# Cascading style sheet

* CSS is the language we use to style an HTML document.
* Responsible for styling the web.
* CSS is referred to as layout and look of the web page.
* CSS means apply design document.

CSS SYNTAX OR CSS RULE :-

* + 1. Selectors
    2. Declaration blocks
    3. Property
    4. Value

*EX* :-h1{color:red}

Classification of CSS :-

|  |  |  |
| --- | --- | --- |
| INLINE CSS | INTERNAL CSS | EXTERNAL CSS |
| In inline CSS ,we are using style attribute inside the opening tag of html element. | In internal CSS , we are writing the style inside <style></style> tag. The better way to write style tag we need write that inside <head> tag of our document. | In external CSS , we are attaching the CSS file at present outside of our document. The extension external css file should be dot CSS . To link that external into our html document we are using link tag with two attribute rel and href .rel= we are providing the relation of the document. href = here providing the path of both absolute and relative.  Advantage :-  1.reusability of the css.  2.cleaness of the html document. |
| <h1style=“color:red,font-size:47px”>Hello world</h> | <style>  h1{ color : red;  }  </style> | <link rel=“stylesheet”  href=“path.css”> |

COMMENTS :-

/\*----------------------------

-----------------------------

----------------------------\*/

NOTE :-

Inline CSS > Internal CSS > External CSS

SELECTORS :-For selecting html tag to provide the style we need selectors. Basically the selectors are classified into 5 types

1.Simple Selectors

2.Combinators selectors

3.Pseudo class selectors

4.Pseudo Element selectors

5.Attribute selectors.

SIMPLE SELECTORS :-Select elements based on name , id , class .Basically the simple selectors are classified into 5 types .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TAG NAME | ID NAME (#) | CLASS NAME(.) | UNIVERSAL SELECTOR (\*) | GROUPING SELECTOR( ,) |
| In tag name selector , we are selecting the element by using the tag names | For selecting an element with its Id name ,we need to use ID attribute and provide some name to that element. To select the ID name we need to use # symbol before that name. ID name should be unique i.e. we cannot take multiple ID name for one html tag element | For selecting an element with CLASS name, we need to use CLASS attribute and provide some name to that element. To select the CLASS name we need to use dot symbol before that name. CLASS name should be duplicate i.e. we can take multiple class name for one html tag element | Using universal selector to apply the styles in every tag of our document. By apply the universal selector we need to use \* symbol. | By the help of group selector selects all the HTML elements with the same style definitions. |
| EX:- h1{color:red} | EX:-#heading {color:red} | EX:- .heading{color:red} | EX:-\*{color:red} | .head , #head {color:red} |

COMBINATOR SELECTORS :-Select elements based on the specific relationship between them. Basically these are classified into 4 type.

|  |  |  |  |
| --- | --- | --- | --- |
| DESCENDANT SELECTOR() | CHILD SLEECTOR( >) | ADJACCENT SIBLING (+) | GENERAL SIBLING (~) |
| For using descendant selectors ,we need parent and child relationship (child may be direct or indirect child) . | For using child selector , we need parent and child relationship (child is direct child) | In adjacent sibling selector, we can be able to select the immediate coming element. The elements must share one parent. | By using general sibling selector we can be able to select all sibling of a given element. The element should share single parent and symbol is tilt (~). |
| EX:- head h1{color:red} | EX:-head>h1{color:red} | EX:-div+p{color:red} | Ex:-div~p{color : red} |

PSEUDO CLASS SELECTORS (:) :- A pseudo-class is used to define a special state of an element.

|  |  |  |
| --- | --- | --- |
| SELECTOR | EXAMPLE | EXAMPLE DESCRIPTION |
| :active | a:active | Selects the active link |
| :checked | a:checked | Selects every checked <input> element |
| :disabled | a:disabled | Selects every disabled <input> element |
| :empty | P:empty | Selects every <p> element that has no children |
| :enabled | Input: enabled | Selects every enabled <input> element |
| :first-child | P:first-child | Selects every <p> elements that is the first child of its parent |
| :first-of-child | P:first-of-child | Selects every <p> element that is the first <p> element of its parent |
| :focus | Input:focus | Selects the <input> element that has focus |
| :hover | Input: hover | Selects links on mouse |
| :invalid | Input:invalid | Selects <input> elements with an invalid value |
| :lang(language) | p:lang(it) | Selects every <p> element with a lang attribute values starting with “it” . |
| :last-child | P:last-child | Selects every <p> elements that is the element child of its parent |
| :link | a:link | Selects all unvisited |
| :not(selector) | :not(p) | Selects every element that is not <p> element |
| :nth-child(n) | P:nth-child(2) | Selects every <p> element that is the second child of its parent. |
| :nth-of-type(n) | P:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent , counting from the last child. |
| :only-of-type | P:only-of-type | Selects every <p> element that is the only <p> element of its parent. |
| :only-child | P:only-child | Selects every <P> element that is the only child of its parent. |
| :optional | Input:optional | Selects <input> element with no “required ” attribute |
| :out-of-range | Input: out-of-range | Selects <input> elements with a value outside a specified range. |
| :read-write | Input :read-write | Selects <input> elements with a “read only” attribute specified |
| :read-only | Input :read-only | Selects <input> elements with a “read only” attribute specified. |
| :required | Input : required | Selects <input> elements with a “required” specified |
| :root | Root | Selects the document’s root element |
| :target | #news :target | Selects the current active #news elements (clicked on a URL containing that anchor name) |
| :valid | Input: valid | Selects all <input> elements with a valid value. |
| :visited | Input :visited | Selects all visited links. |

PSEUDO ELEMENTS SELECTOR (::) :- Selects and style are applied a part of an element.

* Pseudo-elements can be combined with CSS classes.
* Images does not work before and after pseudo-elements.
* The ::before and ::after pseudo-elements can be used to insert some content before the content of an element.

|  |  |  |
| --- | --- | --- |
| SELECTOR | EXAMPLE | EXAMPLE DESCRIPTION |
| ::after | P::after | Insert something after the content of each <p> element. |
| ::before | P:: before | Insert something before the content of each <p> element. |
| ::first-letter | P::first-letter | Select the first letter of each <p> element. |
| ::first-line | P::first-line | Select the first line of each <p> element. |
| ::marker | ::marker | Select the markers of list items. |
| ::selection | ::selection | Selects the portion of an element by a user. |

ATTRIBUTE SELECTORS :- we are selecting the element by the attribute and its value. we are applying the style with attribute and value attribute of the tag within square brackets.

H1[align]{

Color:red;

}

H1[align = “center”]{

Color: blue;

}

</style>

SPECIFICITY :-If there are two or more conflicting CSS rules that point to the same element.

Inline style – An inline style is attached directly to the element to be styled .

IDs – An ID is a unique identifier for the page elements , such as #navbar.

Classes , attribute, and pseudo-classes -This category includes classes [attributes] and pseudo-classes such as :hover , :focus etc …

Elements and Pseudo-elements– This category includes element names and pseudo-elements ,such as h1 , div , :before and :after.

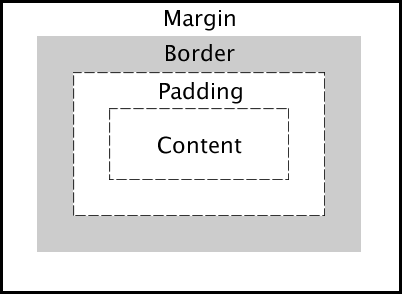
|  |  |
| --- | --- |
| Universal selector | 0 |
| Tag name | 1 |
| Class | 10 |
| id | 100 |
| Inline css | 1000 |

!important :- The !important rule in CSS is used to add more importance to a property / value than normal.

INHERITANCE :-The concept of the inheritance we can apply the css styles , it will apply   
to the parent whenever the child has no styles .If the child has styles it will be applied to this child .It is concept of inheritance in css.

LAST RULE :-If we can same css are apply to same selectors at the time .But the styles are applied based on the last rule. It complies styles line by line only.

CSS BOX MODEL :-



PADDING :- Padding is used to create space around an element’s content , inside of any defined borders. CSS has properties for specifying the padding for each of an element .

NOTE :-Negative values are not allowed.

* If the padding property has four values.
* Padding : top right bottom left
* If the padding property has three values.
* Padding : top right/left bottom
* If the padding property has two values.
* Padding : top/bottom right/left
* If the padding property has one values.
* padding : top/bottom/right/left

PADDING VS MARGIN :-

|  |  |
| --- | --- |
| The space between the content and border. | The space between the around of the border. |
| Negative values are not allowed | Negative values are allowed. |
| Padding has no auto value. | Margin has auto value. |
| Padding property accepts all the inline-elements. | Margin-top and margin-bottom not possible inline elements. |

BORDER :- The CSS border properties allow you to specify the style , width , and color of an element’s border.

|  |  |  |
| --- | --- | --- |
| Border-style | Border-top-style , Border-bottom-style , Border-right-style , Border-left-style | Dotted, dashed , solid , double , groove , ridge , inset , outset , non  e , hidden |
| Border-width | Border-top-width , Border-bottom-width , Border-right-width , Border-left-width | length , values |
| Border-color | Border-top-color , border-bottom-color , border-right-color , border-left-color | colors |
| Border-radius | Border-top-radius , Border-bottom-radius , Border-right-radius, Border-left-radius | values |

* The border property is a shorthand property for the following individuals border properties.

Border : border-width border-style border-color

MARGIN :- Margin are used to create space around element , outside of any defined borders. CSS has properties for specifying the margin for each side of an element.

NOTE :-Negative values are allowed.

* If the Margin property has four values.
* Margin : top right bottom left
* If the Margin property has three values.
* Margin : top right/left bottom
* If the Margin property has two values.
* Margin : top/bottom right/left
* If the Margin property has one values.
* Margin : top/bottom/right/left

OUTLINE :- An outline is a line drawn outside the element’s border. CSS has the following outline properties.

|  |  |
| --- | --- |
| Outline-style | Dotted ,dashed, solid, double, groove, ridge, inset, outset, none, hidden |
| Outline-color | Colors |
| outline-width | values |
| Outline-offset | The property adds space between an outline and the edge/border of an element. The space between an element and its outline transparent. |

TEXT STYLES :- The styles which are applied to the text.

|  |  |  |
| --- | --- | --- |
| PROPERTIES | EXPLANAITON | VALUES |
| Direction | Specifies the text direction / writing direction | Rtl , ltr , inherit |
| Unicode-bidi | Used together with the direction property to set or return whether the text should be overridden to support multiple language in th same document. | Bidi-override |
| Text-align | Specifies the horizontal alignment of text. | None, start, left, right, end, centre, justify |
| Vertical-align | The vertical-align property sets the vertical alignment of a center. | Baseline , text-top ,text-bottom , sub , super. |
| Text-align-last | The text-align-last property specifies how to align the last line of text. | Right , center , left , justify |
| Text-transform | Controls the capitalization of text. | None, uppercase, lowercase, capitalize |
| Letter-spacing | The letter-spacing property is used to specify the space between the characters in a text. | Normal, length, values |
| Text-indent | The property is used to specify the indentation of the first line of a text. | Length, value, auto |
| Line-height | The line-height property is used to specify the space between lines. | Normal values. |
| Word-spacing | The word-spacing property is used to specify the space between the word in a text. | Normal, values |
| White-space | The white-space property specifies how white-space inside an element is handled. | Normal, pre-wrap, pre-line, no-wrap, pre |
| cursor | Whenever the mouse is hover then the pointer is like hand symbol. | pointer |
| Text-overflow | Specifies how overflowed content that is not displayed should be signalled to the user. | Ellipsis , clip |
| Writing-mode | Specifies whether lines of text are laid out horizontally or vertically. | Horizontal-tb , vertical-rl |
| Text-decoration-line | The text-decoration-line property is used to add a decoration line to test. | Overline , line-through ,underline , overline underline |
| Text-decoration-color | The text-decoration-color property is used to set the color of the decoration line | color |
| Text-decoration-style | The text-decoration-style property is used to set the style of the decoration line | Solid , double , dotted , dashed , wavy |
| Text-decoration thickness | The text-decoration-thickness property is used to set thickness of the decoration line. | Auto , value |
| Text-decoration | shorthand property for text-decoration-line (required) , color(optional) , style(optional) , thickness(optional)  Text-decoration : line color style thickness. |  |

HEIGHTS AND WIDTH :-

|  |  |
| --- | --- |
| Property | Description |
| Height | Sets the height of an element. |
| Max-height | Sets the maximum height of an element. |
| Min-height | Sets the minimum height of an element. |
| Width | Sets the width of an element. |
| Max-width | Sets the maximum width of an element. |
| Min-width | Sets the minimum width of an element. |

COLORS :-

|  |  |
| --- | --- |
| COLOR NAMES | Colors are specified using predefined color name. |
| RGB VALUE | In css , a color can be specified an RGB value , using the formula. Each parameter defines the intensity of the color between 0 to 255.  Rgb(red , green , blue) 0 🡪 black and 255 🡪 white |
| RGBA value | RGBA color values are an extension of RGB color values with alpha channel which specifies the opacity for a color. The alpha parameter is a number between 0.0 to 1.0. |
| HEX value | In CSS , a color can be specified using a hexadecimal(0 to 9 & A-f) value in the form #rrggbb. Where the highest value is FF and the lowest value is 00. |
| HSL value | In CSS , a color can be specified using hue , saturation, and lightness (HSL) in the form . Hue is 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 full color. Lightness is also a percentage . 0% is black , 50% is neither light nor dark , 100% is white. |
| HSLA value | HSLA color value are an extension of HSL color values with alpha channel-which specifies the opacity for a color. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all ). |

BACKGROUNDS :- The CSS background properties are used to add background effects for elements.

|  |  |  |
| --- | --- | --- |
| Background-color | Sets the background color of an element. | Color |
| Background-image | Sets the background image for an element. | URL (“path”) |
| Background-repeat | Sets how a background image will be repeated. | No-repeat , repeat-x ,  Repeat-y , space , repeat (default). |
| Background-attachment | Sets whether a background image is fixed or scrolls with the rest of the page. | Fixed , scrolled(default). |
| Background-position | Sets the starting position of a background image. | Top , bottom , right , left , right top , left top , right bottom , left bottom , x axis and y axis. |
| Background-size | Sets the size of the background images. | Cover , auto , contain , length ,Width height, |
| Background-origin | Specifies where the background image is /positioned | Padding-box , border-box content-box |
| Background-clip | Specifies the painting area of the background. | Padding-box , border-box, content-box |

SHORTHAND :-

Background : color url repeat attachment position

FONT STYLES :- Th css styles which are applied to the text of the elements .

|  |  |  |
| --- | --- | --- |
| Font-family | Specifies the font family for text. | Serif , Sans-serif , Monospace , cursive , Fantasy etc… |
| Font-variant | Specifies whether or not a text should be displayed in a small-caps font. | Normal , small-caps |
| Font-weight | Specifies the weight of a font | Normal , bolder , bold , lighter , values ,100-900 values |
| Font-size | Specifies the font size of text. | Values , XX-small , X-small , small , XX-large , X-large , inherit , length |
| Font-style | Specifies the font style for text. | Normal , italic , oblique , inherit |
| font | Sets all the font properties in one declaration. | Font : font-style font -variant font-weight font-size/line-height font-family. |

LIST :-

|  |  |  |
| --- | --- | --- |
| List-style-type | Specifies the position of the list-item markers. | Circle , square , upper-roman , lower-alpha |
| List-style-image | Specifies an image as the list-item markers | URL(“path”) |
| List-style-position | Specifies the position list-item markers (bullet points) | Inside , outside |
| List-style | Sets all the properties for a list in one declaration | List-style : style position url(“ ’) |

TABLE :-

|  |  |  |
| --- | --- | --- |
| Caption-side | Specifies the placement of a table caption | Bottom , top , initial , inherit |
| Empty-cells | Specifies whether or not to display borders and background on the empty cells in a table. | Show , hide , initial , inherit |
| Table layout | Sets the layout algorithm to be used for a table. | Fixed , auto |

SHADOW EFFECTS :- The shadow effects will work on the based on x-axis and y-axis.

Text-shadow :-adds one or more shadow to an element.

Syntax :-

Text-shadow : x-axis y-axis blur spread radius color;

Box-shadow :- Adds one or more shadows to an element.

Syntax :-

Box-shadow : x-axis y-axis blur spread radius color;

GRADIENTS :- Combination of two or more colors is known as gradients. CSS defines two type of gradients .

* Linear Gradients (goes down/up/right/diagonally)
* Radial Gradients (defines by their center)

Linear Gradients :- Linear-gradients are used to mix multiple colors in linear direction.

SYNTAX :-

Background-image : linear-gradient (direction, color-stop1, color-stop2,…..)

Background-image : linear-gradient (angle , color-stop1, color-stop2 , ………)

Radial Gradients :-Radial gradients is defined by its center . To create gradient you must also define at least two colors.

SYNTAX :-

Background-image : radial-gradients (shape size at position , start-color ,…… , last-color)

BOX-SIZING :- Defines how the width and height of an element are calculated : should they include padding and border or not .

Box-sizing : Border-box

VISIBILITY :- It is also hidden an element. However the element will still take up the same space as before. The element will be hidden , but still affect the layout.

Visibility : Hidden and Visibility : Visible

OPACTIY : The opacity property specifies the opacity/transparency of an element. The opacity property can take a value from 0.0 -1.0 .The lower values, the more transparent.

Opacity : 0.0 -1.0

CSS FORMS :-The look of an HTML form can be greatly improved with CSS with the attribute of form.

|  |  |
| --- | --- |
| Input[type=text] | Will only select text fields. |
| Input[type=password] | Will only select password fields. |
| Input[type=number] | Will only select number fields. |

CSS FLEX BOX :-

* one dimension 🡪 row or column
* Display: flex , it will take block level elements.
* Display : inline-flex , it will take inline elements.
* In container flex is container but in items flex is a property.

PARENT ELEMENT (CONTAINER) :-

|  |  |  |
| --- | --- | --- |
| Flex-direction | Specifies the direction of the flexible items inside a flex container. | row (default) ,row-reverse , column , column-reverse. |
| Flex-wrap | Specifies whether the flex items should wrap or not , if there is not enough room for them on one flex line. | no wrap (default) , wrap ,  wrap-reverse |
| Justify-content | Horizontally aligns the flex items when the items do not use all available space on the main-axis. | Flex-start(default) , flex-end , center , space-between , space-evenly , space-around |
| Align-items | Vertically aligns the flex items when the items do not use all available space on the cross-axis. | Stretch (default) , center , flex-start ,flex-end ,baseline |
| Align-content | Modifies the behaviour of the flex-wrap property. It is similar to align-items , but instead of aligning flex-items , it aligns flex-lines. | Stretch (default) , center , flex-start ,flex-end ,space-around , space-between , space-evenly |
| Justify-items | The Justify-items property is used to align the flex items a vertically. | Stretch (default) , center , flex-start ,flex-end ,baseline |
| Flex-flow | A shorthand property of flex-direction and flex-wrap  Flex-flow : direction wrap |  |
| Row-gap | It is used to gap between the rows. | values |
| Column-gap | It is used to gap between the columns | Values |
| Gap | It is the shorthand property for the rows and columns | Row-gap column-gap |

MAIN-AXIS AND CROSS-AXIS :-

Flex-direction : row (default)

1.Main-axis 🡪

2.cross-axis

Flex-direction : column

1.Main-axis

CHILD ELEMENTS (ITEMS) :- The direct child elements of a flex container automatically becomes flexible(flex) items.

|  |  |
| --- | --- |
| Order | The order property specifies the order of the flex items. The order value must be a number , default value is 0. Here we can rearrange the order of the element and more order value of an item that will comes last and lesser value comes first. |
| Flex-grow | It gives the ability or access to one item to grow inside the container and take the empty spaces left .Negative value does not accept default value is 0. If we give one item more value, it will occupy more space according to less value items. |
| Flex-shrink | By using flex shrink property, we can decide one item will shrink or not in our container, default value is 1 . We  cannot take negative value. If we are taking flex-shrink 0 now the item will not shrink with initial width. |
| Align-self | The align-self property overrides the align-items property value for specific item. Properties like start / center /end /stretch /baseline |
| Flex-basis | The flex-basis property specifies the initial width of a flex item. |
| flex | Grow shrink basis |

MEDIA QUERIES :- Media Queries are used make screen size responsive . We have 4 type points (prints) , screen (every digital screen) , speech(all speech devices) , all( for all types of devices).

Smart phone : @media screen and (max-width :767px){

}

Tablet :@media screen and (min-width:767px) and (max-width :1023px){

}

Desktop : @media screen and (min-width : 1024px){

}

Z-INDEX :- Using z-index property we can specify the stacking order the elements. An element with greater stack order is always in front of an element with lower stack order. Z-index only works on positioned elements only. It will work for negative or positive values.

OVERFLOW :- The property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.

|  |  |  |
| --- | --- | --- |
| Overflow | Specifies what happens if content overflows an element’s box. | Visible(default) , hidden , scroll , auto |
| Overflow -wrap | Specifies whether or not the browser can break lines with along words , if they overflow its container. | Break-word |
| Overflow-x | Specifies what to with the left /right edges of the content if it overflows the element ‘s content area. | Hidden , scroll |
| Overflow-y | Specifies what to do with the top/bottom edges of the content if in overflows the element | Hidden , scroll |

CSS POSITION :- The CSS position property specifies the positioning of an element it seems the authority to change the original position of an element. For changing the position we have four additional properties such as left , right , bottom , top . These four properties behaves differently for different position value. We have five types of position value.

1.poisiton : static 🡪 These is the default position value for every element present in our html document. In position static we can’t change for one element’s position i.e. left , right , bottom, and top properties will not work for static position element.

2.position:relative 🡪 Using position relative we can be able to move an element inside our web page i.e. top , right , bottom , left properties will work. In position relative the element is relative to its original origin.

3.position:fixed 🡪 Using position fixed to one element we can fixed the element in our webpage view port.

4.position:sticky 🡪 In position sticky ,the element will not change its original position but when we scroll the element at first it will scroll then it’s reached its left , right , bottom , top value it will east there. Position sticky is combination of position relative and position fixed.

5.position: obsolute 🡪 position obsolute we need parent and child relationship between elements. If we provide position obsolute to one element it first search for immediate coming relative or fixed parent. We need to provide the child position obsolute and parent element position relative or fixed . Now child element can move within the parent. If there is no parent relative presented it assume the body tag as it parent .

DISPLAY :-Default display property decides how to represent/display one element inside the web page. We have two types of values.

Inline-block : It’s formatted just like the inline element where it doesn’t start a new line. But you can set width and height values.

BLOCK-LEVEL ELEMENTS :- A block-level element always starts on a new line. A block-level element always takes up the full width available. A block-level element has a top and bottom margin whereas an inline element does not.

|  |  |  |  |
| --- | --- | --- | --- |
| <address> | <main> | <noscript> | <blockquote> |
| <aside> | <hr> | <p> | <dd> |
| <canvas> | <h1>--<h6> | <section> | <dl> |
| <div> | <footer> | <video> | <fieldset> |
| <dt> | <figcaption> | <article> | <form> |
| <li> | <pre> | <table> |  |
| <nav> | <ol> | <ul> |  |

INLINE ELEMENTS :- An inline element does not start on a new line .A line element only takes up as much as width necessary.

|  |  |  |  |
| --- | --- | --- | --- |
| <a> | <dfn> | <sup> | <time> |
| <acronym> | <i> | <strong> | <var> |
| <bdo> | <input> | <small> | <abbr> |
| <br> | <label> | <script> | <b> |
| <cite> | <object> | <q> | <big> |
| <code> | <button> | <em> | <img> |
| <map> | <kdb> | <output> | <samp> |
| <select> | <span> | <sub> | <textarea> |
| <tt> |  |  |  |

None : Removes the element from our webpage. When one element is removed using display property and other elements takes its position.

NOTE :- Block -level elements are used to center based on the using Margin of the elements.

|  |  |  |
| --- | --- | --- |
| Block | inline | Inline-block |
| Block elements are laid out vertically by default. | Inline elements are laid out horizontally by default. | Inline-block makes a block elements are laid out horizontally and Inline elements are laid out vertically. |
| Block level elements are starts with new line | Inline elements does not start with new inline | It’s just like inline element where it doesn’t start on a new line . But you set width and height values |
| It takes width the entry available. | It takes width as the element available. |  |
| You can set the width and height. | You can’t set width and height. |  |

TRANSFORM :- Applies a 2D or 3D transformation to an element. It transfer element from one position to another position. In CSS , for changing one element static position we are apply transform property.

|  |  |  |  |
| --- | --- | --- | --- |
| Property | 1D | 2D | 3D |
| translate | translateX()  translateY()  translateZ() | Translate(x-axis , y-axis) | Translate3d (x,y,z) |
| rotate | rotateX(deg)  rotateY(deg)  rotateZ(deg) | Rotate(deg) | Rotate3d (x,y,z, deg) here x,y,z values are 1:2:3 ratio |
| Scale | ScaleX()  ScaleY()  ScaleZ() | Scale(x,y) , Here it will 1 means 100% value. | Scale3D(x,y,z) by default it will have 1 numeric value . |
| Skew | SkewX(deg)  SkewY(deg) | Skew(deg) |  |

TRANSFORM ORGIN :- right , left ,center , x-axis , y-axis ;

PERSPECTIVE :- In transform perfect text of an element. It specifies the distance between object and the observer is known perspective.

TRANSITION :- Transition allows you to change property value smoothly over given duration.

|  |  |
| --- | --- |
| Transition-property | Specifies the name of the CSS property the transition effects. |
| Transition-duration | Specifies how many seconds or milli seconds a transition effects a delay for the complete. |
| Transition-delay | Specifies a delay for the transition. |
| Transition-timing-function | Specifies the speed curve of the transition effects.  Ease = slow start, fast , slow end  Linear = same speed  Ease-in =slow start  Ease-out = slow end  Ease-in-out = slow start , fast , slow end |
| Transition | Property duration timing-function delay |

ANIMATIONS :- continues change of element form one state another state. In css we for animation we are using special selector @keyfarmes

|  |  |  |
| --- | --- | --- |
| Animation-name | Specifies the name of the @keyframes animation | identifiers |
| Animation-delay | Specifies a delay for the start of an animation | Time |
| Animation-direction | Specifies whether an animation should be played forwards , backwards or in alternate cycles | Normal ,alternative , reverse |
| Animation-duration | Specifies how long time an animation should take to complete one cycle | time |
| Animation-fill-mode | Specifies a style for the element when the animation is not playing(before it starts , after it ends , or both) | None , forwards backwards , both |
| Animation-iteration-count | Specifies the number times an animation should be played. | infinite , numbers |
| Animation-play-state | Specifies whether the animation is running or paused. | Running , paused |
| Animation-timing-function | Specifies the speed curve of the animation | Ease, linear, ease-in, ease-out, ease-in-out |
| Animation | A shorthand property for setting all the animation properties . | Name duration timing delay count direction |

UNITS :- CSS has several different units for expressing a relative , absolute , frequency , angles, and Time.

Time :-

|  |  |
| --- | --- |
| Time | Timing actions |
| Ms | Milliseconds |
| S | Seconds |

Frequency :-

|  |  |
| --- | --- |
| HZ | Hertz |
| KHZ | Kilo-hertz |

Absolute Measurement :-

|  |  |
| --- | --- |
| Absolute measurements | It will works based on the parent. |
| Cm | Centimetres |
| Mm | Millimetres |
| Pc | Pica (1pc = 12 points) |
| In | Inch (1in = 96px = 2.54cm) |
| pt | Point (1pt = 1/72 inch) |

Relative Measurement :-

|  |  |
| --- | --- |
| Relative measurements | It will work on the roots root element of the html. |
| ch | Width of the “0” glyph found in the font for the font size used to render |
| Em | 1em = current font size of current element (1em =16px default) |
| Ex | x-height of the element’s font |
| Gd | The grid defined by ‘layout-grid’ |
| Px | Pixel of the viewing device |
| Rem | The font size of the root element |
| Vh | The viewport’s height |
| Vw | The viewport’s width |
| vm | View port’s height or width whichever is smaller of the two. |
| % | Relative to the parent |
| Fr | It will take fraction of available space. |
| Auto | It will take automatically remaining width and height which is available |

Angles :-

|  |  |
| --- | --- |
| Angles | It will denote the angle of the styles |
| Deg | degrees |
| Grad | Grads |
| Rad | Radians |
| turn | Turns |

CSS GRID :- Display grid is two dimensional layout . It will work both row and column. Display grid is combination of table property ,media queries property , float property and position property

* Display : grid , make a block level grid container.
* Display : inline-grid make an inline level grid container.

CONTAINER:-

|  |  |
| --- | --- |
| Grid-template-columns | Specifies the size of the columns and how many columns in a grid items. |
| Grid-template-rows | Specifies the size of the rows in a grid layout. |
| Grid-row-gap | Specifies the size of the gap between rows. |
| Grid-column-gap | Specifies the size of the gap between columns |
| Grid-gap | A shorthand property for the grid-row-gap and grid-column-gap properties. |
| Row-gap | Specifies the gap between grid rows. |
| Column-gap | Specifies the gap between grid columns. |
| gap | A shorthand property for the row-gap and column-gap properties. |
| Grid-column-start | Specifies where to start the grid item. |
| Grid-column-end | Specifies where to end the grid item. |
| Grid-row-start | Specifies where to start the grid item. |
| Grid-row-end | Specifies where to end the grid item. |
| Grid-row (start/end) | A shorthand property for the grid-row-start and grid-row-end properties. |
| Grid-column(start/end) | A shorthand property for the grid-column-start and grid-column-end properties. |
| Grid-template-areas | Specifies how to display columns and rows using grid items. |
| Gird-area | Specifies a name for the area or this property is a shorthand property for the grid-row-start , grid-column-start , grid-row-end , grid-column-end properties. |
| order | The order property is used for the child and default order is 0 . |

IMPLICIT GRID :- It will takes automatically based on the grid default values .

REPEAT :- The repeat function is used to repeat the columns or rows . It is used repeatedly same value we can repeat function

Syntax :- repeat (times, value).

|  |  |  |
| --- | --- | --- |
| Justify-content | The justify-content property is used to align the flex items a horizontally center . | Flex-start(default) , flex-end , center , space-between , space-evenly , space-around |
| Align-items | The align-items property is used to align the flex items a vertically . | Stretch (default) , center , flex-start ,flex-end ,baseline |
| Align-content | The align-content property is used to align the flex lines as vertically . | Stretch (default) , center , flex-start ,flex-end ,space-around , space-between , space-evenly |
| Justify-items | The Justify-items property is used to align the flex items a horizontally. | Stretch (default) , center , flex-start ,flex-end ,baseline |
| Align-self | The align-self property specifies the alignment for the selected items inside the flexible container(vertically). | Flex-start , flex-end , center |
| Justify-self | The Justify-self property specifies the alignment for the selected items inside the flexible container(Horizontally). | Flex-start , flex-end ,center |

Minmax() :-It will take exactly value based on min-value and max-value.

Syntax :- Minmax(minimum-value, maximum-value)

Calc() :- Allows you to perform calculations to determine CSS property values.

Auto-fill and Auto-Fit :- auto-fill fills the row with as many columns as it can fit. So it creates implicit columns whenever a new column can fit , because it’s trying to fill the row with as many columns as it can. Auto-fir fits the currently available columns into the space by expanding them so they take up any available space.

ROOT ELEMENTS :-

The root element in the html style or common styles. We can access them hole web pages.

:root {

styles

}

We can arrange the values based on the root element. We can change them based on the root element.

VARIBALES CSS :-

CSS variables are also known as custom property.

Syntax :-

--VarName :value

property : var(--VarName)

Scope of variables :-

:root{ } 🡺global

Element 🡺local

🡪If we can create the variable in the parent and applied to child anywhere.

🡪If we can create the variable in the global and applied to hole program of webpages.

|  |  |
| --- | --- |
| Name | Required . The variable name(must start with two dashes) |
| value | Optional . The fallback value (used if variable is not found) |

CSS COLUMN PROPERTIES :-

|  |  |
| --- | --- |
| Column-count | The column-count property specifies the number of columns an element should be divide into. |
| Column-gap | The column-gap property specifies the gap between the columns. |
| Column-rule-style | The column-rule-style property specifies the style of the rule between columns. |
| Column-rule-width | The column-rule-width property specifies the width of the rule between columns. |
| Column-rule-color | The column-r  ule-color property specifies the color of the rule between columns. |
| Column-rule | Width style color |
| Column-width | Specifies a suggested , optimal width for the columns |
| Column-span | Specifies how many columns an element should span across. |
| columns | Width count |

CSS LAYOUT -FLOAT AND CLEAR :-

* The CSS Float property specifies how an element should be float.
* The CSS Clear property specifies what element should be float beside the cleared element and on which side.

1. FLOAT PROPERTY :- The float property is used for positioning and formatting content. The float property can have one of the following values:

* Left 🡪 The element float to the left of its container.
* Right 🡪 The element float to the right of its container.
* None 🡪 The element does not float (will be displayed just where it occurs in the text ).
* Inherit 🡪 The element inherits the float value of its parent.

1. CLEAR PROPERTY :- When we use the float property and we want the nest element below (not on right or left), we will have used the clear property. The clear property specifies what should happen with the element that is next to a floating element. The clear property can have one the following values:

* None 🡪 The element is not pushed below left or right floated elements . This is default
* Left 🡪 The element is pushed below left floated elements.
* Right 🡪 The element is pushed below right floated elements.
* Both 🡪 The element is pushed below both left and right floated elements.
* Inherit 🡪 The element inherits the clear value from its parent.

OBJECT-FIT :- The CSS object-fit property is used to specified how an <img> or <video> should be resized to fit its container. The object-fit property can take one if the following values.

|  |  |
| --- | --- |
| fill | This is default. The image is resized to fill the given dimension. If necessary , the image will be stretched or squished to fit. |
| contain | The image keeps its aspect ratio but is resized to fit within the given dimension/ |
| cover | The image keeps its aspect ratio and fills the given dimension. The image will be clipped to fit. |
| none | The image is not resized. |
| Scale-down | The image is scaled down to smallest version of none or contain. |

OBJECT-POSITION PROPERTY :- The CSS object-position property is used to specify how an <img> or <video> should be positioned within its container based on the x/y coordinates inside its “own content box”.

CSS RESIZE :- The resize property specifies if an element should be resizable by the user . The resize property has four values such as horizontal , vertical , both and none.