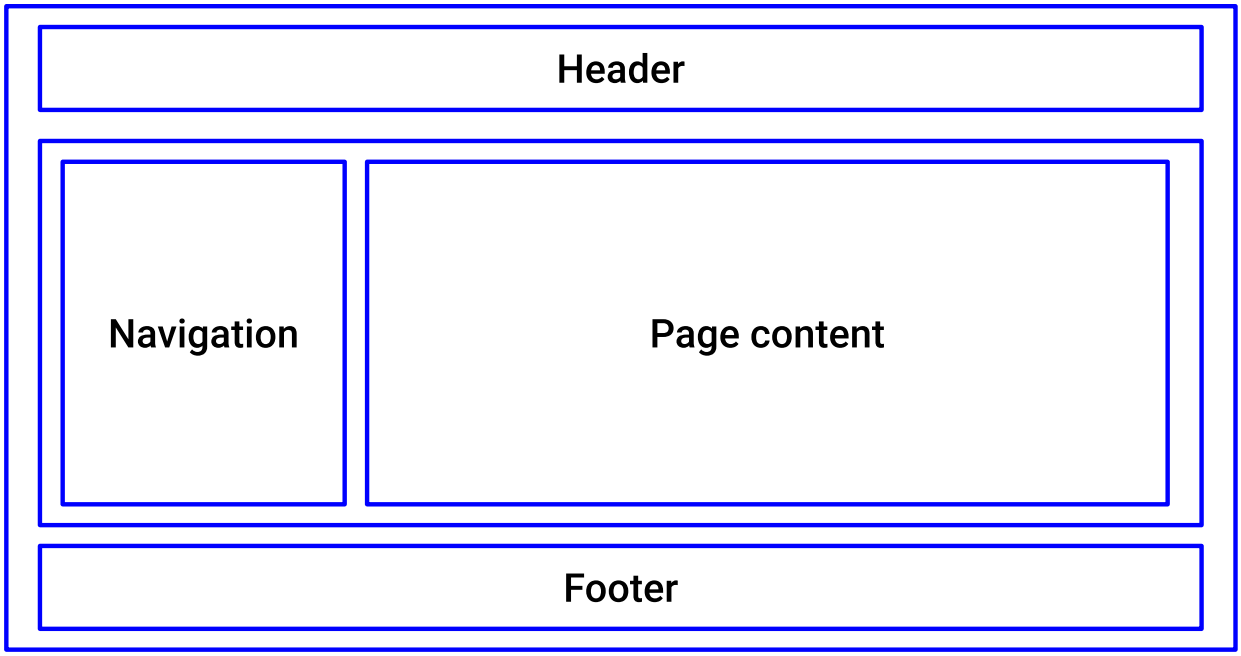
Important Concepts in CSS

Dr Harshad Prajapati
13 Nov 2023

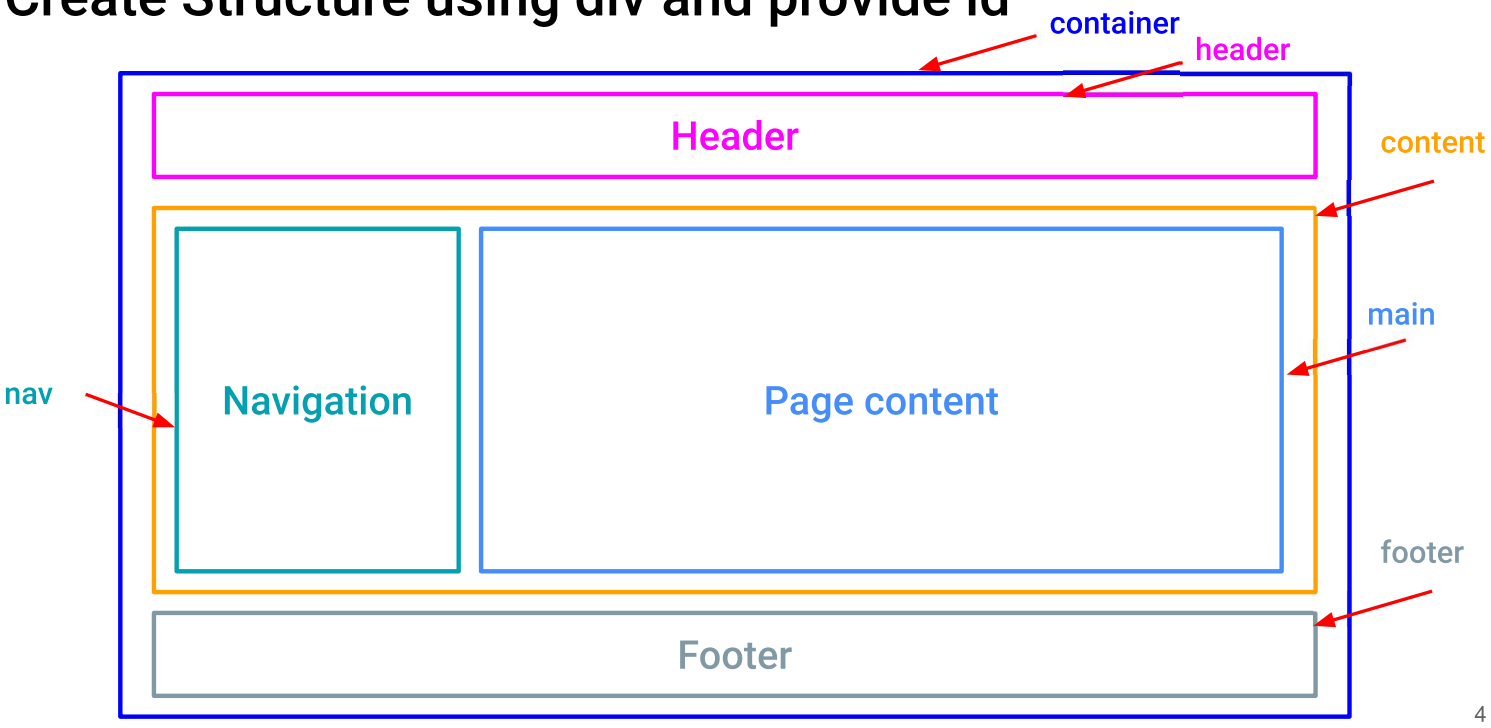
How is CSS Used?

Typical Webpage Structure



When semantic HTML elements were not available, developers were using for sectioning a document.

Create Structure using div and provide id



HTML Code for Structure of the Document

article.html

```
<html>
    <head>
        <title>Demo for website design</title>
    </head>
    <body>
        <div id="container">
            <div id="header">
                <h1>Demo for website design</h1>
            </div>
            <div id="content">
                <div id="nav">
                <h3>Navigation</h3>
                <ul>
                    <li><a href="" >Home</a></li>
                    <li><a href="">About</a></li>
                    <li><a href="">Contact</a></li>
                </ul>
            </div>
```

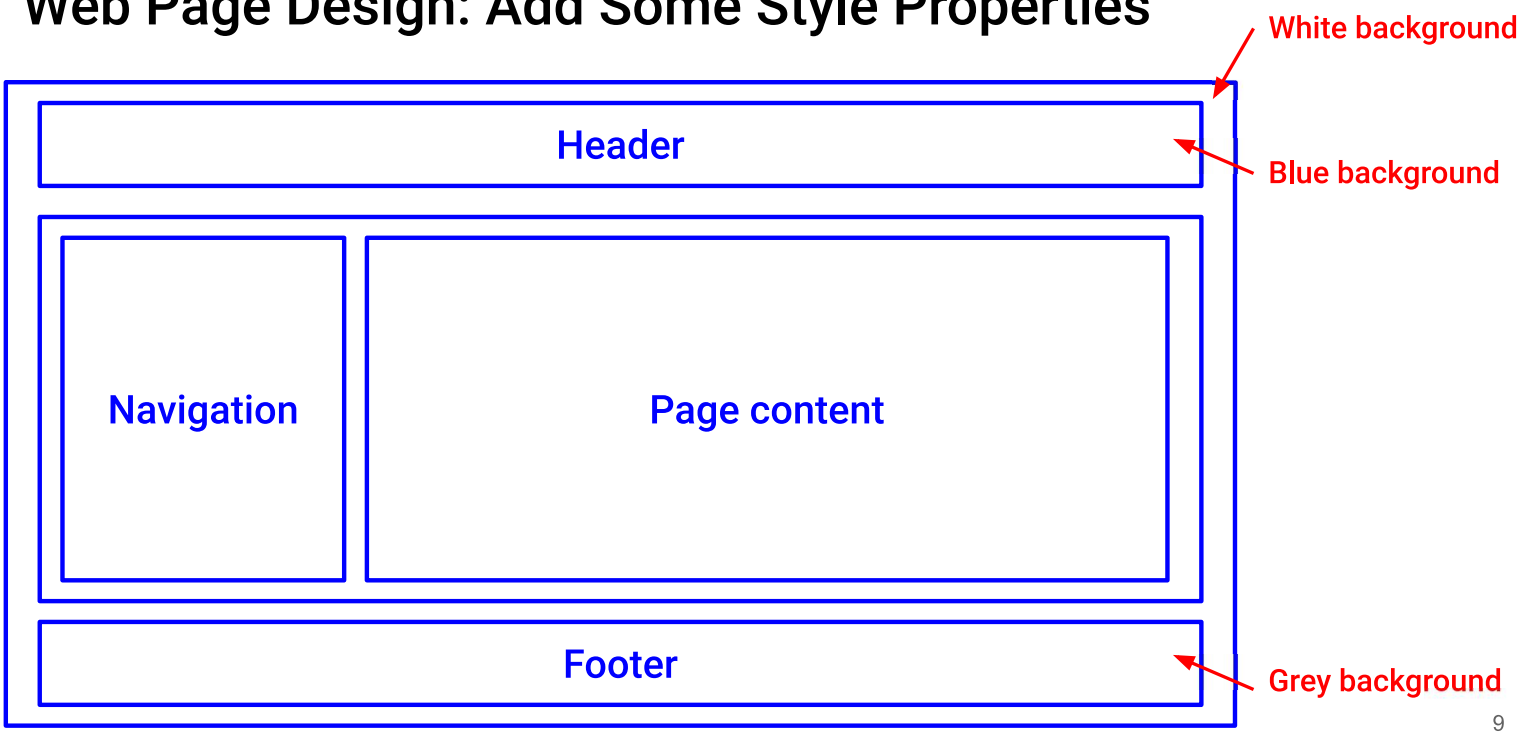
Developers used to use for sectioning.

HTML Code for Structure of the Document

```
19                  <div id="main">
20                      <h3>Main content</h3>
21                          Here is main content.
22                  </div>
23              </div>
24              <div id="footer">
25                  Copyright &copy; 2023 Demo Website
26              </div>
27          </div>
28      </body>
29  </html>
```


Look of Webpage Without any Style

Web Page Design: Add Some Style Properties



Create a Style File in the Same Folder

styles.css

```css
body {
    background-color:#EEEEEE;
    font-family: Helvetica, Arial, sans-sarif;
}

a {
    text-decoration: none;
    color: red;
}

#nav ul {
    list-style-type: none;
    padding: 0;
}

h1,h2, h3 {
    margin:0;
}
```

Without any special symbol at the front indicate, element or tag selector

With a # symbol at the front indicate, id selector


```css
#header{
    background-color: #66CCFF;
    color:white;
    text-align: center;
    padding: 10px;
}

#container{
    background-color: white;
    width: 800px;
    margin-left: auto;
    margin-right: auto;
}

#content{
    padding:10px;
}

#nav{
    width:180px;
    float:  left;
}
```

```css
#main{
    width:600px;
    float: right;
}

#footer{
    clear:both;
    padding: 0px;
    background-color: #999999;
    color: white;
    text-align: center;
}
```


Add Link to Stylesheets in HTML

article.html

Add this code

```html
<html>
    <head>
        <title>Demo for website design</title>
        <link rel="stylesheet" type="text/css" href="styles.css" />
    </head>
    <body>
        <div id="container">
            <div id="header">
                <h1>Demo for website design</h1>
            </div>
            <div id="content">
```

Same Webpage after Styling

Webpage before styling

Webpage after styling

Cascading Style Sheets (CSS)

Why CSS?

- A webpage should have separation of data from its presentation.
- HTML:
  - HTML is used to structure the document.
- CSS: Cascading Style Sheets:
  - CSS is used to present the data.

Advantages of CSS

- Saves development time:
  - We can prepare CSS and use it in multiple HTML pages of a web app.
- Pages load faster:
  - Same presentation code (.css or style part) can be used for multiple occurrences of a tag. (less code, so faster loading)
- Easy Maintenance due to global changes:
  - Changes at one place (.css) can affect to all HTML pages of a web app.
- Compatibility of same web app for multiple devices:
  - Desktop, Laptop, Smartphone, Tablet
  - Screen, Projection, Print, etc.

Brief History of CSS

- Early Development (1994-1996):
  - Håkon Wium Lie working at CERN proposed the idea of CSS in 1994.
  - Bert Bos responded to the first draft.

Håkon Wium Lie, 12 December 1995

Bert Bos, 12 December 1995

Source: https://www.w3.org/Style/CSS20/history.html

Brief History of CSS

- CSS Level 1 (1996):
  - Cascading Style Sheets Level 1 (CSS1) as W3C recommendation on December 17, 1996.
- CSS Level 2 (1998-2009):
  - CSS2 as W3C recommendation on 12 May 1998.
    - There were several revisions.
- CSS Level 2 Revision 1 (CSS 2.1) (2011):
- CSS3 (2001-present):
  - The development of CSS3 has been modular, with individual modules addressing specific features.

Source: https://www.w3.org/Style/CSS20/, https://www.w3.org/Style/CSS/

Elements in HTML and CSS

- In HTML and CSS, elements are classified as either inline or block-level.

Inline elements:	Block-level elements:
take up only required space.	take up the full width available.
do not start on a new line.	start on a new line.
generally accept width and height values.	can accept width and height values
can be nested within block-level or inline elements.	can contain both inline and block-level elements.
Examples: , <a>, <img>, , form elements (<input>)	Examples: , <p>, <h1> to <h6>, <table>, ,

Categories of various style properties

- Box Model
- Typography
- Colors and Backgrounds
- Layout
- Flexbox
- Grid
- Positioning
- Transform
- Transitions and Animations
- Responsive Design
- Lists
- Tables
- User Interface
- Print Styles

Units of Length

- There are many units. Only widely used are shown:
  - px: pixels. One dot on screen.
  - em: a relative measurement. If the font size of an element is 16px, 1em is equal to 16px.
  - rem: rem is relative to the font size of the root element.
  - %: percentage is relative to the parent element. If we specify width as 50%, it takes up 50% of the parent width.
  - vw (viewport width), vh (viewport height), vmin (smaller of vw or vh), vmax (smaller of vw or vh): 1vw is 1% of viewport width.
- No space is allowed between the number and the unit specification
  - For example, 2.5in is legal, whereas 2.5 in is illegal.

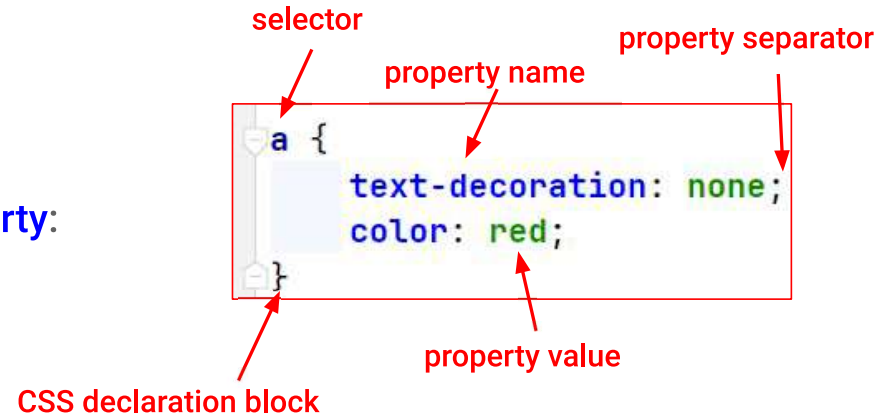
Syntax of CSS

- Syntax:
  - selector {declarations}
- For single property:

selector {property:value}

- For using more than one property:

```
selector {
        property2:value;
        property2:value;
        property3:value;
}
```



Levels of Style Definition

- There are three levels at which we can define styles for elements:
  - Inline.
  - Embedded.
  - External.

Inline Style

- Inline style is tag level.
- Style is assigned to the element that is defining the style.
  - Inline style is used to modify the appearance of a particular element.
- Example:
  - <h1 style="background-color:yellow; font-family:'Arial' ">
    - If property-value is a string, we can use single quotes to enclose it.
- The style attribute of an element contains key-value pairs of style properties, separated by semicolon, and whole style enclosed within double quotes.

Embedded Style

- Embedded style is page level.
- Style is defined within <style> block in html page itself.

```
< style type="text/css">
    h1 {
        background-color:yellow;
        font-family:'Arial'
    }
</style>
```

- This style can be used by multiple elements within the page in which it is defined.

External Style

- External style is website level. It is also called linked style.
- Style is defined in a separate file that has extension as .css.
- This style file can be used by multiple pages within web application.
    - If style file is accessible on Internet, it can be used by any other web applications.
        - Example: Bootstrap and Tailwind CSS.
- If the name of a style file is say styles.css, it can be embedded in a page using the following:
    - <link rel="stylesheet" type="text/css" href="styles.css" />

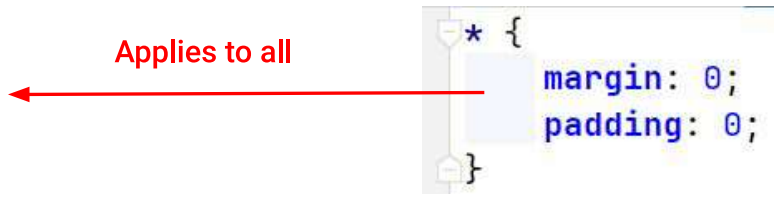
CSS Selectors

Selectors

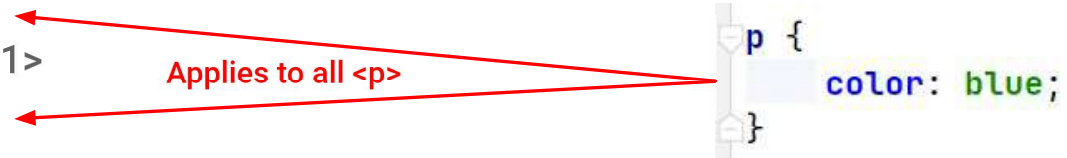
- There are different types of selectors in CSS:
  - Universal selector.
  - Type selector (tag selector).
  - Class selectors.
  - ID selectors.
  - Descendant Selector.
  - Child Selector.
  - Adjacent Sibling Selector.
  - Attribute Selector.
  - Pseudo-class selector.
  - Pseudo-element selector.

Selectors

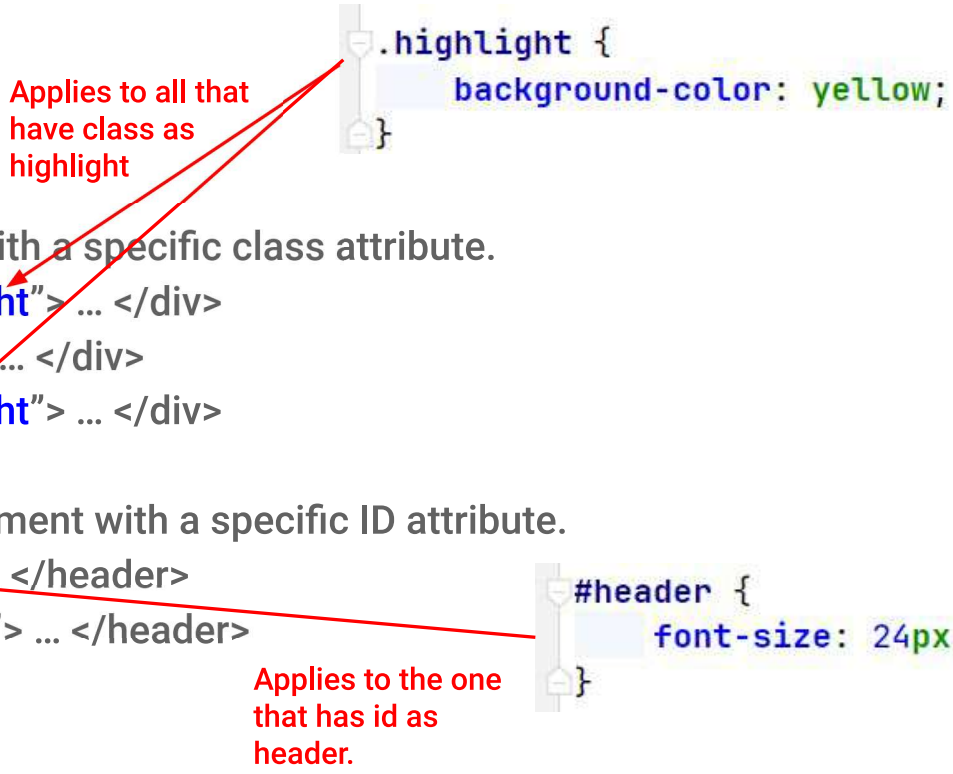
- Universal Selector (*):
  - Selects all elements on the page.
  - <body> …
    <h1> … </h1>
     …
    </body>



- Type or Element Selector:
  - Selects all instances of a specific HTML element.
  - …
    <h1> … </h1>
     …



Selectors



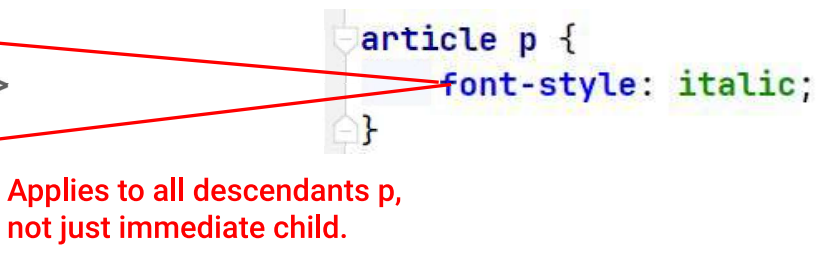
- Class Selector (.):
  - Selects elements with a specific class attribute.
  - …
     …
     …
- ID Selector (#):
  - Selects a single element with a specific ID attribute.
  - … </header>
     … </header>

Selectors

- Descendant Selector ( ):
  - Selects an element that is a descendant of another specified element.
  - It selects all elements that are descendants of a specified element, no matter how deep they are nested.
  - <article> …
     …

     …

    </article>



Selectors

- Child Selector (>):
  - A child selector selects only the immediate children of a specified element.
  - It does not target elements that are nested further than one level deep.
  - <article> …
     …

     …

    </article>

Applies to all that are immediate child of <article>

```
article > p {
    font-style: italic;
}
```

Does not apply to this as it is not immediate child of <article>
It is: article > div > p

Selectors

- Adjacent Sibling Selector (+):
  - Selects an element that is immediately preceded by a specified element.
  - <article> …
    <h2> … </h2>
     …

    <h1> … </h1>
     …
    <h2> … </h2>
     …

    </article>

Applies to all that are preceded by <h2>

```
h2 + p {
    font-weight: bold;
}
```

Does not apply to this as it is not preceded by <h2>

Selectors

- Attribute Selector ([ ]):
  - Selects elements based on the presence or value of their attributes.
- <form> …
  <input type= “radio” />
  <input type= “radio” />
  <input type= “text” />
  <input type= “checkbox” />
  </form>

Applies to all <input> that have text attribute.

```
input[type="text"] {
    border: 1px solid #ccc;
}
```

Does not apply to this <input> as it is not having text attribute.

Selectors

- Pseudo-class Selector (:):
  - Selects elements based on their state or position.
  - Pseudo-classes select and style entire elements based on their state or position.
- <nav> …

  …
  …

  </nav>

```
a:hover {
    color: red;
}
```

Applies to an <a> when we hover on the <a> element.

Selectors

- Some other pseudo-class Selector (:):

```css
li:nth-child(odd) {
    background-color: #f0f0f0;
}

input:focus {
    border: 2px solid blue;
}
```

```css
a:link {
    color:black;
}

a:visited {
    color:grey;
}

a:active {
    color:blue;
}
```


Selectors

- Pseudo-element Selector (::):
  - Selects a specific part of an element.
  - Pseudo-elements select and style specific parts of elements, allowing for more fine-grained control.
- <nav> …

  ……
  ……
  ……

  </nav>

```css
p::first-line {
    font-weight: bold;
}
```

Applies to the first line of paragraph element.

Selectors

- Some other pseudo-element Selector (::):

```
p::before {
    content: ">> ";
}

p::after {
    content: " <<";
}

p::first-letter {
    font-size: 150%;
}
```


Combining Two Selectors

- In the following, we combine element selector and class selector.

```
p.highlight {
    background-color: yellow;
}
```

Applies to only <p> elements that have class as highlight.

- The following selector (class selector), applies to all elements (not just <p>) that have highlight as class.

```
.highlight {
    background-color: yellow;
}
```

Cascading in CSS

CSS Rule for Cascade

- CSS means Cascading Style Sheets.
  - So, we need to understand meaning of Cascading word.
- When there are multiple styles definitions present for a particular element, then all those styles will apply in cascade manner.
  - First external styles.
  - Then embedded styles.
  - At last, inline styles.
- For any conflicting styles, the style that is more specific or declared later in the document will take precedence.

Example: CSS Rule for Cascade

```css
h1 {
    margin:0;
    color: red;
    background-color: black;
    font-style: italic;
}
```

This is external style for <h1>

These are embedded styles for <h1>

```html
<style>
    h1{border: 4px solid green; background-color: gold; color: pink;}
    h1{border: 4px solid black; color: green;}
</style>
```

This is inline style for <h1>

```html
<div id="container">
        <div id="header">
            <h1 style="background-color: pink;">Demo App</h1>
        </div>
```


This will be effective style.

Example: CSS Rule for Cascade

```css
h1 {
    margin:0;
    color: red;
    background-color: black;
    font-style: italic;
}
```

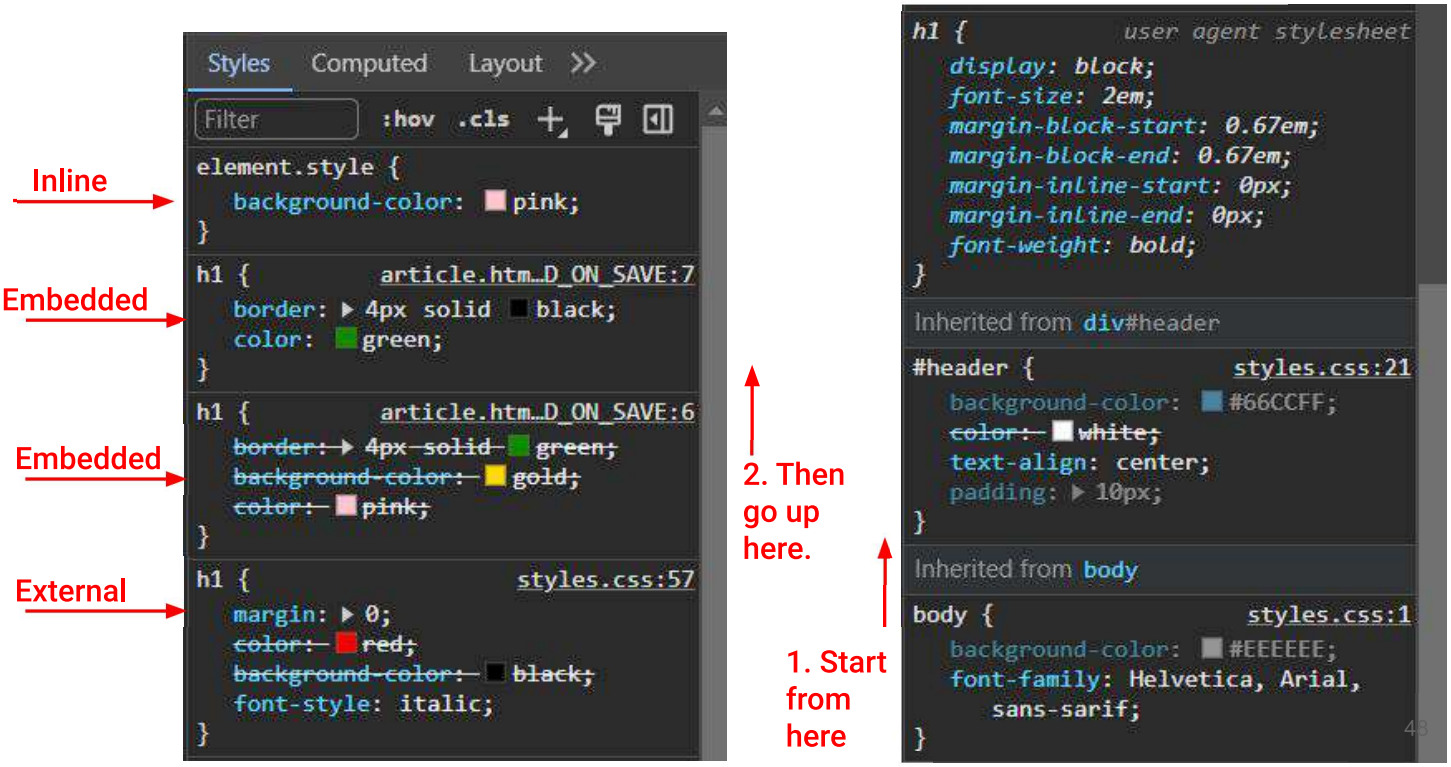
```css
{
    margin:0;
    font-style: italic;
    border: 4px solid black;
    color: green;
    background-color: pink;
}
```

```html
<style>
    h1{border: 4px solid green; background-color: gold; color: pink;}
    h1{border: 4px solid black; color: green;}
</style>
```

```html
<div id="container">
        <div id="header">
            <h1 style="background-color: pink;">Demo App</h1>
        </div>
```

Example: CSS Rule for Cascade

We can see effective styles using Developer Tool.
Right click <h1> in browser and select Inspect.

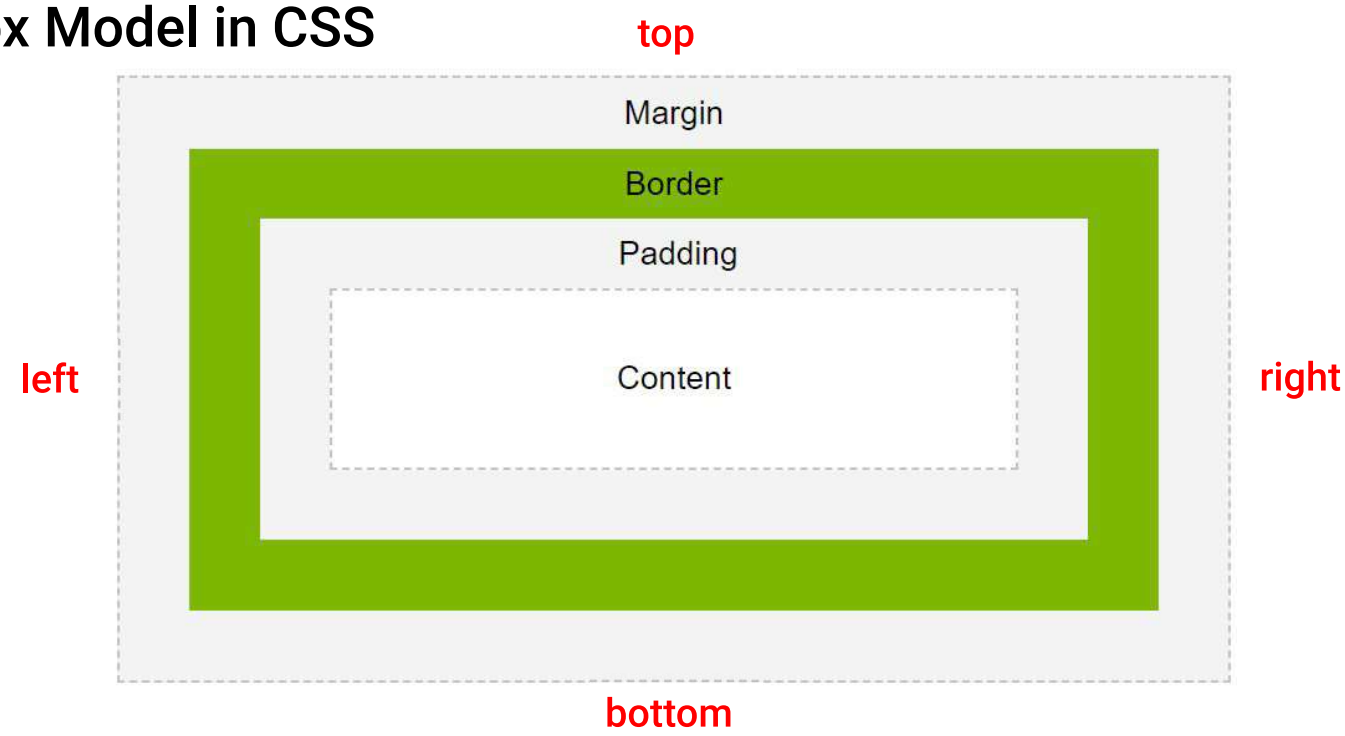


CSS Rule for Cascade

- Precedence sequence (lowest to highest)
  - Browser default.
  - Imported stylesheets.
  - External style sheets are included.
  - Embedded styles (inside the <head> tag) override external styles.
  - Inline styles (inside an HTML element) override both embedded and external styles.
  - Styles modified with JavaScript override all other styles.

Box Model in CSS

Box Model in CSS



Source: https://www.w3schools.com/css/css_boxmodel.asp

Box Model in CSS

- In CSS, **box model** is used to refer **design** and **layout**.
- A box wraps around every HTML element.
- The box has the following parts:
  - **content**
  - **padding**
  - **border**
  - **margin**

Source: https://www.w3schools.com/css/css_boxmodel.asp

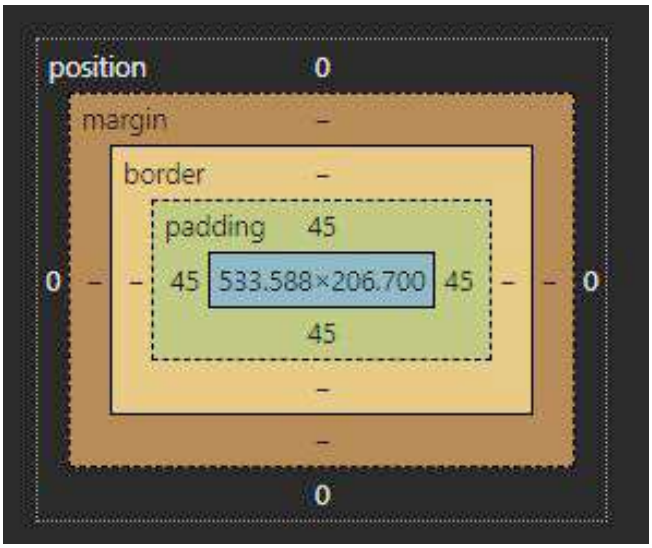
Box Model in CSS

- content:
  - Content represents the content of the box.
  - Here text and images appear.
- padding: (padding-top, padding-right, padding-bottom, padding-left)
  - Padding is an area around the content.
  - The padding is transparent.
- border: (border-top, border-right, border-bottom, border-left)
  - Border goes around the padding and content.
- margin: (margin-top, margin-right, margin-bottom, margin-left)
  - Margin is space around the border.
  - The margin is transparent.

Source: https://www.w3schools.com/css/css_boxmodel.asp

Box Model in Developer Tool of Browser

- We can inspect box model for any element in developer tool of a browser.



Important Points While Using Box Model

- The box width is the sum of the widths of the following:
  - margin-left
  - border-left
  - padding-left
  - content
  - padding-right
  - border-right
  - margin-right
- The box height is the sum of the heights of the following:
  - margin-top
  - border-top
  - padding-top
  - content
  - padding-bottom
  - border-bottom
  - margin-bottom

Important Points While Using Box Model

- We can write value for all four direction using single property.
- We need to write values for four direction starting from top then go in clockwise direction.
  - top, right, bottom, left.
- Vertical margins of two block boxes collapse.
  - If there are two block elements, then margin-bottom of the first element will collapse with the margin-top of the second element.

Important Points While Using Box Model

- How value for a single property works?
  - One value: assign this value to all four properties (directions).
  - Two values: assign first value to top and bottom properties and second value to right and left properties.
  - Three values: assign first value to top, second to right, and third to bottom.
  - Four values: assign first value to top, second to right, third to bottom, and fourth to left.

References

- https://developer.mozilla.org/en-US/docs/Web/CSS
- https://www.w3.org/Style/CSS20/history.html
- https://www.bu.edu/lernet/artemis/years/2020/projects/FinalPresentations/HTML/historyofcss.html