3. Introduction to CSS

CSS → Cascading Style sheet

↓

A rule based language. Rules are defined by specifying groups of styles that should be applied to particular elements or groups of elements on the web page.

Every single thing you see on a HTML page can be considered as an element.

CSS works at element level

Tell CSS → Apply style on a particular element or group of elements.

HTML is rendered from top to bottom.

Style needs to be defined & ready before the content loads.

```
<head>
    <style>
        Step1: Select the element
        h2 {  //all the h2's in
                HTML doc will have
                this style
            color: red;
            font-size: 20px;  → property
        }
          ↓      ↓
         key    value
    </style>
</head>
```

Defining CSS,

→ External

→ Internal

→ Inline

User Agent Style Sheet (UASS):

Even if we don't do any styling, every HTML element had some default styling. That styling is coming from UASS.

They are default styles that web browsers apply to HTML doc's when no specific styles are provided by the author.

Each browser has its own UASS.

User Agent → Browser (acts on behalf of the user when interacting with web content)

External CSS:

New CSS file is created. It contains all rules for the app.

To link this external CSS to HTML doc,

```
<head>
    <link rel="stylesheet" href=
    "relative path" />
```

style.css

```
h3{
    color: green;
}
```

Internal CSS:

CSS in the same as HTML file using <style>.

Inline CSS:

CSS for a particular element using <style>.

```
<div style = "background-color: aqua;
              font-size: 30px"
```

Style is available for all HTML elements.

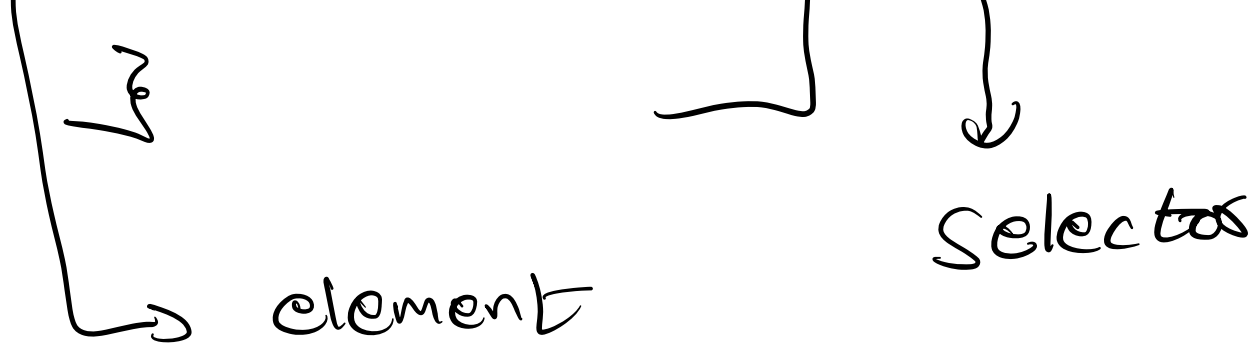
If a HTML element has style defined in all 3 places (external, internal, inline) what style will be applied!

The style of the last CSS to be applied.

Element Selector:

Selecting group of elements by a name.

```
h3 {
    color: red;
}
```



Classes:

Let's say we want to apply same styling to specific elements like h3, h2, input. Applying the style in internal/external CSS will apply to all h3, h2 & input but we only want for particular elements. Applying inline CSS will make the style repetitive.

Classroom → Students with different interests.

Science

Literature

Blue Badges

Red Badges

A class is an attribute that can be used to assign a specific name to an HTML element.

This name can be used to apply CSS styles or to manipulate the element with Java script.

Id's:

Library → Each book has UIN.

↓

Unique Identification Numbers

Library → HTML Doc

Book → HTML element

UIN → Id

Id's are used to identify HTML elements uniquely.

Id's are meant to be unique in the entire HTML doc.

No two elements can share the same id.

Classes vs Id's → can be assigned to only element

↓

can be assigned to multiple elements and styling will be applied to all the elements with that class name.

HTML won't throw errors when two elements have same id.

But it is considered invalid according to HTML specification.

Some browsers may render the style for all elements with same id & some may only render the style for first element with the id.

Issues with shared Id.

→ CSS conflicts : Only first occurence of id will be styled.

→ Javascript Issues: getElementById() only return the first element with that ID.

→ Accessibility Problems: Duplicate Id's confuse screen readers & other

assistive technologies

Class selector:

```
. class-name {

}
```

Id selector:

```
# id-name {

}
```

Selector is easy to select an element or group of elements from a HTML doc.

Descendant Selector:

Descendant → A child is a descendant of father as well as his grand father.

```
<ol>
    <li> I </li>
    <li> am </li>
    <li> a </li>
    <li> list </li>
</ol>

<div>
    <ol>
        <li> select </li>
        <li> me </li>
    </ol>
</div>
```

```
div li { // All li elements in
                div should be applied
}               this style.
```

Children Selector:

```
.cl > h1 { // h1 is direct child of
                cl.
}
```

One element can be given multiple classes separated by a space.

```
<div class="m1 m2"> </div>
```

Not one class. Two classes separated by a space.

```
.m1.m2 {  // Selecting multiple
                classes at the same
                time.
}
```

Attribute Selector:

Selecting an element from HTML doc based on attribute.

```
input [value = "select me"] {

}
```

Pseudo-Class Selectors:

Select and style elements based on their state or position in the document, without needing to add additional classes or ID's to those elements.

```
a:hover {
```

element : state

```
    color: greenyellow;

}
```