**Element Selector**: We can target multiple paragraphs at the same time and add a slide to them. It is very limitation (Can’t add a size of different paragraphs)  
Example:

p {

    color: rgb(221, 31, 31);

    background-color: rgb(228, 173, 230);

}

**Class Selector:** Class selector can be used to apply size to any elements independent of the type name.

Example:

.highlight{

    color: rgb(221, 31, 31);

    background-color: rgb(186, 230, 173);

}

We can also use multiple classes in one HTML elements.

.highlight{

    color: yellow;

    background-color: rgb(186, 230, 173);

}

.text-center{

    text-align: center;

}

<p class="highlight text-center">This is a paragraph 2.</p>

**Id Selector:** It is used by adding an attribute tothe HTML elements. It add a specific styling to just one element. Must be unique within a page. An id name cannot start with a number.

#id{

    padding: 20px;

}

**Descendant selector:**  It matches all elements that are descendants of a specified element.

.dec a{

    text-decoration: none;

    font-size: 30px;

    font-weight: 700;

}

<div class="dec">

    <div class="dec-child">

        Lorem ipsum dolor <a href="#">Link1</a>

    </div>

    <p>

        Lorem ipsum dolor sit amet <a href="#">Link2</a>

    </p>

    <a href="#">Link3</a>

</div>

**Universal Selector:** The universal selector (\*) selects all HTML elements on the page.

\* {

  background-color: rgb(254, 255, 255);

}

**DRY (Don’t Repeat Yourself):** Avoid repeating the same lines of code.

h1,h2,h3{

    background-color: rgb(40, 175, 175);

 }

**Combinator**

**Combinator:** It explains the relationship between the selectors.

**The child selector (>):** It selects all elements that are the children of a specified element .

Example:

div > a {

  background-color: rgb(89, 22, 245);

}

**Adjacent Sibling Selector (+):** It is used to select an element that is directly after another specific element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

h1 + p {

    border: 5px dotted rgb(219, 25, 25);

}

**General Sibling Selector (~):** It basically selects all elements that are next siblings of a specified element.

h1 ~ p {

    border: 5px dotted rgb(219, 25, 25);

}

# **Basic Attribute Selectors:** It allows us to target elements based on their specific attributes.

Specific

img[alt="Image of attribute?"]{

    border: 2px solid red;

}

Bigins with value

img[alt^="Image "]{

    border: 2px solid red;

}

Ends with value

img[alt$=" attribute?"]{

    border: 2px solid red;

}

Occurs anywhere

img[alt\*=" attrib"]{

    border: 2px solid red;

}

Space before and after of that value

img[alt~="of"]{

    border: 2px solid red;

}

Specific Value

input[type='text']{

    border: 3px solid green;

}

## Pseudo-class Selector:

* Style an element when a user mouse over it
* Style visited and unvisited links differently
* Style an element when it gets focus

## Example:

## Unvisited link

  a:link{

        background-color: royalblue;

    }

## Visited link

    a:visited{

        background-color: teal;

        color: #fff;

    }

## Hover

p:hover{

        background-color: seagreen;

    }

## Active: Describe the moment that press the link

 p:active{

        border: 2px solid tomato;

    }

## Focus

a:focus{

        border: 4px dashed rgb(13, 66, 212);

    }

Pseudo-classes Selector with Forms

    p{

    background-color: coral;

    color: rgba(123, 255, 0, 0.918);

    }

    input[type='checkbox']:checked ~ p{

        background-color: red;

        text-decoration: line-through;

    }

Valid Pseudo-class Selector

 input:valid + q{

        color: honeydew;

        background-color: lawngreen;

        border: 1px solid green;

    }

    input:invalid + q{

        color: honeydew;

        background-color: red;

        border: 1px solid darkred;

    }

<!-- Valid  Pseudo-class Selecto -->

    <label for="gender">Gender:</label>

    <input type="number" id="age" name="age"

    min='18' max=25 >

    <q>If valid it Green otherwise Red</q>

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

Required Pseudo-class Selector

   input:required + q{

        border: 2px solid black;

    }

 <input type="text" placeholder="Required"

    required>

    <q>If valid it Green otherwise Red</q><br><br>

Optional Pseudo-class Selector

   input:optional + q{

        color: blueviolet;

    }

:not() & :empty

   p:not(#sec){

      padding: 20px;

      background-color: brown;

    }

    q:empty{

      display: none;

    }

Target Pseudo-class Selector

    p:target{

      padding: 20px;

      background-color: brown;

    }

    div{

        height: 500px;

    }

   <a href="#first">First Par</a>

    <a href="#sec">Second Par</a>

    <div></div>

    <p id="first">Paragraph1 11111111

        111111111111111111111111

        1111111111111111111111

        111111111111111111111111

    </p>

    <p id="sec">Paragraph2 222222222

        2222222222222222

        22222222222222222222

        22222222222222222222

    </p>

First-child Pseudo-class Selector

   p:first-child{

      color: tan;

      background-color: brown;

    }

Last-child Pseudo-class Selector

p:last-child{

        background-color: rgb(42, 165, 114);

    }

Only-child Pseudo-class Selector

  p a:only-child{

        background-color: rgb(147, 165, 42);

    }

Nth-child Pseudo-class Selector

  p:nth-child(2n+1){

        background-color: rgb(42, 165, 114);

    }

Nth-type Pseudo-class Selector

p:nth-of-type(2){

        background-color: rgb(42, 165, 114);

    }

First line & First Letter & Selection Text

    p::first-letter{

        font-size: 1cm;

    }

    p::first-line{

        color: tomato;

    }

    ::selection{

        background-color: rgb(224, 221, 64);

        color: rgb(27, 12, 240);

    }

::before & ::after

  p::before{

        content: 'Hi,' ;

        display: block;

        color: yellowgreen;

    }

    p::after{

        content: 'Regards' ;

        display: block;

        color: yellowgreen;

    }