Example2:

```
<!DOCTYPE html>
<html>
<head>
<title>Class Selector</title>
<style type="text/css">
p.abc{
   color:red;
   text-align: center;
}
</style>
</head>
<body>
   <h1 class="abc">This Heading is not Affected. </h1>
   <p class="abc">This Paragraph is in red color and center
align.</p>

</body>
</html>
```

PREVIEW

This Heading is not Affected.

This Paragraph is in red color and center align.

The style rule inside the selector **p.abc** renders the text in red color and text-align is center only of those <p> elements that has class attribute set to red, center and has no effect on other paragraphs.

सलेक्टर p.abc के अंदर स्टाईल नियम केवल उन <p> एलिमेन्टो को लाल रंग के टेक्स्ट में प्रस्तुत करता है जिनमें क्लास ऐट्रीब्यूट लाल रंग एवं सेंटर सेट है, और किसी अन्य पैराग्राफ पर कोई प्रभाव नहीं पड़ेगा।

4.8 Sub selector - सब सेलेक्टर

A CSS selector can contain more than one simple selectors, it means that, it combines the other selectors in a way that gives them a useful relationship to each other and the location of content in the document. Between the simple selectors, we can include a combinator.

CSS सेलेक्टर में एक से अधिक सिंपल सेलेक्टर शामिल हो सकते हैं, इसका मतलब यह है कि, यह अन्य चयनकर्ताओं को इस तरह से जोड़ता है जो उन्हें एक–दूसरे के लिए उपयोगी संबंध और डॉक्यूमेंट में कंटेंट का स्थान देता है। सिम्पल सेलेक्टर के बीच, हम एक कॉम्बिनेटर शामिल कर सकते हैं।

There are four different combinators in CSS:

सीएसएस में चार अलग–अलग संयोजन होते हैं

1. Descendant selector (space) - डिसेण्डेन्ट सेलेक्टर (स्पेस)

Descendant selector matches all the elements that are descendants of a specified elements, it means that selectors can select element that are descendants of others selector. The element does not have to be a direct child, but rather any descendant down the line.

डिसेण्डेन्ट सेलेक्टर उन सभीं एलिमेन्टों से मेल खाता है जो एक स्पेसिफाईड एलिमेन्ट के डिसेण्डेन्ट हैं। इसका मतलब है कि चयनकर्ता ऐसे एलीमेंट का चयन कर सकते हैं जो दूसरे चयनकर्ता के डिसेण्डेन्ट हैं। एलिमेंट के पास सीधा चाईल्ड से सम्बन्ध होना जरुरी नहीं है, बल्कि लाईन के नीचे कहीं भी हो सकता है।

Syntax:

```
element1 element2
{
 style_properties
 }
```

element1: The first element type to match. मैच करने वाला पहला एलिमेंट टाइप।

element2: The second element type to match that must be a descendant of element1. मैच के लिए दूसरा एलिमेंट टाइप जो एलिमेंट 1 का डिसेण्डेन्ट होना चाहिए।

style_properties: The CSS styles to apply to the targeted element2. सीएसएस स्टाईल लक्षित एलीमेंट 2 पर लागू करने के लिए।

Note: The descendant selector uses a space character to target an element that is a descendant of another element.

डिसेण्डेन्ट सेलेक्टर एक एलिमेंट को target करने के लिए एक space कैरेक्टर का उपयोग करता है जो किसी अन्य एलिमेन्ट का डिसेण्डेन्ट है।

Example 1:

h1 h2 h3 ¶p b i u span link img ul •li

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  ol li { color: red; }
6  </style>
7  </head>
8  <body>
9  <ol>
10   <li>Databases</li>
11   <ul>
12     <li>Oracle/PLSQL</li>
13     <li>SQL</li>
14   </ul>
15   <li>ASP.Net</li>
16   <li>Java</li>
17 </ol>
18 </body>
19 </html>
```

PREVIEW

1. Databases
   - Oracle/PLSQL
   - SQL
2. ASP.Net
3. Java

In this descendant selector example, all <li> tags that are descendants of the <ol> tag will be styled by the descendant selector (ie: styled with red font). So in the case of two <li> tags under the <ul> tag:

इस डिसेण्डेंट सेलेक्टर उदाहरण में, सभी टैग जो कि टैग के डिसेण्डेंट हैं, को डिसेण्डेंट सेलेक्टर (यानीः लाल फ़ॉन्ट के साथ स्टाइल) द्वारा स्टाइल किया जाएगा। इसी तरह <ul> टैग के अन्दर दो टैग है।

Oracle/PLSQL
SQL

Although the two <li> tags are not direct children of the <ol> tag, they are still descendants of the <ol> tag and thus are styled by the descendant selector.

यद्यपि दो टैग <ol> टैग के डायरेक्ट चाईल्ड नहीं हैं, फिर भी वे टैग के डिसेण्डेंट हैं और इस प्रकार डिसेण्डेंट सेलेक्टर द्वारा स्टाइल किया जाता है।

Example 2: The descendant selector matches all elements that are descendants of a specified element. The following example selects all <p> elements inside elements:

डिसेण्डेंट सेलेक्टर उन सभी एलिमेन्ट से मेल खाता है जो एक स्पेशिफाईड एलिमेंट के डिसेण्डेंट हैं। निम्न उदाहरण में सभी <p> एलिमेंट्स कें अंदर एलिमेंट्स को सेलेक्ट किया गया हैः

```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
  background-color: yellow;
}
</style>
</head>
<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <span><p>Paragraph 3 in the div.</p></span>
</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>

</body>
</html>
```

PREVIEW

Paragraph 1 in the div.
Paragraph 2 in the div.
Paragraph 3 in the div.
Paragraph 4. Not in a div.
Paragraph 5. Not in a div.

Example 3:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
h1 em{     color: blue; }
ul.menu    { padding:5;  list-style:none;  margin:5px;  }
ul.menu li  { display:inline;  }
ul.menu li a{ margin:8px;  text-decoration:none;          }
</style>
</head>
<body>
     <h1>This is <em>Heading</em> </h1>
     <ul class="menu">
        <li><a href="#">Home</a></li>
        <li><a href="#">About Us</a></li>
        <li><a href="#">Sell</a></li>
        <li><a href="#">Contact</a></li>
     </ul>
 </body>
</html>
```

PREVIEW

This is *Heading*

Home About Us Sell Contact

2. Child selector (>) - चाइल्ड सेलेक्टर

The CSS child selector uses the > character to target an element that is a direct child of an element type.

CSS चाइल्ड सेलेक्टर > कैरेक्टर का उपयोग किसी एलिमेंट को टारगेट करने के लिए करता है जो एलिमेंट टाइप के डाइरेक्ट चाइल्ड हैं।

Syntax:

```
element1 > element2 {
style_properties
}
```

element 1: The first element type to match.

पहला एलिमेंट मिलान करने के लिए -

element 2: The second element type to match that must be a direct child of element1.

दूसरा एलिमेंट, जो एलिमेंट 1 से सीधा संबंधित होना चाहिए।

style_properties: The CSS styles to apply to the targeted element2.

CSS स्टाइल टारगेट एलिमेंट 2 पर लागू होती हैं।

Note: The child selector uses the **> combinator** to target an element that is a direct child of another element.
चाइल्ड सेलेक्टर उस एलिमेंट को टारगेट करने के लिए > कॉम्बिनेटर का उपयोग करता है जो किसी अन्य एलिमेंट का सीधा चाइल्ड होता है।

Example: We will discuss the child selector below, exploring examples of how to use this selector in CSS. With <li> tag, let's look at a CSS child example where we apply the child selector to the <li> tag.

हम नीचे दिए गए चाइल्ड सेलेक्टर पर चर्चा करेंगे, और इस सेलेक्टर का CSS में कैसे उपयोग करेंगे उनके उदाहरणों को देखेंगे। टैग के साथ, CSS में चाइल्ड सेलेक्टर के उदाहरण को देखते हैं, जहां हम चाइल्ड सेलेक्टर को टैग पर लागू करते हैं।

h1 h2 h3 ¶p b i u span link img ul •li </h1>

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
ol > li {
        color: blue;
        padding:5px;
      }
</style>
</head>
<body>
  <ol>
       <li>Tbalaji Publication</li>
   <ul>
       <li>C Language</li>
       <li>IT Tools</li>
       <li>Python</li>
   </ul>
       <li>www.tbalaji.com</li>
       <li>www.W3C.com</li>
  </ol>
 </body>
</html>
```

PREVIEW

1. Tbalaji Publication
   - C Language
   - IT Tools
   - Python
2. www.tbalaji.com
3. www.W3C.com

In this child selector example, only <li> tags that are direct children of an <ol> tag will be styled by the child selector (ie: styled with blue font color). So in the case of the three <li> tags under the <ul> tag:

इस चाइल्ड सेलेक्टर के उदाहरण में, केवल <li> टैग जो कि एक <ol> टैग का direct children हैं, को चाइल्ड सेलेक्टर द्वारा स्टाइल किया जाएगा (यानीः नीले फ़ॉन्ट के साथ स्टाइल किया गया)। इसलिए इस केस में टैग के तहत तीन <li> टैग का प्रयोग होगा।

```
<li>C Language</li>
<li>IT Tools</li>
<li>Python</li>
```

They are not direct children of the <ol> tag and therefore are not styled by the child selector.
ये टैग के डायरेक्ट चाइल्ड नहीं हैं और इसलिए चाइल्ड सेलेक्टर द्वारा स्टाइल नहीं किया जाता है।

3. Adjacent sibling selector (+) - एडजेसेन्ट सिबलिंग सेलेक्टर (+)

Adjacent sibling selector uses "+" sign to combine two sequences of simple selectors. It is a selector for separating two selectors and matches the second element only when it immediately follows the first element. In this case, both are the children of the same parent element.

एडजेसेन्ट सिबलिंग सेलेक्टर सिम्पल सेलेक्टर्स के दो अनुक्रमों को संयोजित करने के लिए "+" चिह्न का उपयोग करता है। यह दो सेलेक्टर्स को अलग करने के लिए एक सेलेक्टर है और दूसरे एलीमेंट से तभी मेल खाता है जब वह पहले एलीमेंट का तुरंत अनुसरण करता है। इस मामले में, दोनों एक ही मूल एलीमेंट के चिल्ड्रेन हैं।

Syntax:

```
Element1 + Element2 {
style_properties
}
```

Here, Element2 is the target of the selector. The selector h2 + p in the example below will select the <p> elements only if both the <h2> and <p> elements share the same parent in the document tree and <h2> is immediately precedes the <p> element. That means only those paragraphs that come immediately after each <h2> heading will have the associated style rules.

यहां, Element2 सेलेक्टर का टारगेट है। नीचे दिए गए उदाहरण में सेलेक्टर h2 + p <p> एलिमेन्ट को सेलेक्ट तभी करेगा जब दोनों <h2> और <p> एलिमेंट एक ही parent को डॉक्यूमेन्ट ट्री को साझा करें और <h2> तुरंत <p> एलिमेंट से पहले हो। इसका मतलब है कि केवल उन पैराग्राफ जो प्रत्येक <h2> हेडिंग के तुरंत बाद आते हैं, संबंधित स्टाइल नियम होंगे।

Example:

h1 h2 h3 ¶p b i u span link img ul •li </h1>

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
  h2 + p {
    color: red;
    font-size: 25px;
    text-align:center;
  }
  ul.tbp + p {
    color: green;
    text-indent: 180px;
  }
</style>
</head>
<body>
  <h2>Welcome to Tbalaji Publication</h2>
  <p>We are giving the following items.....</p>
  <p>Some of our books are given below:</p>
  <ul class="tbp">
    <li>1-> Python</li>
    <li>2-> IT Tools</li>
    <li>3-> C Language</li>
  </ul>
  <p>That's Set</p>
  <p>Goto Our Next Page></p>
</body>
</html>
```

PREVIEW

Welcome to Tbalaji Publication

We are giving the following items.....

Some of our books are given below:

- 1-> Python
- 2-> IT Tools
- 3-> C Language

That's Set

Goto Our Next Page>

4. General sibling selector (~) - सामान्य सिबलिंग सेलेक्टर (~)

The general sibling combinator (~) separates two selectors and matches the second element only if it follows the first element (though not necessarily immediately) and both are children of the same parent element.

सामान्य सिबलिंग कॉम्बीनेटर (~) दो सेलेक्ट्स को अलग करता है और केवल दूसरे एलिमेंट से मैच करता है केवल तभी जब यह पहले एलिमेंट (हालांकि जरूरी नहीं कि तुरंत) का अनुसरण करता है और दोनों एक ही पैरेन्ट एलिमेंट से संबंधित होते हैं।

Syntax:

```
Former_Element ~ Target_Element {
style properties
}
```

Former_Element ~ Target_Element, where Target_Element is the target of the selector.

Former_Element ~ Target_Element, जहां Target_Element सेलेक्टर का टारगेट है।

Example 1: The selector h2 ~ p in the example below will select all the <p> elements that proceeded by the <h2> element, where all the elements share the same parent in the document tree.

उदाहरण 1: सेलेक्टर h2 ~ p नीचे दिए गए उदाहरण में <h2> एलिमेन्ट द्वारा आगे बढ़ने वाले सभी एलिमेन्टों को सेलेक्ट करेगा, जहाँ सभी एलिमेन्ट एक ही parent डॉक्यूमेंट ट्री में साझा करते हैं।

```
<!DOCTYPE html>
<html>
<head>
<style>
  h2 ~ p {
    color: red;
    font-size: 25px;
  }
  ul.tbp ~ p {
    color: green;
    text-indent: 20px;
  }
</style>
</head>
<body>
  <h2>Welcome to Tbalaji Publication</h2>
  <p>We are giving the following items.....</p>
  <p>Some of our books are given below:</p>
  <ul class="tbp">
    <li>1-> Python</li>
    <li>2-> IT Tools</li>
    <li>3-> C Language</li>
  </ul>
  <p>That's Set</p>
  <p>Goto Our Next Page></p>
</body>
</html>
```

PREVIEW

Welcome to Tbalaji Publication

We are giving the following items.....

Some of our books are given below:

- 1-> Python
- 2-> IT Tools
- 3-> C Language

That's Set

Goto Our Next Page>

Activate Windows

Example 2:

```
<!DOCTYPE html>
<html>
<head>
<style>
  div ~ p {
    background-color: red;
    font-size: 20px;
  }
</style>
</head>
<body>
 <h2>Welcome to Tbalaji Publication</h2>
 <div>
  <p>Start Code section 1</p>
  <code>Write some code here</code>
  <p>End of code section 1</p>
 </div>
  <p>Start Code section 2</p>
  <code>Write some code here</code>
  <p>End of Code section 2</p>
</body>
</html>
```

PREVIEW

Welcome to Tbalaji Publication

Start Code section 1

```
Write some code here
```

End of code section 1

```
Write some code here
```

Example 3:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p ~ span {
     color: red;
     font-size: 15px,
  }
</style>
</head>
<body>
 <span>Welcome Tbalaji Publication</span>
  <p>Para1</p>
  <code>Para 2> Write Some Code here</code>
  <span>This is Red Span!</span>
  <code>More Code here</code>
  <span>This is Red Span!</span>
</body>
</html>
```

PREVIEW

Welcome Tbalaji Publication
Para1

Para 2> Write Some Code here This is Red Span! More Code here This is Red Span!

4.9 Attribute Selector - एट्रीब्यूट सेलेक्टर

Attribute selectors are a special kind of selector that will match elements based on their attributes and attribute values. Their generic syntax consists of square brackets ([ ]) containing an attribute name followed by an optional condition to match against the value of the attribute.

एट्रीब्यूट सेलेक्टर एक विशेष प्रकार का सेलेक्टर है जो एलिमेंट को उनके ऐट्रीब्यूटो और ऐट्रीब्यूट वैल्यू के आधार पर मिलान करता है। उनके जेनेरिक सिंटैक्स में स्क्वायर ब्रेकेट्स ([ ]) होते हैं, जिसमें ऐट्रीब्यूट के नाम के साथ एक वैकल्पिक स्थिति होती है, जो ऐट्रीब्यूट के मूल्य के साथ मेल खाती है।

Attribute selectors can be divided into two categories depending on the way they match attribute values:

ऐट्रीब्यूट वैल्यू के आधार पर ऐट्रीब्यूट सेलेक्टर को दो श्रेणियों में विभाजित किया जा सकता है:

- **[attribute] [ऐट्रीब्यूट]:**
  This selector is used to select elements with a specified attribute. The following example selects all <a> elements with a target attribute:
  इस सेलेक्टर का उपयोग एक स्पेशिफाईड ऐट्रीब्यूट वाले एलिमेंट का चयन करने के लिए किया जाता है। निम्नलिखित उदाहरण में सभी <a> एलिमेंट का चयन टारगेट ऐट्रीब्यूट के साथ में किया गया है:

```
<!DOCTYPE html>
<html>
<head>
<style>
  a[target]{
     background-color: yellow;
  }
</style>
</head>
<body>
 <h3 style="text-align:center;">Welcome Tbalaji Publication</h3>
  <p>The link with target attribute is in blue background.....</p>
  <a href="http://www.facebook.com" target="_p">Facebook</a>|
  <a href="http://www.google.com" target="_b">Google</a>|
 <a href="http://www.twitter.com">Twitter</a>
</body>
</html>
```

PREVIEW

Welcome Tbalaji Publication

The link with target attribute is in blue background.....

Facebook| Google| Twitter

- **[attribute = "value"] :**
  This selector is used to select elements with a specified attribute and value. The following example selects all <a> elements with a target="_b" attribute:

(एट्रीब्यूट = "वैल्यू"): इस सेलेक्टर का उपयोग एक स्पेशिफाईड ऐट्रीब्यूट और वैल्यू वाले एलिमेन्टो का चयन करने के लिए किया जाता है। निम्न उदाहरण मे सभी <a> एलिमेंट एक target="_b" ऐट्रीब्यूट के साथ का चयन किया गया है:

```
<!DOCTYPE html>
<html>
<head>
<style>
   a[target=_b]{
      background-color: yellow;
   }
</style>
</head>
<body>
 <h3 style="text-align:center;">Welcome Tbalaji Publication</h3>
  <p>The link with target attribute is in blue background.....</p>
  <a href="http://www.facebook.com" target="_p">Facebook</a>|
  <a href="http://www.google.com" target="_b">Google</a>|
 <a href="http://www.twitter.com">Twitter</a>
</body>
</html>
```

PREVIEW

Welcome Tbalaji Publication

The link with target attribute is in blue background.....

Facebook| Google| Twitter

- **[attribute ~= "value"] :**

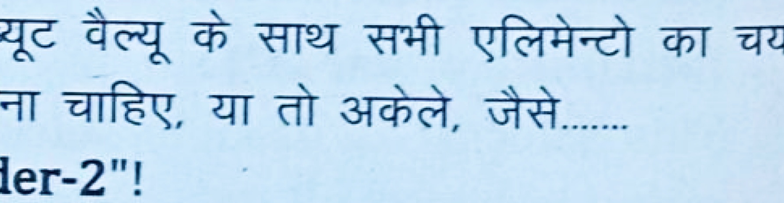
This selector is used to select elements with an attribute value containing a specified word. The following example selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower":

(एट्रीब्यूट ~ = "वैल्यू"): यह सेलेक्टर एक स्पेशिफाईड वर्ड वाले एट्रीब्यूट वैल्यू के साथ एलिमेन्टों का चयन करने में उपयोग किया जाता है। निम्न उदाहरण एक टाईटिल ऐट्रीब्यूट के साथ सभी एलिमेन्टों का चयन करता है जिसमें शब्दों की एक स्पेस–सेपरेटेड सूची होती है, जिनमें से एक "flower" है:

```
1  <!DOCTYPE html>
2  <html>
3  <head><style>
4  [title~=flower] {
5    border: 5px solid yellow;
6  }
7  </style></head>
8  <body>
9
10 <p>All images with the title attribute containing the word "flower" get a
   yellow border.</p>
11
12 <img src="klematis.jpg" title="klematis flower" width="150"
   height="113">
13 <img src="img_flwr.gif" title="flower" width="224" height="162">
14 <img src="img_tree.gif" title="tree" width="200" height="358">
15
16 </body></html>
```

PREVIEW

All images with the title attribute containing the word "flower" get a yellow border.



The example above will match elements with title="flower", title="sun flower" and title= "flower card", but not title="Sun-flower" or title="flowers".

- **[attribute |= "value"] :**

This selector is used to select elements with the specified attribute starting with the specified value. The following example selects all elements with a class attribute value that begins with "header": **Note:** The value has to be a whole word, either alone, like ...

(ऐट्रीब्यूट |= "वैल्यू"): यह सेलेक्टर स्पेशिफाईड वैल्यू के साथ शुरू होने वाले स्पेशिफाईड ऐट्रीब्यूट वाले एलिमेंट का चयन करने के लिए उपयोग किया जाता है। निम्न उदाहरण में एक क्लास ऐट्रीब्यूट वैल्यू के साथ सभी एलिमेन्टो का चयन करता है जो "**header**" से शुरू होता है: **नोट:** वैल्यू को एक संपूर्ण शब्द होना चाहिए, या तो अकेले, जैसे..... class="**header**", or followed by a hyphen(-), like class=" **header-2**"!

```
<!DOCTYPE html>
<html>
<head>
<style>
 [class|=header]
      {
      background-color: yellow;
      }
</style></head>
<body>
 <h3 class="header-1">Welcome Tbalaji Publication</h3>
 <p class="header-2">Hello Everyone!</p>
 <p class="header3">Where Are You?</p>
</body>
</html>
```

PREVIEW

Welcome Tbalaji Publication

Hello Everyone!

Where Are You?

- **[attribute ^= "value"] :**

This selector will select all elements with the attribute for which the value starts with value. The following example selects all elements with a class attribute value that begins with "**header**".

(एट्रीब्यूट ^= "वैल्यू"): यह सेलेक्टर उस एट्रीब्यूट के साथ सभी एलिमेंट का चयन करेगा जिसके लिए वैल्यू की शुरूआत वैल्यू से होती है। निम्न उदाहरण में एक क्लास एट्रीब्यूट वैल्यू के साथ उन सभी एलिमेंट्स का चयन करता है जो "**header**" से शुरू होता है।

```
<!DOCTYPE html>
<html>
<head>
<style>
 [class^="header"]
      {
      background-color: yellow;
      }
</style></head>
<body>
 <h3 class="header-1">Welcome Tbalaji Publication</h3>
 <p class="header-2">Hello Everyone!</p>
 <p class="header3">Where Are You?</p>
 <p class="Third header">Are You Finding Me (Yes or No ?)</p>
</body>
</html>
```

PREVIEW

Welcome Tbalaji Publication

Hello Everyone!

Where Are You?

Are You Finding Me (Yes or No ?)

- **[attribute $= "value"]** : This selector will select all elements with the attribute for which the value ends with value. The following example selects all elements with a class attribute value that ends with "header".

(एट्रीब्यूट $= "वैल्यू"): यह सेलेक्टर उस ऐट्रीब्यूट वाले सभी एलिमेन्टो का चयन करेगा जिसके लिए वैल्यू, वैल्यू के साथ ही समाप्त होता है। निम्नलिखित उदाहरण एक क्लास ऐट्रीब्यूट वाले सभी एलिमेन्टो का चयन करता है जो "header" के साथ समाप्त होता है।

```
<!DOCTYPE html>
<html>
<head>
<style>
 [class$="header"]
      {
      background-color: yellow;
      }
</style></head>
<body>
 <div class="first_header">Welcome Tbalaji Publication</div>
 <div class="header second">Hello Everyone!</div>
 <div class="third-header">Where Are You?</div>
 <p class="fourthheader">Are You Finding Me (Yes or No ?)</p>
</body>
</html>
```

PREVIEW

Welcome Tbalaji Publication

Hello Everyone!

Where Are You?

Are You Finding Me (Yes or No ?)

- **[attribute *= value] :**

This selector will select all elements with the attribute for which the value contains the substring value. (Substring is the part of a string. i.e. "pub" is the substring in the string "publication".)

(एट्रीब्यूट *= "वैल्यू"): यह सेलेक्टर उस ऐट्रीब्यूट वाले सभी एलिमेंट का चयन करेगा, जिसके लिए वैल्यू में सबस्ट्रिंग वैल्यू है। (एक सबस्ट्रिंग स्ट्रिंग का एक हिस्सा है, उदाहरण के लिए "पब्लिकेशन" स्ट्रिंग मे सबस्ट्रिंग "पब" है।)

```
<!DOCTYPE html>
<html>
<head>
<style>
 [class*="fo"]
      {
      background-color: yellow;
      }
</style></head>
<body>
  <div class="first_header">Welcome Tbalaji Publication</div>
  <div class="header second">Hello Everyone!</div>
  <div class="third-header">Where Are You?</div>
  <p class="fourthheader">Are You Finding Me (Yes or No ?)</p>
</body>
</html>
```

PREVIEW

Welcome Tbalaji Publication
Hello Everyone!
Where Are You?
Are You Finding Me (Yes or No ?)

Here, we see some example of all the attributes together.

सभी attributes के कुछ उदाहरण निम्नलिखित हैं–

Example 1: In this example all elements with the attribute "data-books" are given blue text, all elements with the attribute "data-books" with the exact value "ccc" are given a yellow background color and all elements with the attribute "data-books", containing the value "easy" are given red text color.

इस उदाहरण में सभी एलिमेन्ट जो ऐट्रीब्यूट "data-books" के साथ है नीला रंग दिया गया है, एलिमेन्ट जो ऐट्रीब्यूट "data-books" जो वास्तविक वैल्यू "ccc" के साथ हैं उन्हें पीला रंग एवं एलिमेन्ट जो ऐट्रीब्यूट data-books के वैल्यू "easy" का साथ हैं उन्हें लाल रंग दिया गया है।

```
<!DOCTYPE html>
<html>
<head>
<style>
 [data-books]
      {     color: blue;          }
 [data-books="ccc"]
      {     background-color:yellow;      }
 [data-books~="easy"]
      {     color:red;          }
</style></head>
<body>
 <h2 style="text-align:center;">Welcome Tbalaji Publication</h2>
 <p>Our Books Are.....</p>
 <ul>
      <li data-books>C Language</li>
      <li data-books>IT</li>
      <li data-tough>Python</li>
      <li data-tough>Web</li>
      <li data-books="easy">Internet</li>
      <li data-books="easy">Multimedia</li>
      <li data-books>DCA/ADCA</li>
      <li data-books="ccc">CCC Book</li>
      <li data-books="ccc">Exam Booster</li>

 </ul>
</body>
</html>
```

PREVIEW

Welcome Tbalaji Publication

Our Books Are.....

- C Language
- IT
- Python
- Web
- Internet
- Multimedia
- DCA/ADCA
- CCC Book
- Exam Booster

Activate Windows
Go to Settings to activate Windows.