

1- Form Validation

Khá thường xuyên bạn gặp một website mà ở đó người dùng nhập các thông tin vào một biểu mẫu (form) trước khi gửi tới máy chủ. Chẳng hạn biểu mẫu đăng ký tài khoản. Các thông tin mà người dùng nhập vào biểu mẫu cần phải được xác thực (validate) để đảm bảo sự hợp lý của dữ liệu.

Register

Create your account. It's free and only takes a minute.

First Name

Last Name

Email

Password

Confirm Password

☐ I accept the [Terms of Use & Privacy Policy](#).

Register Now

Already have an account? [Sign in](#)

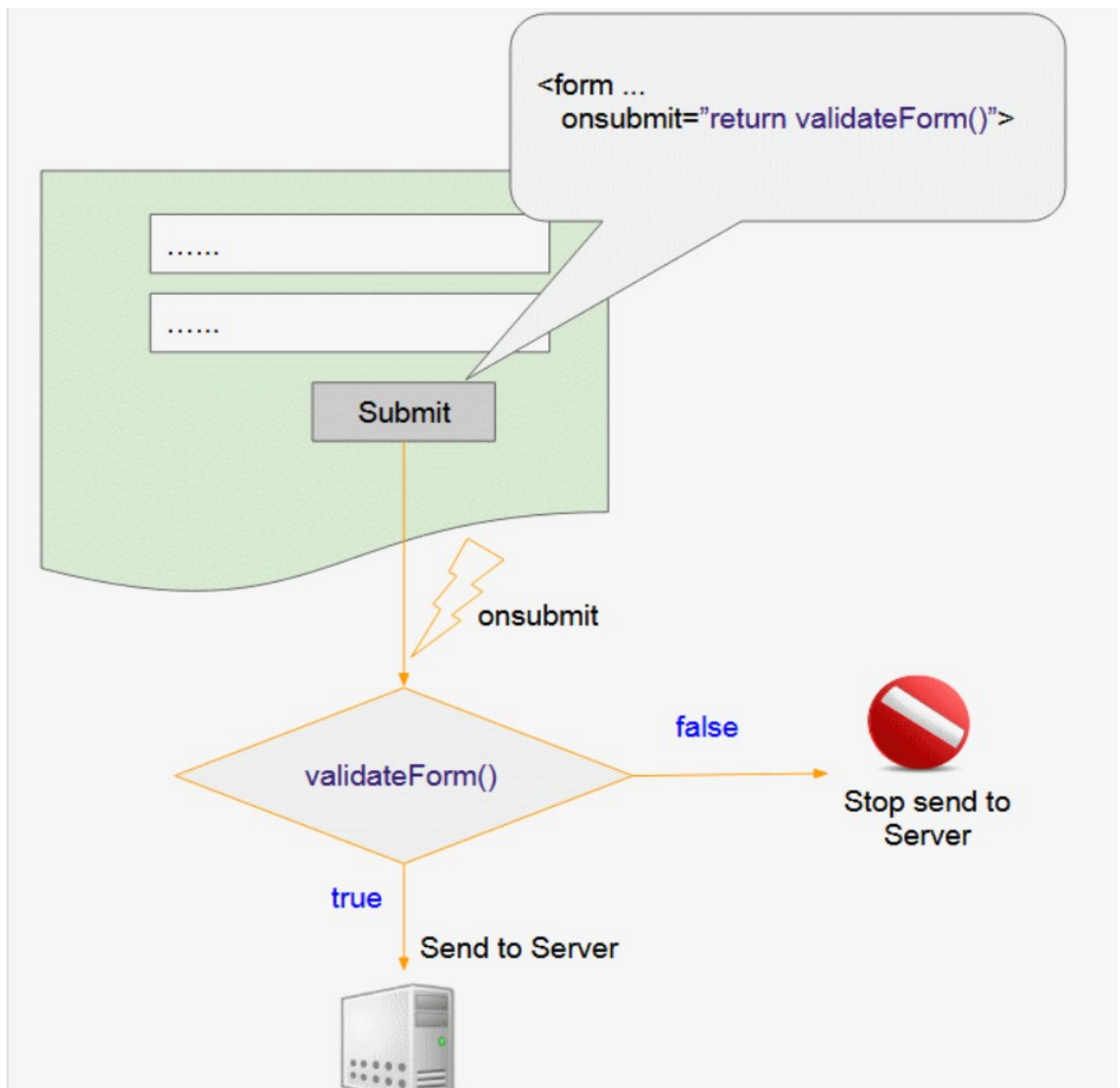
Về cơ bản có 3 cách để xác thực dữ liệu:

Dữ liệu của form sẽ được gửi tới server (máy chủ), và việc xác thực (validate) sẽ được thực hiện tại phía máy chủ.

Dữ liệu của form sẽ được xác thực tại phía client bằng cách sử dụng Javascript, điều này giúp server không phải làm việc quá nhiều, và tăng hiệu năng cho ứng dụng.

Sử dụng cả 2 phương thức trên để xác thực form.

Trong bài học này tôi sẽ thảo luận về việc sử dụng Javascript để xác thực (validate) form. Dưới đây là hình minh họa mô tả hành vi của chương trình khi người dùng nhấn vào nút Submit.



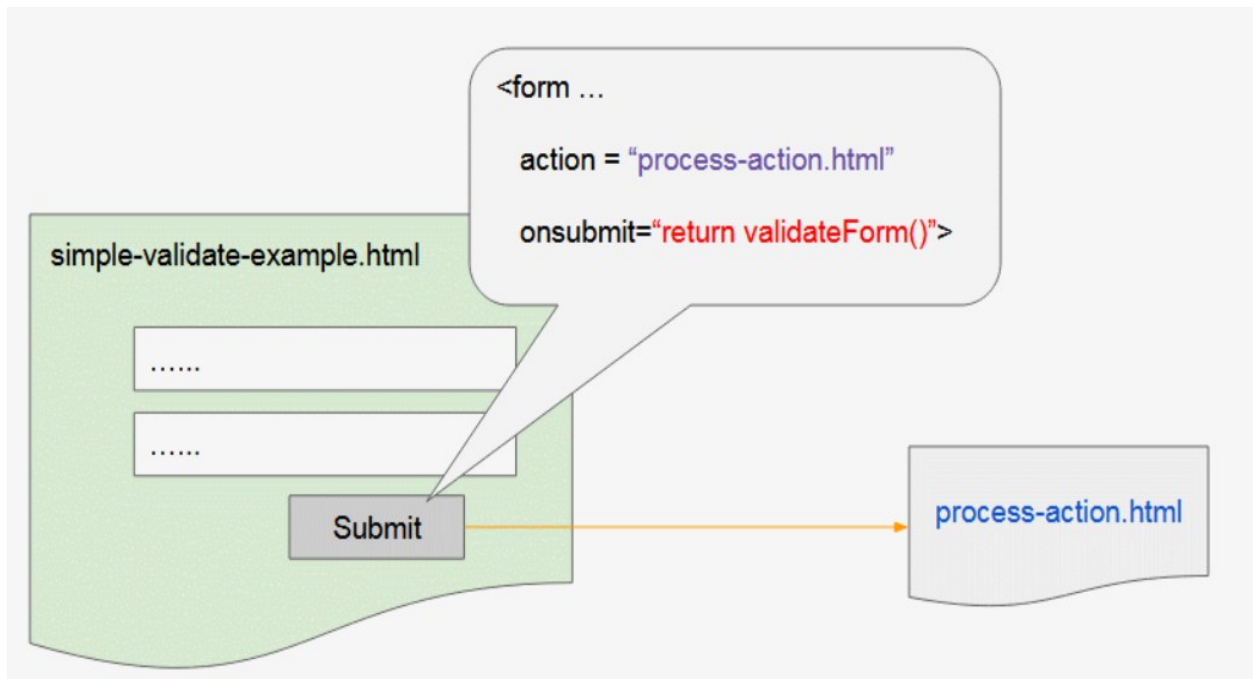
Bạn phải đăng ký một hàm liên hợp với sự kiện `onsubmit` của form. Nhiệm vụ của hàm này là kiểm tra dữ liệu mà người dùng đã nhập vào form, và trả về `true` nếu tất cả các thông tin người dùng nhập vào hợp lệ, ngược lại trả về `false`.

Khi người dùng nhấn vào nút Submit, hàm liên hợp với sự kiện `onsubmit` sẽ được gọi.

Nếu hàm liên hợp với sự kiện `onsubmit` trả về `true` dữ liệu của form sẽ được gửi tới server. Ngược lại hành động Submit sẽ bị hủy bỏ.

2- Ví dụ đơn giản

OK, đây là một ví dụ đơn giản giúp bạn hiểu về nguyên tắc hoạt động của Form trước khi thực hành các ví dụ phức tạp hơn.



Thuộc tính (attribute) `action` của `<form>` được sử dụng để chỉ định trang mà dữ liệu sẽ được gửi đến, hay nói cách khác đây chính là trang sẽ xử lý dữ liệu được gửi đến từ `<form>` của trang hiện tại.

Các trang xử lý dữ liệu gửi đến từ form thường được viết bởi công nghệ Servlet/JSP, PHP hoặc một công nghệ nào đó ở phía Server thay vì một trang HTML. Tuy nhiên tôi không đề cập tới việc xử lý dữ liệu trong bài học này.

| | |
|------------------------------------|--|
| simple-validation-example.html | |
| <code><!DOCTYPE html></code> | |
| <code><html></code> | |

```
<head>

<title>Hello Javascript</title>

<script type = "text/javascript">

    function validateForm() {

        var u = document.getElementById("username").value;

        var p = document.getElementById("password").value;


        if(u== "") {

            alert("Please enter your Username");

            return false;

        }

        if(p == "") {

            alert("Please enter you Password");

            return false;

        }


        alert("All datas are valid!, send it to the server!")


        return true;

    }

</script>

</head>

<body>


<h2>Enter your Username and Password</h2>


<div style="border:1px solid #ddd;padding: 5px;">

    <form method="GET" action="process-action.html" onsubmit =

"return validateForm()">
```

| | |
|---|--|
| <pre> Username: <input type="text" name="username" id="username"/> Password: <input type="password" name = "password" id="password"/> <button type="submit">Submit</button> </form> </div> </body> </html> </pre> | |
|---|--|

| | |
|---|--|
| process-action.html | |
| <pre> <!DOCTYPE html> <html> <head> <title>Process Action</title> </head> <body> <h3>Process Action Page</h3> OK, I got data! [Go Back] </pre> | |

| | |
|--|--|
| <pre></body> </html></pre> | |
|--|--|

3- Truy cập vào các dữ liệu của form

Truy cập vào dữ liệu của một trường (field) thông qua ID của trường.

```
<input type="text" id="username"/>
```

```
<input type="password" id="password"/>
```

// Access field via ID:

```
var field = document.getElementById("fieldId");
```

```
var value = field.value;
```

Truy cập vào các trường của Form thông qua thuộc tính name:

```
<form name="myForm" ...>
```

```
<input type="text" name="username"/>
```

```
<input type="password" name = "password"/>
```

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

```
</form>
```

// Get form via form name:

```
var myForm = document.forms["myForm"];
```

```
var u = myForm["username"].value;
```

```
var p = myForm["password"].value;
```

Khi người dùng nhập dữ liệu không chính xác trên một trường của form, bạn nên thông báo cho người dùng đồng thời focus vào trường đó.

| | |
|--|--|
| validation-example1.html | |
| <pre><!DOCTYPE html> <html> <head> <title>Validation</title> <script type = "text/javascript"> function validateForm() { // Get form via form name: var myForm = document.forms["myForm"]; var u = myForm["username"].value; var p = myForm["password"].value; if(u== "") { alert("Please enter your Username"); myForm["username"].focus(); // Focus return false; } if(p == "") {</pre> | |

| | |
|---|--|
| <pre> alert("Please enter you Password"); myForm["password"].focus(); // Focus return false; } alert("All datas are valid!, send it to the server!") return true; } </script> </head> <body> <h2>Enter your Username and Password</h2> <div style="border:1px solid #ddd;padding: 5px;"> <form name="myForm" method="GET" action="process-action.html" onsubmit = "return validateForm()"> Username: <input type="text" name="username"/> Password: <input type="password" name = "password"/> <button type="submit">Submit</button> </form></pre> | |
|---|--|

| | |
|--|--|
| <pre> } </script> </head> <body> <h2>Enter a Number between 0 and 10</h2> <div style="border:1px solid #ddd;padding: 5px;"> <form name="myForm" action="process-action.html" onsubmit = "return validateForm()"> Number: <input type="text" id= "myNumber"/> <button type="submit">Submit</button> </form> </div> </body> </html> </pre> | |
|--|--|

4- Submit thông qua Javascript

Nhấn vào nút `<button type="submit">` hoặc `<input type="submit">` bên trong form giúp bạn gửi dữ liệu của form này tới máy chủ, tuy nhiên bạn cũng có thể làm điều đó thông qua Javascript.

| | |
|--------------------------------|--|
| javascript-submit-example.html | |
| <!DOCTYPE html> | |

```
<html>

<head>
  <title>Javascript Submit</title>
  <script type = "text/javascript">

    function validateForm() {
      var name =
document.forms["myForm"]["fullName"].value;
      if(name == "") {
        alert("Please enter your name");
        return false;
      }
      return true;
    }

    function submitByJavascript() {

      var valid = validateForm();
      if(!valid) {
        return;
      }

      var myForm = document.forms["myForm"];

      myForm.submit();
    }
  </script>
</head>
<body>
```

| | |
|--|--|
| <pre> <h2>Submit a form with Javascript</h2> <div style="border:1px solid #ddd;padding: 5px;"> <form name="myForm" action="process-action.html" onsubmit = "return validateForm()"> Your Name: <input type="text" name = "fullName" value =""/> <button type="submit">Submit</button> </form> </div> <!-- A Button outside the form --> Button outside the form: <button onclick="submitByJavascript()">Click Me to submit form</button> </body> </html> </pre> | |
|--|--|

5- Validate tự động

Trình duyệt có thể tự động xác thực (validate) một vài loại dữ liệu trên form, chẳng hạn thêm thuộc tính required vào một trường của form để nói với trình duyệt rằng trường này là bắt buộc, trình duyệt sẽ tự động kiểm tra và thông báo cho người dùng nếu người dùng không nhập vào trường đó.

Chú ý: Các trình duyệt quá cũ, chẳng hạn IE phiên bản 9 hoặc cũ hơn không hỗ trợ validate tự động

| | |
|---|--|
| required-example.html | |
| <pre><!DOCTYPE html> <html> <head> <title>Required</title> </head> <body> <h2>Required attribute</h2> <div style="border:1px solid #ddd; padding: 5px;"> <form name="myForm" action="process-action.html" onsubmit = "return validateForm()"> Your Name: <input type="text" name = "fullName" value = "" required/> <button type="submit">Submit</button> </form> </div> </body> </html></pre> | |

Một vài loại phần tử `<input>` mới được giới thiệu trong HTML 5, chẳng hạn `color`, `date`, `datetime-local`, `email`, `month`, `number`, `range`, `search`, `tel`, `time`, `url`, `week`,.. Các phần tử này có các thuộc tính (attribute) đặc biệt giúp trình duyệt biết cách để validate dữ liệu của nó một cách tự động. Dưới đây là danh sách một vài thuộc tính như vậy:

| Attribute | Description |
|-----------------------|---|
| <code>disabled</code> | Chỉ định rằng phần tử Input này sẽ bị vô hiệu hóa (disabled). |
| <code>max</code> | Chỉ định giá trị lớn nhất của phần tử Input này. |
| <code>min</code> | Chỉ định giá trị nhỏ nhất của phần tử Input này. |
| <code>pattern</code> | Chỉ định pattern của phần tử Input này. |
| <code>required</code> | Chỉ định rằng trường đầu vào là bắt buộc. Người dùng phải nhập dữ liệu. |
| <code>type</code> | Chỉ định kiểu của phần tử Input . |

Ví dụ: Một phần tử `<input type="number">` với các thuộc tính `min`, `max`, trình duyệt sẽ thông báo cho người dùng nếu họ nhập vào một con số nằm ngoài phạm vi cho phép.

Lưu ý: Xem chi tiết danh sách các phần tử `<input>` và các thuộc tính tương ứng với từng phần tử này:

| | |
|---|--|
| attr-min-max-example.html | |
| <pre><!DOCTYPE html> <html> <head> <title>Min Max Attributes</title> </head> <body> <h2>Attribute min, max</h2></pre> | |

| | |
|--|--|
| <pre> <div style="border:1px solid #ddd;padding: 5px;"> <form name="myForm" action="process-action.html"> Enter your score (0-100): <input type="number" name = "score" min= "0" max = "100"/> <button type="submit">Submit</button> </form> </div> </body> </html> </pre> | |
|--|--|

Ví dụ: Yêu cầu người dùng nhập vào một mã số quốc gia có 2 ký tự.

| | |
|---|--|
| attr-pattern-example.html | |
| <pre> <!DOCTYPE html> <html> <head> <title>pattern attribute</title> </head> <body> <h2>Attribute: pattern</h2> <div style="border:1px solid #ddd;padding: 5px;"> </pre> | |

| | |
|--|--|
| <pre> <form name="myForm" action="process-action.html"> Country code: <input type="text" name = "countryCode" pattern="[A-Za-z]{2}" title="Two letter country code"/> <button type="submit">Submit</button> </form> </div> </body> </html> </pre> | |
|--|--|

Ví dụ: Yêu cầu người dùng nhập vào mật khẩu có ít nhất 8 ký tự.

| | |
|---|--|
| attr-pattern-example2.html | |
| <pre> <!DOCTYPE html> <html> <head> <title>pattern attribute</title> </head> <body> <h2>Attribute: pattern</h2> <div style="border:1px solid #ddd; padding: 5px;"> <form name="myForm" action="process-action.html"> </pre> | |

| | |
|--|--|
| <pre> Password: <input type="password" name = "password" pattern=".{8,}" title="8 or more characters"/> <button type="submit">Submit</button> </form> </div> </body> </html> </pre> | |
|--|--|

Ví dụ: Yêu cầu người dùng nhập vào một mật khẩu mạnh, có ít nhất 8 ký tự, có ít nhất một chữ hoa (uppercase), và có ít nhất một chữ thường (lowercase).

| | |
|---|--|
| attr-pattern-password-example.html | |
| <pre> <!DOCTYPE html> <html> <head> <title>pattern attribute</title> </head> <body> <h2>Attribute: pattern</h2> Password must contain 8 or more characters that are of at least one number, and one uppercase and lowercase letter: </pre> | |

| | |
|---|--|
| <pre> <div style="border:1px solid #ddd; padding: 5px;" <form name="myForm" action="process-action.html"> Password: <input type="password" name = "password" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}" title="Invalid password!"/> <button type="submit">Submit</button> </form> </div> </body> </html> </pre> | |
|---|--|

Ví dụ: Yêu cầu người dùng nhập vào địa chỉ email, sử dụng thuộc tính pattern để đảm bảo người dùng nhập vào một email đúng định dạng.

| | |
|--|--|
| attr-pattern-email-example.html | |
| <pre> <!DOCTYPE html> <html> <head> <title>pattern attribute</title> </head> <body> <h2>Attribute: pattern</h2> </pre> | |

| | |
|--|--|
| <pre><div style="border:1px solid #ddd; padding: 5px;"> <form name="myForm" action="process-action.html"> Email: <input type="password" name = "password" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}\$" title="Invalid password!"/> <button type="submit">Submit</button> </form> </div> </body> </html></pre> | |
|--|--|