Cascading Style Sheets

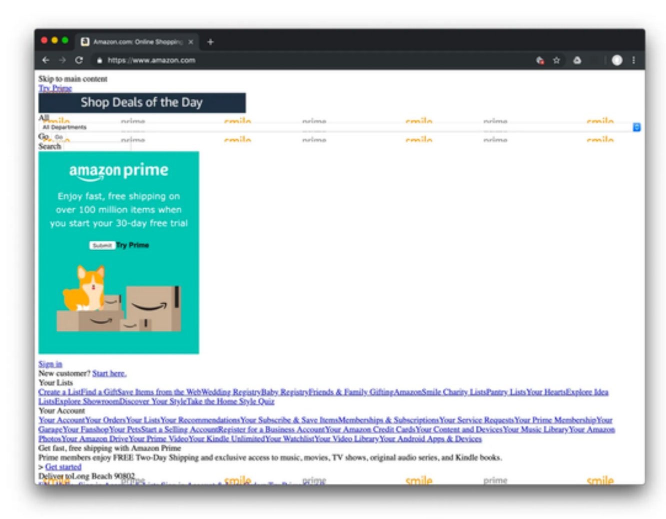
CSS: Cascading Style Sheets

- Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML).
- CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.
- CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications.
- Previously, various parts of CSS specification were developed synchronously, allowing the versioning of the latest recommendations.
- You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; everything is now CSS without a version number.

Why do we use CSS?

CSS (Cascading Style Sheets) is used to style and layout web pages — for example, to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

WITHOUT CSS

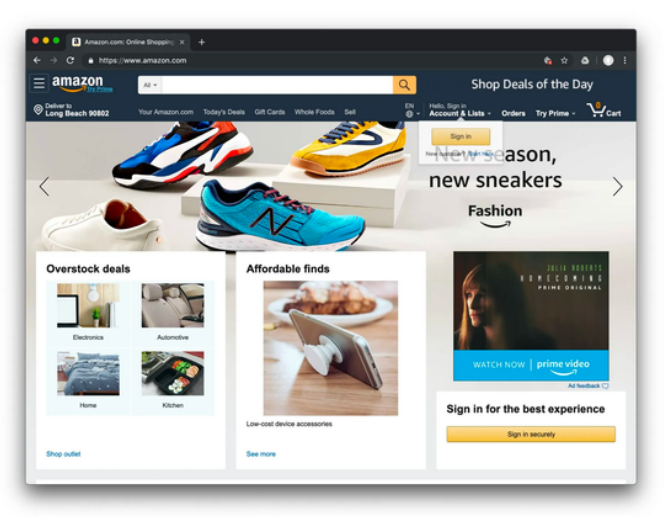


Skip to main content
Try Prime

Enjoy fast, free shipping on over 100 million items when you start your 30-day free trial

Sign in
New customer? Start here.
Your Lists
Create a ListFind a GiftSave Items from the WebWedding RegistryBaby RegistryFriends & Family GiftingAmazonSmile Charity ListsPantry ListsYour HeartsExplore Idea ListsExplore ShowroomDiscover Your StyleTake the Home Style Quiz
Your Account
Your AccountYour OrdersYour ListsYour RecommendationsYour Subscribe & Save ItemsMemberships & SubscriptionsYour Service RequestsYour Prime MembershipYour GarageYour FanshopYour PetsStart a Selling AccountRegister for a Business AccountYour Amazon Credit CardsYour Content and DevicesYour Music LibraryYour Amazon PhotosYour Amazon DriveYour Prime VideoYour Kindle UnlimitedYour WatchlistYour Video LibraryYour Android Apps & Devices
Get fast, free shipping with Amazon Prime
Prime members enjoy FREE Two-Day Shipping and exclusive access to music, movies, TV shows, original audio series, and Kindle books.
> Get started
Deliver toLong Beach 90802

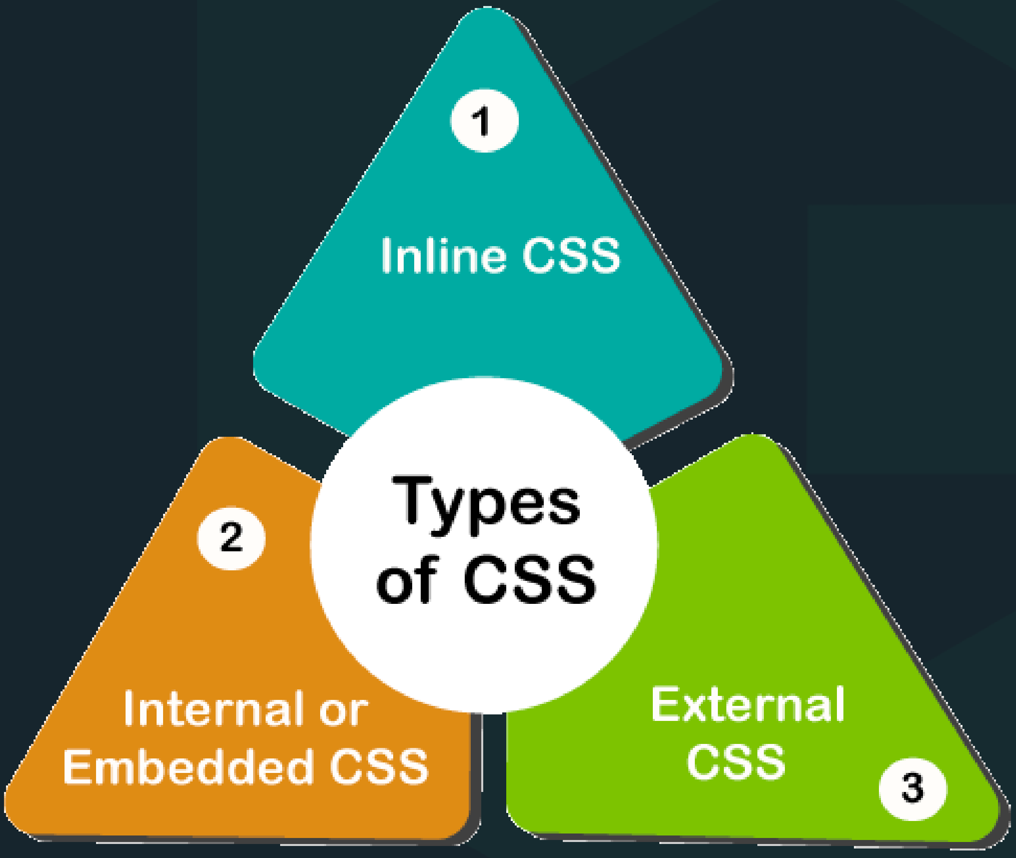
WITH CSS



Types of Stylesheet

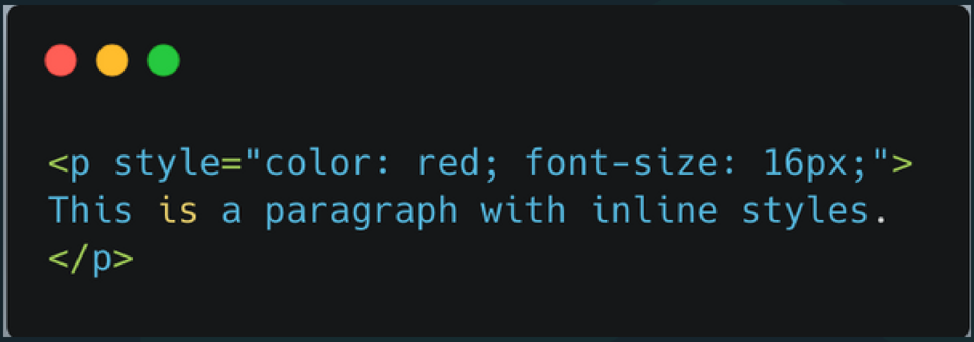
There are three main types of style sheets in web development: inline styles, internal stylesheets, and external stylesheets. Here's an overview of each type:

- **Inline Styles**
- **Internal Stylesheets:**
- **External Stylesheets:**



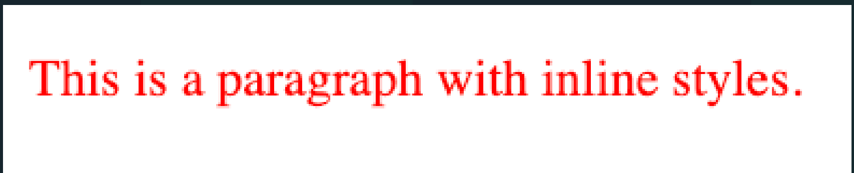
Inline CSS

Inline styles are applied directly to individual HTML elements using the "style" attribute. This approach allows you to define styles within the HTML tags themselves. Here's an example:



```
<p style="color: red; font-size: 16px;">
This is a paragraph with inline styles.
</p>
```

Output

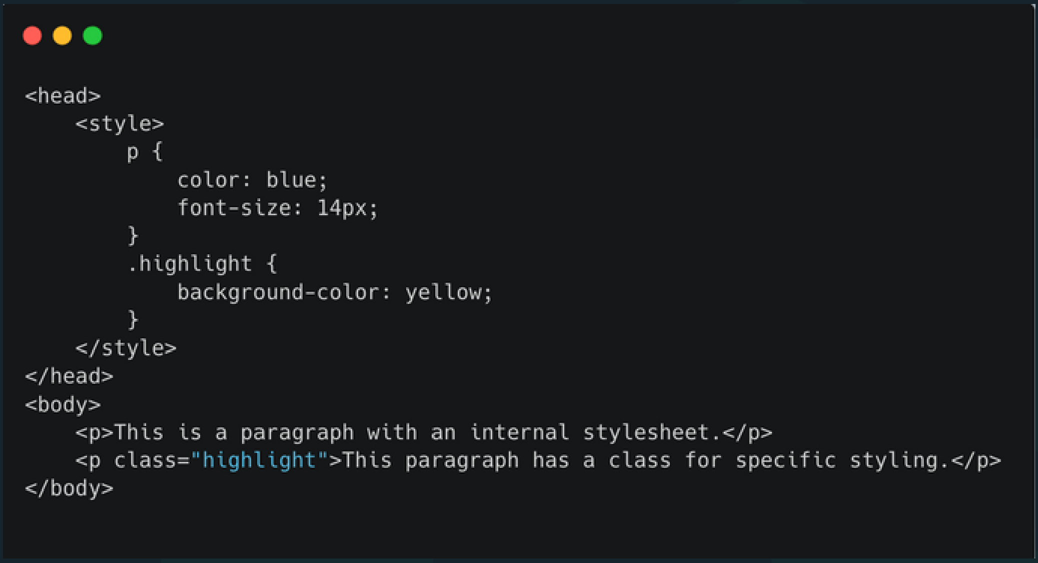


This is a paragraph with inline styles.

Inline styles are useful for applying unique styles to specific elements, but they can become cumbersome to manage if you have many elements or complex styles.

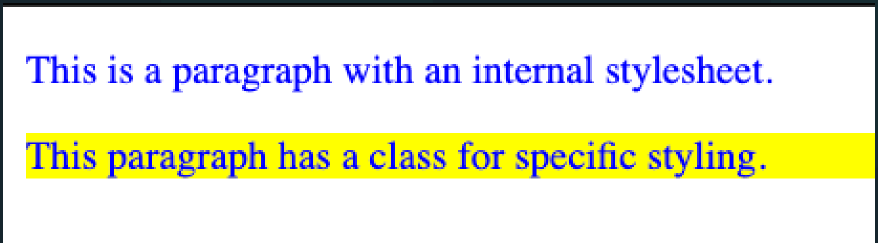
Internal Stylesheets

Internal style sheets are defined within the HTML file using the <style> tag in the head section. This approach allows you to define styles for multiple elements within the same HTML file. Here's an example:



```
<head>
    <style>
        p {
            color: blue;
            font-size: 14px;
        }
        .highlight {
            background-color: yellow;
        }
    </style>
</head>
<body>
    <p>This is a paragraph with an internal stylesheet.</p>
    <p class="highlight">This paragraph has a class for specific styling.</p>
</body>
```

Output



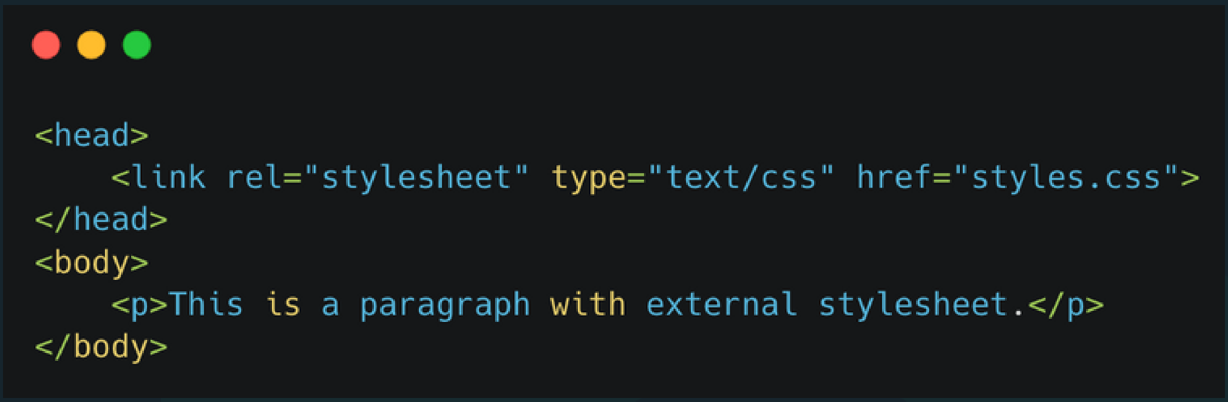
This is a paragraph with an internal stylesheet.

This paragraph has a class for specific styling.

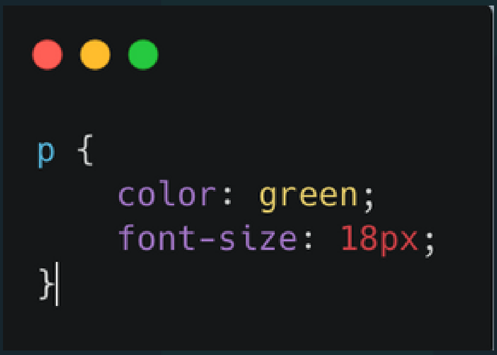
In the above example, the styles defined within the <style> tag will be applied to all <p> elements, giving them a blue color and font size of 14 pixels. The paragraph with the class "highlight" will have a yellow background color.

External Stylesheets

External stylesheets are separate CSS files that are linked to the HTML file using the <link> tag in the head section. This approach allows you to define styles in a separate file that can be shared across multiple HTML files. Here's an example:

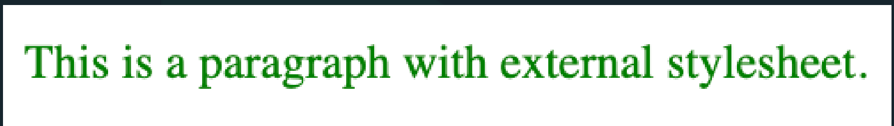


```
<head>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <p>This is a paragraph with external stylesheet.</p>
</body>
```



```
p {
    color: green;
    font-size: 18px;
}
```

Output



This is a paragraph with external stylesheet.

In this example, the styles defined in the external "styles.css" file will be applied to all <p> elements, making the text green with a font size of 18 pixels.

Using external stylesheets is a recommended approach for larger projects as it promotes the separation of concerns and allows for easier maintenance and reusability of styles.

The line <link rel="stylesheet" href="style.css" /> is an HTML tag used to link an external CSS stylesheet to an HTML document.

Let's break down the different parts of this line:

<link>: This is an HTML tag used to define a link between the current document and an external resource.

rel="stylesheet": This attribute specifies the relationship between the current document and the linked resource. In this case, rel="stylesheet" indicates that the linked resource is a stylesheet.

href="style.css": This attribute specifies the location or URL of the linked resource. In this example, the linked resource is a CSS file named "style.css". The href attribute value can be a relative or absolute URL.

/: The closing forward slash is used in self-closing tags in HTML to indicate that the tag does not have any content.

Overall, the line **<link rel="stylesheet" href="style.css" />** is used to link an external CSS file named "style.css" to the current HTML document. The CSS file will contain the styles that will be applied to the HTML elements within the document, allowing you to separate the style definitions from the HTML structure for better organisation and maintainability.

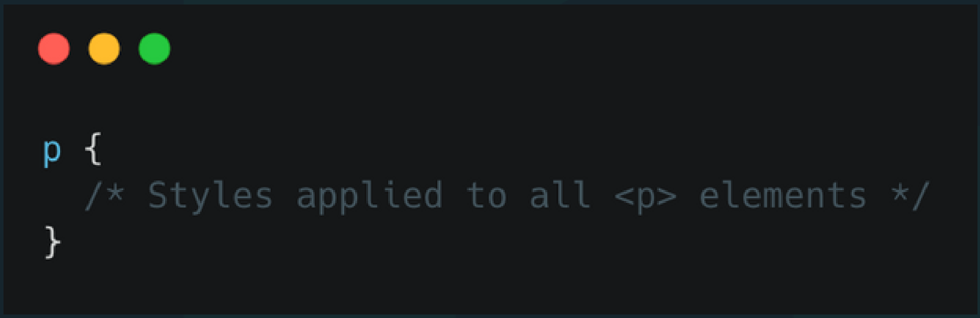
Each type of stylesheet has its own advantages and use cases, and the choice depends on the specific requirements of your project.

CSS Selectors

In CSS, selectors are used to target and select specific HTML elements to apply styles to. Selectors determine which elements on a web page should receive the defined styles. Here are some common types of CSS selectors:

Element Selector:

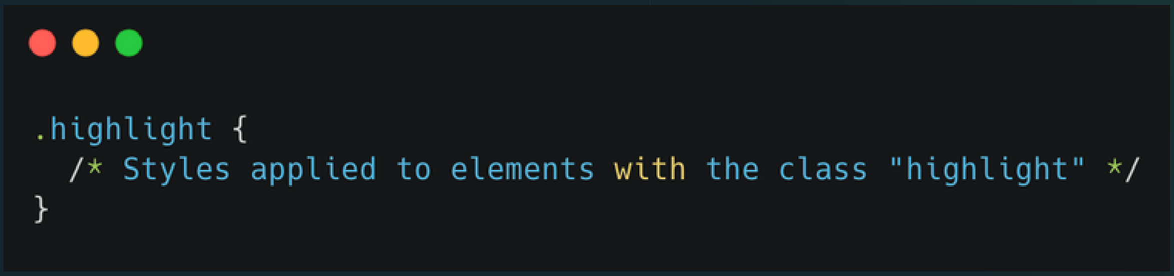
An element selector targets elements based on their HTML tag name. It is represented by the tag name itself. For example, to select all <p> elements, you would use the following selector:



```css
p {
  /* Styles applied to all <p> elements */
}
```

Class Selector:

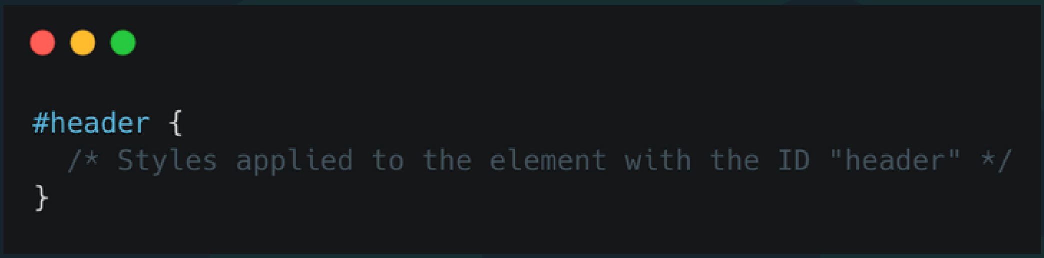
A class selector targets elements based on the value of their "class" attribute. It is represented by a dot followed by the class name. For example, to select all elements with the class "highlight", you would use the following selector:



```
.highlight {
  /* Styles applied to elements with the class "highlight" */
}
```

ID Selector:

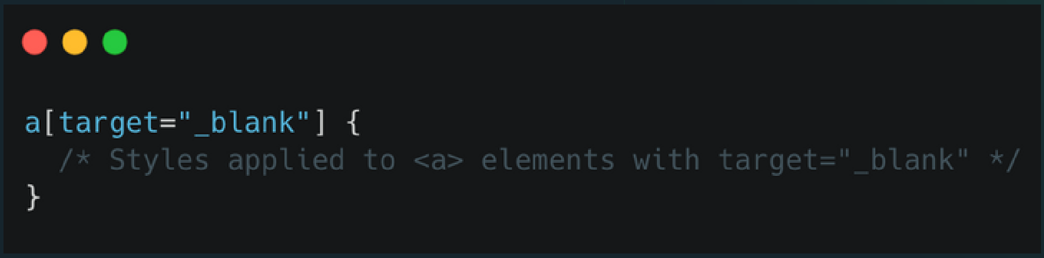
An ID selector targets a single element based on its "id" attribute. It is represented by a hash symbol (#) followed by the ID value. IDs should be unique within a document. For example, to select an element with the ID "header", you would use the following selector:



```
#header {
  /* Styles applied to the element with the ID "header" */
}
```

Attribute Selector:

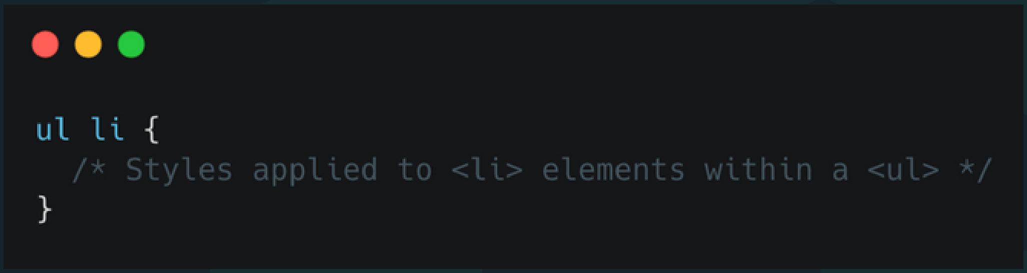
An attribute selector targets elements based on their attribute values. It is represented inside square brackets. For example, to select all <a> elements with the "target" attribute set to "_blank", you would use the following selector:



```css
a[target="_blank"] {
  /* Styles applied to <a> elements with target="_blank" */
}
```

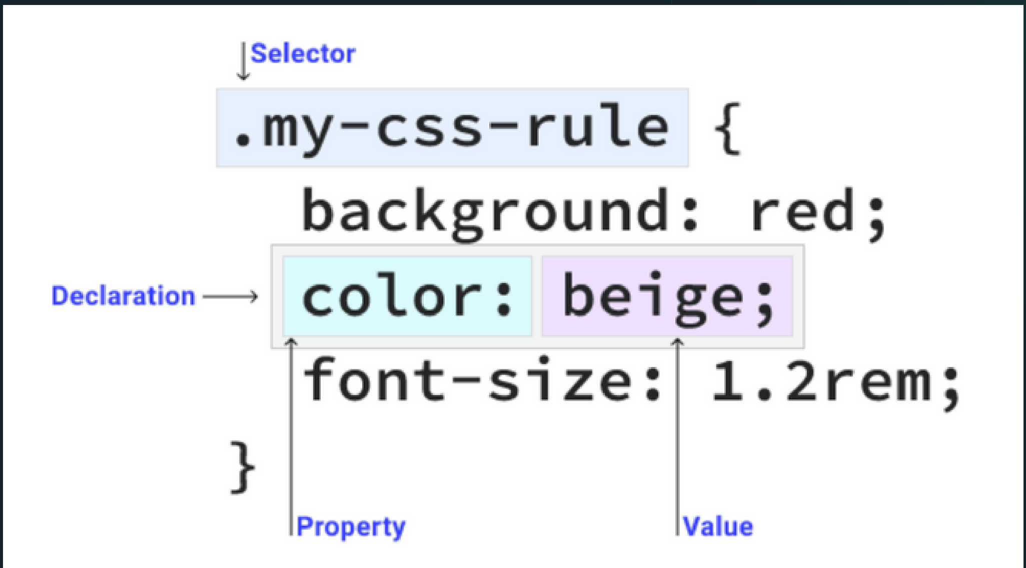
Descendant Selector:

A descendant selector targets elements that are descendants of a specific parent element. It is represented by a space between selectors. For example, to select all <li> elements that are descendants of a <ul> element, you would use the following selector:



```css
ul li {
  /* Styles applied to <li> elements within a <ul> */
}
```

These are just a few examples of CSS selectors. CSS provides a wide range of selectors and combinators that allow you to target elements based on their relationships, attributes, and other criteria. Selectors help you apply styles to specific elements and create targeted styling rules for different parts of your web page.



```
.my-css-rule {
  background: red;
  color: beige;
  font-size: 1.2rem;
}
```

Universal Selectors:

universal selector that targets all elements on the page and sets both the margin and padding properties to 0%.

Here's an explanation of the rule:

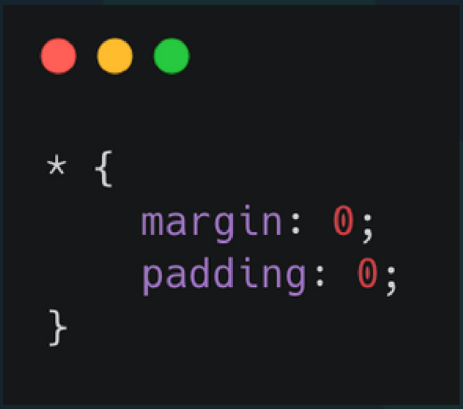
*: The asterisk (*) is a universal selector that matches any element on the page.

margin: 0%;: This property sets the margin of all elements to 0%. The margin is the space outside the element, which affects the spacing between elements.

padding: 0%;: This property sets the padding of all elements to 0%. The padding is the space inside the element, which affects the spacing between the element's content and its border.

By setting both margin and padding to 0%, this rule effectively removes any default spacing or gaps between elements. It can be useful in cases where you want to have more control over the spacing and positioning of elements by starting with a clean slate.

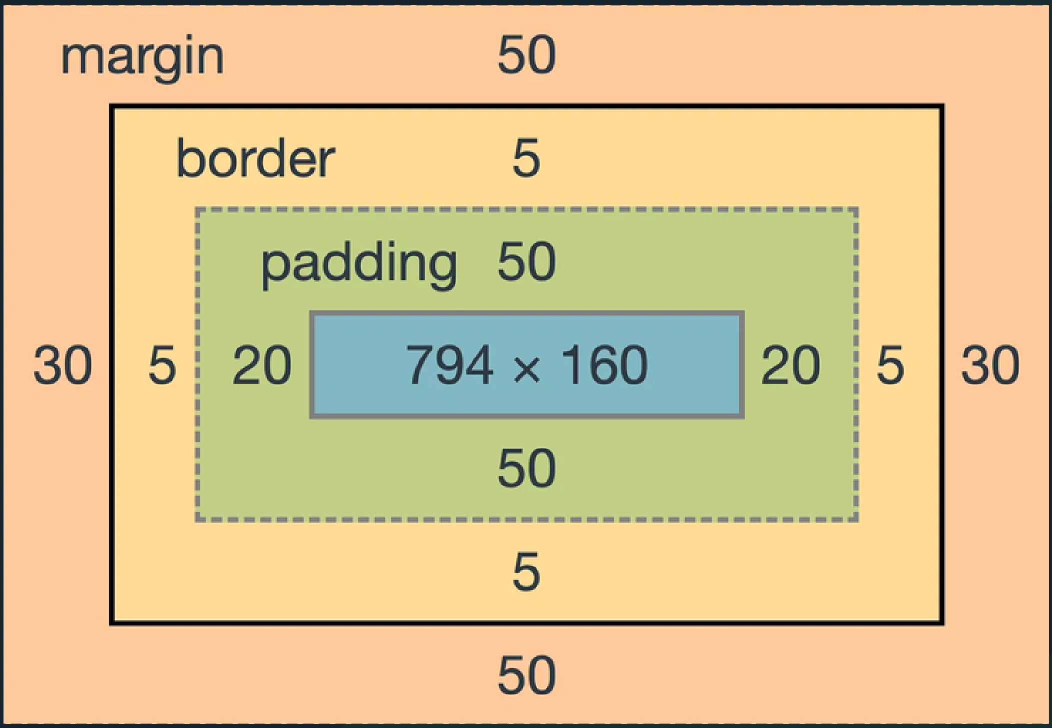
It's important to note that using a percentage value of 0% for margin and padding may not be necessary. Typically, using just 0 without the percentage symbol achieves the same result. So the rule can be simplified to:



```
* {
    margin: 0;
    padding: 0;
}
```

This rule is often used as a part of a CSS reset or normalise stylesheet, which helps in creating consistent cross-browser stylesheets by resetting or normalising the default styles applied by different browsers to different elements.

CSS Box Model



01 Margin

Margin is the space around an element, outside of its border. It creates a gap between the element and neighboring elements. Margins do not have a background color or content and do not affect the size of the element.

02 Padding

Padding is the space between the content of an element and its border. It provides a buffer zone between the content and the border. Padding is used to create visual separation and improve readability.

03 Border

Border is the line or outline that surrounds the content and padding of an element. It defines the visual boundary of the element. Borders can be customized with properties like color, width, and style.