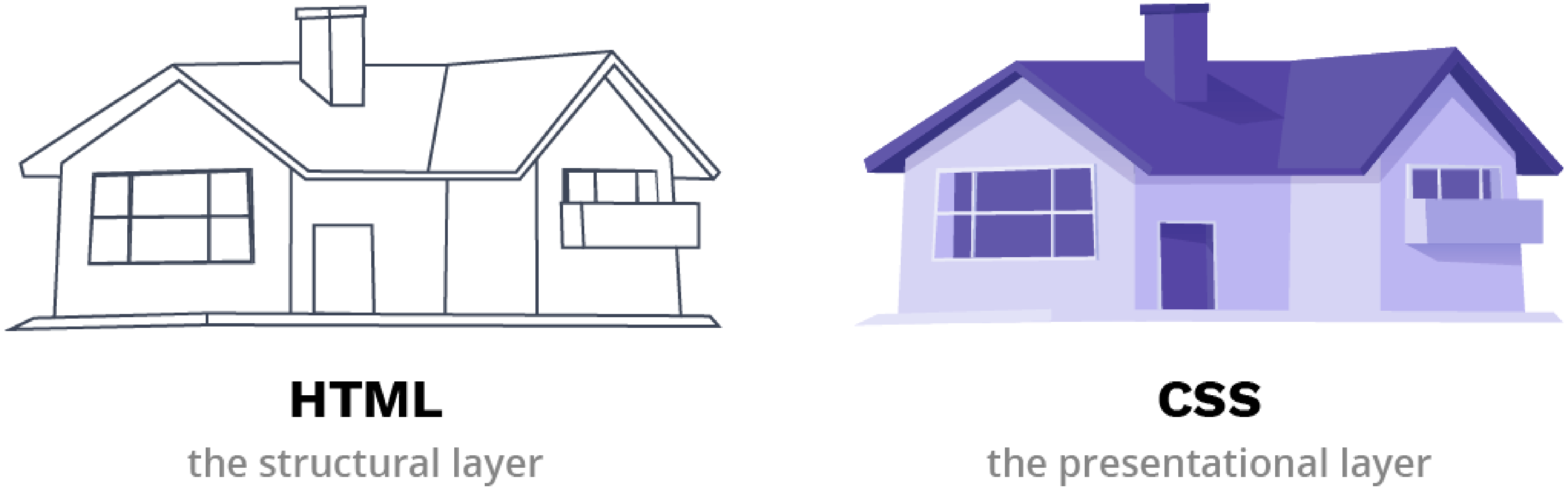
CSS- Cascading Style Sheets

- A **style sheet** is a document that **contains style information** about **one or more documents** written in markup language and used to control the styles such as **fonts, colors, size, spacing, margins etc.**

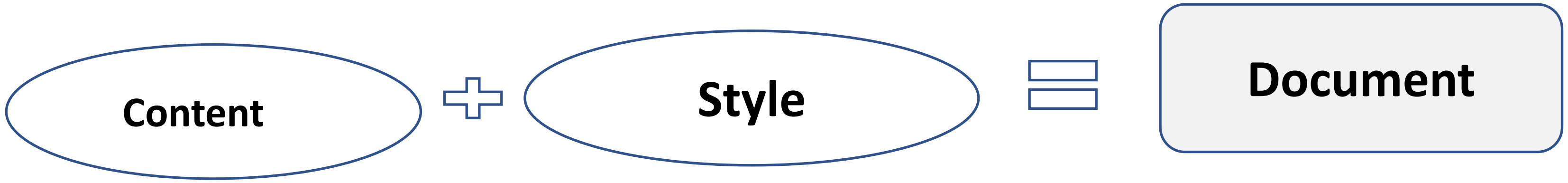
- **CSS** stands for **Cascading Style Sheets is a** style sheet language that describes the style of an HTML document.



- Describes the **appearance, layout, and presentation** of information on a web page
- Describes ***how* information is to be displayed**, not *what* is being displayed

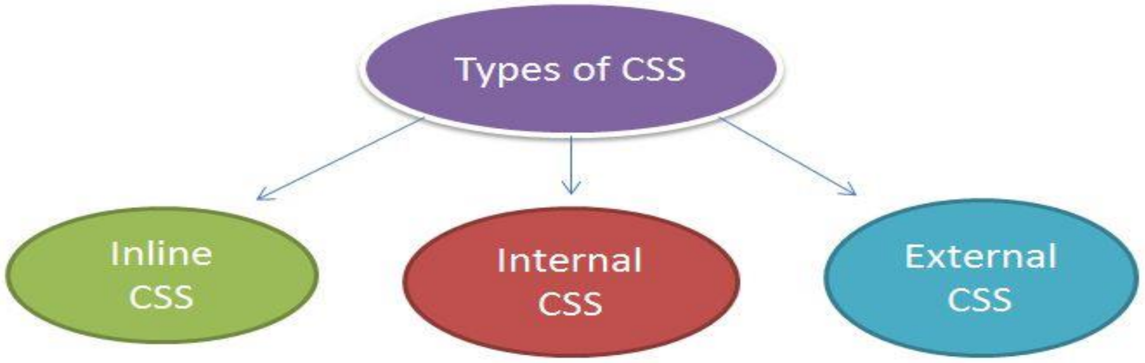
CSS- Cascading Style Sheets

- **CSS adds the styles to the HTML document**


Content + Style = Document

3 WAYS TO ADD CSS TO HTML:

THREE TYPES OF CSS


Types of CSS

Inline CSS

Internal CSS

External CSS

CSS can be added to HTML elements in 3 ways:

1.Inline - by using the **style** attribute in HTML elements

Applied to only one element

2. Internal - by using a **<style>** element in the **<head>** section

Applied to entire web page

3. External - by using an **external .CSS file**

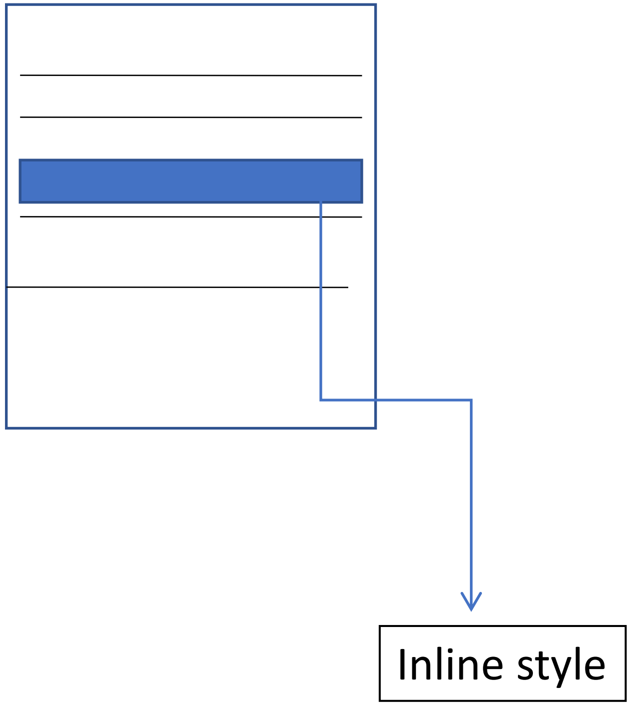
CSS- Cascading Style Sheets

1. Inline CSS

- Inline CSS is used to apply CSS on a **single line or element.**
- An inline CSS uses the **style** attribute of an HTML element.

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

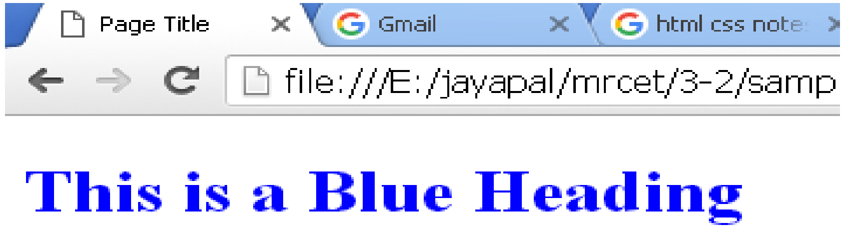
This example sets the **text color** of the < h1> element to blue:



```
<html>
<body>

<h1 style="color:blue;">This is a Blue Heading</h1>

</body>
</html>
```



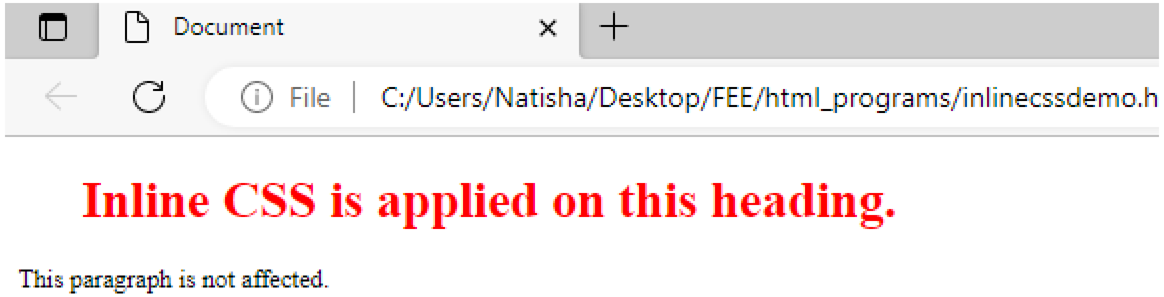
This is a Blue Heading

CSS- Cascading Style Sheets

1. Inline CSS

Style is applicable to only one tag

```
<html>
<body>
<h1 style="color:red;margin-left:40px;">Inline CSS is applied on this heading.</h1>
<p>This paragraph is not affected.</p>
</body>
</html>
```



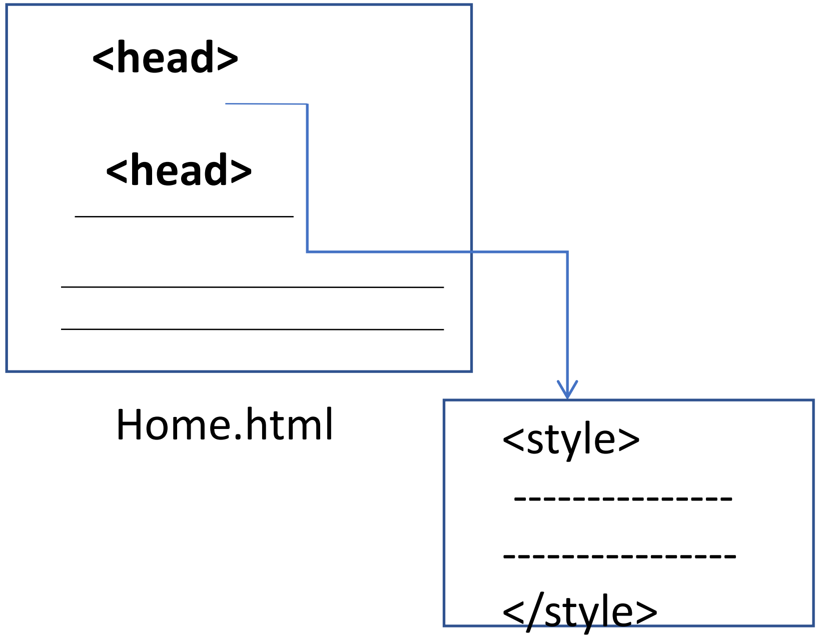
C:/Users/Natisha/Desktop/FEE/html_programs/inlinecssdemo.h

Inline CSS is applied on this heading.

This paragraph is not affected.

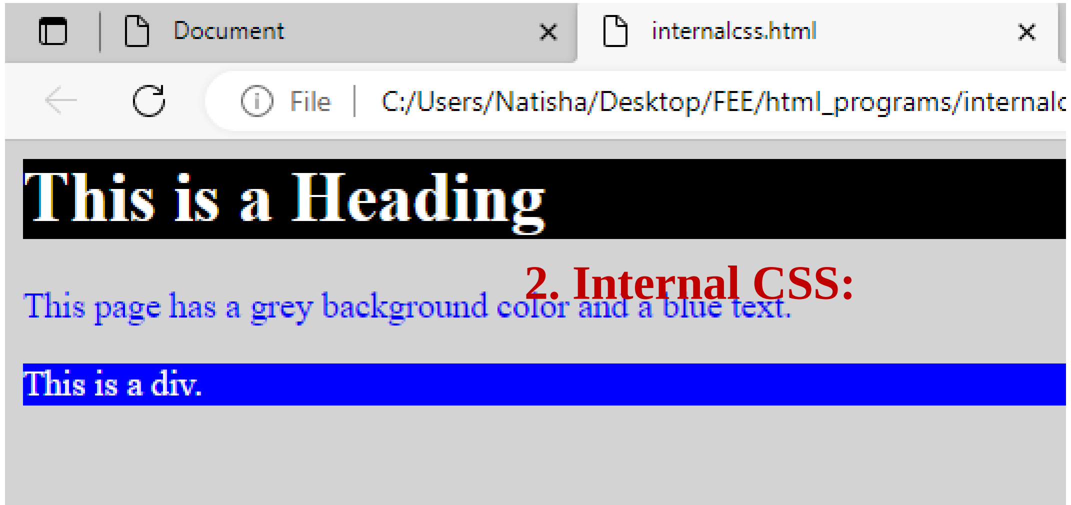
2. Internal CSS: An internal CSS is used to define a **style for a single HTML page**.

The internal style is defined inside the **<style>** element, inside the **<head>** section.



```
<html>
<head>
<style>
  body {background-color: blue;}
 h1 {color: blue;}
 p{color: red;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

2. Internal CSS



This is a Heading

This page has a grey background color and a blue text.

This is a div.

2. Internal CSS:

```
<html>
<head>
 <style>
                    body {
                              background-color: lightgrey;
                             color: blue;
                             }
                        h1 {
                             background-color: black;
                             color: white;
                             }
                       div {
                             background-color: blue;
                             color: white;
                             }
</style>
</head>
<body>
<h1>This is a Heading</h1>
<p>This page has a grey background color and a blue text.</p>
<div>This is a div.</div>
</body>
</html>
```

CSS SELECTORS

- **A CSS selector selects the HTML element(s) you want to style.**

- There are several different types of selectors in CSS.

1. Element Selector(element name)
2. Id Selector(#)
3. Class Selector(.)
4. Universal Selector(*)
5. Group Selector(h1,p,.)

1) CSS Element Selector

- The element selector selects the HTML element by name.
- The style is applicable to the tags that are same .

```
<!– EEMENT SELECTOR-->
<html>
<head>
<style>
 p{
   text-align: center;
   color: blue;
}
</style>
</head>
<body>
<p>This style will be applied on every
paragraph.</p>
<p >here also applie</p>
<p>And me!</p>
</body>
</html>
```

CSS SELECTORS

2) CSS Id Selector(#)

- The **id** selector selects the **id** attribute of an HTML element to select a specific element.
- An id is always unique within the page so it is chosen to select a single, unique element.
- It is written with the **hash character (#),** followed by the id of the element.

```
<html>
<head>
<style>
#para1 {
   text-align: center;
   color: blue;
}
</style>
</head>
<body>
<p id="para1">applicable to this element only</p>
<p>This paragraph will not be affected.</p>
</body>
</html>
```

CSS SELECTORS

3) CSS Class Selector

- The class selector **selects HTML elements with a specific class attribute.**
- It is used with a **period character** . (full stop symbol) followed by the class name.
- A class name should not be started with a number.

```
<html>
<head>
<style>
.center {
   text-align: center;
   color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is blue and center-aligned.</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```

CSS Class Selector for specific element

If you want to **specify that only one specific HTML element should be affected**, then you should use the element name with class selector.

```
<html>
<head>
<style>
p.center {
   text-align: center;
   color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is not affected</h1>
<p class="center">This  is blue and center-aligned.</p>
</body>
</html
```

CSS SELECTORS

4) CSS Universal Selector (*)

- It selects all the elements on the pages.

```
<html>
<head>
<style>
* {
  color: green;
  font-size: 20px;
  }
</style>
</head>
<body>
        <h2>This is heading</h2>
        <p>This style will be applied on every paragraph.</p>
        <p id="para1">Me too!</p>
        <p>And me!</p>
</body>
</html>
```

CSS SELECTORS

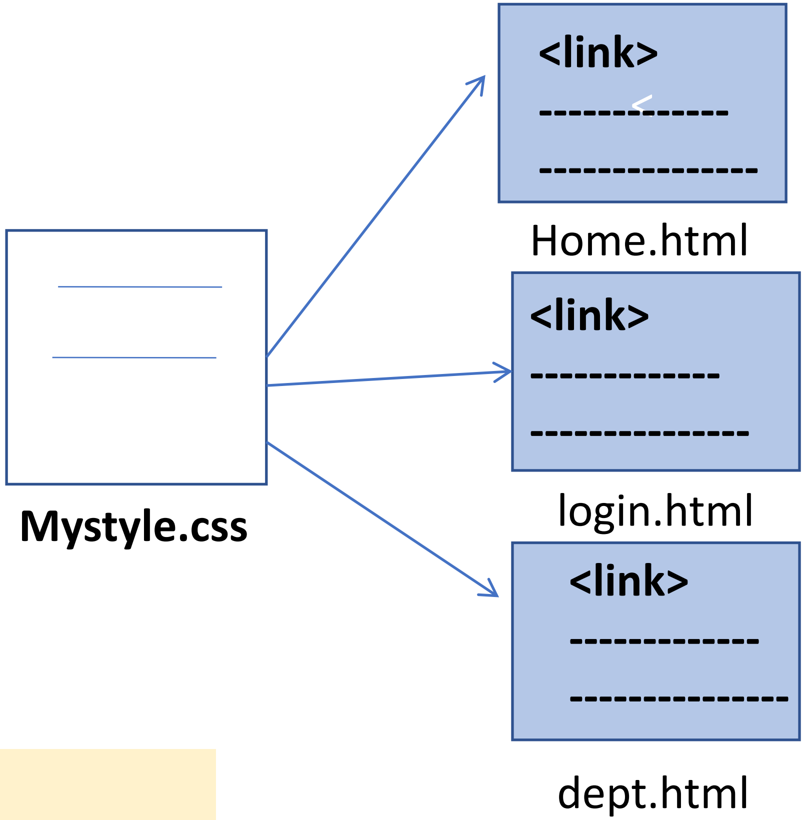
5) CSS Group Selector:

- The grouping selector is used to select all the elements with the same style definitions.
- Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

```
<html>
<head>
<style>
h1, h2, p {
                text-align: center;
                 color: blue;
               }
</style>
</head>
<body>
        <h1>Hello..This is heading one /h1>
        <h2>Hello   this is  heading two </h2>
        <p>This is a paragraph.</p>
</body>
</html>
```

3. External CSS:-

- An external style sheet is used to **define the style for many HTML pages**.
- With an external style sheet, we can **change the look of an entire web site**, by changing **one file**
- An External style sheet is written in **separate file with extension .css** and **referenced in multiple HTML documents**
- To use an **external style sheet**, add a **<link>** in the <head> section of the HTML page:



Syntax to Link external CSS file:

<link rel="stylesheet" type="text/css" href="mystyle.css">

CSS- Cascading Style Sheets

- An external style sheet can be written in **any text editor,** and must be saved with a **.css extension**.
- The external **.css file** should not contain any HTML tags.

```
body
{background-color:lightblue;
}
h1{
    color:navy;
    margin-left:20px;
}
```

"mystyle.css“

```
<html>
<head>
        <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

“home.html“

Attributes of <link> tag:

1. The **“rel”** attribute is **compulsory** and it is used to **specify the relationship** between the current file and the linked file.
2. The **“href”** → it is **compulsory** used to **specify the file** location.
3. The “**type**” →attribute is **optional**, it is used to **define the type of content** that we are linking.

CSS Properties

- **CSS Fonts:**
  - **color:** The CSS color property **defines the text color** to be used.

```
h1 { color: red; }     //specify color name
h2 { color: #9000A1; }   //specify by hexadecimal value
p { color:rgb(0, 220, 98); }   //by RGB color combinations
```

  - **font-family:** The CSS font-family property defines the **font to be used.**
  - **font-size** property defines the **text size** to be used.

```
font-size: 30px; or  font-size: 200%;
```

  - **font-style:** The CSS font-family property defines the **font to be used**.

```
font-style: italic
```

- **border**: defines a border around an HTML element.
  - border-style {border-style: none;} {border-style: dotted;}
  - border-color border-color: red;
  - border-width border-width: 1px;
- **padding**: defines a padding (space) between the text and the border.

{ padding-top: 50px; padding-right: 100px; padding-bottom: 150px; }

- **margin:** defines a margin (space) outside the border.

{ margin-top: 50px; margin-bottom: 50px; margin-right: 100px; }

```
h1
   { color: blue;
    font-family: verdana;
  }
p {
 color: red;
  font-family: courier;
  font-size: 160%;
 }
```

```
h1{ color: blue;
    font-family:'Courier New';
 }
p { color: red;
  font-size: 200%;
  border: 2px solid powderblue;
  padding: 30p x;
  margin: 50p x; }
```

CSS Background

- CSS background property is used to define the background effects on element.

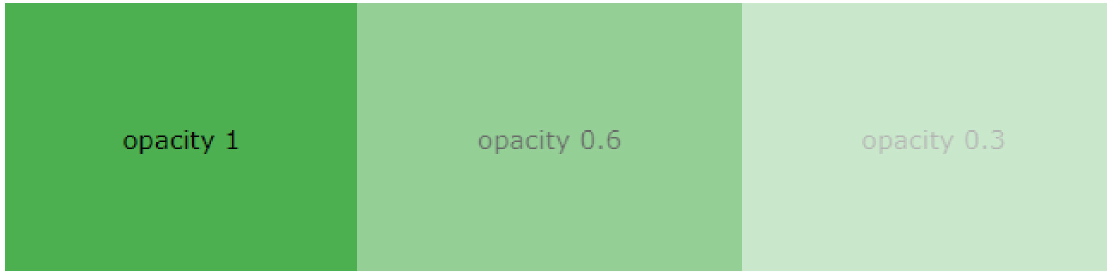
CSS background properties: 5 BG properties

1. background-color

```
background-color: lightblue;
```

Opacity / Transparency

```
div {
  background-color: green;
  opacity: 0.3;
}
```



opacity 1

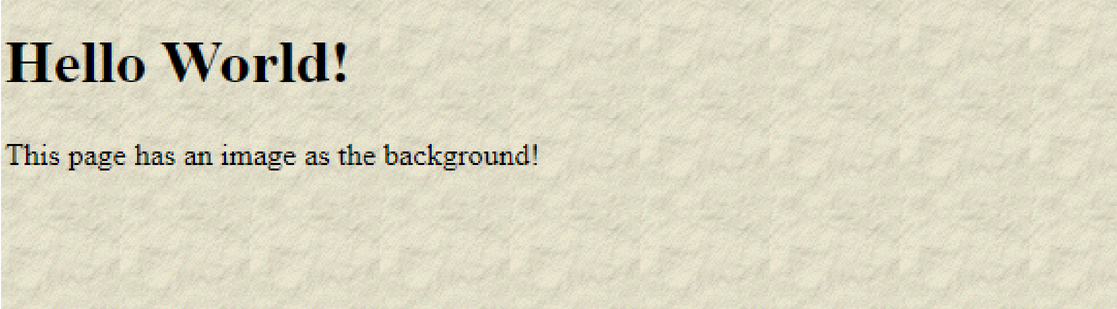
opacity 0.6

opacity 0.3

2 . background-image:

- specifies an image to use as the background of an element
- By default, the image is repeated so it covers the entire element.

```
body {
  background-image: url("paper.gif");
}
```

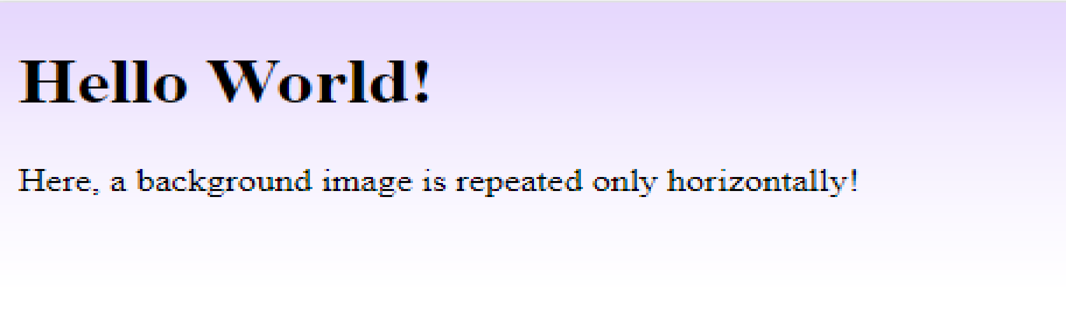


Hello World!

This page has an image as the background!

3. . background-repeat: repeats the background image horizontally and vertically.

```
body {        background-image: url("gradient_bg.png");
background-repeat: repeat-x;      }
```



Hello World!

Here, a background image is repeated only horizontally!

CSS Background

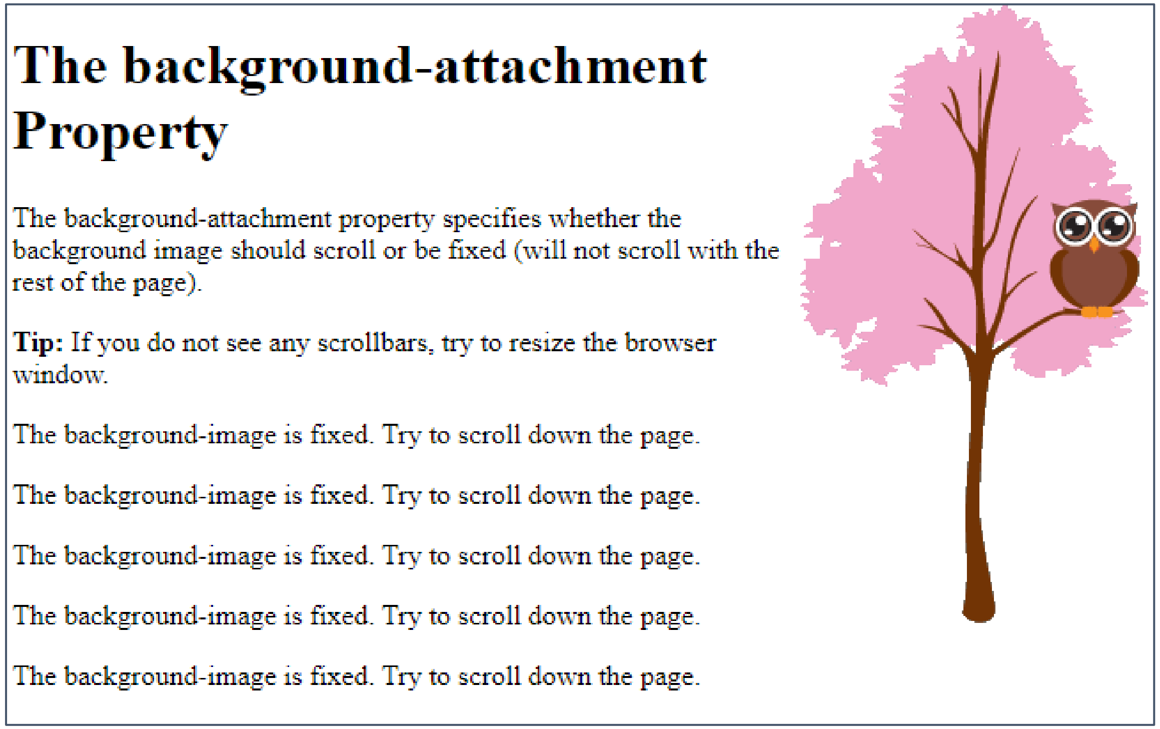
- CSS background property is used to define the background effects on element.

3. **background-attachment** → specifies whether the background image should scroll or be fixed

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  margin-right: 200px;
  background-attachment: fixed;
}
```

3. **background-position** → specifies initial position of the background image

  - center
  - top
  - bottom
  - left
  - right



The background-attachment Property

The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).

Tip: If you do not see any scrollbars, try to resize the browser window.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

CSS- Multiple Classes

- Multiple classes can be applied to a single HTML element by **adding each class name** to the class attribute, **separated by a space.**

```
<style>
   .class_1{
          /* some styles */
       }

   .class_2{
      /* some styles */
   }
</style>
```

Here two classes are defined

```
<tag_name class="class_1 class_2">
```

Here the two class style are applied to tag

```
<style>
          .para1 {
                    font-size: larger;
                    margin-bottom: 35px;
                    background-color: lightgreen;
                 }
          .para2 {
                    color: red;
                 }
</style>
<body>
          <p class="para1"> only one style is applied </p>
          <p class="para1 para2">Two classes styles are applied
          </p>

</body>
```

SPECIFICITY IN CSS

- Specificity in CSS is decides which styles to apply when multiple styles are defined for the same element
- i.e. if there are two or more CSS styles applied to the same element, then the selector with the **highest specificity value will be applied to that HTML element.**
- **[DIFFERENT SELECTRS are id selector(#), class selector(.), element selector, etc.]**

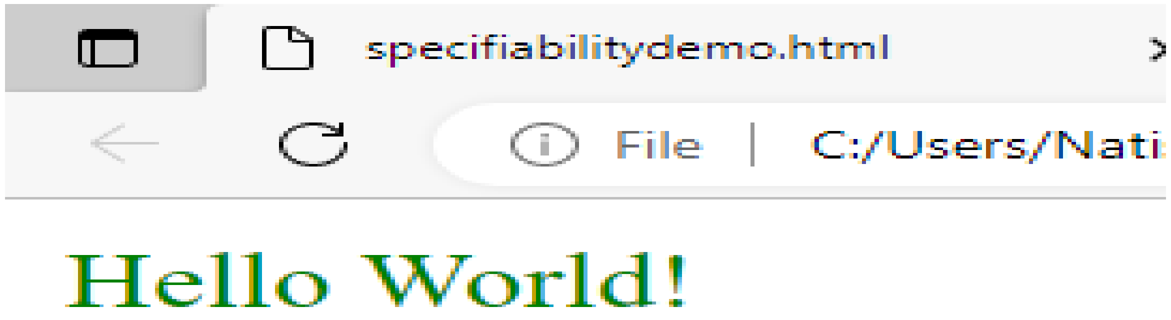
```
<html>
<head>
<style>

  .test {color: green;}   /*class selector*/

  p {color: red;}         /*element selector */

</style>
</head>
<body>

<p class="test">Hello World!</p>
            <!- - here the class selector is applied -- >
</body>
</html>
```



Hello World!

SPECIFICITY FOR DIFFETENT SELECTORS

//ID SELECTOR IS GIVEN HIGHER PRIORITY in [ID, CLASS, ELEMENT]

Ex1:

```
<html>
<head>
<style>
       #demo {color: blue;}  /* id selector */
       .test {color: green;}     /* class selector */
       p {color: red;}             /* element selector*/
</style>
</head>
<body>
<p id="demo" class="test">Hello World!</p>

</body>
</html>
```

- Here **ID selector is given higher priority,** **so Hello World will be displayed in Blue**

//INLINE STYLE IS GIVEN THE HIGHEST PRIORITY IN [ID ,CLASS, ELEMENT, INLINE}

Ex 2.

```
<html>
<head>
<style>
       #demo {color: blue;} /* id selector */
       .test {color: green;}   /* class selector */
        p {color: red;}        /* element selector*/
</style>
</head>
<body>
<p id="demo" class="test" style="color: pink;">Inline style</p>
                                       <! - - inline selector  is appled-->
</body>
```

- Here we have added an inline style for the "p" element.
- The text will now be pink, because the **inline style is given the highest priority**

Specificity hierarchy

- Each selector has a place in the hierarchy.

FOUR categories define the selector's specificity level:

PRIORITY 1: INLINE STYLES (highest priority)

Example: **<h1 style="color: pink;">**

PROORITY 2: ID SELECTOR (#)

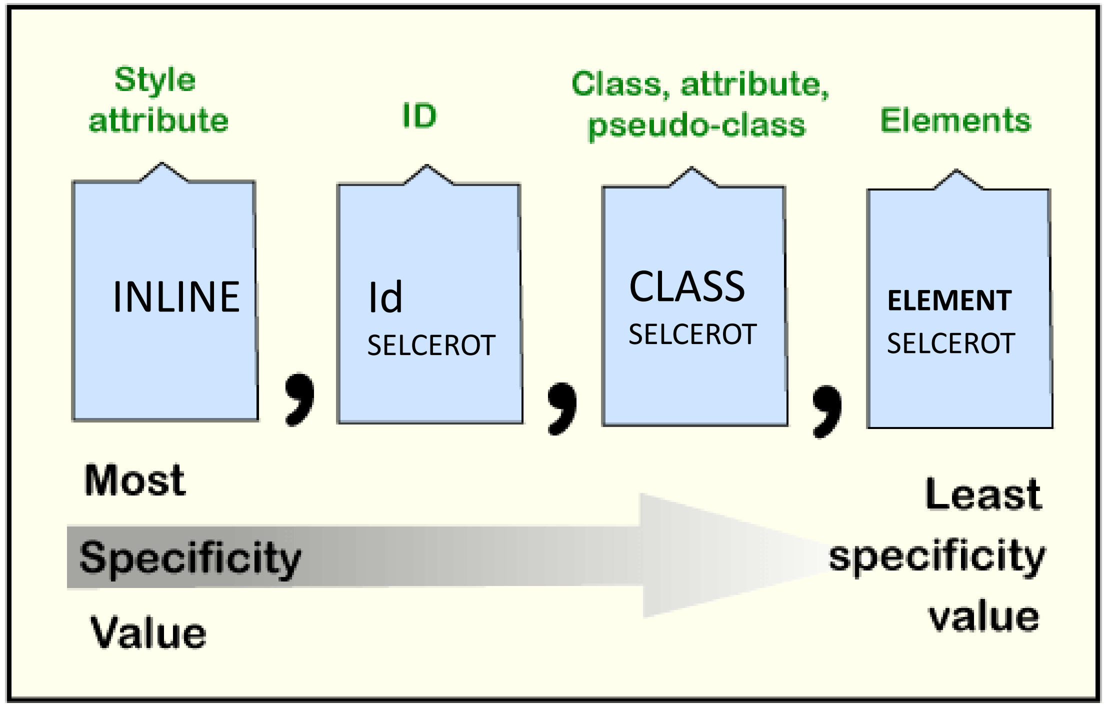
- Example: **#navbar**

PROIRITY 3: CLASS SELECTOS, pseudo-classes, attribute selectors -

Example: **.test,** :hover, [href]

PRIORITY 4 ELEMENT SELECTOR and pseudo-elements -

Example: **h1, ::before**



CSS Pseudo-classes

- A pseudo-class is used to define a **special state of an element.**

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

```
selector:pseudo-class {
  property: value;
}
```

```
/* unvisited link */
a:link {
  color: #FF0000;
}

/* visited link */
a:visited {
  color: #00FF00;
}

/* mouse over link */
a:hover {
  color: #FF00FF;
}

/* selected link */
a:active {
  color: #0000FF;
}
```

Specificity rules

- **Equal specificity: the latest rule wins -** If the same rule is written twice, **then the latest(last) rule will be applied.**

```
    h1 {background-color: yellow;}
    h1 {background-color: red;}   /* highest priority */
```

- The specificity of **class selector is greater than the element selectors**

```
<style>
    .intro {background-color: yellow;}
    h1 {background-color: red;}
</style>
<body>
<h1 class="intro">This is a heading</h1>
</body>
```

- **The specificity of ID selectors is higher than attribute selectors**

```
        div#a {background-color: green;} /* highest priority */
        #a {background-color: yellow;}
        div[id=a] {background-color: blue;}
```

- The universal selector (*) and inherited values have a specificity of 0 - The universal selector (*) and inherited values are ignored!

CSS Combinators

- A combinator is something that explains the **relationship** between the selectors.

FOUR DIFFERENT COMBINATORS IN CSS:

1. Descendant selector (space)

div p → <p> is descendant of <div>

2. child selector (>)

div > p --> <p> is child of <div>

3. adjacent sibling selector (+)

div + p ---> <p> is adjacent sibling of <div>

4. general sibling selector (~)

div ~ p --> <p> is general sibling selector

```
<div>      <!-- Descendant selector-- >
 <p>Para 1 in the div.  It is descendant  of div </p>
 <p>Para 2 in the div. It is descendant  of div </p>
  <section>    <p>Para3 in the div. it  is descendant</p>    </section>
</div>
```

```
<div>              <!-- child selector -->
 <p>Para 1 in the div. it is child of div</p>  → CHILD OF DIV
 <p>Para 2 in the div. it is child of div </p> → CHILD OF DIV
  <section>    <p>Para 3 in the div. but not a child</p>    </section>
</div>
```

```
<div>  <!-- adjacent sibling selector -->
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>
<p>Para 3. After a div. it is adjacent sibling</p>  → ADJUSENT SIBLING
```

```
<div>   <p>Paragraph 2.</p> </div>     <!--  general sibling selector -->
<p>Paragraph 3.</p>     →  GENERAL SIBLING
    <code>Some code.</code>
<p>Paragraph 4.</p>  → GENERAL SIBLINE
```

CSS Combinators

1.DESCENDANT SELECTOR(SPACE)

- The descendant selector **matches all elements that are descendants** of a specified element.
- The example selects all <p> elements inside <div> elements:

```
<!– DESCENDANT SELECTOR -- >
<html>
<head>
<style>
div p       /* p is descendant of div*/
 {
  background-color: yellow;
}
</style>
</head>
<body>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>
</body>
</html?
```

CSS Combinators

2. Child Selector (>)

- The child selector **selects all elements that are the children** of a specified element.
- The example selects all **<p>** elements that are **children of a <div>** element:

Example

```
div > p
{
  background-color: yellow;
}
```

```
<style>
div > p      /* p is child  of div*/
{
 background-color: yellow;
}
</style>
.
.
<div>
 <p>Paragraph 1 in the div.</p>
 <p>Paragraph 2 in the div.</p>
 <section>   <!-- not Child but Descendant -->
  <p>Paragraph 3 in the div (inside a section element).</p>
 </section>
 <p>Paragraph 4 in the div.</p>
</div>
```

CSS Combinators

3. Adjacent Sibling Selector (+)

- →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".

```
<style>
div + p    /* p is adjacent sibling of div*/
{

  background-color: yellow;
}
</style>
```

```
..
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>
<p>Paragraph 3. After a div.</p>
<p>Paragraph 4. After a div.</p>
<div>
  <p>Paragraph 5 in the div.</p>
  <p>Paragraph 6 in the div.</p>
</div>

<p>Paragraph 7. After a div.</p>
<p>Paragraph 8. After a div.</p>

  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section>   <!-- not Child but Descendant -->
    <p>Paragraph 3 in the div (inside a section element).</p>
  </section>
  <p>Paragraph 4 in the div.</p>
</div>
```

CSS chaining Selectors

- Chaining selectors is a technique in CSS where **multiple selectors are combined to apply styles to elements that meet all the specified criteria.**

- Chaining selectors uses the combinators.

Ex.

.container > p ,

Applies on the **p** element that is a **direct child** of an element with **class "container"**.

- It applies a font size of 24px to the **p** element.

```
.container > p      /* chaining selector */
  {
  font-size: 24px;
}

<div class="container">

         <p>Paragraph 1</p>
         <p>Paragraph 2</p>
</div>
```

CSS chaining Selectors

- **.container p + ul li**,

Applies style on the

1. <**li> elements that are direct children of a <ul>**
2. **<uL> element** that **is adjacent sibling of <p> element**
3. **<P>** that is a **descendant of an element with class = "container**".

- It applies a red color to these li elements.

```
.container p + ul li
 {
  color: red;
}
```

```
<div class="container">
  <h1>Title</h1>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <ul>
    <li>List item 1</li>
    <li>List item 2</li>
  </ul>
</div>
```

CSS chaining Selectors

- Chaining selectors can also be used with classes, IDs, and attribute selectors to target specific elements with a particular set of properties.

- .container .special
- we first specify the parent element with a class of **"container",** then we specify the child element with a class of **"special“**
- **The style is applied on Paragraph2**

```
.container .special
 {
  font-weight: bold;
  color: blue;
}

<div class="container">
  <h1>Title</h1>
  <p>Paragraph 1</p>
    <p class="special">Paragraph 2</p>
</div>
```

Nested Elements in css

- In CSS, nested elements refer to the **use of selectors** that are **nested within other selectors** to target specific elements within the HTML document.

- Selectors can be nested using the **parent-child relationship.**
- For example, you can target all <**li>** elements that are children of a **ul** element using the following selector:
- **ul li**

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>
    Item 3
    <ul>
      <li>Subitem 1</li>
      <li>Subitem 2</li>
    </ul>
  </li>
</ul>
```

```
ul li ul li {
  color: blue;
}
```