**HTML <input> Tag with pattern Attribute**

**1. Introduction**

HTML provides various attributes to validate form inputs on the client side before they are submitted to the server. One of the most powerful attributes is the pattern attribute, which allows developers to define a **regular expression** (regex) that the input must match.

This document explains the syntax, usage, examples, and benefits of using the pattern attribute in HTML <input> elements.

**2. What is the pattern Attribute?**

The pattern attribute is used with <input> elements of type text, search, url, tel, email, or password. It specifies a **regular expression** the input's value must conform to in order to be considered valid.

**Syntax:**

<input type="text" pattern="regular\_expression">

**3. Why Use the pattern Attribute?**

* To enforce **input validation** on the client side.
* To improve **form reliability and security**.
* To ensure data consistency before submission.
* To reduce invalid entries and user errors.

**4. Common Use Cases**

**a) Validating a Phone Number:**

<input type="tel" pattern="[0-9]{10}" title="Enter 10-digit phone number">

✔ Accepts exactly 10 digits.

**b) Validating an Email Format:**

<input type="email" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$" title="Enter a valid email address">

✔ Accepts common email formats like example@mail.com.

**c) Validating a Postal Code (e.g., 6-digit Indian PIN code):**

<input type="text" pattern="[1-9][0-9]{5}" title="Enter a valid 6-digit PIN code">

✔ Ensures the PIN does not start with 0 and has exactly 6 digits.

**d) Validating a Password with Special Characters:**

<input type="password" pattern="(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}" title="Must contain at least 8 characters, including UPPER/lowercase and number">

✔ Requires at least one number, one uppercase, one lowercase, and minimum 8 characters.

**5. title Attribute (Optional but Recommended)**

The title attribute helps the user understand what format is expected. This message appears when the user hovers over the input field or when validation fails.

**6. Important Notes**

* The pattern attribute **only works on form submission**.
* It **does not provide complete security** — always validate on the server too.
* If the input does not match the pattern, the form will not be submitted, and a **browser error message** is shown.
* Not supported for <input type="number"> — instead, use min and max.

**7. Complete Example Form Using pattern**

<form>

<label for="username">Username (only letters):</label>

<input type="text" id="username" pattern="[A-Za-z]+" title="Only letters allowed" required><br><br>

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

<input type="email" id="email" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$" title="Enter a valid email" required><br><br>

<label for="phone">Phone:</label>

<input type="tel" id="phone" pattern="[0-9]{10}" title="Enter 10-digit phone number" required><br><br>

<label for="password">Password:</label>

<input type="password" id="password" pattern="(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}"

title="At least 8 characters with UPPER/lowercase and a number" required><br><br>

<button type="submit">Submit</button>

</form>

**8. Benefits of Using pattern in HTML**

* Enhances user experience by providing immediate feedback.
* Reduces load on the server with front-end validation.
* Makes form input more structured and readable.
* Works without JavaScript.

**9. Conclusion**

The pattern attribute in HTML is a powerful way to validate input using regular expressions. It helps ensure form data is entered in the correct format before submission, improving both user experience and application quality. However, it should always be used in conjunction with server-side validation for maximum security.