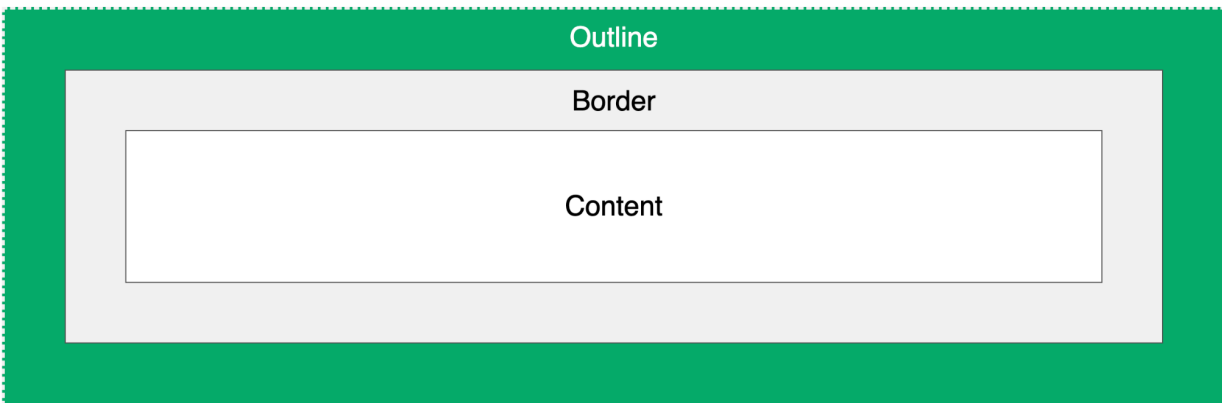
```
div {  
    height: 100px;  
    width: 500px;  
}
```

## Box Model :



## Outline:

- `outline-style`
- `outline-color`
- `outline-width`
- `outline-offset`
- `outline`



## Text :

### Text Color -

```
body {  
    color: blue;  
}
```

### Text Alignment and Text Direction

- `text-align`
- `text-align-last`
- `direction`
- `unicode-bidi`
- `Vertical-align`

## Text Decoration

- `text-decoration-line`
- `text-decoration-color`
- `text-decoration-style`
- `text-decoration-thickness`
- `text-decoration`

## Text Transformation

- `text-transform: uppercase;`
- `text-transform: lowercase;`
- `text-transform: capitalize;`

## Text Shadow

```
h1 {  
  
    text-shadow: 2px 2px red;  
  
}
```

## Fonts

- `font-family`
- `font-style`
- `font-size`
- `font-weight`

## Icons

### Font Awesome Icons

```
<script src="https://kit.fontawesome.com/yourcode.js"  
crossorigin="anonymous"></script>
```

## Link

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

List:

**list-style-type** (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot be displayed)

- **list-style-position** (specifies whether the list-item markers should appear inside or outside the content flow)
- **list-style-image** (specifies an image as the list item marker)

Table:

**border-collapse** - Specifies whether table borders are collapsed into a single border or separated.

**border-spacing** - Defines the space between table cells when borders are not collapsed.

**empty-cells** - Determines whether borders and backgrounds are displayed on empty table cells.

**width** - Sets the width of the table or cells.

**height** - Sets the height of the table or cells.

**background-color** - Specifies the background color of the table, rows, or cells.

**text-align** - Aligns text horizontally within table cells.

**vertical-align** - Aligns content vertically within table cells.

**padding** - Adds space inside table cells between content and borders.

**margin** - Adds space around the table.

**border** - Sets the border style, width, and color for the table or cells.

**border-width** - Specifies the thickness of the table or cell borders.

**border-color** - Sets the color of the table or cell borders.

**border-style** - Specifies the style of the table or cell borders (e.g., solid, dotted).

**nth-child** - A selector for styling specific table rows or columns (e.g., even/odd rows).

**overflow** - Determines how content is displayed when it overflows a table cell.

**white-space** - Controls how white space is handled within table cells.

**box-shadow** - Adds shadow effects to the table or cells.

**font-size** - Sets the size of the text in table elements.

**font-weight** - Defines the thickness of the text in table elements.

**font-family** - Specifies the font style for table text.

**cursor** - Changes the mouse cursor appearance when hovering over table elements.

**clip-path** - Clips the table or cell to a specific shape or path.

## Display

**display: block** - The element is rendered as a block-level element, starting on a new line and taking up the full width.

**display: inline** - The element is rendered as an inline element, taking up only as much width as necessary without breaking to a new line.

**display: inline-block** - Combines features of **block** and **inline**; the element flows inline but respects block-level dimensions like width and height.

**display: none** - Hides the element; it is removed from the document flow and does not take up space.

**display: flex** - Enables a flexible layout container that aligns items in rows or columns.

**display: inline-flex** - Similar to **flex**, but the container itself behaves like an inline element.

**display: grid** - Enables a grid layout container, allowing items to align along both rows and columns.

**display: inline-grid** - Similar to **grid**, but the container behaves like an inline element.

**display: table** - Behaves like a **<table>** element; used for table-like layouts.

**display: table-row** - Behaves like a **<tr>** element in a table.

**display: table-cell** - Behaves like a **<td>** element in a table.

**display: table-caption** - Behaves like a **<caption>** element in a table.

**display: table-column** - Behaves like a **<col>** element, defining column properties.

**display: table-column-group** - Behaves like a **<colgroup>** element, grouping columns.

**display: table-header-group** - Behaves like the **<thead>** element in a table.

**display: table-footer-group** - Behaves like the **<tfoot>** element in a table.

**display: table-row-group** - Behaves like the **<tbody>** element in a table.

**display: contents** - Makes the element itself disappear, but its children remain visible and part of the document flow.

**display: run-in** - Acts as a block or inline element depending on context (rarely used).

**display: list-item** - Makes the element behave like a list item with a bullet or number.

**display: initial** - Resets the **display** property to its default value for that element.

**display: inherit** - Inherits the **display** value from the parent element.

**display: revert** - Resets the **display** property to the browser's default stylesheet value.

## max-width

1. **max-width: value;** - Sets the maximum width an element can have. Prevents the element from exceeding this width regardless of its content or parent container size.
  - Example: **max-width: 500px;**

## width

2. **width: value;** - Sets the width of an element. Can be defined using pixels (**px**), percentages (**%**), viewport units (**vw**), or other units.
  - Example: **width: 100%;**

## margin

3. **margin: value;** - Sets the outer space around an element. Accepts one value for all sides or specific values for individual sides (top, right, bottom, left).
  - Example: **margin: 10px;**
4. **margin-top: value;** - Sets the margin above an element.
  - Example: **margin-top: 20px;**
5. **margin-right: value;** - Sets the margin to the right of an element.
  - Example: **margin-right: 15px;**
6. **margin-bottom: value;** - Sets the margin below an element.
  - Example: **margin-bottom: 10px;**



7. `margin-left: value;` - Sets the margin to the left of an element.
  - Example: `margin-left: 15px;`
8. `margin: auto;` - Centers the element horizontally within its container (requires the container to have a defined width).
  - Example: `margin: auto;`

## Position Property

1. `position: static;`
  - Default value. The element is positioned according to the normal document flow (not affected by top, right, bottom, or left).
2. `position: relative;`
  - The element is positioned relative to its normal position. It can be offset using `top`, `right`, `bottom`, and `left`.
3. `position: absolute;`
  - The element is positioned relative to the nearest positioned (non-static) ancestor. If no such ancestor exists, it is positioned relative to the document's initial containing block.
4. `position: fixed;`
  - The element is positioned relative to the browser's viewport, so it stays in place even when the page is scrolled.
5. `position: sticky;`
  - The element toggles between `relative` and `fixed` positioning based on the scroll position. It is relative until a specified offset is met, then it becomes fixed.

## Offset Properties

6. `top`
  - Specifies the vertical offset of a positioned element from its containing block.
  - Example: `top: 10px;`

## 7. `right`

- Specifies the horizontal offset of a positioned element from the right edge of its containing block.
- Example: `right: 15px;`

## 8. `bottom`

- Specifies the vertical offset of a positioned element from the bottom edge of its containing block.
- Example: `bottom: 20px;`

## 9. `left`

- Specifies the horizontal offset of a positioned element from the left edge of its containing block.
- Example: `left: 25px;`

## z-index Property

### 10. `z-index`

- Specifies the stack order of an element. Elements with a higher `z-index` appear in front of those with a lower `z-index`.
- Example: `z-index: 10;`

## inset Property

### 11. `inset`

- Shorthand for `top`, `right`, `bottom`, and `left`.
- Example: `inset: 10px 20px 30px 40px;`

## `overflow` Property

### 1. `overflow: visible;`

- Default value. Content is not clipped and may overflow outside the element's box.

### 2. `overflow: hidden;`

- Content is clipped and any overflow is not visible.

### 3. `overflow: scroll;`

- Content is clipped, and scrollbars are always visible, regardless of whether scrolling is necessary.

#### 4. `overflow: auto;`

- Content is clipped, and scrollbars appear only when necessary (if the content overflows).
- 

### `overflow-x` Property

#### 5. `overflow-x: visible;`

- Specifies horizontal overflow behavior. Content is not clipped horizontally.

#### 6. `overflow-x: hidden;`

- Content is clipped horizontally, and no horizontal scrollbar is shown.

#### 7. `overflow-x: scroll;`

- Horizontal scrollbars are always visible, regardless of content width.

#### 8. `overflow-x: auto;`

- Horizontal scrollbars appear only if the content overflows horizontally.
- 

### `overflow-y` Property

#### 9. `overflow-y: visible;`

- Specifies vertical overflow behavior. Content is not clipped vertically.

#### 10. `overflow-y: hidden;`

- Content is clipped vertically, and no vertical scrollbar is shown.

#### 11. `overflow-y: scroll;`

- Vertical scrollbars are always visible, regardless of content height.

#### 12. `overflow-y: auto;`

- Vertical scrollbars appear only if the content overflows vertically.

## Shorthand Syntax

13. `overflow: {overflow-x} {overflow-y};`
  - Example: `overflow: hidden scroll;` sets horizontal overflow to `hidden` and vertical overflow to `scroll`.

## Anchor Pseudo-Classes

1. `a:link` - Styles a normal, unvisited link.
  2. `a:visited` - Styles a link that has been visited by the user.
  3. `a:hover` - Styles a link when the user hovers over it.
  4. `a:active` - Styles a link at the moment it is clicked.
- 

## User Interaction Pseudo-Classes

5. `:hover` - Styles an element when the user hovers over it with the mouse.
  6. `:focus` - Styles an element when it gains focus, such as an input field.
  7. `:active` - Styles an element at the moment it is being clicked.
  8. `:visited` - Styles links that have already been visited.
  9. `:focus-visible` - Styles elements that are focused but only when it's apparent to the user (e.g., keyboard navigation).
  10. `:focus-within` - Styles an element if it or any of its descendants has focus.
- 

## Structural Pseudo-Classes

11. `:first-child` - Styles the first child of an element.
12. `:last-child` - Styles the last child of an element.

13. `:nth-child(n)` - Styles the nth child of an element (n can be a number, keyword, or formula).
  14. `:nth-last-child(n)` - Styles the nth child from the end of the parent element.
  15. `:only-child` - Styles an element that is the only child of its parent.
  16. `:first-of-type` - Styles the first child of a specific type within its parent.
  17. `:last-of-type` - Styles the last child of a specific type within its parent.
  18. `:nth-of-type(n)` - Styles the nth child of a specific type within its parent.
  19. `:nth-last-of-type(n)` - Styles the nth child of a specific type from the end of its parent.
  20. `:only-of-type` - Styles an element that is the only child of its type in the parent.
- 

## Input/Form Pseudo-Classes

21. `:checked` - Styles an input element that is checked (e.g., radio button, checkbox).
22. `:disabled` - Styles an input element that is disabled.
23. `:enabled` - Styles an input element that is enabled.
24. `:required` - Styles an input element that is required.
25. `:optional` - Styles an input element that is not required.
26. `:valid` - Styles an input element with valid data.
27. `:invalid` - Styles an input element with invalid data.
28. `:in-range` - Styles an input element with a value within a specified range.
29. `:out-of-range` - Styles an input element with a value outside a specified range.
30. `:read-only` - Styles an input element that is read-only.
31. `:read-write` - Styles an input element that is editable.
32. `:placeholder-shown` - Styles an input element when its placeholder is visible.

---

## Other Pseudo-Classes

- 33. `:not(selector)` - Styles elements that do not match the given selector.
- 34. `:root` - Styles the root element of the document (usually `<html>`).
- 35. `:empty` - Styles elements with no children (including text nodes).
- 36. `:target` - Styles an element that is the target of a URL fragment.
- 37. `:lang(language)` - Styles elements with a specific language attribute.
- 38. `:first-line` - Styles the first line of text in an element.
- 39. `:first-letter` - Styles the first letter of text in an element.
- 40. `:before` - Inserts content before an element.
- 41. `:after` - Inserts content after an element.
- 42. `:visited` - Styles a link the user has visited.
- 43. `:hover` - Applies when a user interacts with an element with a pointing device without activating it.

### 1. `input[type="text"]`

- Selects text input fields (i.e., allows the user to input text).
- Example: `<input type="text" />`

### 2. `input[type="password"]`

- Selects password input fields (i.e., hides the input text for security).
- Example: `<input type="password" />`

### 3. `input[type="number"]`

- Selects number input fields (i.e., allows the user to input numbers).

- Example: `<input type="number" />`
- 4. `input[type="email"]`
  - Selects email input fields (i.e., allows the user to input email addresses).
  - Example: `<input type="email" />`
- 5. `input[type="tel"]`
  - Selects telephone input fields (i.e., allows the user to input telephone numbers).
  - Example: `<input type="tel" />`
- 6. `input[type="url"]`
  - Selects URL input fields (i.e., allows the user to input URLs).
  - Example: `<input type="url" />`
- 7. `input[type="date"]`
  - Selects date input fields (i.e., allows the user to input a date).
  - Example: `<input type="date" />`
- 8. `input[type="time"]`
  - Selects time input fields (i.e., allows the user to input a time).
  - Example: `<input type="time" />`
- 9. `input[type="datetime-local"]`
  - Selects date and time input fields (i.e., allows the user to input both date and time).
  - Example: `<input type="datetime-local" />`
- 10. `input[type="week"]`
  - Selects week input fields (i.e., allows the user to input a week and year).
  - Example: `<input type="week" />`
- 11. `input[type="month"]`
  - Selects month input fields (i.e., allows the user to input a month and year).
  - Example: `<input type="month" />`
- 12. `input[type="color"]`
  - Selects color input fields (i.e., allows the user to pick a color).
  - Example: `<input type="color" />`

13. `input[type="checkbox"]`
  - Selects checkbox input fields (i.e., allows the user to select one or more options).
  - Example: `<input type="checkbox" />`
14. `input[type="radio"]`
  - Selects radio button input fields (i.e., allows the user to select one option from a group).
  - Example: `<input type="radio" />`
15. `input[type="file"]`
  - Selects file input fields (i.e., allows the user to select files from their device).
  - Example: `<input type="file" />`
16. `input[type="submit"]`
  - Selects submit buttons (i.e., allows the user to submit a form).
  - Example: `<input type="submit" />`
17. `input[type="reset"]`
  - Selects reset buttons (i.e., allows the user to reset a form).
  - Example: `<input type="reset" />`
18. `input[type="button"]`
  - Selects generic button input fields (i.e., allows the user to interact with a button but does not submit or reset the form).
  - Example: `<input type="button" />`
19. `input[type="range"]`
  - Selects range input fields (i.e., allows the user to select a value within a specified range).
  - Example: `<input type="range" />`
20. `input[type="search"]`
  - Selects search input fields (i.e., allows the user to input a search query).
  - Example: `<input type="search" />`



