**INDEX**

1. Syntax
2. How to add links or add CSS to HTML.
   1. Inline, Internal, External
3. Priority -> order
4. Sectors, priority ->set before
   1. Element, id, class
   2. Id > class > element > universal
5. Comment.
6. Color
7. Background, multiple background
8. Border
9. Border-radius.
10. Height & width
    1. Max-width, min-width.
    2. Max-height, min-height
11. Padding.
12. Margin, margin collapse
13. Box model
14. Text
15. Link
16. Unit in CSS – responsive font
17. Opacity
18. Filter (image)
19. Image sprites
20. Gradient
21. Overflow – works with inline-block, and block elements, height is a must
22. Resizable- works with inline-block, block element
23. List
24. Table
25. CSS function - calc ()
26. Box sizing
27. Initial vs inherit
28. Object-fit.
29. Pseudo element and class
30. Display property
31. Position
32. Float
33. Transformation – works with inline-block and block element
34. CSS Transition
35. Animation
36. Print Specific Style
37. Responsive with design
38. Viewport meta in CSS
39. Media query
40. Advance selectors
41. Specificity
42. Flexbox
43. Grid
44. Box shadow

**Text**

* 1. Color
  2. Text-align
  3. Text-decoration
  4. Text-decoration
  5. Text-transformation
  6. Line-height
  7. Letter-spacing
  8. Word-spacing
  9. Text-indent
  10. White-space
  11. Word-wrap
  12. shadow

**Media Query**

Syntax

@media screen and(max-width:600px){

}

@media(max-width:600px){

    p{color:green}

}

Note: 0px-600px

@media(min-width:600px){

    p{color:green}

}

Note- above 600px

@media(min-width:600px) and (max-width:768px){

    p{color:green}

}

Note: between 600px – 768px

**Display**

1-block

Start with a new line

Takes full with

Height and width can be applied

2- inline

Not start from a new line.

Don’t take full width (fit-content)

Height and width can not be applied

Padding top/bottom can not be applied.

Margin-top/bottom can not be applied.

3- inline-block

Do not start from a new line.

Don’t take full width(fit-content).

Height and width can be applied.

Padding top/bottom can be applied.

Margin-top/bottom can be applied.

4-none

Inline>inline block>flex

**Position**

**Type of position**

1. static (default)
2. Relative

start positioning from its place, leave its place but not allow any element to take its place.

1. Absolute (out of flow)

Start from its nearest positioned parent(relative, absolute, fixed) if not then consider the body as the positioned parent and leave its place, allowing other elements to take its place

1. Fixed (out of flow)

As per the viewport, leave its place and allow other elements to take its place

1. Sticky (out of flow)

**Property**

1. Top
2. Bottom
3. Left
4. Right

**Transformation**

(works with inline-block and block element)

1. Rotate (out of flow)
2. Move (out of flow)
3. Scale
4. Skew

**Rotate**

transform:**rotate**(20deg);

**Move**

transform: **translate**(10px, 40px);-(x, y)

**Scale**

transform: **scale**(2, 2) */\* current width multiplyx2, and  current height multiply by 2 \*/* ;

**Scew**

transform: **skew**(20deg, 30deg) */\*x exits and y exits\*/*

**Transition**

Apply to main element and effect on hover

  p{

            border:1px solid green;

            text-align: center;

            line-height:60px;

            font-size:15px;

            transition-property: all; */\*width, height\*/*

            transition-duration: 2s;

            transition-timing-function: ease;

            transition-delay:1s;

       }

       P: hover{

        background: red;

        color: white;

        cursor: pointer;

       }

        p{

            border:1px solid green;

            text-align: center;

            line-height:60px;

            font-size:15px;

            transition: property, duration, transition-timing-function, delay;

            transition: all .3s linear;

            transition: 1s ease

       }

       P:hover{

        background: red;

        color:white;

        cursor: pointer;

       }

Note- transition does not work with display property like display none or display block, use opacity , visible instead of display

**Animation**

@keyframes changeColor{

    0%{background-color:red;}

    25%{background-color:green;}

    50%{width:400px; }

    750%{background-color:pink;}

    100%{background-color:brown;}

}

#animate{

    background: red;

    border:1px solid green;

    height:200px;

    width:200px;

    animation-name:changeColor;

  animation-duration:5s;

    animation-timing-function:linear;

    animation-iteration-count:infinite;

animation-delay:2s;

animation-direction:reverse;/\*alternate, alternate-reverse\*/

}

<style>

  @keyframes example {

    0% { background-color: red; }

    50% { background-color: yellow; }

    100% { background-color: blue; }

  }

  .box {

    width: 100px;

    height: 100px;

    background-color: red;

    animation: example 3s infinite;

*/\* Name, duration, timing function, delay, iteration count direction \*/*

  }

</style>

</head>

<body>

<div *class*="box"></div>

</body>

</html>

**Pseudo-element and class**

It is used to style specific portions of elements and insert them through the CSS

**Pseudo Class**

<style>

*/\* unvisited link \*/*

    a:link {color: red;}

*/\* visited link \*/*

    a:visited {color: green;}

*/\* mouse over link \*/*

    a:hover {color: hotpink;}

*/\* selected link \*/*

    a:active {color: blue;}

    li:first-child {color:red;}

    li:last-child {color:blue; }

    li:nth-child(even) {background:gray}

    li:nth-child(odd) {color:orange}

    li:nth-child(4){color:red;}

    input:focus {

        outline:1px solid red}

</style>

</head>

<body>

    <a *href*="default.asp" *target*="\_blank">This is a link</a>

    <ul>

        <li>Home</li>

        <li>About</li>

        <li>service</li>

        <li>blog</li>

        <li>forum</li>

        <li>Service</li>

    </ul>

    <input *type*="text">

</body>

**Pseudo-element**

:: after

:: before

:: first-letter

:: first-line

:: selection

:: placeholder

:: marker

**Before and After**

<style>

    p::before{

        content:'hello world';

        color: white;

        background-color: red;

        padding:10px;

        display:block;

    }

    p::after{

        content:'bye world';

        color:white;

        background-color:green;

        padding:10px;

        display:block;

    }

    P:hover::before{

        display:block;

        cursor: pointer;

    }

</style>

</head>

<body>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Earum, aperiam!</p>

</body>

**First-letter and First-line**

    p::first-letter{

        color: red;

        font-size:20px;

    }

    p::first-line{

        color:green;

    }

</style>

</head>

<body>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Earum, aperiam!</p>

</body>

**Flex**

.container{

            display: flex;

gap:10px;

 row-gap:10px;

            flex-direction: row;

            flex-direction:row-reverse;

            flex-direction: column;

            flex-direction:column-reverse;

            flex-wrap:no-wrap;

            flex-wrap: wrap;

            flex-flow: row wrap;

            justify-content:flex-start;

            justify-content:flex-end;

            justify-content: center;

            justify-content:space-between;

            justify-content:space-around;

*/\*-------container height-------\*/*

            align-items: stretch;

            align-items:start;

            align-items: center;

            align-items: end;

*/\*-------container height-------\*/*

align-content: stretch;

         align-content:flex-start;

            align-content: center;

            align-content:flex-end;

            align-content:space-between;

            align-content:space-around;

justify-content:center;

            align-items:center;

            place-content: center;

}

.item1{

  order:3

        }

        .item2{

order:1

        }

        .item3{

order:2

        }

.item{

        flex-grow:1;

        }

  .item1{

            flex-basis:300px;

        }

*/\*-------container height-------\*/*

  .item1{

            align-self: stretch;

            align-self: start;

            align-self:center;

            align-self:end;

        }

 .item2{

            width:200px;

            flex-grow:1;

            flex:1 0 200px; */\*flex-grow, flex-shrink, flex-basis \*/*

            flex:200px;

        }

**Random**

**Black Overlay**

main {

  min-height: 100vh;

  background-image: **url**("assests//images/bg.jpg");

  background-repeat: no-repeat;

  background-position: center;

  background-size: cover;

  position: relative;

}

main .overlay {

  width: 100%;

  height: 100vh;

  background: **rgba**(0, 0, 0, 0.7);

  position: absolute;

  top: 0;

}

<body>

    <main>

        <div *class*="overlay">

        </div>

    </main>

</body>

**To chang color of svg img**

Fill=”red”

background-size: **max**(1200px, 100vw);

**or**

*/\* Default background size for smaller screens \*/*

.container {

    background-size: 100vw; */\* Sets the background size to cover the full viewport width \*/*

  }

*/\* Media query for larger screens \*/*

  @media (min-width: 1200px) {

    .container {

      background-size: 1200px; */\* Sets the background size to a maximum of 1200 pixels on wider screens \*/*

    }

  }