HTML <section> एलिमेन्ट एक स्टैन्डएलोन सेक्शन को रिप्रजेन्ट करता है – जिसके पास इसे दर्शाने के लिए अधिक विशिष्ट एलिमेन्ट नहीं है – जो HTML डॉक्यूमेंट में शामिल है। <section> टैग किसी डॉक्यूमेन्ट में सेक्शनो को डिफाईन करता है, जैसे चैप्टर, headers, footers या डॉक्यूमेन्ट के किसी भी अन्य सेक्शन। लेकिन आमतौर पर, सेक्शन के पास एक शीर्षक हो ऐसा हमेशा नहीं होता है।

See the below example:

```
<!DOCTYPE html>
<html>
<body>

<section>
<h1>HTML5 Introduction</h1>
<p>HTML5 is the latest development of the standard that defines HTML. <br>
    The term represents two different concepts.</p>
</section>

<section>
<h1>Features</h1>
<p>•Semantics: allowing you to describe more precisely what your content is.<br>
    •Connectivity: allowing you to communicate with the server in new and innovative
            ways.<br>
</p>
</section>
<body>
</html>
```

Preview of above example look like ... उपरोक्त उदाहरण का पूर्वावलोकन जैसा दिखता है

PREVIEW

HTML5 Introduction

HTML5 is the latest development of the standard that defines HTML.
The term represents two different concepts.

Features

- Semantics: allowing you to describe more precisely what your content is.
- Connectivity: allowing you to communicate with the server in new and innovative ways.

Usage notes Section - नोट्स का उपयोग करें

- ✓ Each <section> should be identified, typically by including a heading (<h1>-<h6> element) as a child of the <section> element.

  प्रत्येक <section> को आम तौर पर <section> के एक बच्चे के रूप में शीर्षक (<h1>-<h6> element) शामिल करके पहचाना जाना चाहिए।

- ✓ *If it makes sense to separately syndicate the content of a <section> element, use an <article> element instead.* यदि यह एक <section> तत्व की सामग्री को अलग से सिंडिकेट करने के लिए समझ में आता है, तो इसके बजाय ढलेखझ तत्व का उपयोग करें।

- ✓ *Do not use the <section> element as a generic container; this is what is for, especially when the sectioning is only for styling purposes. A rule of thumb is that a section should logically appear in the outline of a document.*

जेनेरिक कंटेनर के रूप में <section> तत्व का उपयोग न करें य यह वह है जो के लिए है, खासकर जब सेक्शन केवल स्टाइलिंग उद्देश्यों के लिए है। अंगूठे का एक नियम यह है कि किसी अनुभाग को दस्तावेज की रूपरेखा में तार्किक रूप से प्रकट होना चाहिए।

2. **Nav tag:** The <nav> tag is a new element in HTML5. It is used to define a block of navigation links, either within the current document or to other documents. Examples of navigation blocks are menus, tables of contents, and indexes. One HTML document may contain several <nav> tags, for example, one for site navigation and one for intra-page navigation.Note that not all links in the HTML document can be placed inside the <nav> element; it can only include major navigation blocks. For example, the <nav> tag is not placed in the <footer> tag for defining links in the footer of the website.

HTML5 में <nav> टैग एक नया एलिमेन्ट है। इसका उपयोग नेविगेशन लिंक के एक ब्लॉक को डिफाईन करने के लिए किया जाता है, या तो वर्तमान डॉक्यूमेंट के भीतर या अन्य डॉक्यूमेंट के लिए। नेविगेशन ब्लॉकों के उदाहरण मेनू, टेबिल ऑफ कन्टेंट और इंडेक्स हैं। एक HTML डॉक्यूमेंट में कई <nav> टैग हो सकते हैं, उदाहरण के लिए, साइट नेविगेशन के लिए एक और इंट्रा–पेज नेविगेशन के लिए एक। ध्यान दें कि HTML डॉक्यूमेंट में सभी लिंक <nav> एलिमेंट के अंदर नहीं रखे जा सकते हैं; इसमें केवल प्रमुख नेविगेशन ब्लॉक शामिल हो सकते हैं। उदाहरण के लिए, वेबसाइट के फुटर में लिंक डिफाईन करने के लिए <nav> टैग को <footer> टैग में नहीं रखा गया है।

Syntax: The <nav> tag comes in pairs; the content is written between the opening (<nav>) and closing (</nav>) tags.

Example:

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

```
1  <!DOCTYPE html>
2  <html>
3   <head>
4    <title>Title of the document</title>
5   </head>
6   <body>
7    <header>
8     <h1>Programming Courses</h1>
9    </header>
10   <nav>
11    <a href="Html.html">HTML</a> | <a href="CSS.html">CSS</a> | <a
    href="javascript.html">JavaScript</a> | <a href="/php.html">PHP</a>
12   </nav>
13   <h2>Welcome to Balaji Publication</h2>
14  </body>
15 </html>
```

PREVIEW

Programming Courses

HTML | CSS | JavaScript | PHP

Welcome to Balaji Publication

3. **Article tag:** The <article> tag has been added to HTML5. It is used to define an independent, self-contained content. An article should have its own meaning and be easily differentiated from the rest of the web page content.

HTML5 में <article> टैग जोड़ा गया है। इसका उपयोग एक स्वतंत्र, सेल्फ–कन्टेंट को डिफाईन करने के लिए किया जाता है। एक आर्टिकल का अपना अर्थ होना चाहिए और बाकी वेब पेज के कन्टेंट से आसानी से अलग होना चाहिए।

The <article> element can include: - <article> एलीमेंट में शामिल हो सकते हैं

- ✓ Blog entry ब्लॉग इंट्री

- ✓ Forum post फोरम पोस्ट
- ✓ News न्यूज
- ✓ Comment कमेंट

Multiple <article> tags can be used in one HTML document.

Syntax: The content of the <article> tag is written between the opening (<article>) and the closing (</article>) tags.

Headers from <h2> to <h6> can be nested inside the <article> tag. If the <h1> tag was not used before, then its placement is allowed.

Example:

```
h1 h2 h3 ¶p b i u span link img ul •li
1 <!DOCTYPE html>
2 <html>
3  <head>
4   <title>Title of the document</title>
5  </head>
6  <body>
7   <article>
8    <h1>Title of the article</h1>
9    <p>Text of the article</p>
10  </article>
11 </body>
12 </html>
```

PREVIEW

Title of the article

Text of the article

4. **Aside:** The <aside> tag has been added to HTML5 to define a section with additional information related to the content around the aside element. It is generally used to enhance an article with additional information, or highlighting parts that can be interesting to the user. Content in aside is a stand-alone, non-essential part of the web page, and if you delete it, the main content will not be affected.

Endnotes comments, reference information, lists of terms, a collection of links, pull-quotes, etc. are a kind of information that can appear in the <aside> element. Do not confuse the <aside> element with sidebar. Sidebar is just a visual element, and the <aside> tag doesn't systematically look like a sidebar.

HTML 5 में <aside> टैग को एक सेक्शन में डिफाईन करने के लिए जोड़ा गया है ताकि एक तरफ के एलिमेंट से संबंधित अतिरिक्त जानकारी हो। यह आमतौर पर अतिरिक्त जानकारी के साथ एक आर्टिकल को बढ़ाने के लिए उपयोग किया जाता है, या उन हिस्सों को उजागर करता है जो उपयोगकर्ता के लिए दिलचस्प हो सकते हैं। एक तरफ कन्टेंट वेब पेज का एक स्टैंड–अलोन, गैर–आवश्यक हिस्सा है, और यदि आप इसे हटाते हैं, तो मेन कंटेट प्रभावित नहीं होगी।

एन्डनोट्स कमेंट, रेफरेंस इनफार्मेशन, लिस्ट ऑफ टर्म, लिंक का एक संग्रह, पुल–कोट्स, आदि एक तरह की जानकारी है जो <aside>element में दिखाई दे सकते हैं। साइडबार के साथ <aside> एलिमेंट को कनफ्यूज न करें। साइडबार केवल एक विजुअल एलिमेंट है, और <aside> टैग व्यवस्थित रूप से साइडबार की तरह नहीं दिखता है।

Syntax: The <aside> tag comes in pairs, which means that the content is written between the opening (<aside>) and the closing (</aside>) tags.

Example:

```
1  <!DOCTYPE html>
2  <html>
3   <head>
4     <title>Title of the document</title>
5   </head>
6    <body>
7        <p>I like watching Game of Thrones.</p>
8        <aside>
9           <h4>Game of Thrones</h4>
10          <p>Game of Thrones is an American fantasy drama television<br>
    series created by David Benioff and D. B. Weiss. It is an adaptation of
    A<br> Song of Ice and Fire, George R. R. Martin's series of fantasy
    novels</p>
11       </aside>
12    </body>
13 </html>
```

PREVIEW

I like watching Game of Thrones.

Game of Thrones

Game of Thrones is an American fantasy drama television series created by David Benioff and D. B. Weiss. It is an adaptation of A Song of Ice and Fire, George R. R. Martin's series of fantasy novels

5. **Audio Tag:** The<audio> has been added to HTML5 to allow embedding audio files to a web page. Since not all browsers support all audio formats, the audio file is encoded using special codecs. The <source> tag or the src attribute is used to indicate the variations of the same audio file. The path to an audio file can contain absolute or relative URLs.

एक वेब पेज पर ऑडियो फाइलों को एम्बेड करने की अनुमति देने के लिए <audio> को HTML5 में जोड़ा गया है। चूंकि सभी ब्राउजर सभी ऑडियो प्रारूपों को सपोर्ट नहीं करते हैं, ऑडियो फाइल विशेष codecs का उपयोग करके encoded है। <source> टैग या src ऐट्रीब्यूट का उपयोग उसी ऑडियो फाइल की विविधताओं (वैरिएशन) को दर्शाने के लिए किया जाता है। ऑडियो फाइल के पाथ में ऐबसॉल्युट या रिलेटिव URL हो सकते हैं।

Syntax: The <audio> tag comes in pairs; the content is written between the opening (<audio>) and the closing (</audio>) tags.

```
1  <html>
2   <head>
3     <title>Title of the document</title>
4   </head>
5   <body>
6     <audio controls>
7       <source src="/build/audios/jingle_bells.ogg" type="audio/ogg">
8       <source src="/build/audios/audio.mp3" type="audio/mpeg">
9     </audio>
10    <p>Click the play button</p>
11  </body>
12 </html>
```

PREVIEW

0:00 / 0:00

Click the play button

6. **Video Tag:** The HTML 5 <video> tag is used to embed video on an HTML document. Browsers don't all support the same video formats, so you should provide multiple video formats for correct rendering. A path to the video file is nested inside <source> tag, or src attribute.You can also include an alternate text in the <video>tag, that will be displayed in case if the browser doesn't support the video format.

HTML 5 <video> टैग का उपयोग HTML डॉक्यूमेंट पर वीडियो embed करने के लिए किया जाता है। ब्राउजर सभी एक ही जैसी वीडियो फार्मेटो को सपोर्ट नहीं करते हैं, इसलिए आपको सही पिक्चर के लिए कई वीडियो फार्मेट प्रदान करना चाहिए। वीडियो फाइल के लिए एक पाथ <source> टैग, या src ऐट्रीब्यूट के अंदर नेस्टेड है। आप <video> टैग में एक वैकल्पिक text भी शामिल कर सकते हैं, अगर ब्राउजर वीडियो फार्मेट को सपोर्ट नहीं करता है तो यह प्रदर्शित किया जाएगा।

There are 3 supported video formats for the <video> element: MP4/MPEG 4, WebM, and Ogg. For the compression/decompression of large video files special programs, codecs, are used.MP4/MPEG 4 files are used with H264 video and AAC audio codecs, WebM files - with VP8

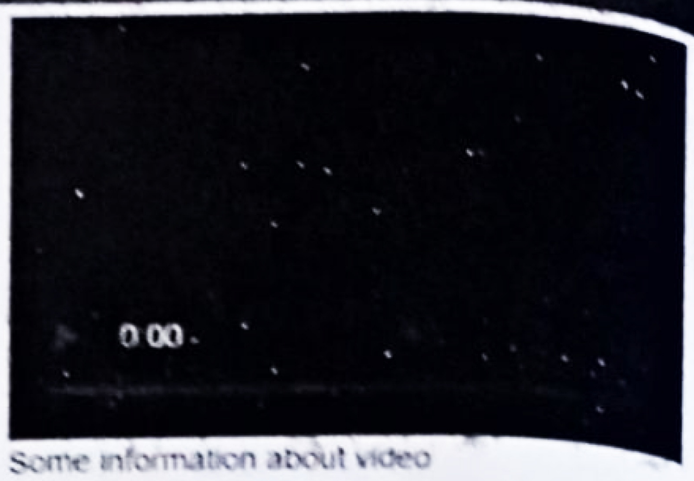
video codec and Vorbis audio codec; and Ogg files - with Theora video codec and Vorbis audio codec.

<video> एलिमेंट के लिए 3 सपोर्टेड वीडियो फार्मेट हैं: MP4/MPEG 4, WebM, and Ogg। बड़े वीडियो फाइलों के कंप्रेसन/डिकंप्रेसन के लिए विशेष प्रोग्राम, कोडेक्स का उपयोग किया जाता है। H264 वीडियो और AAC ऑडियो कोडेक, वेबएम फाइलों के साथ VP8 वीडियो कोडेक और वोरबिस ऑडियो कोडेक के साथ -MP4/MPEG 4 फाइलें उपयोग की जाती हैं; और Ogg files – थेरा वीडियो कोडेक और वोरबिस ऑडियो कोडेक के साथ।

Example

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Title of the document</title>
5      <style>
6        video {
7          width: 300px;
8          height: 200px;
9          border: 1px solid #666;
10       }
11     </style>
12   </head>
13   <body>
14     <video controls muted src="/videos/spiritualreality.mp4">
15       <track default kind="subtitles" srclang="en"
     src="/videos/spiritualreality.mp4"/>
16     </video>
17     <p>Some information about video</p>
18   </body>
19 </html>
```

PREVIEW



Some information about video

3.10.2 HTML5 Form Validations - HTML5 फॉर्म वेलिडेशन

Go to any popular site with a registration form, and you will notice that they give you feedback when you don't enter your data in the format they are expecting.

रजिस्ट्रेशन फॉर्म के साथ किसी भी लोकप्रिय साइट पर जाएं, और आप नोटिस करेंगे कि आपको फीडबैक देना है जब आप उस फार्मेट में अपना डेटा इन्टर नहीं करेंगे, जिसकी वे अपेक्षा कर रहे हैं।

You'll get messages such as: - आपको संदेश प्राप्त होंगे जैसे

- ✓ "This field is required" (you can't leave this field blank) - "यह फील्ड आवश्यक है" (आप इस फील्ड को खाली नहीं छोड़ सकतें)
- ✓ "Please enter your phone number in the format xxx-xxxx" (it enforces three numbers followed by a dash, followed by four numbers) - "कृपया xxx-xxxx प्रारूप में अपना फोन नंबर दर्ज करें" (इसमें डैश के बाद तीन नंबर और उसके बाद चार नंबर लागू होते हैं)
- ✓ "Please enter a valid e-mail address" (if your entry is not in the format of "somebody@ example.com") - "कृपया एक वैध ई–मेल पता दर्ज करें" (यदि आपकी प्रविष्टि "somebody@ example.com" के प्रारूप में नहीं है)
- ✓ "Your password needs to be between 8 and 30 characters long, and contain one uppercase letter, one symbol, and a number" - "आपका पासवर्ड 8 और 30 वर्णों के बीच लंबा होना चाहिए, और इसमें एक अपरकेस अक्षर, एक प्रतीक और एक संख्या शामिल है"

This is called form validation — when you enter data, the web application checks it to see that the data is correct. If correct, the application allows the data to be submitted to the server and (usually)

saved in a database; if not, it gives you an error message explaining what corrections need to be made. Form validation can be implemented in a number of different ways.

We want to make filling out web forms as easy as possible. So why do we insist on validating our forms?

इसे फॉर्म का वैलिडेशन कहा जाता है – जब आप डेटा इन्टर करते हैं, तो वेब एप्लिकेशन यह देखने के लिए जांचता है कि डेटा सही है। यदि सही है, तो एप्लिकेशन सर्वर को डेटा सबमिट करने की अनुमति देता है और (आमतौर पर) डेटाबेस में सेव हो जाता है। यदि नहीं, तो यह आपको एक त्रुटि संदेश देता है जो बताता है कि क्या सुधार करने की आवश्यकता है। फार्म वैलिडेशन को कई अलग–अलग तरीकों से लागू किया जा सकता है।

हम यथासंभव आसान वेब फॉर्म भरना चाहते हैं। तो हम अपने फार्म को वैलिड करने पर जोर क्यों देते हैं?

There are three main reasons: इसके तीन मुख्य कारण हैं–

1. We want to get the right data, in the right format — our applications won't work properly if our user's data is stored in the incorrect format, or if they don't enter the correct information, or omit information altogether.

   हम सही डेटा प्राप्त करना चाहते हैं, सही फार्मेट में – यदि हमारे उपयोगकर्ता का डेटा गलत फार्मेट में संग्रहीत है, या यदि वे सही जानकारी इन्टर नहीं करते हैं, या पूरी तरह से जानकारी को छोड़ना चाहते हैं, तो हमारे एप्लिकेशन ठीक से काम नहीं करेंगे।

2. We want to protect our users' accounts — by forcing our users to enter secure passwords, it makes it easier to protect their account information.

   हम अपने उपयोगकर्ताओं के खातों की सुरक्षा करना चाहते हैं – हमारे उपयोगकर्ताओं को सुरक्षित पासवर्ड इन्टर करने के लिए मजबूर करके, इससे उनके खाते की जानकारी को सुरक्षित करना आसान बनाता है।

3. We want to protect ourselves — there are many ways that malicious users can misuse unprotected forms to damage the application.

   हम अपनी सुरक्षा करना चाहते हैं – ऐसे कई तरीके हैं जिनसे दुर्भावनापूर्ण उपयोगकर्ता एप्लिकेशन को नुकसान पहुंचाने के लिए असुरक्षित रूपों का दुरुपयोग कर सकते हैं।

Warning: Never trust data passed to your server from the client. Even if your form is validating correctly and preventing malformed input, a malicious user can still alter the network request.

कभी भी ग्राहक को अपने सर्वर पर डेटा पर विश्वास न करें। भले ही आपका फॉर्म सही तरीके से मान्य हो और विकृत इनपुट को रोक रहा हो, फिर भी दुर्भावनापूर्ण उपयोगकर्ता नेटवर्क अनुरोध को बदल सकता है।

Lets see form validation attributes: - आओ हम प्रपत्र सत्यापन एट्रीब्यूट देखें

1. Required Attribute - रिक्वार्यड ऐट्रीब्यूट:

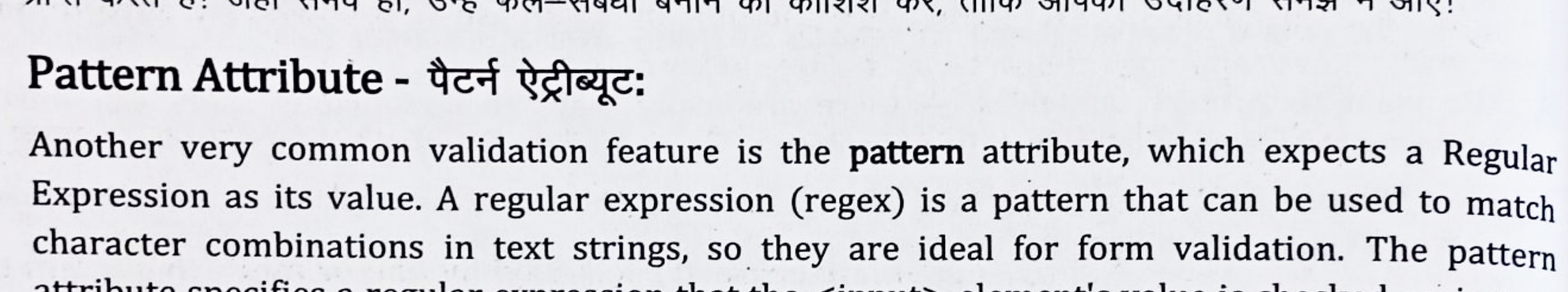
The simplest HTML5 validation feature to use is the **require** attribute — if you want to make an input mandatory, you can mark the element using this attribute. The required attribute is a **Boolean** attribute. When present, it specifies that an input field must be filled out before submitting the form.

उपयोग करने के लिए सबसे सरल एचटीएमएल 5 वैलिडेशन सुविधा का **require** ऐट्रीब्यूट है – यदि आप एक इनपुट अनिवार्य बनाना चाहते हैं, तो आप इस ऐट्रीब्यूट का उपयोग करके एलिमेंट को चिह्नित कर सकते हैं। **require** ऐट्रीब्यूट एक **Boolean** ऐट्रीब्यूट है। उपस्थित होने पर, यह स्पेशिफाई करता है कि फॉर्म सबमिट करने से पहले इनपुट फील्ड को भरना होगा।

Note: The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

रिक्वार्यड ऐट्रीब्यूट निम्न इनपुट प्रकारों के साथ काम करती है: टेक्स्ट, सर्च, यूआरएल, टेल, ईमेल, पासवर्ड, दिनांक पिकर (डेट पिक करने वाला), नंबर, चेकबॉक्स, रेडियो और फाईल।

Add a required attribute to your input, as shown below: -
अपने इनपुट में एक आवश्यक विशेषता जोड़ें, जैसा कि नीचे दिखाया गया है



In this example, the <input> element accepts one of two possible values: the string "banana" or the string "cherry". At this point, try changing the value inside the pattern attribute to equal some of the examples you saw earlier, and look at how that affects the values you can enter to make the input value valid. Try writing some of your own, and see how you get on! Try to make them fruit-related where possible, so your examples make sense!

इस उदाहरण में, <input> element दो संभावित वैल्यू में से एक को स्वीकार करता है: स्ट्रिंग "banana" या स्ट्रिंग "cherry"। इस बिंदु पर, आपके द्वारा पहले देखे गए कुछ उदाहरणों को बराबर करने के लिए पैटर्न ऐट्रीब्यूट के अंदर वैल्यू को बदलने का प्रयास करें, और देखें कि इनपुट वैल्यू को वैध बनाने के लिए आप उन मूल्यों को कैसे प्रभावित कर सकते हैं जो आप इन्टर कर सकते हैं। आप खुद कुछ लिखने की कोशिश करें, और देखें कि आप कैसे प्राप्त करते हैं! जहां संभव हो, उन्हें फल–संबंधी बनाने की कोशिश करें, ताकि आपको उदाहरण समझ में आएं!

2. Pattern Attribute - पैटर्न ऐट्रीब्यूट:

Another very common validation feature is the **pattern** attribute, which expects a Regular Expression as its value. A regular expression (regex) is a pattern that can be used to match character combinations in text strings, so they are ideal for form validation. The pattern attribute specifies a regular expression that the <input> element's value is checked against on form submission.

एक और बहुत ही सामान्य वैलिडेशन पैटर्न attribute है, जो अपने value के रूप में एक रेगुलर एक्सप्रेशन की उम्मीद करता है। एक रेगुलर एक्सप्रेशन (रेगेक्स) एक ऐसा पैटर्न है, जिसका इस्तेमाल टेक्स्ट स्ट्रिंग्स में कैरेक्टर कॉम्बिनेशन को मैच करने के लिए किया जा सकता है, इसलिए वे फॉर्म वैलिडेशन के लिए आदर्श हैं। पैटर्न ऐट्रीब्यूट एक रेगुलर एक्सप्रेशन स्पेशिफाई करता है कि फार्म सबमिशन करने पर <input> element's की वैल्यू को जांचा जाता है।

Note: The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.

पैटर्न ऐट्रीब्यूट निम्नलिखित इनपुट प्रकारों के साथ काम करती है: text, date, search, url, tel, email, and password.

Syntax: <input pattern="regexp">

Below are some examples to give you a basic idea of how they work: -
नीचे कुछ उदाहरण दिए गए हैं जिनसे आपको पता चलता है कि वे कैसे काम करते हैं

a — matches one character that is a (not b, not aa, etc.)

abc — matches a, followed by b, followed by c.

a* — matches the character a, zero or more times (+ matches a character one or more times).

[^a] — matches one character that is not a.

a|b — matches one character that is a or b.

[abc] — matches one character that is a, b, or c.

[^abc] — matches one character that is not a, b, or c.

[a-z] — matches any character in the range a–z, lower case only (you can use [A-Za-z] for lower and upper case, and [A-Z] for upper case only).

a.c — matches a, followed by any character, followed by c.

a{5} — matches a, 5 times.

a{5,7} — matches a, 5 to 7 times, but no less or more.

You can use numbers and other characters in these expressions too, such as:

[-] — matches a space or a dash.

[0-9] — matches any digit in the range 0 to 9.

You can combine these in pretty much any way you want, specifying different parts one after the other:

[Ll].*k — A single character that is an upper or lowercase L, followed by zero or more characters of any type, followed by a single lowercase k.

एक सिंगल कैरेक्टर जो एक अपर या लोवर केस L है, उसके बाद शून्य या किसी भी प्रकार का उससे बड़ा कैरेक्टर, उसके बाद एक सिंगल लोअरकेस k।

[A-Z][A-Za-z' -]+ — A single uppercase character followed by one or more characters that are an upper or lower case letter, a dash, an apostrophe, or space. This could be used to validate the city/town names of English-speaking countries, which need to start with a capital letter but don't contain any other characters. Examples from the UK include Manchester, Ashton-under-lyne, and Bishop's Stortford.

एक या एक से अधिक कैरेक्टरों के बाद एक सिंगल अपरकेस कैरेक्टर जो एक अपर या लोवर केस लेटर है, एक डैश, एक एपॉस्ट्रॉफी या स्पेस है। इसका उपयोग अंग्रेजी बोलने वाले देशों के शहर/शहर के नामों को वैलिड करने के लिए किया जा सकता है, जिन्हें एक बड़े अक्षर से शुरू करने की आवश्यकता होती है, लेकिन इसमें कोई अन्य कैरेक्टर नहीं होता है। यूके के उदाहरणों में मैनचेस्टर, एश्टन–अंडर–लिने और बिशप स्टॉर्टफोर्ड शामिल हैं।

[0-9]{3}[-][0-9]{3}[-][0-9]{4} — A simple match for a US domestic phone number — three digits, followed by a space or a dash, followed by three digits, followed by a space or a dash, followed by four digits. You might have to make this more complex, as some people write their area code in parentheses, but it works for a simple demonstration.

एक अमेरिकी घरेलू फोन नंबर के लिए एक साधारण मैच – तीन अंक, उसके बाद एक स्थान या एक डैश, तीन अंकों के बाद, एक स्थान या एक डैश, उसके बाद चार अंक। आपको इसे और अधिक जटिल बनाना पड़ सकता है, क्योंकि कुछ लोग अपना क्षेत्र कोड कोष्ठक में लिखते हैं, लेकिन यह एक साधारण प्रदर्शन के लिए काम करता है।

Anyway, let's implement an example — update your HTML to add a pattern attribute, like so:

Example 1: An <input> element with type="password" that must contain 6 or more characters.

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

```
1 <!DOCTYPE html>
2 <html><body>
3 <p>A form with a password field that must contain 6 or more
  characters:</p>
4 <form>
5  Password: <br><input type="password" name="pw" pattern=".{6,}"
  title="Six or more characters"><br><br>
6  <input type="submit">
7 </form>
8 </body></html>
```

PREVIEW

A form with a password field that must contain 6 or more characters:

Password:

Submit

Example 2: An <input> element with type="password" that must contain 8 or more characters that are of at least one number, and one uppercase and lowercase letter.

एक <input>element के साथ type="password" जिसमें 8 या अधिक कैरेक्टर होने चाहिए जो कि कम से कम एक नंबर और एक अपरकेस और लोअरकेस कैरेक्टर हों।

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

```
1 <!DOCTYPE html>
2 <html><body>
3
4 <p>A form with a password field that must contain 8 or <br>more
  characters that are of at least one number, <br>and one uppercase
  and lowercase letter:</p>
5
6 <form>
7 Password:
8  <input type="password" name="pw" pattern="(?=.*\d)(?=.*[a-z])(?=.*
  [A-Z]).{8,}" title="Must contain at least one number and one uppercase
  and lowercase letter, and at least 8 or more characters">
9  <br><br>
10  <input type="submit">
11  </form>
12 </body></html>
```

PREVIEW

A form with a password field that must contain 8 or more characters that are of at least one number, and one uppercase and lowercase letter:

Password:

Submit

Example 3: An <input> element with type="email" that must be in the following order: characters@characters.domain (characters followed by an @ sign, followed by more characters, and then a "."

After the "." sign, add at least 2 letters from a to z.

```
h1 h2 h3 ¶p b i u span link img ul •li
1 <form>
2 E-mail: <input type="email" name="email" pattern="[a-z0-
  9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$">
3  <br><br>
4 <input type="submit">
5 </form>
```

PREVIEW

E-mail:

Submit

Example 4: An <input> element with type="url" that must start with http:// or https:// followed by at least one character.

```
h1 h2 h3 ¶p b i u span link img ul •li
1 <form>
2 Homepage:
3  <input type="url" name="website" pattern="https?://.+"
  title="Include http://">
4 <input type="submit">
5 </form>
```

PREVIEW

Homepage: www.pss.org Submit

Please enter a URL.

3. Autofocus Attribute - ऑटोफोकस ऐट्रीब्यूट

The autofocus attribute is a **Boolean** attribute. When present, it specifies that an <input> element should automatically get focus when the page loads. Only one form-associated element in a document can have this attribute specified. If there are several, the first element with the attribute set inserted, usually the first such element on the page, gets the initial focus.

ऑटोफोकस ऐट्रीब्यूट एक **Boolean** ऐट्रीब्यूट है। वर्तमान में, यह स्पेशिफाई करता है कि पेज लोड होने पर एक <input> एलिमेंट को स्वचालित रूप से फोकस प्राप्त करना चाहिए। किसी डॉक्यूमेंट में केवल एक फार्म–एसोसियेटेड एलिमेंट में यह ऐट्रीब्यूट स्पेशिफाई हो सकता है। यदि कई हैं, तो ऐट्रीब्यूट सेट के साथ पहला एलिमेंट, आमतौर पर पृष्ठ पर पहला ऐसा एलिमेंट होता है, जो प्रारंभिक ध्यान केंद्रित करता है।

Syntax: <input autofocus>

Example:

```
<!DOCTYPE html>
<html>
<body>
<h3>Autofocus Demo</h3>
<form>
  <input type="text"><br><br>
<input type="checkbox" autofocus><br><br>
<select id="mySelect">
  <option>Option 1</option>
  <option>Option 2</option>
</select><br>
<input type="submit">
</form>
</body>
</html>
```

Autofocus Demo

Option 1 ▾

Submit