CSS NOTES

CSS selectors including simple, combinator, pseudo-class, pseudo-element, and attribute selectors:

\*Simple Selectors:\*

1. \*Type Selector:\* Selects elements based on their element type. Example: div, p, span.

2. \*Class Selector:\* Selects elements based on their class attribute. Example: .classname.

3. \*ID Selector:\* Selects elements based on their id attribute. Example: #idname.

4.\*Universal Selector:\*Selects all elements in webpage at once.

5.\*Grouping Selector:\* Selects multiple elements same time separated by comma. Example: h1,p,div{//css}

\*Combinator Selectors:\*

1. \*Descendant Selector:\* Selects an element that is a descendant of another specified element. Example: div p.

2. \*Child Selector:\* Selects an element that is a direct child of another specified element. Example: ul > li.

3. \*Adjacent Sibling Selector:\* Selects an element that is immediately preceded by a sibling element. Example: h2 + p.

4. \*General Sibling Selector:\* Selects elements that are siblings of a specified element. Example: h2 ~ p.

\*Pseudo-class Selectors:\*

1. \*:hover:\* Selects an element when the mouse pointer is over it.

2. \*:active:\* Selects an element when it is being activated by the user.

3. \*:focus:\* Selects an element when it is in focus.

4. \*:first-child:\* Selects an element that is the first child of its parent.

5. \*:last-child:\* Selects an element that is the last child of its parent.

6. \*:nth-child():\* Selects elements based on their position within a parent.

7. \*:not():\* Selects elements that do not match a specific selector.

8. \*:nth-of-type():\* Selects elements based on their position within a parent, counting only elements of the same type.

9. \*:checked:\* Selects input elements that are checked.

\*Pseudo-element Selectors:\*

1. \*::before:\* Inserts content before the selected element.

2. \*::after:\* Inserts content after the selected element.

3. \*::first-line:\* Selects the first line of text within the selected element.

4. \*::first-letter:\* Selects the first letter of text within the selected element.

5. \*::selection:\* Selects the portion of an element that is selected by a user.

\*Attribute Selectors:\*

1. \*[attribute]:\* Selects elements that have the specified attribute.

2. \*[attribute=value]:\* Selects elements with the specified attribute and value.

3. \*[attribute~=value]:\* Selects elements with an attribute that includes the specified value as one of its space-separated values.

4. \*[attribute|=value]:\* Selects elements with an attribute that exactly matches the specified value or starts with the specified value followed by a hyphen.

notes on the priorities between inline CSS, internal CSS, external CSS, class selectors, ID selectors, tag name selectors.

\*\* Inline CSS:\*\*

- Applied directly to individual HTML elements using the style attribute.

- Highest specificity.

- Overrides external and internal styles for the targeted element.

- Example: <div style="color: red;">.

\*\* Selector Specificity:\*\*

- Inline styles have the highest specificity, followed by ID selectors, class selectors, and tag name selectors.

- Specificity is calculated based on the combination of selectors used to target an element.

- The more specific selector takes precedence over less specific selectors.

\*\* ID Selectors:\*\*

- Defined using the id attribute in HTML elements.

- More specific than class and tag name selectors.

- Should be used for unique elements.

- Example: <div id="uniqueElement">.

\*\* Class Selectors:\*\*

- Defined using the class attribute in HTML elements.

- Less specific than ID selectors but more specific than tag name selectors.

- Can be applied to multiple elements.

- Example: <div class="container">.

\*Tag Name Selectors:\*

- Targets all HTML elements of a specific type.

- Least specific selector.

- Example: p { color: green; }.

\*\* Importance:\*\*

- The !important declaration overrides normal specificity rules.

- Should be used sparingly as it can make styles harder to override and maintain.

CSS selectors including simple, combinator, pseudo-class, pseudo-element, and attribute selectors:

\*Simple Selectors:\*

1. \*Type Selector:\* Selects elements based on their element type. Example: div, p, span.

2. \*Class Selector:\* Selects elements based on their class attribute. Example: .classname.

3. \*ID Selector:\* Selects elements based on their id attribute. Example: #idname.

4.\*Universal Selector:\*Selects all elements in webpage at once.

5.\*Grouping Selector:\* Selects multiple elements same time separated by comma. Example: h1,p,div{//css}

\*Combinator Selectors:\*

1. \*Descendant Selector:\* Selects an element that is a descendant of another specified element. Example: div p.

2. \*Child Selector:\* Selects an element that is a direct child of another specified element. Example: ul > li.

3. \*Adjacent Sibling Selector:\* Selects an element that is immediately preceded by a sibling element. Example: h2 + p.

4. \*General Sibling Selector:\* Selects elements that are siblings of a specified element. Example: h2 ~ p.

\*Pseudo-class Selectors:\*

1. \*:hover:\* Selects an element when the mouse pointer is over it.

2. \*:active:\* Selects an element when it is being activated by the user.

3. \*:focus:\* Selects an element when it is in focus.

4. \*:first-child:\* Selects an element that is the first child of its parent.

5. \*:last-child:\* Selects an element that is the last child of its parent.

6. \*:nth-child():\* Selects elements based on their position within a parent.

7. \*:not():\* Selects elements that do not match a specific selector.

8. \*:nth-of-type():\* Selects elements based on their position within a parent, counting only elements of the same type.

9. \*:checked:\* Selects input elements that are checked.

\*Pseudo-element Selectors:\*

1. \*::before:\* Inserts content before the selected element.

2. \*::after:\* Inserts content after the selected element.

3. \*::first-line:\* Selects the first line of text within the selected element.

4. \*::first-letter:\* Selects the first letter of text within the selected element.

5. \*::selection:\* Selects the portion of an element that is selected by a user.

\*Attribute Selectors:\*

1. \*[attribute]:\* Selects elements that have the specified attribute.

2. \*[attribute=value]:\* Selects elements with the specified attribute and value.

3. \*[attribute~=value]:\* Selects elements with an attribute that includes the specified value as one of its space-separated values.

4. \*[attribute|=value]:\* Selects elements with an attribute that exactly matches the specified value or starts with the specified value followed by a hyphen.

* **Color Property :-**The color property specifies the color of the text.

1. **Color Name:** It will target the text color by directly specifying the name of the color like blue, green, yellow, white, black, etc.

**Syntax:**

color: name-of-the-color;

1. **RGB/RGBA Value:**Here R stands for Red, G stands for Green, and B stands for Blue. The color will be assigned to the text by using the range of these values. These values range from 0 to 255. And, A stands for Alpha channel. Which represents the opacity or opaque of the color.

**Syntax:**

color: RGB(value, value, value);

color: RGBA(value, value, value, value);

1. **Hexa-Decimal Value:**It represents the value of the color in hexadecimal format. It should start with the prefix #. These values range from #000000 to #FFFFFF. And, If there is an alpha channel that defines the opacity of the color, then we will represent it by adding FF (if 100%) after the hex code.

**Syntax:**

color: #RRGGBBFF;

1. **HSL/HSLA values:**HSL stands for Hue, Saturation, and Lightness. The range of hue will be from (0 to 360 degree), saturation means the Grey effect it ranges from (0 to 100%), and Lightness means the effect of light which ranges from (0 to 100%).

**Syntax:**

color: HSL(value, value, value);

color: HSLA(value, value, value, value);

* **Font Property :-**The font property specifies the look of the texts.

1. **Font Size:** It is used to control the size of the text. Or

It is used to increase or decrease the size of the text.

**Syntax:**

font-size:24 px; [Values in Px or %]

1. [**Font-Family**](https://www.geeksforgeeks.org/css-font-family-property/)**:** It is used to set the font type of an HTML element. It holds several font names as a fallback system.

There are two ways to add google fonts.

1. <link> Tag (**Html Way)**
2. @import **(CSS Way)**

**Syntax:**

font-family: "font family name";

1. [**Font-**](https://www.geeksforgeeks.org/css-font-family-property/)**Weight:** It is used to set the boldness of the font. Its value can be “normal, bold, lighter, bolder”. We can control the thickness of the font.

**Syntax:**

font-weight: bold/light/normal;

1. [**Font-variant**](https://www.geeksforgeeks.org/css-font-variant-property/)**:** It is used to create the small-caps effect. It can be “normal or small-caps”.

**Syntax:**

font-variant: normal/small-caps;

1. [**Font-style**](https://www.geeksforgeeks.org/css-font-style-property/)**:** It is used to specify the font style of an HTML element. It can be “normal, italic or oblique”.

font-style: normal; [Default]

font-style: italics;

font-style: oblique 0-90deg;

* **Background Property :-** The CSS background properties are used to define the background effects for elements.

1. [Background-color Property](https://www.geeksforgeeks.org/css-background-color-property/#:~:text=The%20background%2Dcolor%20property%20in,to%20read%20for%20the%20user.): The background-color property in CSS is used to specify the background color of an element.

**Syntax:**

body {

background-color:color-name;

}

**6 ways to add color by using**:- 1. Color-name

1. By Hexadecimal(#)
2. RGB(0-255,0-255,0-255)
3. HSL(0-360,0-100%,0-100%)
4. RGBA(0-255,0-255,0-255,0-1)
5. HSLA(0-360,0-100%,0-100%,0-1)

2. [**Background Image Property**](https://www.geeksforgeeks.org/css-background-image-property/)**:**This property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.

**Syntax:**

body {

background-image : URL(“link”);

}

**Syntax:  Linear Gradient**

body {

background-image : linear-gradient(direction,color1,color-2…color-n);

}

Directions:-

* + 1. to bottom v. to top left
    2. to top vi. to top right
    3. to right vii. to bottom left
    4. to left viii. To bottom right

**Syntax: Radial Gradient**

body {

background-image : radial-gradient(direction,color1,color-2…color-n);

}

Directions:-

1. Circle at bottom v. Circle at top left
2. Circle at top vi. Circle at top right
3. Circle at right vii. Circle at bottom left
4. Circle at left viii. Circle at bottom right
5. [**Background-repeat Property**](https://www.geeksforgeeks.org/css-background-repeat-property/#:~:text=The%20background%2Drepeat%20property%20in,will%20be%20repeated%20or%20not.&text=The%20last%20image%20will%20be,fit%20in%20the%20browser%20window.)**:** The background-repeat property in CSS is used to repeat the background image both horizontally and vertically.

**Syntax:**To repeat an image horizontally

body {

background-image: URL(“link”);

background-repeat: repeat-x;

}

**Syntax:**To repeat an image vertically

body {

background-image: URL(“link”);

background-repeat: repeat-y;

}

**In case of no repetition :-**

**Syntax:**

body {

background-image: URL(“link”);

background-repeat: no-repeat;

}

**In case of round :-** Here, background is stretched and shrinks according to container space and original aspect ratio is not maintained.

**Syntax:**

body {

background-image: URL(“link”);

background-repeat: round;

}

**In case of space :-** Here, background is displayed with original aspect ratio, background is not stretched aor shrinks and background is applied with equal spaces

**Syntax:**

body {

background-image: URL(“link”);

background-repeat: space;

}

1. [**Background-size Property**](https://www.geeksforgeeks.org/css-background-repeat-property/#:~:text=The%20background%2Drepeat%20property%20in,will%20be%20repeated%20or%20not.&text=The%20last%20image%20will%20be,fit%20in%20the%20browser%20window.)**:** The **background-size** property in CSS is used to set the size of the background image.

**Syntax:**

background-size: auto|length|cover|contain

1. **auto:** It is used to set the background-size property to its default value. It is used to display the [background-image](https://www.geeksforgeeks.org/css-background-image-property/) to its original size.

**Syntax:**

body {

background-image: URL(“link”);

background-repeat: no-repeat;

background-size: auto;

}

1. **length:** It is used to set the width and height of the background-image. The first value indicates the width, and the second value indicates the height of the background image in terms of px, pt, em, etc. If any value is not given then it is set to auto.

**Syntax:**

body {

background-image: URL(“link”);

background-repeat: no-repeat;

background-size: width height;

}

1. **cover:** It is used to resize the background image to cover a whole container element.

**Syntax:**

body {

background-image: URL(“link”);

background-repeat: no-repeat;

background-size: cover;

}

1. **contain:** Here image is displayed with maximum size. Image is completely visible.

**Syntax:**

body {

background-image: URL(“link”);

background-repeat: no-repeat;

background-size: contain;

}

1. [**Background-position Property**](https://www.geeksforgeeks.org/css-background-repeat-property/#:~:text=The%20background%2Drepeat%20property%20in,will%20be%20repeated%20or%20not.&text=The%20last%20image%20will%20be,fit%20in%20the%20browser%20window.)**:** The **background-position** property  is used to set an image at a certain position.

**Note:-**The [background-image](https://www.geeksforgeeks.org/css-background-image-property/) is placed default to the top-left corner of an element with a repetition on both horizontally & vertically.

Top-left Corner

**(Default)**

Top-right Corner

Top center

Image



Right center

Left center

center center

Bottom center

Bottom-left Corner

Bottom-left Corner

Values :-

1. Top center v. top left ix) center center
2. Bottom center vi. top right x) 20px 40px
3. Right center vii. bottom left xi) top 50% left 50%
4. Left center viii. bottom right
5. [**Background-attachment Property**](https://www.geeksforgeeks.org/css-background-repeat-property/#:~:text=The%20background%2Drepeat%20property%20in,will%20be%20repeated%20or%20not.&text=The%20last%20image%20will%20be,fit%20in%20the%20browser%20window.)**:** The **property background-attachment property** in CSS is used to specify the kind of attachment of the background image with respect to its container. It can be set to scroll or remain fixed.

**Values:-**

**scroll:** It sets the background image to get fixed at its position with respect to the containing element and scroll with the page. It is the default value.

**fixed:** This property is used to set the background image to fixed at its position with respect to the viewport.

**local:** This property is used to set the background image to scroll along with the content of its container element instead of the page.

**Syntax:**

background-attachment: scroll| fixed| local;

1. [**Background-clip Property**](https://www.geeksforgeeks.org/css-background-repeat-property/#:~:text=The%20background%2Drepeat%20property%20in,will%20be%20repeated%20or%20not.&text=The%20last%20image%20will%20be,fit%20in%20the%20browser%20window.) **:-** The **background-clip property** in CSS is used to define how to extend the background (color or image) within an element.

**Values:-**

* **border-box:** The border-box property is used to set the background color spread over the whole division.
* **padding-box:** The padding-box property is used to set the background inside the border.
* **content-box:** The content-box property is used to set the background color up to the content only.
* **Text:** The background is painted within (clipped to) the foreground text.

**Syntax :-**

background-clip: border-box |padding-box |content-box |text

* **Vendor Prefix :-** Vendor prefix are provided by browser vendors because of addition of new new css features, web developers facing problem of browser compatibility issue that is browser don’t know how to implement new css features.

To implement new features before manufacturer updates the browser. We send these new features to the engine and to do so vendor provides prefix.

|  |  |
| --- | --- |
| Browser | Vendor Prefix |
| Chrome, Safari | -webkit- |
| Firefox | -moz- |
| Ms Edge | -ms- |
| Opera | -o- |

**Syntax :-**

-webkit-background-clip: text;

1. **Text Properties :-** It is used to format text, style the text .

* [text-align](https://www.geeksforgeeks.org/css-text-align-last-property/): This property in CSS is used to specify the horizontal alignment of text in an element inside a block element or table-cell box.

**Syntax:**

  h2 {

   text-align: values; [values:- start,end,center,justify]

         }

* [text-transform](https://www.geeksforgeeks.org/css-text-transform-property/): It is used to control the capitalization of the text.

**Syntax:**

h2 {

   text-transform: values;

[values:- Uppercase,lowercase,capitalize,none]

}

* + [text-decoration](https://www.geeksforgeeks.org/css-text-decoration-property/): text-decoration property is used to “decorate ” the content of the text.

**Syntax:**

h2 {

   text-decoration: values;

[values:- Underline, overline, line-through,none]

}

* + Line-height: It specifies the height of a line. It is used to control the space between line.

**Syntax:**

h2 {

   line-height: values in px,%;

}

* + Letter-spacing: It is used to control the space between letters.

**Syntax:**

h2 {

letter-spacing: values in px,%;

}

* + Word-spacing: It is used to control the space between words.

**Syntax:**

h2 {

word-spacing: values in px,%;

}

* + Text-shadow :- it is used to add shadow to the text.

**Syntax:**

h2 {

Text-shadow: Offset-x Offset-y blur-radius color;

}

* **Offset-x:**This property is required & used to specify the position of horizontal shadow. It accepts the negative values.
* **Offset-y:** This property is required & used to specify the position of vertical shadow. It also accepts the negative values.
* **blur-radius:** It is used to set the blur radius. Its default value is 0 & is optional.
  + **color:** It is used to set the color of the shadow. It is optional.

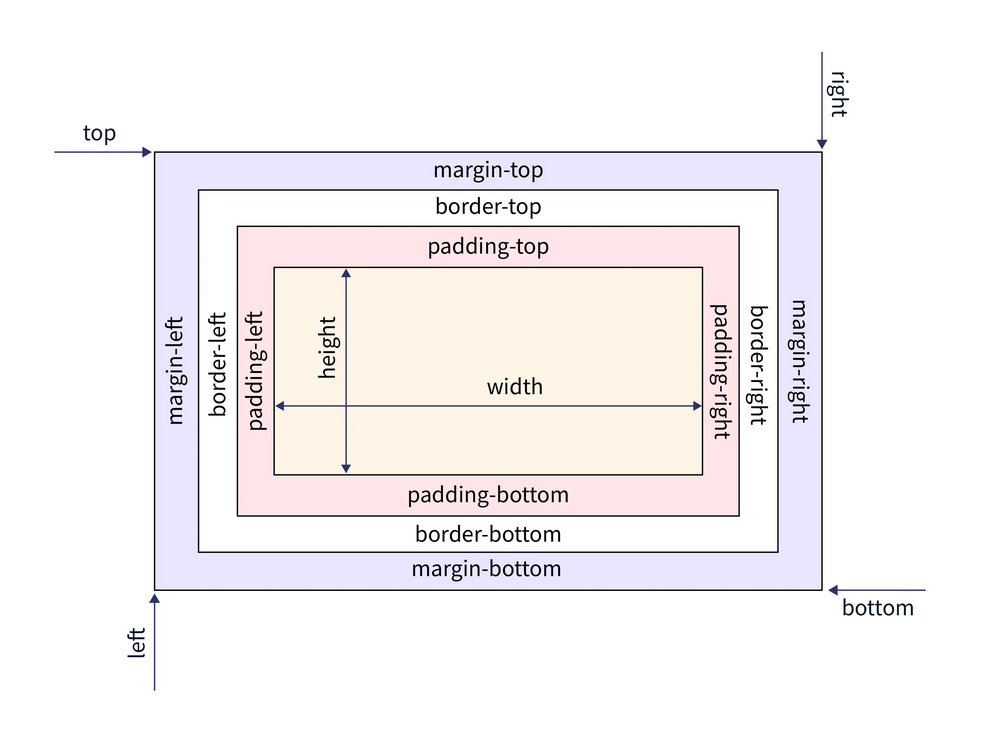
h2 {

Text-shadow: 20px 20px 2px color;

}

* **CSS Box Model** :-
* It is recently added in CSS-3.
* According to CSS Box Model if we display anything on browser will be wrapped by one rectangular box.
* CSS box model helps us to determine position size etc of text or element.
* CSS box model provides 4 parts:-

1. **Margin Area :-** The space outside the border box is known as Margin Area.
2. **Border Area :-** It is the edge of element boundary. Here we can control the boundary width.
3. **Padding Area :-** The space between border box and content box is known as padding area.
4. **Content Area :-** The area where content is available is known as content area. Here content will be wrapped by one box called as content box.



* **Box-sizing :-** The **box-sizing** property in CSS defines how the user should calculate the total width and height of an element i.e padding and borders, are to be included or not.

**Syntax:**

box-sizing: content-box|border-box;

**content-box:** This is the default value of the box-sizing property. In this mode, the width and height properties include only the content. Border and padding are not included in it.

Here height and width will be applied on the content box.

The dimensions (width x height) of content will be constant(will not changed).

**Syntax:**

**\* {**

box-sizing: content-box;

}

**border-box: Here height and width will be applied on border box.**

**The dimensions of border-box (width x height) will be constant (will not changed).**

* **Margin Property:**CSS margins are used to create space around the element. We can set the different sizes of margins for individual sides(top, right, bottom, left).
* [**margin-top**](https://www.geeksforgeeks.org/css-margin-top-property/)**:**It is used to set the top margin of an element.
* [**margin-right**](https://www.geeksforgeeks.org/css-margin-right-property/)**:** It is used to set the right margin of an element.
* [**margin-bottom**](https://www.geeksforgeeks.org/css-margin-bottom-property/)**:**It is used to specify the amount of margin to be used on the bottom of an element.
* [**margin-left**](https://www.geeksforgeeks.org/css-margin-left-property/)**:** It is used to set the width of the margin on the left of the desired element.
* Shorthand :- **If the margin property has 4 values:**

div {

margin: 40px 100px 120px 80px;

}

**top**= 40px

**right** = 100px

**bottom** = 120px

**left** = 80px

* Shorthand :- **If the margin property has 3 values:**

div {

margin: 40px 100px 120px;

}

**top**= 40px

**right** **and left**= 100px

**bottom** = 120px

* Shorthand :- **If the margin property has 2 values:**

div {

margin: 40px 100px;

}

**top and bottom =** 40px;

**left and right =** 100px;

* Shorthand :- **If the margin property has 1 values:**

div {

margin: 40px;

}

**top, right, bottom and left =**40px

* Shorthand :- **Equal margin from left and right . It will used to** **horizontally center the element within its container.**

**For Horizontally center**

div {

margin:auto;

}

**right and left =**auto

**For vertically center :-**

div {

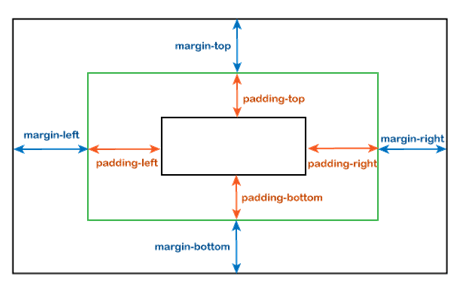
margin: 150px auto;

}

**Top and bottom =**150px

**Right and left =** auto

* **Padding Property :-** CSS paddings are used to create space around the element, inside any defined border.



* [padding-top](https://www.geeksforgeeks.org/css-padding-top-property/): It is used to set the width of the padding area on the top of an element.
* [padding-right](https://www.geeksforgeeks.org/css-padding-right-property/): It is used to set the width of the padding area on the right of an element.
* [padding-bottom](https://www.geeksforgeeks.org/css-padding-bottom-property/): It is used to set the height of the padding area on the bottom of an element.
* [padding-left](https://www.geeksforgeeks.org/css-padding-left-property/): It is used to set the width of the padding area on the left of an element.
* Shorthand :- **If the padding property has 4 values**

Div{

padding: 40px 100px 120px 80px;

 }

**top =** 40px

**right =** 100px

**bottom =**120px

**left =**80px

* Shorthand :- **If the padding property has 3 values**

Div{

padding: 40px 100px 120px;

}

**top =** 40px

**right and left =**100px

**bottom =** 120px

* Shorthand :- **If the padding property has 2 values**

Div{

padding: 100px 150px;

}

**top and bottom =**100px;

**left and right =**150px;

* Shorthand :- **If the padding property has 1 value**

Div{

padding: 100px;

}

**top, right, bottom and left =**100px

* ­Border :- CSS border properties allow us to set the style, color, and width of the border.

1. **Border-Width** :- Border width sets the width of the border. Value in Px.
2. **Border-Style** :- The border-style property specifies the type of border.

Example:-

* dotted – It describes a dotted border
* dashed – It describes a dashed border
* solid – It describes a solid border
* double – It describes a double border
* groove – It describes a 3D grooved border.
* ridge – It describes a 3D ridged border.

1. [**Border Color**](https://www.geeksforgeeks.org/css-border-color-property/)**:** This property is used to set the color of the border. Color can be set using the color name, hex value, or RGB value. If the color is not specified border inherits the color of the element itself.
2. **Individual Border sides** :-

Border-top : width style color ;

Border-bottom : width style color ;

Border-right : width style color ;

Border-left : width style color ;

1. **Border radius property:**It is used to round the corner of the border that looks more attractive.

**Example:-**

1. **For complete circle**

**Border-radius: 50% ; (height=width)**

1. **For Round Corners**

**Border-radius: 8px ;** **(value in Px)**

1. **Individual Border radius :**

* **Border-top-left-radius :- This property rounds the bottom left corner of an element.**
* **Border-top-right-radius** :- This property rounds the bottom right corner of an element.
* **Border**-**bottom-right-radius** : This property rounds the bottom right corner of an element.
* **Border**-**bottom-left-radius** : This property rounds the bottom left corner of an element.

Bottom-left Corner

Bottom-left Corner

Top-right Corner

Top-left Corner

* **Results of Border Radius** :-
* **Overflow Property** **: The overflow property is used to control the element overflow.**

**Values :- Hidden,visible,scroll**

* **Hidden:** The overflow is clipped and the rest of the content is invisible.
* **Visible:** The content is not clipped and visible outside the element box.
* **Scroll:** The overflow is clipped but a scrollbar is added to see the rest of the content. The scrollbar can be horizontal or vertical.
* **Display Property** **:** The **Display property** in CSS defines how the components (div, hyperlink, heading, etc) are going to be placed on the web page. this property is used to define the display of the different parts of a web page.

**Values:-**

1. Inline
2. Block
3. Inline-Block
4. Flex
5. none

|  |  |
| --- | --- |
| **inline** | **It is used to displays an element as an inline element.** |
| **block** | **It is used to displays an element as a block element** |
| **inline-block** | **It is used to display an element as an inline-level block container.**  **Occupy space required by content. We can set height and width.** |
| **flex** | **It is used to display an element as a block-level flex container.** |
| **none** | **It is used to remove the element.** |

* **Position Property** **:** The CSS [position property](https://www.geeksforgeeks.org/css-positioning-elements/) is used to define the position of the element on the web page. By using the top, left, bottom, right, and z-index, we can identify the exact position of the element.

Values:-

**static (default)**

**relative**

**absolute**

**fixed**

**sticky**

1. **Static:-**

It is the default position value for the element. Under static position, elements are positioned according to the normal flow of the page.

**Note:**left, right, top, and bottom properties will not affect if the position is static.

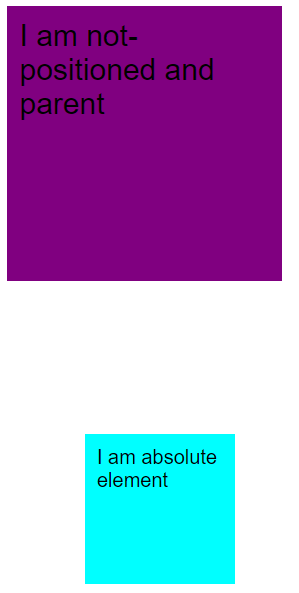
1. **Relative:-**

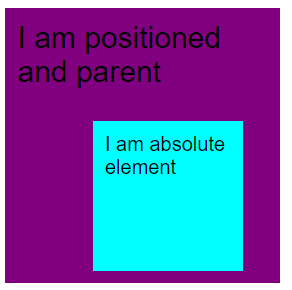
* Position element w.r.t its original position.
* Elemnet will remain in browser, empty space is not occupied.
* Left,Right,Top,Bottom is used to shift the position of an element.

1. **Absolute :-**

* Position Element w.r.t its nearest parent.
* Element will come out of the browser flow, empty space will be occupied by succeeding element.
* Left,Right,Top,Bottom is used to shift the position of an element.

**Note:-** If you want to positioned an element w.r.t its nearest parent then you have to provide position property to the parent.



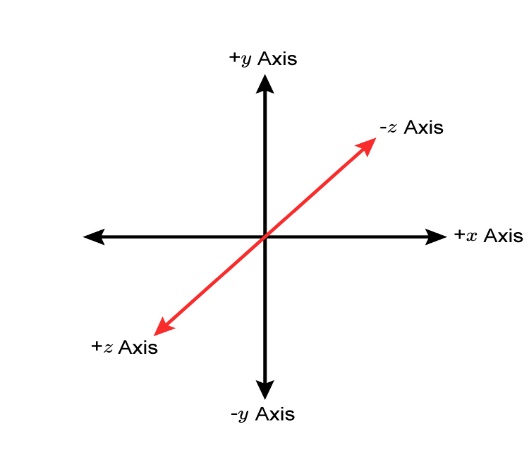


If you not provide position property to the parent then abolute element will be positioned w.r.t body element

1. **Fixed :-**

* Fixed element does not follow normal document flow and positions themselves relative to <HTML> tag. This element always stick to the screen.
* Positioned element w.r.t Viewport (Ex:- Chat with us).

1. **Sticky :-**

* Position an element according to user scroll
* Example:- Navigation/Header in websites.
* **Z-index :-**
* The z-index property is used to displace elements on the z-axis i.e in or out of the screen. It is used to define the order of elements if they overlap on each other.
* This Property is used to position element on z-axis.
* **Example :-** z-index: number(1,2,3,4…);
* **Transition Property :-** The transition property in css is used to make some transition effect.

The transition effect can be defined in two states.(:hover and :active) using pseudo class selectors.

The Transition property is the combination of 4 properties:-

1. Transition-property
2. Transition-duration
3. Transition-timing-function
4. Transition-delay

* **Transition-Property :** To specify on which property transition should be applied.

**Values :- width , Height , all, Background, none.**

* **Transition-duration :** This property allows you to determine how long it will take to complete the transition from one CSS property to the other.

**Or**

Time to be taken to complete the transition. Value in Seconds**(s)** or Milliseconds **(ms)**.

**Example:-** **transition-duration : 2s;**

* **Transition-timing-function :** This property allows you to determine the speed of change during the transition effect. Like, the change should be fast at the beginning and slow at the end, etc.

**Example:-**

transition-timing-function: ease|ease-in|ease-out|ease-in-out|linear|step-start|step-end;

1. Linear : (Default Value) Same speed throughout the transition.
2. Ease : Slow Start + Fast + Slow End
3. Ease-in : Slow Start
4. Ease-out : Slow End
5. Ease-in-out : Slow Start + Linear + Slow End

* **Transition-delay :** This property allows you to determine the amount of time to wait before the transition actually starts to take place.

**OR**

Time to be taken to start the transition. Value in Seconds**(s)** or Milliseconds **(ms)**.

**Example:-** **transition-delay : 2s;**

* **Shorthand Property :**

**transition**: (property name) | (duration) | (timing function) | (delay);

* **Animation Property :-** The  [CSS](https://www.geeksforgeeks.org/css/) ***animation***property is used to specify the animation that should be applied to an element.
* **Animation Name :**
* To animate an element we have to provide animation to the element.
* Example: **Animation-name** : Box1 ;
* By this name {Box1} @keyframe will target the element for animation.
* **@Keyframes :-** Keyframes defines the style to be applied for that moment with in the animation.

We can apply style in two ways :-

1. **From and To :-** It will target starting and ending of animation.

@keyframes identifier {

From{

Here you have to mention **box-name** whichismentioned in Animation-nameProperty

}

To {

}

}

1. **Percentage :-**

@keyframes identifier {

We can use percentage. By using percentage (%) we can target any moment in the animation

0%{

background-color : red;

}

50% {

background-color : green;

}

100%{

background-color : red;

}

}

* **animation-duration:** It is used to specify the time duration and it takes animation to complete one cycle.
* **animation-timing-function:** It is used to specify how the animation makes transitions through keyframes.
* **animation-delay:** It is used to specify the delay when the animation starts.
* **animation-iteration-count:** It is used to specify the number of times the animation will repeat. It can specify as **infinite** to repeat the animation indefinitely.

**Example :-** **animation-iteration-count:** value**;** [ **value :-** 1,2,3,infinite]

* **animation-direction:** It is used to specify the direction of the animation.

**Value :-**

* 1. normal

Forward

* 1. Reverse

Reverse

* 1. Alternate

Forward

Reverse

* 1. alternate-reverse

Reverse

Forward