CSS:

Cascading style sheets used to design the layouts and give the style property to the html document.

CSS was invented by hakon wium lee in 1994 but actual version of css was released in 1996 by both hakon wium lee and bert bos

Previously the CSS webpages are layout by using tables.

Basics:

A Selector is an any html tag at which a style will be applied.

A Property is an type of attribute of the html tag.

A Values are assign to the properties.

Syntax: Selector{
Property : value;
}

Ex: <h1>Hello</h1>

```
h1{
color : red:
}
```

How to add a CSS into your html document:

CSS Can be added into a html document by 3 ways

*Inline CSS

*Internal CSS

*External CSS

Inline CSS:

Inline CSS is a way of defining the styling of an html element by adding CSS rules directly to the elements tag using the style attribute.

Inline CSS has a first preferences among the 3 attributes.

Ex: <h1 style="color : red;">Hello</h1>

Internal CSS:

Internal CSS is also known as Embedded Style. It's a method of adding CSS rules directly to the head section of an html document.

It uses <style> tag to enclose CSS code within the html document.

Internal CSS is mandatorily placed inside the header part of the html document.

Ex:

```
<head>
<style>
h1{
color: red;
}
</style>
</head>
```

External CSS:

External CSS used to place CSS code in a separate file and link that file in the html document by using <link> tag at the head section.

We have to create an external style sheet with the .CSS extension.

Ex:

```
<head>
<link rel: "stylesheet" href="./external style sheet name with extension/.">
</head>
```

How to commenting in CSS: /* Statements….*/

Selectors:

Selectors in CSS is used to select the html element for styling purpose.

Selectors are classified into,

*Simple Selectors
*Combinator Selectors
*Psuedo-Class Selectors
*Psuedo-Element Selectors
*Attribute Selectors

Simple Selectors:

Simple Selectors will select html elements based on id, class, element, etc.,

Simple selectors are categorized into,

*Element selector
*Id Selector
*Class Selector
*Universal Selector
*Group Selector

Element Selector:

The Element Selector select the html elements based on the element name or tag name.

Ex: <h1><p><img>

(i) h1{
Statements……
}

(ii) p{
Statements…..
}

Id Selectors (#):

The Id Selector uses the id attribute of an html element to select a specific of an element.

Note: A id of an element is unique on the page or entire document.

The id attribute should define with # symbol.

Ex:

<p id="demo">

CSS Ex:

#demo{
Statements…..
}

Class Selectors(.):

Class Selectors denoted with (.) operator selects the html element with the specific class attribute.

Note: We can take multiple class names for the single class.

Class attribute should define with . Symbol

Ex:

<p class="test">

CSS Ex:

.test{
Statements…..
}

Universal Selectors(*):

Universal Selector is denoted with * symbol is used to select all the elements in the html document.

CSS Ex:

```
*{
Statements…..
}
```

Group Selectors(,):

Group Selectors is used to style the elements in the html document by using (,) operator which Given a same style property for all the selected elements.

CSS Ex:

```
h1,p1,div,span{
Statements…..
}
```

Combinator Selectors:

*Combinator Selectors is used to select HTML Element or tag based on the relationship.

*Combinator Selectors are Categorized into four types:

*Descendent Selector (parent child{})

*Child Selector (parent**>**child{})

*Adjacent Sibling Selector (parent **+**child{})

*General Sibling Selector (parent **~**child{}) ~ - Tilda Symbol

Descendent Selectors:

This selector is used to select all the child elements of the specified tag.

The tags can be the direct child of the specified tag or can be very deep in the specified tag.

Ex: parent child{}

Descendent selectors will select p1,p2,p3,p4 and p5.

Child Selectors (>):

Child selectors are used to select the element that is the immediate child or direct child of the specified tag.

Ex: parent>child{}

Adjacent Sibling Selectors (+):

The adjacent sibling selector is used to select the element that is adjacent or the element immediately next to the specified element.

The adjacent sibling selector will select only one tag that is present just next or immediately after closing the specified or parent tag.

Ex: parent +child{}

General Sibling Selector (~):

General sibling selector is used to select all the elements which is present after closing of the specified element.

This can be used to select group of elements that shares same parent element.

Ex: parent ~child{}

Pseudo Class Selectors:

A Pseudo Class Selector in CSS is used to define the special state of an element.

Pseudo Class Selector is denoted with single :

Pseudo Class Selectors Are classified into,

*Anchor Pseudo Class

*Hover Pseudo Class

*Focus

*First Child

*Last Child

*'n' th Child

*Only Child

Anchor Pseudo:

Anchor Pseudo Class is used to change the state of colors in anchor tag.

Anchor Pseudo Class the state can be changed by,

*link-(Unvisited)

*Active

*Visited

Link:

Unvisited or link matches the link that have not been visited.

Active:

Pseudo Class is used to select an element which is activated when the user clicks on it.

Visited:

Visited Pseudo class is apply one matches link that have been visited.

Hover Pseudo Class:

Hover Pseudo Class is used to add special effort to an element when mouse point to over on it.

Ex: div : hover { color : red; }

Focus:

Focus Pseudo Class is used to select an element which is currently focused by the user.

It Works on user input element used in forms and it's trigger as soon as the user clicks on it.

Ex: input : focus{}

First Child:

First Child Matches an element that is the first child of it siblings. That means among the siblings who it is first.

Last Child:

Last Child matches an element that is the last child of it siblings.

'n' th child:

'n' th child is used to select elements from a list of sibling element.

Only Child:

Only Child will select an element that has no sibling, like only child.

Enabled: Ex: input : enabled

Selects every enabled <input> element.

Disabled: Ex: input: disabled

Select every disable <input> element.

In-range: Ex: input: in-range

Selects <input> elements with a values specified in the range.

Invalid: Ex: input : invalid

Selects all the <input> elements with an invalid value.

Not-selector: Ex: input : not(p)

Select every element that is not <p> element.

Required: Ex: input : required

Selects <input> elements with a required attribute specified

Optional: Ex: input : optional

Select <input> elements with no required attribute.

Pseudo Element Selectors:

CSS Pseudo Element Selector is used to select and style a specific part of the selected element.

Ex: Selector:: Pseudo element{ Property : value }

Pseudo Selector is used to denoted with double ::

Pseudo Element Selectors is Classified into,

*First line
*First letter
*Marker
*Selection
*Before
*After

First Line:

First line pseudo element applies styles to the first line of an block level element.
We Can't apply to the inline element.
Pseudo Element we can apply CSS properties like font properties, colour, background, word spacing, Letter Spacing, Text Definition, Text Decoration, vertical align, etc.,

Ex: <p>:: First line { color : red;}

First Letter:

It applies to the style to the first letter of the first line of a block level element. But, only one not preceded by the other content.
Pseudo Element we can apply CSS properties like font properties, color, background ,word spacing, Letter Spacing, Text Definition, Text Decoration, vertical align, etc.,

Ex: <p>:: First letter{ color :red;}

Before:

It creates before pseudo element that's the first child of the selected element.
It's often used to add content like images and text.

Ex: p::before{content:"add text/image";}

After:

It create pseudo element that's the last child of the selected element.
It's often used to add content like images and text.

Ex: p::after{content:"add text/image";}

Marker:

Marker pseudo class element selects a marker box of a list items.
List item which typically contents bullet or number.
It works on any element or pseudo element set to display: list item such as the <li> tag and summary element.

Ex: <ul> {::colour : red;}

Selection:

This pseudo element applies styles to the part of an document that has been highlighted by the user such as clicking and dragging the mouse across the text.

Ex: p::selection{ color : red ;}

Attribute Selector:

Attribute selector is used to select an element with some specific attribute or attribute value.
There are several types of attribute selectors are there,

[attr]
[attr="value"]
Attr~="specified word"]
[attr|="value"]
[attr^="value"]
[attr$="value"]
[attr*="value"]

[attr]:

This type of attribute selector is used to select all the element that have the specified[element] attribute and applies the CSS property to that attribute.

Ex:

```
<a href="#" target="_self">Link</a>
[_self]{
…….
''}
```

[attr="value"]:

This type of selector is used to select all the elements where attribute has the value exactly same as the specified value.

Ex:

```
[target="_self"]{
color:red;
}
```

[attr~="value"]:

This type of selector is used to select all the element where attribute value is a list of space separated values, one of which is exactly equal to the specified value.

Ex:

```
[class~="nav"]{
background color: red;
}
```

[attr|="value"]:

This selector is used to select all the element whose attribute has a (-) separated list of values beginning with the specified value.

Ex:

```
[class|="nav"}{
color:red;
}
```

[attr^="value"]:

This selector is used to select all the element whose attribute value begins with the specified value.

The value does not mean to be a whole word.

Ex:

```
[class^="nav"]{
color:red;
}
```

[attr$="value"]:

This selector is used to select all the element whose attribute value ends with the specified value.

The value doesn't need to be a whole word.

Ex:

```
[class$="nav"]{
color:red;
}
```

[attr*="value"]:

This selector is used to select all the element whose attribute values contains the specified value which present anywhere.

The values doesn't need to be a whole word.

Ex;

```
[attr*="nav]{
color:red;
}
```

Color Property:

CSS Color property is used to set the color of element. This property is used to set font color, background color.

Color of an element can be defines in the following ways,

*Built-in color
*RGB Format
*RGB-A Format
*Hexa-Decimal[code format]notation;
*HSL Format
*HSL-A format

Built-In Color/Direct color:

There are a set of pre-defines colors which are used by its name.

Ex: color: red;

RGB-Format:

This format is used to define a color of an HTML element by specifying the RGB values range between [0-255].

Ex: color:(255,0,0) for RED
color:(0,255,0) for Green
color:(0,0,255)for Blue

RGB-A Format:

This format is similar to the RGB. But, the difference is RGB-A Contains (alpha) which specifies the opacity/Transparency of an element. The values of alpha lies between 0-1.

0-Fully Transparent
1-Fully Visible

Hexa-Decimal Notation:

It begins with # symbol followed by its size or their characters. Each ranges from [0-9, A-F] totally 16 characters.

For Red: color: #FF0000

HSL-Format:

HSL Stands for Hue, Saturation, Lightens.

This format uses the cylindrical co-ordinate system.

Hue:

Hue is the degree of the color wheel(0-360).

It values lies between (0-360); where 0-red, 120-Green, 240-Blue.

Saturation:

It takes percentage values; where 100% represents Saturated and 0% represents UnSaturated.

Lightens:

It takes percentage values where 100% represents white and 0% represents black.

Ex:

```
h1{
Color: hsl(_,_,_)
}
```

HSL-A Format:

This color property is similar to HSL property; but the difference is HSL-A Contains a Alpha, which specifies the opacity/transparency of an element. The value of alpha lies between 0-1.

0-Fully Transparency
1-Fully Visible

*Margin
*Border
*Padding

*Content

Background Property:

Background properties are used to define the background effort for the element.

Background Color:

Background color property is used to specify the background color of an element.

```
Ex: body{
Background color:  red;
}
```

Background Image Property:

Background image property is used to set an images to an element.

```
Ex: div{
Background image :"path of an image";
}
```

Background repeat:

Background repeat property is used to repeat the background image both horizontally and vertically.

And background no repeat is used to stop the repeating of images in the element.

```
Ex: div{
        Background -repeat: repeat, repeat x, repeat y, no-repeat;
        }
```

Background size:

Background size property specifies size of the background images.

```
Ex: div{
        Background size: auto, contain, cover;
        }
```

Background Attachment:

Background attachment is use to fix background image.

The image will not scroll with the page one background attachment is fixed.

By default the value is scroll.

```
Ex: div{
        Background attachment: scroll, fixed;
        }
```

Background Position:

This property is used to set the image to a particular position.

The positions are top left(default), top center, top right, center left, center right center center, bottom left, bottom right, bottom center.

Background Clip:

This property is used to define how to extend the background color or image within an element.

```
Ex: div{
Background-clip : border-box, padding-box, content-box;
}
```

Background origin:

The background origin is the property defined in CSS which helps in adjusting the background image of the webpage.

```
Ex: div{
        Background origin: border - box :
        }
```

Gradient:

Linear Gradient:

It includes the smooth color transitions to going up to left, to right, to up, to down and to diagonally.
The minimum 2 color required to create a linear gradient.
More than 2 color element can be possible in linear gradient.
The starting point and the directions are needed for the gradient effect.
Ex: Background-image : linear-gradient (direction, color stop1, color stop2,.....);

Radial-Gradient:

Radial Gradient differs from linear-Gradient.
It starts at a single point and emanate outwards.
By default, the gradient will be ellipse shape and other gradient value is circle.
Ex: Background-image: radial-gradient(shape, color stop1, color stop2,…)

Repeating -linear-gradient:

CSS Allows the user to implement to multiple linear gradient using a single function repeating linear gradient.
Ex: background-image: repeating-linear-gradient(45deg,#000,green 20%);

Font Property:

CSS font defines font related properties and how font resources are loaded.
It led you to defines the style of the font such as family, size, style, weight, etc.,.
Font Size and font family font weight font style font stretch font variant

Font-Size:

Font-size property is used to set the font size of text in html document.
Syntax: font-size: large, small, 100px, 200px, 1rem, 2rem

Font-family:

Font -family is used to set font type of an html element.
It holds several font names as a built-in font families.
We can add google font to the html document by import in CSS or link in html.
Syntax: font-family: times new roman, monoscope, sans-serif;

Font-weight:

It's used to select boldest of the font.
Its value can be 100-900, bold, bolder, lighter, normal.
Syntax: font-weight: bolder, lighter;

Font-variant:

It's used to create the small cap effect in capitals.
It can be normal or small caps.
Syntax: font-variant: small-caps;

Font-Stretch:

The font-stretch property is used to set the text wider or narrower.

Font-style:

Font-style is used to set to change italic or oblique.
Note: Italic is cursive in nature
Whereas oblique default font with slanting direction

Border Property:

This property allows you to specify how the border as a box representing an element should look.
There are 3 properties of an Border,
*border-width
*border-style
*border-color

Border-Width:

This property allows you to set the width of an element border.

We can individually change the width of the bottom, top, left, right borders of an element using the following properties such as,

*border-bottom-width
*border-top-width
*border-left-width
*border-right-width

Border-Style:

This property allows you to select one of the following styles of border such as,

*none - no border
*solid - border is an single solid line
*dotted - border is a series of dots.
*Dashed - border is a series of short lines
*double - Border is 2 solid lines
*group
*ridge
*inset
*outset
*hidden

We Can individually change the style of the bottom, left, right, top borders of an element using the following properties such as,

*border-bottom-style
*border-top-style
*border-left-style
*border-right-style.

Border-Color:

This property allows you to change the color of the border surrounding an element.
We can individually change the color of top, left, right, bottom sides of an element using,

*border-bottom-color
*border-top-color
*border-left-color
*border-right-color

We can take shorthand property for border like,

Ex: Border: width style color;
Ex: border: 2px solid red;

Border-Radius Property:

This property is used to round the corner of the border edges of an element.
This property contains 1,2,3 or 4 values that can be individually given by the properties such as,

*border-top-left-radius
*border-top-right-radius
*border-bottom-left-radius
*border-bottom-right-radius.

Syntax:

Give 4 values - border-radius: 10px 20px 30px 40px

*first value top left
*2nd value top right
*3rd bottom right
*4th bottom left

Give 3 values - border-radius: 10px 20px 30px

*1st value top left
*2nd value top right and bottom left diagonally

*3rd value bottom right

Give 2 values - border-radius: 10px 20

*1st value top left and bottom right

*2nd top right and bottom left

Border-Collapse:

This property is used to create an collapsed border in html.

The border-collapse property is used to set the table border to collapse into a single border.

Syntax: border-collapse: collapse;

Border-Spacing:

Border spacing property is used to set an distance between borders of the neighbouring cells in the table.

This property works only when the border collapse property is set to separate or no collapse or default collapse.

Syntax: border-spacing: 20px;

Outlines:

Outlines are very similar to borders but there are few major differences such as,

*Outlines don't have to be rectangular.

*Outline is always the same on all sides.

*You can't specify different values for difference sides of an element.

*We can set the following outline properties such as,

*outline-width

*outlines-offset

*outlines-colour

*outline-style

Outline-width:

Outline width property specify the width of the outlines to be added to the box.

Syntax: outline-width:2px;

Outline-Style:

The outline style specifies style of an element that goes around an element.

It can take one of the following values,

none, solid, dashed, dotted, etc.,

Syntax: outline-style: dashed;

Outline-Color:

Outline color allows you to specify the color for the outline.

Syntax: outline-color: blue;

Outline-Offset:

The outline-offset property adds space between an outline and the border of an element.

Syntax: outline-offset:5px;

CSS Text Property:

CSS Text properties are used to format the text, style the text and perform different types of manipulations like woard-spacing, alignment, justification and text transformation.

CSS Text formatting properties are text color, text align, text decoration, text indent, text-transform, text overflow, text shadow, letter-spacing, word-spacing, line height and direction.

Text-Align:

This property is used to specify the horizontal alignment of text in an element inside a block-level element or in table-cell.

The values can be pass left by default, right, centre, end, start, justify.

Syntax: text-align: justify;

Text-Decoration:

This property is use to decorate the content of the text with the values such as,

*underline
*overline
*line-through
*none

Syntax: text-decoration: underline;

Text-Indent:

This property is used to indent the text or giving the space for the first line of a paragraph.

Syntax: text-indent: 10% or 5px;

Text-Transform:

This property is used to capitalize the text or convert the text to uppercase or lowercase letters.

Syntax: text-transform: capitalize;

Text-Overflow:

This property of text formatting specify the some text has overflow and its hidden from the view.

Syntax: text-overflow: ellipsis;

Text-Shadow:

This property is used to set the text shadow around a text.
It takes four values,

Syntax: text-shadow: 1px 2px 3px 4px;

First value x is for shadow on x direction
2nd values is for shadow on y direction.
3rd value is for blur.
4th value is for colour.

Letter-Spacing:

This property is used to add a subtract space between the letters that make a word.

Syntax: letter-spacing: 2px;

Word-Spacing:

This property is used to specify the space between the words of the line.

Syntax: word-spacing: 2px;

Line-Height:

This property is used to set space between the lines in a paragraph.

Syntax: line-height: 2px;

Direction:

This property is used to set the direction of the text.

Syntax: direction: rtl or ltr;

White-space:

This property is used to control the flow and formatting of the text.

Syntax: white-space: no-wrap;

Word Break:

This property is used to break the words.

Syntax: word break: break all;

Cursor:

Cursor property allows you to specify the type of cursor that should be displayed to the user. The following values for the cursor properties.

*auto
*cross-hair

*default
*pointer
*move
*wait
*zoom-in
*zoom-out etc.,
Syntax: cursor: pointer;

Float Property:

Float property positions the element to the left or right in its parent container.
It enables the text, images all other inline elements to wrap around the floating elements.
Syntax: float: left or right;
Note: Elements are not allow to flow out vertically. We can only in horizontally.

!important:

The important property in CSS means that all the subsequent rules of an element all to be ignored, and the rule denoted by important is to be apply.
This rule overrides all previous styling rules, the important property increases its priority.
The important property is mentioned immediately before the semi-colon(;).

Inline	Internal	External

l<a href=""=link>	h1{
	color: blue;
	}
		h1{
		color: green;
		}

Box-Shadow property:

Its property used to give a shadow line effects to the frames of an element.
The box shadow can be described using x and y offset relative to the element. 3rd value blur 4th value spread 5th value color.
Syntax: box-shadow: H-offset, V-offset, blur, spread values, color;

Visibility:(Independent property):

This property is used to specify whether an element is visible or not in an web document. But, the hidden element take up space in the web document.
Syntax: visibility: visible;(default)
visibility: hidden;
visibility: collapse;
Note: Display none is used to remove an element or delete an element completely from the browser and it won't take up the space from the browser.

CSS Box-Model:

The box-model is a container that contains multiple properties including borders, margins, paddings and the content itself.
It's used to create a design and layout of web pages.
The web browser renders every element as a rectangular box according to the CSS Box-model.
For each and every CSS browser will create box-model.

Content:

This contains the actual data in the form of text, images or other media forms and it can be sized using width and height.

Padding:

This property is used to create space around the element, inside any defined border.
When you're giving 4 values like 10px, 20px, 30px, 40px it will took top, right, bottom and left.
In the same way if we give 3 values like 10px, 20px, 30px it will took top, right-left diagonally and bottom.

If we give 2 values like 10px, 20px it will took top-bottom and 2nd value will be right-left.

Syntax: padding: 10px 20px 30px 40px;

Border:

This property is used to cover the content and any padding and also allows setting the style, color and width of the border.

Syntax: border: 10px solid red;

Margin:

This property Is used to create space around the element or outside the border.

When you're giving 4 values like 10px, 20px, 30px, 40px it will took top, right, bottom and left.

In the same way if we give 3 values like 10px, 20px, 30px it will took top, right-left diagonally and bottom.

If we give 2 values like 10px, 20px it will took top-bottom and 2nd value will be right-left.

Syntax: margin: 10px 20px 30px 40px;

Box-Sizing Property:

Box sizing property in CSS defines how the user should calculate Total width and height of an element.

Therefore, padding and borders are to be included or not.

Content-Box:(Default value of box-sizing property)

This is the default value of the box Sizing property.

The width and height property include only to the content.

Border and padding are not included in it.

Syntax: box-sizing: content-box;

Border-Box:

The width and height properties included the content, padding and border.

Syntax: box-sizing: border-box;

Overflow Property:

The overflow property controls the element content overflows from its width and height.

The values that can be passed for overflow property is,

*Visible

*Hidden

*Scroll

*Auto

Syntax: overflow: visible/hidden/scroll/auto;

Visible:

This is the default value of the overflow property.

No content is clicked when its bigger than its set dimensions.

Hidden:

Content is clipped when it is bigger than its set dimension is not shown.

Content that overflow from the container is called hidden.

Scroll:

Content is hidden, but users can still scroll through and view the hidden content.

Auto:

If the content is lesser than the set dimensions, the auto value will not add scroll bars, whereas when that the content is more than the container, auto value will automatically add scroll bars to the containers.

Object-Fit Property:

This property is used to specify how an image or audio should be resized to fit its content box.

Syntax: object-fit: fill/contain/cover;

Fill:

Fill is the default value, the image is stretched or compressed to fit the content box. The image will completely fill the box.

Contain:

The image preserves object ration of the original image. But, the image will not fit to the entire width and height of the content box.

Cover:

The image preserves aspect ratio of the original image and also it will fitting in the content box.

Position Property:

Position property is the method of positioning for an element or an html entity and the positioning of an element can be done using top, right, bottom, left properties.
Positioning specifies the distance of an html element from the edge of the viewport.
There are 5 different position properties available in CSS,

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

Static:

This method of positioning is set by default.
By defining static the top, bottom, right and left will not have any controls over the element.
Left, right, bottom, top values aren't given for position static.

Syntax: position: static;

Fixed:

An HTML element with position fixed property will be positioned relative to the viewport.
The fixed position allows an element to remain in the same position even we scroll the page.
We can set top, bottom, left and right properties for positioning an element.

Syntax: position: fixed;

Sticky:

If the element is place in the middle of the document then, when the user scrolls the document, the sticky element start scrolling until it touches the top.
When it touches the top it will be fixed at the top place In spite of further scrolling.
For sticky position the top or bottom is mandatory.

```
Syntax: position: sticky;
        top:0px;
```

Relative:

An element with position relative is positioned relative to its normal position(original position) by leaving the whitespace behind it.
Position relative doesn't alter the webpage.

Syntax: position: relative;

Absolute:

An element with position absolute is positioned relative to the nearest position anchestor (parent & grand-parent cycle) element.
Absolute position doesn't leave whitespace behind it.
It will affect the layout of webpage.

Syntax: position: absolute;

And we must write position: relative; on the anchestor block.

Z-Index:

- The Z-Index allows us to define a visual hierarchy on the "3-dimensional plane".
- It is used to specify stacking order of the position element.
- Z-Index can be given to the element whose position value is either fixed, absolute, relative or sticky.
- Z-Index cannot be given to the position static.

Flex Property:

Flex property is a combination of,

*flex-grow
*flex-shrink
*flex-basis

It is used to set the length of the flexible items . The flex property is much responsible and mobile friendly.
It is easy to position child elements and the main container.
Syntax: display: flex;

Flex-Grow:

It is used to specify how much items will grow relative to its set of the flex items.
Syntax: flex-grow:1;

Flex-Shrink:

This property specifies how much items will shrink relative to its flex items.
Syntax: flex-shrink:1;

Flex-Basis:

This property set length or width of flex items.
Syntax: flex-basis: 100px or 100%;

Flex-Wrap:

This property is used to specify whether flex items should be in a single line or multiple lines.
Syntax: flex-wrap: no-wrap;

Flex-Direction:

This property specifies the direction of the flexible items direction values. It can be either column or row. By default the direction is row.
Syntax: flex-direction: row/column;

Justify-Content:

It aligns the flex-items in 'Horizontal Directions'. The values that can be used for justify content is,

*flex-start
*flex center
*flex-end

Syntax: justify-content: flex-start/flex-center/flex-end;

Align-Items:

It aligns the flex-items in 'Vertical Directions'. The values that can be used for align items should be same as justify content.
Syntax: align-items: flex-start/flex-center/flex-end;

Grid Property:

CSS grid is a two dimensional layout system it's used to be designing the webpages.
The elements are arranged in both column and rows.
Grid contains a parent usually referred as a Grid container and one or more child element are called grid items.

Grid lines:

Grid lines are horizontal lines and vertical lines that run through the entire CSS grid.

These lines separate elements from one another.

Grid Column:

The space between any two adjacent vertical grid lines Is known as "Grid-Column".
Grid column can be created by using a property "grid-template-column".
In Grid-Template-Column, the no. of column its mentioned with either %, px, rem, fraction(fr).
Syntax: grid-template-column: 25%/10rem/100px/1fr;

Grid Rows:

The space between any two adjacent horizontal grid lines is known as "Grid Rows".
To create rows in grids; we use a property called "Grid-template-rows".
In Grid-template-rows the no. of rows is mentioned with either %, px, rem, fr.

Grid-Template Area:

Grid-Template area allows to create a named grid width which can choose at different name for every area in the grid and only element that at the same grid area name will be placed there.
Syntax: grid-template-area: item1 item2 item2 item3;

Column-Gap:

This property defines the size of the gap between the column is grid layout.

Row-Gap:

This property defines the size of the gap between the rows in a grid layout.

Grid-Gap:

Grid-gap is a shortend property that sets the size of the gap between grid rows and column.

Grid-column start:

Grid-column-start is defines on the which column line the item will start.

Grid-column-End:

Grid-column-end property defines on the which column line the item will end.

Grid-Row-Start:

Grid-row-start defines on which row line, the item will be start.

Grid-Row-End:

Grid-row-end property defines on which row line, the item will end.

Transform Property:

Transform property used to change the co-ordinate space by visual formatting model.
This is used to add effect like skew, rotate, scale and translate.

Rotate:

It specifies the angle of rotation. Provide values only in degrees.
Rotate can be specified with x axis, y axis and z axis.
Syntax: transform: rotatex(120deg);

Skew:

It specifies the skew(parallelogram) transformation along with x axis, y axis and z axis corresponding to skew angles.
Syntax: transform: skewx(120deg);

Scale:

It specifies the scale transformation along with x axis and y axis. The default value of scale is 1.
Syntax: transform: scale(1.5);

Translate:

It specifies the translation(Movement) across the x and y axis.
Transform can translate across x, y and z axis.
Syntax: transform: translate(20% or 100px);

Transition Property:

CSS transition allows you to change the property value over a given duration.
Transition means a triggering point such as hovering effect.
Transition property consists of transition duration, transition delay and transition timing function.

Transition Duration:

This property allows you to how long it will take to complete the transition from one state to another state.
Syntax: transition-duration: 10s;

Transition-Delay:

This property allows you to determine the amount of time to wait before the transition actually starts.
Syntax: transition-delay:2s;

Transition Timing Function:

This property allows you to determine the speed of change and manner of change during the transition.
Syntax: transition-timing-function: linear/scale/ease-in/ease-out/ease-in-out.

PS5 Animation:

It's a technique to change the appearance and behaviour of the element in the webpage.
It is used to control the elements by changing their motions or display.
It's having two parts, One part the essay property which describes animation properties will be written. Whereas, the second part contains keyframes which indicates the animation properties of the element and the specific time travel at which animation occurs.

@Keyframe:

@keyframe defines how the animation should display with respective of its duration.
Animation properties are animation name, animation duration, animation timing function, animation delay, animation iteration count.

Animation Name:

It is used to specify the name of the @keyframes describing the animations.
Syntax: Animation-Name: Name;

Animation Duration:

It is used to specify the time duration for complete one animation.
Syntax: Animation-duration-timing: 4s;

Animation-Timing-Function:

It Specify how animation makes transition through keyframes.
Then value of animation timing function are
Syntax: Animation-timing-function:[linear/ease/ease-in/ease-out/ease-in-out];

Animation-Delay:

It specifies the delay of Start of an animation.
Syntax: Animation-Delay:3s;

Animation-Iteration-count:

This specify the number of times the animation will be repeated.
Syntax: Animation-iteration-count: infinite;

Media Query:

The Media Query in CSS used to create a responsive web design.
It means the view of a webpage differs from system to system based on screen or media types.
The break point specifies for what device width size the content is just starting to default.
Media queries can be used to check,
*width and height of the viewport.

*width and height of the device.
*Orientation
*Resolution

The media query consist of the media type that can contain one or more expression which can be either true or false.

If the media query is true then, the stylesheet is applied.

Syntax: @media screen and (expression){
…Styles
}

refer mdn web docs for latest update

https://developer.mozilla.org/en-US/