Selectors

Sudhakar Sharma

•

Yesterday

5. Element Validation State Pseudo Classes  
  
:required                It verifies the required attribute set for input.  
:optional                It defines effects when required is not defined.  
:valid                It verifies any validation defined in input  
:invalid                It returns effects when any validation returns  
                invalid.  
:in-range                It is for number validation   [min]  
:out-of-range        It is for number validation   [max]  
  
  
Ex: required and optional are used only to verify the attribute not validate.  
  
  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Marquee</title>  
        <style>  
            #User:required {  
              border:1px solid red;  
              box-shadow: 2px 2px 2px red;  
            }  
            #User:optional {  
              border:1px solid yellow;  
              box-shadow: 2px 2px 2px yellow;  
            }  
        </style>  
         
    </head>  
    <body>  
       User Name : <input type="text"  id="User" required >  
    </body>  
</html>  
  
Ex: valid , invalid etc..  
  
  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Marquee</title>  
        <style>  
           #UserName:valid+div {  
             display: none;  
           }  
           #UserName:invalid+div {  
             display: block;  
           }  
           #Mobile:valid+div {  
             display: none;  
           }  
           #Mobile:invalid+div {  
             display: block;  
           }  
           #Age:in-range+div {  
             display: none;  
           }  
           #Age:out-of-range+div {  
             display: block;  
           }  
  
           #Age:in-range {  
             border:1px solid green;  
           }  
           #Age:out-of-range {  
             border:1px solid red;  
           }  
        </style>  
         
    </head>  
    <body>  
       <h2>Register</h2>  
       <dl>  
         <dt>User Name</dt>  
         <dd>  
           <input type="text" id="UserName" required>  
           <div style="color:red">User Name Required</div>  
         </dd>  
         <dt>Mobile</dt>  
         <dd>  
           <input type="text" id="Mobile" required pattern="\+91\d{10}">  
           <div style="color:red">  
               Mobile Required / Invalid : +91 and 10 digits required  
           </div>  
         </dd>  
         <dt>Age</dt>  
         <dd>  
           <input type="number" min="15" max="30" id="Age">  
           <div style="color:red">  
             Age 15 to 30 Only  
           </div>  
         </dd>  
       </dl>  
    </body>  
</html>  
  
6. Attribute Selectors  
- It allows to apply effects based on the element attribute instead of tag directly.  
  
Syntax:  
        element[attribute=value]  
        {  
        }  
        element[attribute]  
        {  
        }  
  
Ex:  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Selectors</title>  
        <style>  
            input[type="button"] {  
                background-color: yellow;  
            }  
        </style>  
    </head>  
    <body>  
        <dl>  
            <dt>Name</dt>  
            <dd><input type="text"></dd>  
            <dt>Mobile</dt>  
            <dd><input type="text"></dd>  
        </dl>  
        <input type="button" value="Register">  
        <input type="button" value="Cancel">  
    </body>  
</html>  
  
Ex: Only for Attribute  
  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Selectors</title>  
        <style>  
            p[id] {  
                color:red;  
            }  
        </style>  
    </head>  
    <body>  
        <p>Para-1</p>  
        <p id="p2">Para-2</p>  
        <p>Para-3</p>  
        <p id="p4">Para-4</p>  
        <p>Para-5</p>  
    </body>  
</html>  
  
- Attribute selectors have conditions  
  
         attribute="value"                Equal to  
        attribute^="value"                Starts with  
        attribute$="value"                Ends with  
        attribute~="value"                Any location with space  
        attribute|="value"                with - separator and  
                                start with.  
  
Ex:  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Selectors</title>  
        <style>  
            p[class|="Effects"] {  
                color:red;  
            }  
        </style>  
    </head>  
    <body>  
        <p class="Effects">Para-1</p>  
        <p class="EffectsText">Para-2</p>  
        <p class="TextEffects">Para-3</p>  
        <p class="AnimateEffectsText">Para-4</p>  
        <p class="Effects-Text">Para-5</p>  
        <p class="Effects Text">Para-6</p>  
        <p class="Text-Effects">Para-7</p>  
    </body>  
</html>  
  
7. Structural Selectors [Pseudo class]  
- It depends on hierarchy parent-child  
- It is to handle various occurances in hierarchy.  
  
:first-child  
:last-child  
:nth-child()  
:nth-of-type()  
:nth-last-of-type()  
  
Syntax:  
        :nth-child(2)  
        :nth-child(even)  
        :nth-child(odd)  
  
        :nth-of-type(2)        only 2nd  
        :nth-of-type(2n)        for every 2nd occurance  
        :nth-of-type(2n+1)        for every 2nd starting with 1st from top  
        :nth-last-of-type(2n+1)   starting from bottom  
  
Ex  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Selectors</title>  
        <style>  
           li:nth-last-of-type(2n+1) {  
               color:red;  
           }  
           
        </style>  
    </head>  
    <body>  
        <ol>  
            <li>Item-1</li>  
            <li>Item-2</li>  
            <li>Item-3</li>  
            <li>Item-4</li>  
            <li>Item-5</li>  
            <li>Item-6</li>  
            <li>Item-7</li>  
            <li>Item-8</li>  
        </ol>  
    </body>  
</html>  
  
8. Element Classes and Behavioural  
::placeholder  
::before  
::after  
::selection  
  
  
Ex:  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Selectors</title>  
        <style>  
            #UserName::placeholder {  
                color:red;  
            }  
        </style>  
    </head>  
    <body>  
       <input type="text" id="UserName" placeholder="User Name">  
    </body>  
</html>  
  
  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Selectors</title>  
        <style>  
            ul {  
                display:flex;  
                list-style: none;  
            }  
            li::before{  
                content: " / ";  
                margin-left: 5px;  
            }  
            li:first-child::before {  
                content: "";  
            }  
            li:last-child {  
                color:gray;  
            }  
        </style>  
    </head>  
    <body>  
       <ul>  
           <li>Home</li>  
           <li>Categories</li>  
           <li>Electronics</li>  
           <li>Mobiles</li>  
           <li>Samsung</li>  
       </ul>  
    </body>  
</html>  
  
Ex:  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Selectors</title>  
        <style>  
           p::selection {  
               background-color: yellow;  
           }  
        </style>  
    </head>  
    <body>  
      <p>Your use of this software is subject to the terms and conditions of the license agreement by which you acquired this software.  If you are a volume license customer, use of this software is subject to your volume license agreement.  You may not use this software if you have not validly acquired a license for the software from Microsoft or its licensed distributors.  
    </p>  
    </body>  
</html>  
  
- Root Selector  
        :root {  
        }  
  
- Universal Selector  
        \* {  
        }  
- Language Selector  
        :lang {  
  
        }  
Syntax:  
 p:lang(en) {  
                color:red;  
            }  
  
                         CSS Attributes