CSS

* Styles are additional attributes used to make HTML more interactive and responsive.
* Styles provides c CSSOM to make elements effective .

[Cascade Style Sheet Object Model]

* Styles can be defined in 3 ways

1. Inline styles
2. Embedded styles
3. External style sheet

Inline styles :

* Styles are defined for every element individually by using “style ” attribute.
* It is faster in rendering
* It difficult to re-use.

Syntax:

<h2 style = “attribute:value”></h2>

Embedded Styles :

* Styles are defined in head or body section by using <style> tag.
* You can re-use the styles but slow in rendering .

Syntax:

<style>

h2{

}

</style>

**Session - 23**

FAQ: What is difference between styles in <head> or <body> ?

Ans: Styles in head section are intended to load into browser memory first , from there you can apply to elements .

* Styles in body section are intended to load and apply to elements directly.

FAQ: Can we defines styles in both sections ?

Ans: Yes.

Note: You can keep styles outside HTML .

FAQ: What is the MIME type for styles ?

Ans: text/css

Syntax:

<style type=”text/css”></style>

FAQ: What is the media types for styles ?

Ans: screen , print, speech

EX:

<!DOCTYPE html>

<html>

    <head>

        <title>Styles</title>

        <style type="text/css" media="screen">

            h1{

                background-color: green;

                color: white;

                text-align: center;

            }

            aside{

                width:100px;

                height: 50px;

                border: 2px dotted red;

                padding: 5px;

            }

        </style>

        <style type="text/css" media="print">

            h1{

                background-color: red;

                color: white;

                text-align: center;

            }

            aside{

                display: none;

            }

        </style>

    </head>

    <body>

        <aside>

            Ads..

        </aside>

        <h1>HTML</h1>

        <h1>CSS</h1>

        <h1>JavaScript</h1>

    </body>

</html>

* You can not access embedded styles from another page.

3.External Style Sheets

[Cascading Style Sheets]

* styles are maintained in a separate stylesheet.
* Clean separation of markup and styles.
* Reusability across page.
* Using an external style sheet will increase the number of requests for page.
* If number of requests for page increases then page load time will increase.

EX:

1. Create a new style sheet .

effects.css

h1{

    background-color: tomato;

    color: white;

    text-align:center;

}

1. Link the stylesheet to any page

        <link rel="stylesheet" href="../src/styles/effects.css" type="text/css">

FAQ: What is Minification ?

Ans: it is a technique used to reduce the file size.

it is not ZIP or RAR file.

effects.css development

effects.min.css production

EX:



FAQ: Can we edit minified files directly ?

Ans: Yes . but you should not disturb the minified logic.

**Session-24**

FAQ: What is CDN ?

Ans: Content Distribution Network

We keep our resources in a server location.

We access and use in various projects.

We will not download the files , we will use direct server links.

It save the memory space for your project.

Summary

1. Various techniques of defining styles
2. Inline
3. Embedded
4. Css
5. Minification and CDN
6. Media and MIME types

Stylesheet Rules

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

Q1. If we define 3 types of styles in a page , which one will work ?

1. Inline
2. Embedded
3. External file

Ans: If you are writing same attributes in all 3 locations then the priority

1st Inline

2nd Embedded

3rd external

If you are writing different attributes in all 3 locations then all will apply

Q2. Can we link multiple stylesheet to page ?

Ans: Yes

Q3. If all stylesheet have same effect then which one will apply ?

Ans: The latest will apply .

<link href=”a.css”>

<link href=”b.css”> // this will work

If all stylesheets have different effects , then all will apply.

Writing styles for Element

1. If you are writing styles inline

<h1 style=“attribute:value;attribute:value”></h1>

1. If you are writing styles embedded or external style sheet

selector{

attribute:value;

attribute:value;

}

CSS Selectors

1. Primary selector
2. Type selector
3. Id selector
4. Class selector

Type Selector : it defines the element name , to which element you want the styles to apply.

h2{}

p{}

table{}

li{}

you can ignore effects for any specific occurrence.

ID Selector : it uses an id reference for element.

<h2 id=”heading”>

<p id=”para”>

#heading{}

#para{}

Every element can refer to one ID only

You cant multiple categories of styles

Class Selector : it uses an class reference for element.

Class allows to implement multiple categories of styles.

.className{}

<div class=”class1 class2 class3 ..”>

Selector Rules

FAQ: if we are writing effects with all selectors which one will apply ?

Ans: if you are writing same attributes with 3 selectors then

1st priority ID

2nd priority class

3rd priority Type

If you are writing different attributes then all will apply.

Grouping Primary Selector;

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

* You can group the selector using “,” separator

A white background with black text

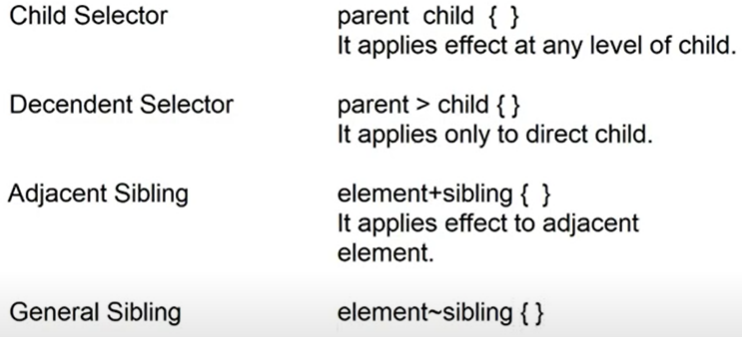
Description automatically generated

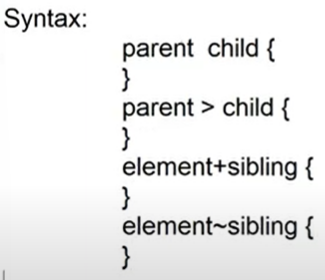
1. Rational Selectors – Combinators
2. Pseudo element selectors
3. Pseudo element classes
4. Element state selectors
5. Element validation state selectors
6. Structural selectors
7. Universal selectors
8. Root selector
9. Language selector

**Session – 25**

Rational Selectors And Combinators

* These selectors used for elements having parent and child hierarchy or siblings.





Dynamic pseudo classes

Dynamic : It changes according to situation.

Pseudo : It is not what it is showing .

Class : it is a program template with data and logsic .

Inheritance : it is a technique of resusing and extending code.

You inherit class by using “:” operator.

: hover : specifies actions on mouse over

:link :specifies effect for anchor in normal state

:visited :specifies effect for anchor in visited state

:active :specifies effect for anchor in active state.

EX:

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Selector</title>

    <style>

        h2:hover{

            color:red;

        }

    </style>

</head>

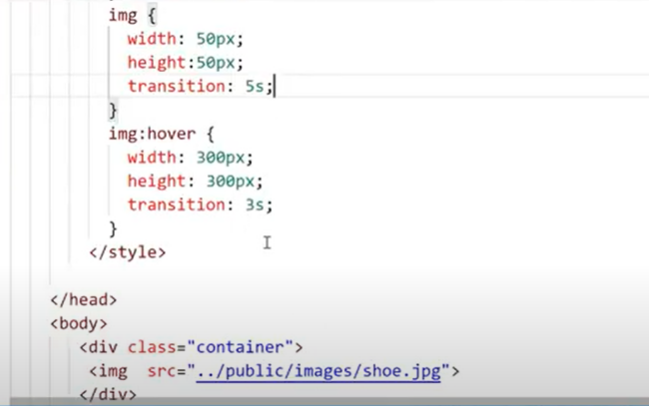
<body>

    <h2>Headings</h2>

</body>

</html>

EX:



A white background with black text

Description automatically generated

Element state pseudo classes

:disabled

:enabled

:read-only

:checked

:focus

EX: readonly

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Selector</title>

    <style>

        #user:read-only+span{

            display: inline;

        }

        #user+span{

            display: none;

        }

        #user:read-only{

            background-color: lightgray;

            border:none;

        }

    </style>

</head>

<body>

    User Name:

    <input type="text" id="user" value="John" readonly><span>You can't modify name</span>

</body>

</html>

Element Validation State classes

:required it verifies the required attributes set for input.

:optional it defines effects when required is not defined.

:valid it verifies any validations defined in input

:invalid it returns effects when any validation returns invalid

:in-range it is for number validations [min]

:out-of-range it is for number validations [max]

EX: required and optional are used only to verify the attribute not validate.

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Selector</title>

    <style>

        #user:required{

            border: 1px solid red;

            box-shadow:2px 2px 2px red;

        }

        #user:optional{

            border: 1px solid yellow;

            box-shadow:2px 2px 2px yellow;

        }

    </style>

</head>

<body>

    User Name:

    <input type="text" id="user" required>

</body>

</html>

EX:

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Selector</title>

    <style>

        #username:valid+div{

            display: none;

        }

        #username:invalid+div{

            display: block;

        }

        #mobile:invalid+div{

            display: block;

        }

        #mobile:valid+div{

            display: none;

        }

    </style>

</head>

<body>

    <h2>Register</h2>

    <dl>

        <dt>User Name</dt>

        <dd>

            <input type="text" id="username" required>

            <div style="color:red">User Name Required</div>

        </dd>

        <dt>Mobile</dt>

        <dd>

            <input type="text" id="mobile" pattern="\+91\d{10}\">

            <div style="color: red">

                Invalid Mobile : +91 and 10 digits required

            </div>

        </dd>

    </dl>

</body>

</html>

Attribute Selector

* It allows to apply effects based on the element attribute instead of tag directly.

Syntax:

element[attribute=value]{

}

element[attribute]{

}

EX:

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Selector</title>

    <style>

        input[type="button"]{

            background-color: green;

        }

    </style>

</head>

<body>

    <h2>Register</h2>

    <dl>

        <dt>User Name</dt>

        <dd>

            <input type="text" id="username">

        </dd>

        <dt>Mobile</dt>

        <dd>

            <input type="text" id="mobile">

        </dd>

    </dl>

    <input type="button" value="Register">

    <input type="button" value="Cancel">

</body>

</html>

EX:

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Selector</title>

    <style>

        p[id]{

            color:red

        }

    </style>

</head>

<body>

    <p>Para-1</p>

    <p id="p2">Para-2</p>

    <p>Para-3</p>

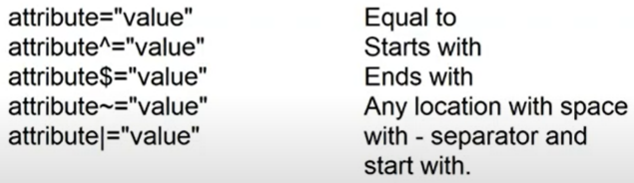
    <p id="p4">Para-4</p>

    <p>Para-5</p>

</body>

</html>

* Attribute selectors have conditions



EX:

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Selector</title>

    <style>

        p[class|="effects"]{

            color:red

        }

    </style>

</head>

<body>

    <p class="effects">Para-1</p>

    <p class="effectstext">Para-2</p>

    <p class="texteffects">Para-3</p>

    <p class="animatedeffectstext">Para-4</p>

    <p class="effects-text">Para-5</p>

    <p class="effects text">Para-6</p>

    <p class="text effects">Para-7</p>

</body>

</html>

Structural Selectors [Pseudo class]

* It depends on hierarchy parent-child
* It is to handle various occurrences in hierarchy

:first-child

:last-child

:nth-child()

:nth-of-type()

:nth-last-of-type()

EX:

li:first-child{

            color: red;

           }

           li:last-child{

            color: green;

           }

           li:nth-child(even){

            color:yellowgreen;

           }

           li:nth-child(odd){

            color:blue;

           }

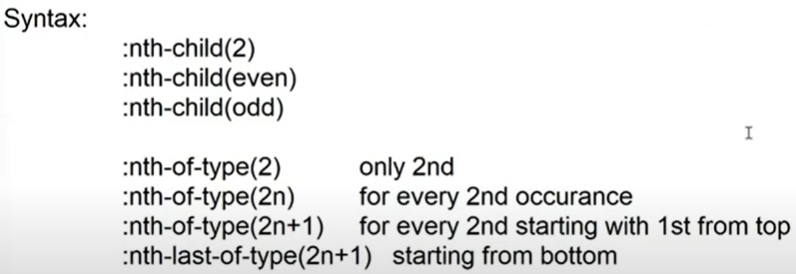
<style>

           li:nth-last-of-type(3n+1){

            color: red;

           }

        </style>



Element Classes and Behavioural

::placeholder

::before

::after

::selection

Root Selector

* :root{

}

Universal Selector

\*{

}

Language Selector

lang{

}