**Pseudo classes**

**Introduction:**

Pseudo classes in CSS are special selectors that target specific states or behaviors of elements that cannot be selected with regular classes or IDs. They begin with a colon (:) followed by the pseudo class name. Pseudo classes are used to add styles to elements based on interactions, such as hover, focus, and visited states.

Pseudo-classes in CSS are used to select and style elements based on a specific state or condition.

Here are some of the commonly used pseudo-classes in CSS:

* :hover - Selects an element when the mouse pointer is over it.
* :active - Selects an element when it is being activated by the user (e.g., clicked).
* :focus - Selects an element when it has received focus (e.g., through keyboard navigation).
* :visited - Selects a link that has been visited by the user.
* :link - Selects a link that has not been visited by the user.
* :first-child - Selects the first child element of its parent.
* :last-child - Selects the last child element of its parent.
* :nth-child(n) - Selects the nth child element of its parent. It can be further refined with different formulas, such as even, odd, or specific numeric values.
* :nth-of-type(n) - Selects the nth element of its type among its siblings.
* :first-of-type - Selects the first element of its type among its siblings.
* :last-of-type - Selects the last element of its type among its siblings.
* :empty - Selects elements that have no children or content.
* :checked - Selects an input element of type "checkbox" or "radio" that is checked.
* :disabled - Selects form elements that are disabled.
* :enabled - Selects form elements that are enabled.
* :not(selector) - Selects elements that do not match the specified selector.
* :nth-last-child(n) - Selects the nth child element counting from the end of its parent's children.
* :only-child - Selects elements that are the only child of their parent.
* :target - Selects an element that is the target of the current URL.
* :root - Selects the root element of the document (i.e., <html>).

**Real-life example:**

Let's imagine you are designing a navigation menu for a website. You want to change the color of the navigation links when the user hovers over them to provide visual feedback. By using the :hover pseudo class, you can easily add a hover effect to the navigation links, making the user experience more engaging and intuitive.

Difference from general:

Pseudo classes are different from regular classes or IDs because they target specific states or behaviors of elements. While regular classes or IDs are used to select and style elements based on their assigned attributes, pseudo classes enable us to target and style elements based on user interactions or element states.

**Code-based example:**

Here's an example of how to use a pseudo class in CSS:

<a href="#" class="nav-link">Home</a>

CSS:-

.nav-link:hover{

color: red;

}

In this example, we have a navigation link with the class "nav-link". By using the :hover pseudo class as part of the selector, we can change the color of the link to red when the user hovers over it.

====================================================

**:active**

In this note, we will learn about how the :active pseudo class works and how to implement it demonstrated by an example.

**Pseudo Classes**

A pseudo-class is a term used in CSS that defines a particular condition or state of an element, like when it's clicked, hovered over, or the first child of its parent element. These are identified by a colon (":") and applied along with a selector to select specific elements on a webpage.

Introduction to :active pseudo class

**Observation**

Have you ever noticed how some buttons or links on a webpage change their appearance slightly when you click on them? That's because they are using the ":active" pseudo-class to apply a temporary style when the element is being clicked on.

The :active CSS pseudo-class represents an element (such as a button) that is being activated by the user.

What is meant by activation?

When using a mouse, "activation" typically starts when the user presses down the primary mouse button.

**Syntax for active pseudo class:-**

The CSS active selector is a tool used to change how a link looks when you click on it. It is usually used for buttons or links that you click on.

The effect of the CSS active selector only lasts for a moment, but it makes the link look different when you click on it. It is used to create interesting styles or to show that the link has been clicked.

**Syntax**

**:active**

The :active pseudo-class is commonly used on <a> and <button> elements.

**Example for active pseudo class:-**

<p>

<a href="#">This is active class link</a>

</p>

CSS:-

a:link{

color:blue;

}

a:active{

color:red

}

=====================================================

**Active form Elements**

Form elements are commonly used for creating forms, with popular ones including the label and button tags. It's possible to style them using the active selector.

**HTML:-**

<form>

<label>A button</label>

<br>

<button type="button"> click here </button>

</form>

**CSS:-**

form button{

background: blue;

}

form:active{

color:red;

}

Because we've used the active selector on the form element as a whole, when a user clicks or selects any of the elements inside the form, the label's color will change.

**Conclusion**

In this note, we learned about how the :active pseudo class works and how to implement it.

================================

**:first-child**

**You**: Hi Sir, I am excited to learn about the :first-child pseudo-class today! I have been applying what we learned about the previous topic, and it has been great so far.

**Sir**: Great to hear, Student! The :first-child pseudo-class is a powerful tool for targeting the first child element of a parent element. It allows you to style or apply functionality to specific elements in a more efficient way.

The :first-child CSS pseudo-class represents the first element among a group of sibling elements

**You**: That sounds interesting, Sir. Can you give me an example of how it is used in real-life situations?

**Real Life Example for :first-child pseudo class**

You can use the :first-child pseudo-class to style the first item in a list differently from the rest of the items. It can also be used to target the first paragraph in a section of text to apply specific formatting.

It can also be used to target the first paragraph in a section of text to apply specific formatting.

**HTML**:-

<div>

<p>First statement</p>

<p>Second Statement</p>

</div>

**CSS**:-

P:first-child{

color:yellow;

background-color: black;

}

======================================================

**:focus:-**

**Introduction**

The :focus pseudo-class is a CSS selector that applies styles to an element when it is in focus. It is commonly used for interactive elements like form inputs and links to provide visual cues to the user that they are interacting with that element.

**Let's Understand with the help of an example:-**

<div>

<input class="class1" value="turn me into blue"/>

</div>

CSS:-

.class1:focus{

background:yellow;

color: red;

}

**Conclusion**

We learned about the :focus selector and the ways of using focus and implemented it with the help of an example.

==============================================

:**required**;-

Let's consider a user registration form. We want to make sure that certain fields, like the username and password, are filled out before the user can proceed. By applying the ":required" pseudo-class, we can style these required fields differently, perhaps by adding a visual indicator like a red border, to draw attention to them and prompt the user to fill them in.

**EXAMPLE**:-

Here's an example of a simple form with two required fields: a name input and an email input.

<form>

<label for="name">Name:</label>

<input type="text" name="name" required>

<label for="email">Email:</label>

<input type="email" name="email" required>

<input type="submit" value="Submit">

</form>

**Css**:-

input:required{

border: 2px solid red;

}

In this example, the "input:required" selector targets all input elements with the "required" attribute and applies a red border to them.

**Conclusion**

**You**: That makes sense! It seems straightforward to implement. I can't wait to try it out myself.

**Sir**: That's the spirit, my student! I encourage you to experiment with the ":required" pseudo-class and explore different styling options. It's a valuable tool to ensure form completion and provide visual cues to users.

=======================================================

:**visited**:-

**Real Life Example**

Imagine you're designing a website for a travel agency. You have a page with different destinations, each represented by a link. Now, when users click on a link to view details about a specific destination, you want to differentiate between visited and unvisited links. This distinction can help users keep track of the places they've already explored.

Ah, I see! So, by using the ":visited" pseudo-class, we can style the visited links differently from the unvisited ones. That's a great way to enhance user experience. Can we try implementing it with some code?

**Code Based Example**

Absolutely! Let's consider a simple example. Assume we have a navigation menu with a list of links. Here's how you can apply the ":visited" pseudo-class in HTML and CSS:

**HTML:-**

</head>

<body>

<a href="#test-visited-link">Have you visited this link yet?</a><br />

<a href="">You've already visited this link.</a>

</body>

**CSS**:-

a {

/\* Specify non-transparent defaults to certain properties,

allowing them to be styled with the :visited state \*/

background-color: white;

border: 1px solid white;

}

a:visited {

background-color: yellow;

border-color: hotpink;

color: hotpink;

}

==================================================

**:hover:-**

**Introduction**

The ":hover" pseudo-class is a CSS selector that allows you to apply specific styles to an element when a user hovers over it. This can be a useful tool for creating interactive and engaging websites that respond to user actions.

**Real-Life Example**

A real-life example of using the ":hover" pseudo-class is on a website that has navigation links. When a user hovers over a navigation link, the link can change color or underline to indicate that it is a clickable option. This can help improve user experience and make the website feel more interactive.

**Here is an example of using the ":hover" pseudo-class in code:**

**HTML**:-

<a href="#">Hover on this link</a>

**css**:-

a{

background-color: rgb(106, 159, 167);

transition: background-color 0.5s;

}

a:hover{

background-color : yellow;}

**Conclusion**

In conclusion, the ":hover" pseudo-class is a useful tool for adding interactive styling effects to your website. By using this selector, you can create a more engaging user experience and improve the overall design of your website.

**Example:- All pseudo class in one example.**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>All Pseudo-Classes</title>

    <link rel="stylesheet" href="./demopsu.css" />

  </head>

  <body>

    <ul>

      <li><a href="##home">Home</a></li>

      <li><a href="##about">About</a></li>

      <li><a href="##services">Services</a></li>

      <li><a href="##contact">Contact</a></li>

    </ul>

  </body>

</html>

CSS:-

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

}

ul {

    list-style-type: none;

    margin: 0;

    padding: 0;

    overflow: hidden;

    background-color: #333;

}

li {

    float: left;

}

a {

    display: block;

    color: white;

    text-align: center;

    padding: 14px 16px;

    text-decoration: none;

    transition: background-color 0.3s, color 5s;

}

/\* Pseudo-class styles \*/

a:hover {

    background-color: #555;

}

a:active {

    background-color: #c23b3b;

}

a:focus {

    outline: 2px solid #fff;

    outline-offset: 4px;

    background-color: #444;

}

a:visited {

    color: yellowgreen;

}

**In this example:**

* :hover changes the background color when the mouse hovers over a link.
* :active changes the background color when a link is being clicked.
* :focus adds an outline and changes the background color when a link is focused (e.g., tabbed to).
* :visited changes the color of visited links.

The transition property is used to create a smooth color transition effect when the links change their background color. You can adjust the colors and other styles according to your preference.

**:nth-child(n)**

Today we shall unravel the mysteries of the ":nth-child(n)" pseudo-class in CSS. It is a powerful tool that allows us to select elements based on their position within a parent element. This pseudo-class is quite handy when we need to style specific elements in a group based on their order.

That sounds intriguing, Sir. Can you provide a real-life example to help me understand it better?

**Real Life Example**

Sir: Certainly! Imagine you have a list of items on a webpage, and you want to style every second item differently. For instance, let's say you have a list of testimonials, and you want to highlight every second testimonials with a different background color to make them stand out. That's where the ":nth-child(n)" pseudo-class comes in handy.

**Code Based Example**

You: Ah, I see. So, if I understand correctly, the ":nth-child(n)" pseudo-class allows us to select elements based on their position within their parent element, and we can apply different styles to them?

**HTML**:-

<p>list of champians</p>

<ul>

<li>Sachin Tendulkar (India):</li>

<li>Sir Don Bradman (Australia):</li>

<li>Sir Vivian Richards (West Indies):</li>

<li>Wasim Akram (Pakistan):</li>

<liShane Warne (Australia):></li>

<li>Jacques Kallis (South Africa):</li>

</ul>

**CSS**:-

li:nth-child(3){

border:2px solid blue;

margin-bottom: 1px;

}

So, the ":nth-child(n)" pseudo-class allows us to apply styles to elements based on their position within their parent element.

==========================================================================

:nth-last-child(n)

The ":nth-last-child(n) pseudo class" is similar to the ":nth-child(n) pseudo class," except that it selects elements based on their position within the parent container, counting from the last child element rather than the first. So, for example, if you wanted to style the second-to-last child element in a container, you would use the ":nth-last-child(2)" selector.

**Example:-**

<p>list of champians</p>

<ul>

<li>Sachin Tendulkar (India):</li>

<li>Sir Don Bradman (Australia):</li>

<li>Sir Vivian Richards (West Indies):</li>

<li>Wasim Akram (Pakistan):</li>

<liShane Warne (Australia):></li>

<li>Jacques Kallis (South Africa):</li>

</ul>

**CSS**:-

li:nth-last-child(2){

border:2px solid blue;

margin-bottom: 1px;

}

The above CSS properties will be applied to the 2nd last child of the list. We would see a blue border around the 2nd last element in the list.

**Note:-**

One thing to keep in mind is that the ":nth-last-child(n) pseudo class" can be used in combination with other selectors, such as ":first-child" or ":last-child," to create more complex selector chains. For example, you could use the selector ":nth-last-child(2):first-child" to target the first child element that is also the second-to-last child element in a container.

====================================================================