Cascaded Style Sheets

anand.jain@pccoepune.org
anandkumar.training@gmail.com

What is CSS?

- Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

- CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects.

- CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Why CSS?

- **CSS SAVE TIME**
  - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **PAGES LOAD FASTER**
  - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply to all the occurrences of that tag. So less code means faster download times.
- **EASY MAINTENANCE**
  - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

Why CSS?

- **SUPERIOR STYLES TO HTML**
  - CSS has a much wider array of attributes than HTML so you can give far better look to your HTML page in comparison of HTML attributes.
- **MULTIPLE DEVICE COMPATIBILITY**
  - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **GLOBAL WEB STANDARDS**
  - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Who Creates and Maintain CSS?

- CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by W3C members, it becomes a recommendation.

- These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

- NOTE: The World Wide Web Association, or W3C is a group that makes recommendations about how the Internet works and how it should evolve.

CSS Types (How to use CSS?)

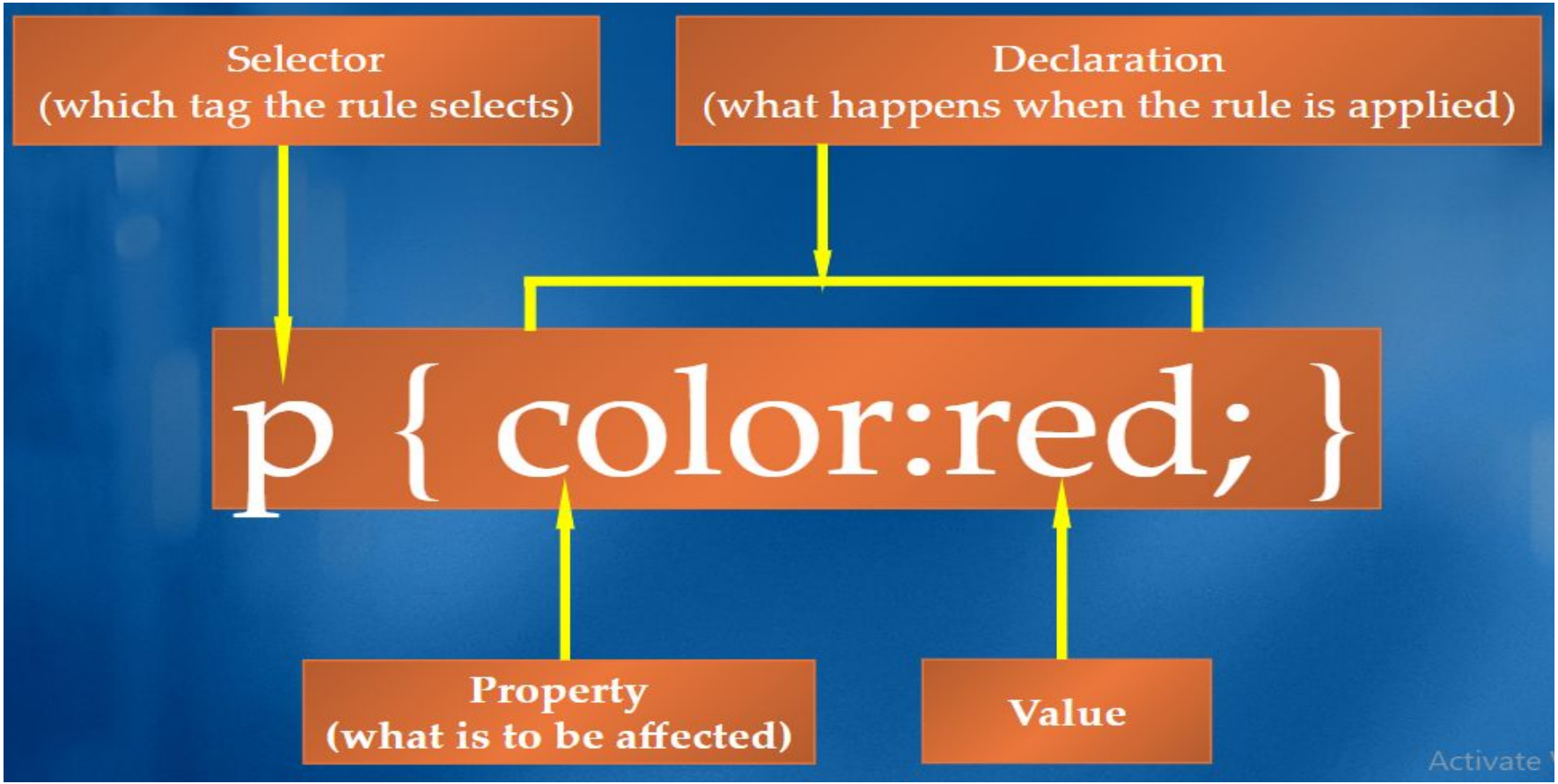
- External / Linked Style
  - Separates presentational styles from structural markup
  - How to?
    - Create a .css file
    - Create a .html file
    - Link .css to .html using link tag
  - The <link> Tag
    - Written inside <head>
    - Attributes
      - href = “nameOfCSSFile.css”
      - media = “screen | print | projection | handheld | all”
      - rel = “stylesheet” (Specifies relation between current document and linked document)
      - type = “text/css” (Specifies the type of linked document)
  - E. g. : <link href="external.css" media="screen" rel="stylesheet" type="text/css" />

CSS Types (How to use CSS?)

- Embedded Style / Internal Style
  - The scope is limited to the page in which it is defined.
  - Overrides external styles.
  - Written inside <head> using <style> tag.
  - The <style> Tag
    - Used to define style information for an HTML document.
    - Attributes
      - type = “text/css”

- Inline Style
  - Added to the tags using style attribute
  - overrides external and internal styles
  - Scope is restricted to tag only.
  - Ex. : —---

CSS Rule


Selector
(which tag the rule selects)
Declaration
(what happens when the rule is applied)
p { color:red; }
Property
(what is to be affected)
Value

CSS Selectors

- CSS selectors are used to "find" (or select) the HTML elements you want to style.

- **The Element Selector** : The element selector selects HTML elements based on the element name.
  - p {color:red; }

- **The CSS id Selector** : The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by the id of the element.
  - Example
    Create ID Selector: **#para1 {color:red; text-align:center; }**
    Apply ID Selector : ** ..... **

CSS Selectors

- **The CSS class Selector :** The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.
  - Example
    Create Class Selector: **.para1 {color:red; text-align:center; }**
    Apply class Selector : ** ..... **

- **The CSS Universal Selector :** The universal selector (*) selects all HTML elements on the page.
  - Example : *** {color:red; text-align:center; }**

- **The CSS Grouping Selector :** The grouping selector selects all the HTML elements with the same style definitions. To group selectors, separate each selector with a comma.

```
h1, h2, p
{
  text-align: center;
  color: red;
}
```

CSS Selectors

Pseudo Classes : A CSS pseudo-class is a keyword added to a selector that specifies a special state of the selected element(s). For example, the pseudo-class :hover can be used to select a button when a user's pointer hovers over the button and this selected button can then be styled.

selector:pseudo {.....}

Pseudo Class Name	Purpose
:link	Used to apply style to an unvisited link
:visited	Used to apply style to a visited link
:hover	Used to apply style to an element when mouse is moved over to the element
:active	Used to apply style to an element when it is active
:focus	Used to apply style when element has focus
:checked	Used to apply style to every checked input element
:disabled	Used to apply style to every disabled input element
:enabled	Used to apply style to every enabled input element

CSS Units

We use length in padding, margin, font-size, height, width etc
Two types of Units : 1) Absolute 2) Relative

Absolute Unit In CSS	Relative Unit In CSS
1. Cm	1. Em
2. Mm	2. Rem
3. Inch	3. Vh
4. Px	4. Vw
5. Pt	5. %
6. Pc	

CSS BOX MODEL

- All HTML elements can be considered as boxes.
- The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

CSS Properties

Color and Background

Name of Property	Values	Purpose
color	• colorName • hex color code • rgb(int, int, int) • rgba(int, int, int, opacity (in float)) • Hue, Saturation, Light : hsl(0 to 360, 0 to 100%, 0 to 100%) • transparent	Apply forecolor
background-color		Apply background color
background-image	• url('Image URL or Path') • none • conic-gradient(), linear-gradient(), radial-gradient(), repeating-conic-gradient(), repeating-linear-gradient(), repeating-radial-gradient()	Used to sets one or more background images for an element. By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

Color and Background

Name of Property	Values	Purpose
background-repeat	• repeat, no-repeat, repeat-x, repeat-y • space • round	Used to set if/how a background image will be repeated. By default, a background-image is repeated both vertically and horizontally.
background-attachment	• scroll • fixed	The background-attachment property sets whether a background image scrolls with the rest of the page, or is fixed.
background-blend-mode	normal, multiply, screen, overlay, darken, lighten, color-dodge, saturation, color, luminosity	The background-blend-mode property defines the blending mode of each background layer (color and/or image).
background-clip	• border-box • padding-box • content-box	The background-clip property defines how far the background (color or image) should extend within an element.

Color and Background

Name of Property	Values	Purpose
background-origin	• border-box • padding-box • content-box	The background-origin property specifies the origin position (the background positioning area) of a background image.
background-position	• x% y% • x_px y_px • x_keyword y_keyword	The background-position property sets the starting position of a background image.
background-position-x	• x% • x_px • x_keyword	The background-position-x property sets the position of a background image on the x-axis.
background-position-y	• y% • y_px • y_keyword	The background-position-y property sets the position of a background image on the y-axis.

Background

background-size : The background-size property specifies the size of the background images.

Value	Description
auto	Default value. The background image is displayed in its original size
length	Sets the width and height of the background image. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto".
percentage	Sets the width and height of the background image in percent of the parent element. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"
cover	Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges
contain	Resize the background image to make sure the image is fully visible

https://www.w3schools.com/cssref/playdemo.php?filename=playcss_background-clip

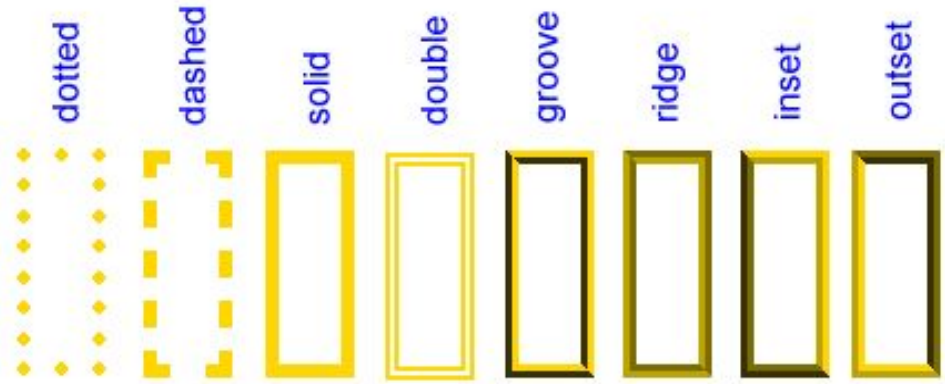
Margin and Padding Properties

- Margin is the distance between the element and the neighboring element.
- Properties
  - margin-top (e.g. : margin-top:10px;)
  - margin-right
  - margin-bottom
  - margin-left
  - margin (e.g.:margin:10px 10px 20px 5px) (top, right, bottom, left)
  - margin:10px □ 10px margin from all sides
- Padding is the inner distance between border of the element and content of the element.
- Properties
  - padding-top (e.g. : padding-top:10px;)
  - padding-right
  - padding-bottom
  - padding-left
  - padding (e.g.: padding:10px 10px 20px 5px) (top, right, bottom, left)
  - padding:10px □ 10px padding from all sides

Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins. This does not happen on left and right margins! Only top and bottom margins!

Border Properties

- border-width
- border-top-width
- border-left-width
- border-right-width
- border-bottom-width
- writing border-width:10px will set thickness of all the borders to 10px.



- border-color
- border-top-color
- border-left-color
- border-right-color
- border-bottom-color
- writing **border-color:red** will set all borders to red color.

- border-style
- border-style-top
- border-style-left
- border-style-right
- border-style-bottom
- writing **border-style:solid** will make all borders solid.

border:width style color
e.g. border: 2px solid blue;

Text

Name of Property	Values	Purpose
color		
direction	• ltr • rtl	The direction property specifies the text direction/writing direction within a block-level element.
text-align	• left • right • center • justify • text-align-last	
text-decoration	text-decoration-line text-decoration-color text-decoration-style text-decoration-thickness	none|underline|overline|line-through solid|double|dotted|dashed|wavy

Text

Name of Property	Values	Purpose
text-transform	lowercase uppercase capitalize none	
text-indent	px (or other units)	
word-spacing	px (or other units)	
letter-spacing	px (or other units)	
line-height	px (or other units)	
text-shadow	x y blur color	

Font

- font-family
- font-size : px | pt | em | rem | vw | %
- font-style : normal | italic | oblique

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

Display

Values of display Property

- none
- inline
- inline-block
- block
- grid
- flex

Flexbox

- flex-direction : row | row-reverse | column | column-reverse
- flex-wrap : wrap | nowrap | wrap-reverse
- flex-flow : flex-direction flex-wrap
- justify-content : flex-start | flex-end | center | space-around | space-between (Horizontal Alignment)
- align-items : stretch | center | flex-start | flex-end (Vertical Alignment)
- align-contents : center | space-between | space-around | stretch | flex-start | flex-end (Flex Line Alignment)

Position

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

- 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.

- 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. The top, right, bottom, and left properties are used to position the element.

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed). However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

- An element with position: sticky; is positioned based on the user's scroll position. A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

List

Name of Property	Values	Purpose
list-style-image	• url(‘imageUrl’) • none	The list-style-image property replaces the list-item marker with an image. Always specify the list-style-type property in addition. This property is used if the image for some reason is unavailable.
list-style-position	• inside • outside	The list-style-position property specifies the position of the list-item markers (bullet points).
list-style-type	none, disc, square, circle, decimal, decimal-leading-zero, lower-alpha, lower-roman, upper-alpha, upper-roman	The list-style-type specifies the type of list-item marker in a list.
list-style	• type position image	The list-style property is a shorthand for the following properties: list-style-type, list-style-position, list-style-image. If one of the values are missing, the default value for that property will be used.

Table

Name of Property	Values	Purpose
border-collapse	• collapse	The border-collapse property sets whether the table borders should be collapsed into a single border.
border-spacing	• x-length y-length	The border-spacing property sets the distance between the borders of adjacent cells.
caption-side	• top • bottom	The caption-side property specifies the placement of a table caption.
empty-cells	• show • hide	The empty-cells property sets whether or not to display borders on empty cells in a table.
table-layout	• fixed • auto	The table-layout property defines the algorithm used to lay out table cells, rows, and columns. The main benefit of table-layout: fixed; is that the table renders much faster. On large tables, users will not see any part of the table until the browser has rendered the whole table. So, if you use table-layout: fixed, users will see the top of the table while the browser loads and renders rest of the table. This gives the impression that the page loads a lot quicker!

CSS Pagination

- CSS pagination is a very useful technique for indexing different pages of a website on the homepage. If your website has lots of pages, you have to add some sort of pagination to each page.

- display:inline-block
- border-radius
- transition
- background-color

CSS Multiple Columns

- The basic idea of multicol, is that you can take a chunk of content and flow it into multiple columns, as in a newspaper. It doesn't matter which elements are inside the content that you turn into a multicol container, everything remains in normal flow, but broken into columns.

column-count	Specifies the number of columns an element should be divided into
column-fill	Specifies how to fill columns
column-gap	Specifies the gap between the columns
column-rule	A shorthand property for setting all the column-rule-* properties
column-rule-color	Specifies the color of the rule between columns
column-rule-style	Specifies the style of the rule between columns
column-rule-width	Specifies the width of the rule between columns
column-span	Specifies how many columns an element should span across
column-width	Specifies a suggested, optimal width for the columns
columns	A shorthand property for setting column-width and column-count

Gradients

- CSS gradient is used to display smooth transition within two or more specified colors.

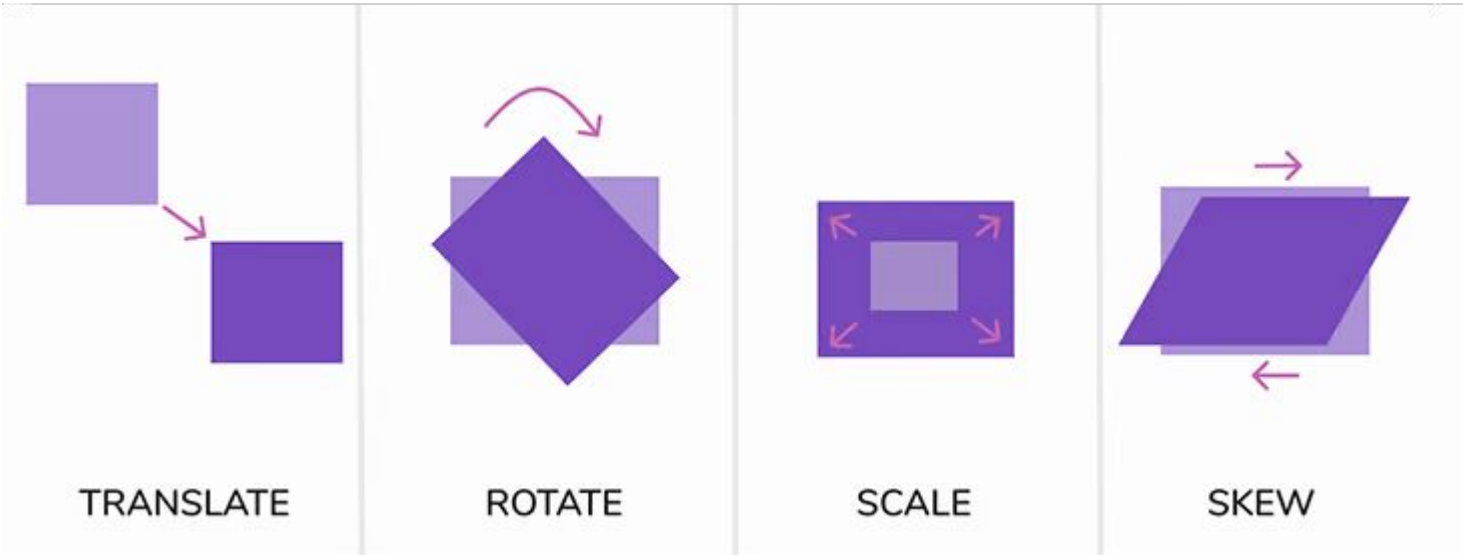
- Linear Gradient
  - The CSS3 linear gradient goes up/down/left/right and diagonally.
  - background-image: linear-gradient (direction, color-stop1, color-stop2.....);
  - Direction = to left, to right, to op, to bottom, to bottom right…
  - Angles can be used instead of direction
    - Top = 0deg
    - Right = 90deg
    - Bottom = 180deg
  - Use rgba() to specify color stops along with transparency
  - repeating-linear-gradient(red, yellow 10%, green 20%);

Gradients

- Radial Gradient
  - A radial gradient is defined by its center. To create a radial gradient you must also define at least two color stops.
  - background-image: radial-gradient(shape size at position, start-color, ..., last-color);
  - By default, shape is ellipse, size is farthest-corner, and position is center.
  - The size parameter defines the size of the gradient. It can take four values:
  - closest-side
    - farthest-side
    - closest-corner
    - Farthest-corner
  - repeating-radial-gradient(red, yellow 10%, green 15%);

2D Transform

- The CSS3 2D transform property allows us to scale, skew, move, translate, and rotate HTML elements. It transforms the element without affecting other elements on the page; in other words, it does not cause other elements on the page to shift and instead gets overlapped over them.



2D Transform

- The **transform** property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew, etc., elements.
- Syntax : transform: transform-function()
- Translate : It moves an element from its current position according to the given x and y-axis values.
- Rotate : It rotates an element clockwise or counter-clockwise according to a given argument.
- Scale : It increases or decreases the size of an element according to the given parameter.
- Skew : It skews an element according to the given parameter.

- **transform-origin** : The transform-origin property sets the origin for an element's transformations.
- Syntax : transform-origin: x-axis y-axis z-axis (left, center, right, top, bottom, px, %)

2D Transform Functions

translate(*x*,*y*)	translation, moving the element along the X- and the Y-axis
translateX(*n*)	translation, moving the element along the X-axis
translateY(*n*)	translation, moving the element along the Y-axis
scale(*x*,*y*)	scale transformation, changing the elements width and height
scaleX(*n*)	scale transformation, changing the element's width
scaleY(*n*)	scale transformation, changing the element's height

2D Transform Functions

rotate(*angle*)	rotation, the angle is specified in the parameter
skew(*x-angle,y-angle*)	skew transformation along the X- and the Y-axis
skewX(*angle*)	skew transformation along the X-axis
skewY(*angle*)	skew transformation along the Y-axis
matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())	transformation, using a matrix of six values

3D Transform Functions and Properties

translate3d(*x,y,z*)	It specifies a 3D translation.
scale3d(x,y,z)	It specifies 3D scale transformation
rotate3d(X,Y,Z,angle)	It specifies 3D rotation along with X-axis, Y-axis and Z-axis.
perspective(n)	It specifies a perspective view for a 3D transformed element.

The **perspective** property is used to give a 3D-positioned element some perspective. The perspective property defines how far the object is away from the user. So, a lower value will result in a more intensive 3D effect than a higher value. When defining the perspective property for an element, it is the CHILD elements that get the perspective view, NOT the element itself.

Transition

- The CSS transitions are effects that are added to change the element gradually from one style to another, without using flash or JavaScript.
- To create a transition effect, you must specify two things: 1) the CSS property you want to add an effect to 2) the duration of the effect

- Properties
  - transition : shorthand property (property duration timing-function delay)
  - transition-delay : specifies a delay (in seconds) for the transition effect.
  - transition-duration : specifies how many seconds (s) or milliseconds (ms) a transition effect takes to complete.
  - transition-property : specifies the name of the CSS property the transition effect is for (the transition effect will start when the specified CSS property changes). Default value : all
  - transition-timing-function : specifies the speed curve of the transition effect

Transition Timing

- transition-timing-function:
  linear | ease | ease-in | ease-out | ease-in-out | step-start | step-end | steps(int,start | end) | cubic-bezier(n,n,n,n)
- ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
- linear - specifies a transition effect with the same speed from start to end
- ease-in - specifies a transition effect with a slow start
- ease-out - specifies a transition effect with a slow end
- ease-in-out - specifies a transition effect with a slow start and end
- cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

Ref : https://www.w3schools.com/cssref/css3_pr_transition-timing-function.php

Animation

- CSS animations make it possible to animate transitions from one CSS style configuration to another. Animations consist of two components, a style describing the CSS animation and a set of keyframes that indicate the start and end states of the animation's style, as well as possible intermediate waypoints.
- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
- To get an animation to work, you must bind the animation to an element.
- Properties
  - animation : shorthand (animation: name duration timing-function delay iteration-count direction fill-mode play-state;)
  - animation-delay : specifies a delay for the start of an animation
  - animation-direction : normal | reverse | alternate | alternate-reverse
  - animation-duration : defines how long an animation should take to complete one cycle
  - animation-fill-mode : property 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

- Properties
  - animation-iteration-count : property specifies the number of times an animation should be played (integer, infinite)
  - animation-name : specifies a name for the @keyframes animation
  - animation-play-state : specifies whether the animation is running or paused (paused | running)
  - The **animation-timing-function** specifies the speed curve of an animation. The speed curve defines the TIME an animation uses to change from one set of CSS styles to another. The speed curve is used to make the changes smoothly.
  - **animation-timing-function:**
    linear|ease|ease-in|ease-out|ease-in-out|step-start|step-end|steps(int,start|end)|cubic-bezier(n,n,n,n)