**Pseudo elements**

**Introduction:**

Pseudo-elements in CSS are a powerful feature that allows you to style and manipulate specific parts of an element's content or layout without adding any additional markup to the HTML code. They are denoted by double colons (::) in CSS and are used to target and style specific parts of an element's structure.

There are several pseudo-elements available in CSS, each serving a specific purpose. Here's an overview of the most commonly used pseudo-elements:

**Commonly used pseudo-elements:**

::**before**: The ::before pseudo-element allows you to insert content before the content of an element. It is often used to add decorative or informational content, such as icons or text, before an element. It is positioned just inside the element but outside its normal content flow.

::**after**: Similar to ::before, the ::after pseudo-element allows you to insert content after the content of an element. It is commonly used for adding decorative elements or additional information after an element.

**::first-letter**: The ::first-letter pseudo-element targets the first letter of the text content within an element. It is often used to style drop caps or apply special formatting to the initial letter of a paragraph.

**::first-line**: The ::first-line pseudo-element targets the first line of the text content within an element. It is commonly used to apply specific styles to the first line, such as different font sizes, colors, or text transformations.

::selection: The ::selection pseudo-element targets the portion of text that has been selected by the user. It allows you to apply styles to the selected text, such as changing the background color or text color.

**Real-life example:**

Imagine you are building a website for a fashion brand. You want to add a small icon next to each product title to indicate its availability, like a green checkmark for available items and a red cross for out-of-stock items. Pseudo-elements provide a perfect solution for this scenario. By using the ::before or ::after pseudo-elements, you can insert and style the icons dynamically, enhancing the visual representation of product availability.

**Difference from pseudo-classes:**

Pseudo-classes are used to select elements based on their state or behavior, such as :hover or :active. On the other hand, pseudo-elements create virtual elements that do not exist in the HTML structure and allow you to style specific parts of an element's content.

**Code-based example:**

**Here's an example of how to use pseudo-elements in CSS:**

//**example**

HTML:-

<div class="product">

<h2>Product Title</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

</div>

CSS:-

.product::before {

content: "#";

color: green;}

margin-right: 5px;

}

.product::after {

content: "X";

color: red; margin-left: 5px; }

h2 { font-size: 20px; }

p { font-size: 14px; }

In this example, we use the ::before pseudo-element to add a green checkmark icon before the product title and the ::after pseudo-element to add a red cross icon after the title. We specify the content of the pseudo-elements using the content property and style them using CSS properties such as color and margin.

**Note:**

Pseudo-elements can be styled using any CSS properties, just like regular elements. However, it's important to note that not all properties are applicable to all pseudo-elements. Some properties may have limited or no effect on certain pseudo-elements.

**Conclusion:**

Overall, pseudo-elements provide a flexible and powerful way to add extra content, style specific parts of an element, or create decorative effects without modifying the HTML structure. They greatly enhance the capabilities of CSS for creating visually appealing and interactive web designs.

::**after**

**Sir**: Absolutely, my eager student! Today, we'll be delving into the topic of "::after." It's a powerful CSS pseudo-element that allows us to add content after an element's content. It's a handy tool for enhancing the visual presentation of webpages.

**Introduction**

The "::after" pseudo-element is used to insert content after the content of an element. It is often used to add decorative elements or additional information to an element, without modifying the HTML structure. By leveraging the "::after" pseudo-element, we can manipulate the appearance of elements and create engaging designs.

**Real Life Example:-**

Imagine you have a blog post with a series of tips. You want to include a checkmark symbol after each tip to make it visually appealing and indicate completion. Instead of manually adding the checkmark symbol to each tip, we can use the "::after" pseudo-element to achieve this effect.

**Code Based Example**

Let's see how we can apply the "::after" pseudo-element in HTML and CSS to create a simple example. Consider the following HTML markup:

**HTML:-**

<div class="box">Hello World </div>

**CSS:-**

.box::after{

content="(after)";

color: red;

}

In this example, the text "Hello, world!" will be displayed within the "box" element, and the "::after" pseudo-element will insert the text " (after)" after the content. Additionally, we've set the color of the inserted content to red.

Conclusion

And that's how you can utilize the "::after" pseudo-element to add content after an element in HTML and CSS. It's a simple yet effective technique to enhance the visual representation of your webpages.

::**before**

**You**: That sounds intriguing! Could you please explain it in more detail?

**Sir**: Of course! The ::before pseudo-element is used to insert content before the content of an element. It creates an additional virtual element that is placed as the first child of the selected element. We can use it to add decorative elements, icons, or even textual content to enhance the presentation of an element.

**You**: I see. So, it's like adding an extra layer of content before the actual content of an element.

**Real Life Example**:-

**Sir**: Exactly! It provides us with a way to visually enhance elements without altering the underlying HTML structure. Now, let's explore a real-life example to understand its practical application.

Imagine you have a website with a list of articles, and you want to display an icon representing the article's category before the title of each article. Using the ::before pseudo-element, you can achieve this by adding a specific class to the title element and applying the necessary CSS.

You: That's really useful! It adds a visual cue to quickly identify the category of each article.

**Code Based Example:-**

**HTML:-**

<ul class="article-list">

<li class="article-title">Intro to css</li>

<li class="article-title">intro to html</li>

<li class="article-title">intro to javascript</li>

</ul>

**Css**:-

.aricle-title::before{

font-family: "Font Awesome 5 Free";

content: "\f015";

display: inline-block;

padding-right: 3px;

vertical-align: middle;

font-weight: 900;

list-style: none;

}

.article-list{

list-style: none;

}