

# Phát triển ứng dụng web

Javascript  
part 2

## Nội dung

- HTML Form
- Regular Expressions
- Javascript Validation
- JQuery

## Nội dung

- **HTML Form**
- Regular Expressions
- Javascript Validation
- JQuery

## Giới thiệu Form

HTML Form



## Giới thiệu Form

- Được dùng để nhận dữ liệu từ phía người dùng
- Giúp gởi yêu cầu của người dùng đến trang xử lý trong ứng dụng web
- Tag **<form>** dùng để chứa các thành phần khác của form
- Những thành phần nhập liệu được gọi là **Form Field**
  - text field
  - password field
  - multiple-line text field
  - .....

Ví dụ

**YAHOO! VIỆT NAM**

**Chào Bạn!**

Chúng tôi sẽ giúp bạn lập tài khoản Yahoo! theo ba bước rất đơn giản! Bạn chỉ cần trả lời một số câu hỏi đơn giản, chọn ID và mật khẩu, và thế là xong.

Tôi thích nhận nội dung bằng tiếng: Yahoo! Việt Nam

**1. Hãy cho chúng tôi biết về bản thân bạn...**

Tên của Tôi:  Tên họ  Tên gọi

Giới Tính: - Chọn Một Mục -

Ngày Tháng Năm Sinh:  Ngày  Chon Tháng  Năm

Tôi sống tại: Việt Nam

Mã Bưu Chính:

**2. Chọn ID và mật khẩu**

Yahoo! ID và Email:  @yahoo.com.vn

Mật Khẩu:  Mức Độ Mật Khẩu

Đánh Lại Mật Khẩu:

**3. Để đề phòng trường hợp bạn quên ID hoặc mật khẩu...**

Email Thay Thế Khác:

Câu Hỏi Bảo Mật: - Chọn Một Mục -

Câu Trả Lời của Bạn:

Chỉ còn một số chi tiết nữa...

☐ Các Lựa Chọn Ưu Tiên về Tiếp Thị

☐ Tôi khi Công ty Compaq có thể liên lạc với tôi về các chương trình chào hàng đặc biệt, các chương trình khuyến mãi và các tính năng Yahoo!

**Đăng nhập**

[Quên mật khẩu hoặc Yahoo! ID của bạn?](#)

## Tag <Form>

- Là container chứa các thành phần nhập liệu khác.

```
<form name="..." action="..." method="...">
    <!-- các thành phần của Form -->
</form>
```

- Các thuộc tính của <form>
  - **NAME** : tên FORM
  - **ACTION** : chỉ định trang web nhận xử lý dữ liệu từ FORM này khi có sự kiện click của button SUBMIT.
  - **METHOD** : Xác định phương thức chuyển dữ liệu (POST,GET)

## Tag <Form> - Ví dụ

```
Dangnhap.htm
<html>
  <body>
    <form name="Dangnhap"
          action="/admin/xlDangnhap.php"
          method="Post">
      .....
    </form>
  </body>
</html>
```

## Các thành phần của Form

HTML Form



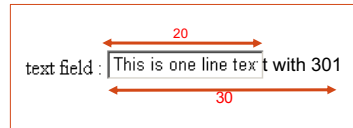
## Các thành phần của Form

- Gồm các loại Form Field sau:
  - Text field
  - Password field
  - Hidden Text field
  - Check box
  - Radio button
  - File Form Control
  - Submit Button, Reset Button, Generalized Button
  - Multiple-line text field
  - Label
  - Pull-down menu
  - Scrolled list
  - Field Set

## Text Field

- Dùng để nhập một dòng văn bản
- Cú pháp

```
<INPUT
  TYPE           = "TEXT"
  NAME           = string
  READONLY
  SIZE           = variant
  MAXLENGTH      = long
  TABINDEX       = integer
  VALUE          = string
  .....
```



- Ví dụ

```
<input type="text" name="txtName" value="This is one line text
with 301" size="20" maxlength="30">
```

## Password Field

- Dùng để nhập mật khẩu
- Cú pháp

```
<INPUT
  TYPE           = "PASSWORD"
  NAME           = string
  READONLY
  SIZE           = variant
  MAXLENGTH      = long
  TABINDEX       = integer
  VALUE          = string
  .....
```

password field :

- Ví dụ

```
<input type="Password" name="txtPassword"
value="123456abc1234" size="20" maxlength="30">
```

## Hidden Text Field

- Dùng để truyền 1 giá trị của thuộc tính value khi form được submit
- Không hiển thị ra trên màn hình
- Cú pháp

```
<INPUT
  TYPE           = "HIDDEN"
  NAME           = string
  READONLY
  SIZE           = variant
  MAXLENGTH      = long
  TABINDEX       = integer
  VALUE          = string
  .....
```

hidden text field:

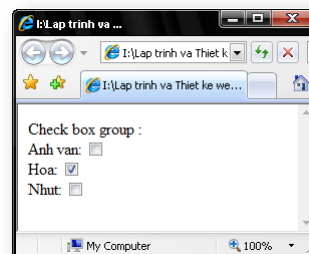
- Ví dụ :  
hidden text field : `<input type="hidden" name="txtHidden" value="This is hidden text. You can't see.">`

## Check box

- Cú pháp

```
<input
  TYPE           = "checkbox"
  NAME           = "text"
  VALUE          = "text"
  [checked]
```

- Ví dụ



```
<html>
<body>
  Check box group : <br>
  Anh van: <input type="checkbox" name="Languages" value="En"><br>
  Hoa: <input type="checkbox" name="Languages" value="Chz" checked><br>
  Nhut: <input type="checkbox" name="Languages" value="Jp"><br>
</body>
</html>
```

## Radio button

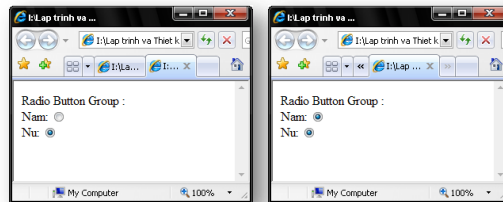
- Cú pháp

```
<input
  TYPE    = "radio"
  NAME    = "text"
  VALUE   = "text"
  [checked]
>
```

- Ví dụ

```
<html>
<body>
  Radio Button Group : <br>
  Nam: <input type="radio" name="sex" value="nam" checked><br>
  Nu: <input type="radio" name="sex" value="nu" checked ><br>
</body>
</html>
```

```
<html>
<body>
  Radio Button Group : <br>
  Nam: <input type="radio" name="sex1" value="nam" checked><br>
  Nu: <input type="radio" name="sex2" value="nu" checked ><br>
</body>
</html>
```



## File upload Control

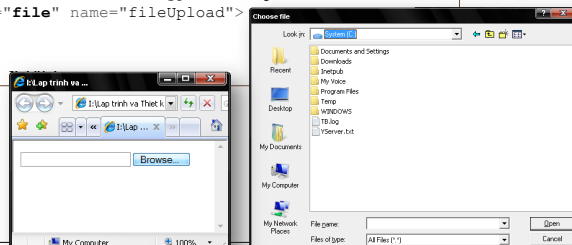
- Dùng để upload 1 file lên server

- Cú pháp

```
<form action="..." method="post" enctype="multipart/form-data"
name="...">
  <input TYPE="FILE" NAME="...">
</form>
```

- Ví dụ

```
<html>
<body>
  <form name="frmMain" action="POST" enctype="multipart/form-data">
    <input type="file" name="fileUpload">
  </form>
</body>
</html>
```





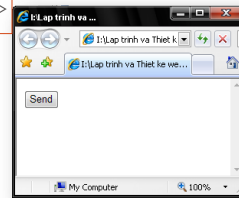
## Submit button

- Nút phát lệnh và gửi dữ liệu của form đến trang xử lý.
- Mỗi form chỉ có một nút submit và nút này được viền đậm
- Cú pháp:

```
<input TYPE="submit" name="..." value="...">
```

- Ví dụ:

```
<input type="submit" name="btnSend" value="Send">
```



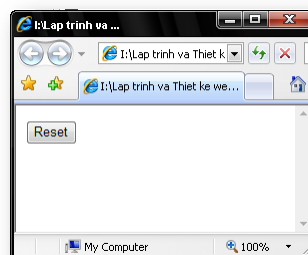
## Reset Button

- Dùng để trả lại giá trị mặc định cho các control khác trong form
- Cú pháp

```
<input TYPE="reset" name="..." value="...">
```

- Ví dụ

```
<input type="reset" name="btnReset" value="Rest">
```



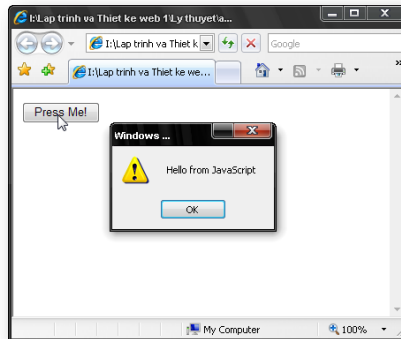
## Generalized Button

- Cú pháp

```
<input type="button" name="..." value="..." onclick="script">
```

- Ví dụ

```
<input type="button" name="btnNormal" value="Press Me!"  
onclick="alert('Hello from JavaScript');" >
```



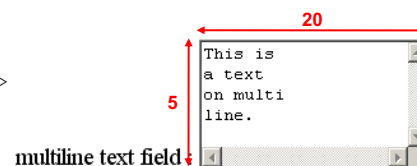
## Multiline Text Field

- Dùng để nhập văn bản nhiều dòng
- Cú pháp

```
<TEXTAREA  
  COLS           = long  
  ROWS           = long  
  DISABLED  
  NAME           = string  
  READONLY  
  TABINDEX       = integer  
  WRAP           = OFF | PHYSICAL | VIRTUAL> .....  
</TEXTAREA>
```

- Ví dụ

```
<textarea cols="20" rows="5" wrap="off">  
  This is a text on multiline.  
</textarea>
```



## Label

- Dùng để gán nhãn cho một Form Field
- Cú pháp

```
<LABEL
  FOR = IDString
  CLASS=string
  STYLE=string
>
```

- Ví dụ

Anh văn: ☐

```
<label for="Languages">Anh văn: </label>
<input type="checkbox" name="Languages" id="Languages" value="Eng">
```

## Pull-down Menu

- Dùng để tạo ra một combo box
- Cú pháp

```
<Select name="...">
  <optgroup label="...">
    <option [selected] value="..." >.....</option>
    .....
  </optgroup>

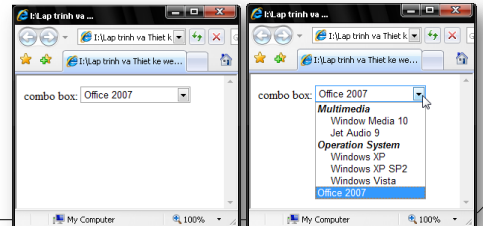
  <option [selected] value="..." >.....</option>
  .....
</select>
```

## Pull-down Menu

```

<html>
  <body>
    combo box:
    <select name="DSSoftware">
      <optgroup label="Multimedia">
        <option value="WM10">Window Media 10</option>
        <option value="JA9">Jet Audio 9</option>
      </optgroup>
      <optgroup label="Operation System">
        <option value="WXP">Windows XP</option>
        <option value="WXPSP2">Windows XP SP2</option>
        <option value="WVT">Windows Vista</option>
      </optgroup>
      <option selected value="Office07">Office 2007</option>
    </select>
  </body>
</html>

```



## Field Set

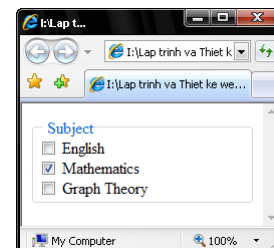
- Dùng để tạo ra Group box, nhóm các thành phần nhập liệu trong form
- Cú pháp

```

<fieldset>
  <legend>GroupBox's Name</legend>
  <input .....>
  ...
</fieldset>

```

- Ví dụ



```

<html>
<body>
  <fieldset>
    <legend>Subject</legend>
    <input type="checkbox" name="Subjects" value="Eng"> English<br>
    <input type="checkbox" name="Subjects" value="Math" checked> Mathematics<br>
    <input type="checkbox" name="Subjects" value="GraphTheory"> Graph Theory<br>
  </fieldset>
</body>
</html>

```

## Phương thức GET/POST

### HTML Form

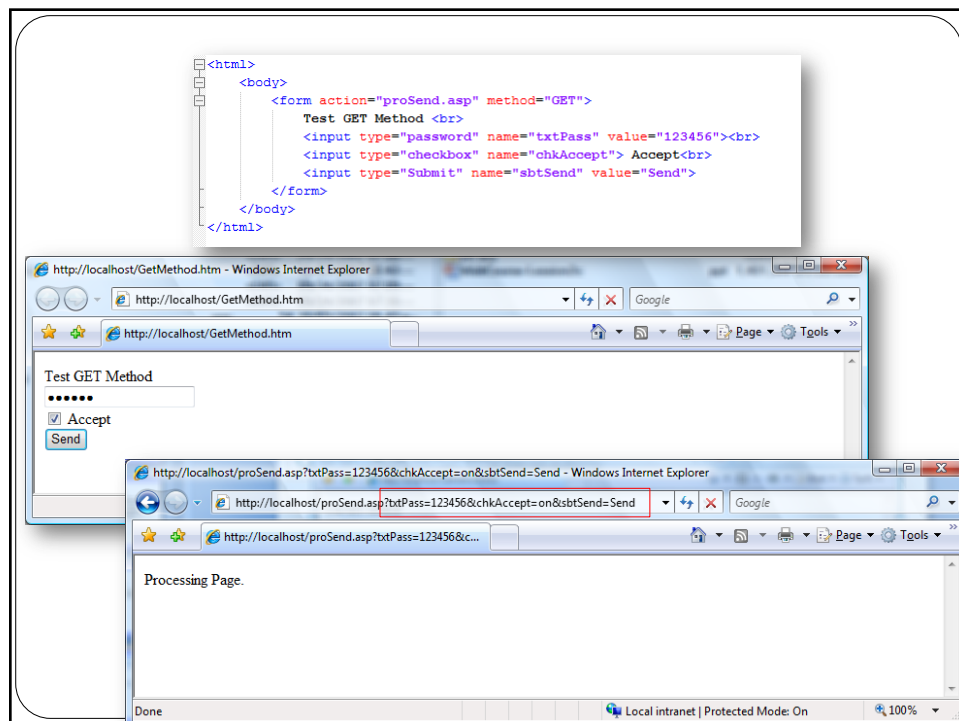


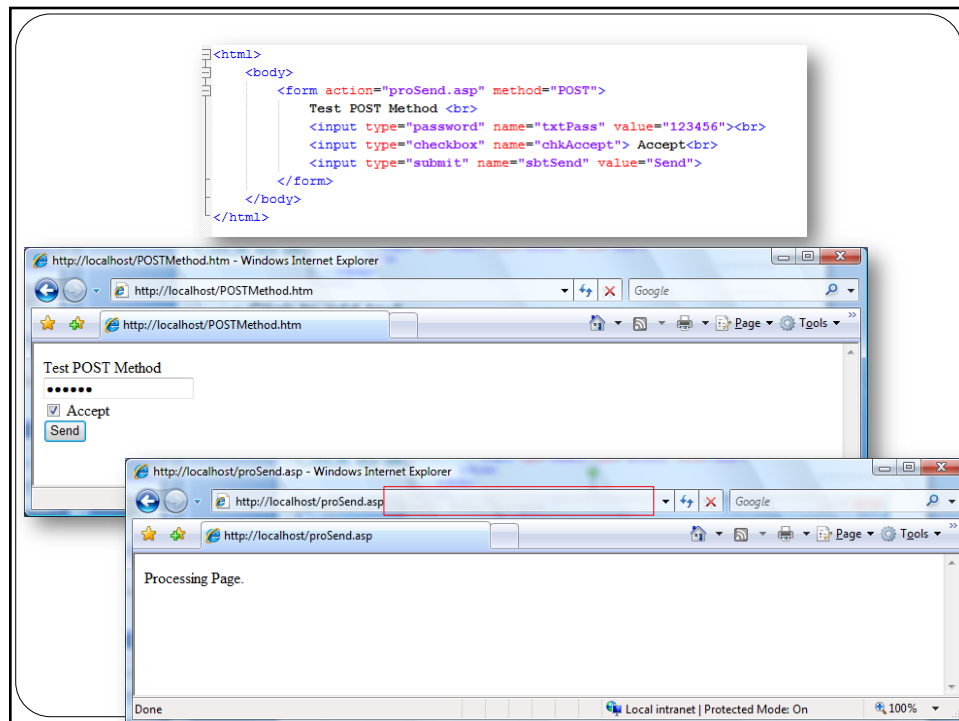
## Phương thức GET

- Các đối số của Form được ghi thêm theo vào đường dẫn URL của thuộc tính Action trong tag <Form>
- Khối lượng dữ liệu đối số được truyền đi của Form bị giới hạn bởi chiều dài tối đa của một URL trên Address bar.
  - IE : Tối đa của một URL là 2.048 ký tự
  - Firefox : Tối thiểu của một URL là khoảng 100.000 ký tự
  - Safari : Tối thiểu của một URL là 80.000 ký tự
  - Opera : Tối thiểu của một URL là 190.000 ký tự
  - Apache Server : Tối đa của một URL là 8.192 ký tự
  - IIS Server : Tối đa của một URL là 16.384 ký tự

## Phương thức POST

- Các đối số của Form được truyền “ngầm” bên dưới
- Khối lượng dữ liệu đối số được truyền đi của Form **không** phụ thuộc vào URL → Không bị giới hạn
- Chỉ sử dụng được phương thức truyền POST khi Action chỉ định đến trang web thuộc dạng trang web **có mã lệnh xử lý trên Server**





## Nội dung

- *HTML Form*
- **Regular Expressions**
- Javascript Validation
- JQuery

## Regular Expression

- **Regular Expression** là một ngôn ngữ cực mạnh dùng mô tả văn bản cũng như thao tác trên văn bản.
- **Regular Expression** là kỹ thuật xác định một chuỗi các ký tự sẽ được sử dụng trong mẫu tìm kiếm. **Regular Expression** được viết theo một cú pháp cụ thể và sau đó thường được áp dụng trên một chuỗi văn bản lớn hơn để xem chuỗi có đáp ứng các điều kiện được xác định trong **Regular Expression** hay không.

## Syntax

- Regular expression có cú pháp chung với 1 ***pattern*** và ***modifier*** như sau:

`/pattern/modifiers`

- **Pattern** là chuỗi ký tự liên tiếp.
- **Modifier** là 1 ký tự xác định cách xử lý của ***regular expression***



## Cấu trúc Regular Expression

- Cơ bản bao gồm 2 thành phần là:
  - **Literal** (trực kiện): đại diện cho ký tự cần so khớp (a, b, ..., Z, 0, 1, ..., 9, \_...).
  - **Meta characters** (siêu ký tự): là ký tự đặc biệt (\, -, [, ], (, ), ^, \$, ...) hoạt động như chỉ thị lệnh trong regular expression.
- Ví dụ kiểm tra toàn số: `/^\d*$/.`

## Ví dụ

```
let str = "Mon hoc phat trien ung dung web";  
let regex = /o/;  
console.log(str.match(regex));
```



```
[  
  "o",  
  index: 1,  
  input: "Mon hoc phat trien ung dung web",  
  groups: undefined  
]
```

## Sử dụng Meta Characters

```
let str = "Mon hoc phat trien ung dung web";
let regex = /.e./;
console.log(str.match(regex));
```



```
[
  "ien",
  index: 15,
  input: "Mon hoc phat trien ung dung web",
  groups: undefined
]
```

## Bộ Metacharacters cơ bản

Ký tự	Ý nghĩa
.	đại diện cho 1 ký tự bất kỳ trừ ký tự xuống dòng \n
\d	ký tự chữ số tương đương [0-9]
\D	ký tự ko phải chữ số
\s	ký tự khoảng trắng tương đương [\f\n\r\t\v]
\S	ký tự không phải khoảng trắng tương đương [^\f\n\r\t\v]
\w	ký tự word (gồm chữ cái và chữ số, dấu gạch dưới _ ) tương đương [a-zA-Z_0-9]
\W	ký tự không phải ký tự word tương đương [^a-zA-Z_0-9]
^	bắt đầu 1 chuỗi hay 1 dòng
\$	kết thúc 1 chuỗi hay 1 dòng
\A	bắt đầu 1 chuỗi
\Z	Kết thúc 1 chuỗi

Ký tự	Ý nghĩa
	ký tự ngăn cách so trùng tương đương với phép or (lưu ý cái này nếu muốn kết hợp nhiều điều kiện)
[abc]	khớp với 1 ký tự nằm trong nhóm là a hay b hay c
[a-z]	so trùng với 1 ký tự nằm trong phạm vi a-z, dùng dấu – làm dấu ngăn cách
[^abc]	sẽ không so trùng với 1 ký tự nằm trong nhóm, ví dụ không so trùng với a hay b hay c
()	Xác định 1 group (biểu thức con) xem như nó là một yếu tố đơn lẻ trong pattern, ví dụ ((a(b))c) sẽ khớp với b, ab, abc.
?	khớp với đứng trước từ 0 hay 1 lần. Ví dụ A?B sẽ khớp với B hay AB.
*	khớp với đứng trước từ 0 lần trở lên. A*B khớp với B, AB, AAB,...
+	khớp với đứng trước từ 1 lần trở lên. A+B khớp với AB, AAB,...
{n}	n là số, khớp đúng với n ký tự đứng trước nó. Ví dụ A{2} khớp đúng với AA.
{n,}	khớp đúng với n ký tự trở lên đứng trước nó, A{2,} khớp với AA, AAA,...
{m,n}	khớp đúng với từ m -> n ký tự đứng trước nó, A{2,4} khớp với AA,AAA,AAAA.

## Sử dụng Pattern Modifiers

```
let str = "Mon hoc phat trien Ung dung web";
let regex = /ung/gi;
console.log(str.match(regex));
```



```
(2) ["Ung", "ung"]
1.0: "Ung"
2.1: "ung"
```

## Các modifier

- i: không quan tâm hoa thường (case-insensitive)
- g: tìm kiếm toàn bộ, không dừng lại khi đã có giá trị đầu tiên so khớp thành công.
- m: tìm kiếm trên nhiều dòng.

## Ví dụ so khớp email

```
let arr = ['matuan@fit.hcmus.edu.vn',  
          'matuan@gmail.com',  
          'matuan.yahoo.com',  
          'matuan@.com'];  
let regex = /^[a-z][a-z0-9_\.]{5,32}@[a-z0-9]{2,}(\.[a-z0-9]{2,}){1,3}$/;  
for (let email of arr) {  
  console.log(`Check: ${email} -> ` + regex.test(email));  
}
```



```
Check: matuan@fit.hcmus.edu.vn -> true  
Check: matuan@gmail.com -> true  
Check: matuan.yahoo.com -> false  
Check: matuan@.com -> false
```

## Nội dung

- *HTML Form*
- *Regular Expressions*
- **Javascript Validation**
- JQuery

## HTML Form Validation tự động

```
<form action="#" method="GET" class="formT">  
  <label for="txtEmail">Email:</label>  
  <input type="text" id="txtEmail" required />  
  <input type="submit" value="submit" />  
</form>
```



Email:

submit

! Please fill out this field.

## Constraint Validation DOM Methods

```
<label for="numAge">Age:</label>
<input type="number" id="numAge" min="10" max="50" required />
<p id="errorMsg"></p>
<input type="button" value="OK" onclick="myValid()" />
```

```
function myValid() {
  const inpObj = document.getElementById('numAge');
  if (!inpObj.checkValidity()) {
    document.getElementById('errorMsg').innerHTML = inpObj.validationMessage;
  }
}
```



Age:

Please fill out this field.

OK

Age: 450

Value must be less than or equal to 50.

OK

## Javascript Validate

```
function valid() {
  const inpAge = document.querySelector('#numAge');
  let str = inpAge.value;
  let eM = document.querySelector('#errorMsg');
  if(str.length === 0) {
    eM.innerHTML = "Can nhap tuoi";
  }
  else {
    let age = parseInt(str);
    if(age < inpAge.min) {
      eM.innerHTML = `Tuoi phai >= ${inpAge.min}`;
    }
    if(age > inpAge.max) {
      eM.innerHTML = `Tuoi phai <= ${inpAge.max}`;
    }
  }
}
```

## Nội dung

- *HTML Form*
- *Regular Expressions*
- *Javascript Validation*
- **JQuery**

## JQuery

- **JQuery** là 1 thư viện JavaScript, mục tiêu giúp cho việc sử dụng javascript dễ dàng hơn với việc “**write less, do more**”.
- Các lợi ích JQuery đem lại là:
  - DOM Traversal and Manipulation
  - Event handling
  - AJAX

## Ví dụ

```
function jqValid() {  
    if($('#numAge').val().length === 0) {  
        $('#errorMsg').html('Can nhap tuoi');  
    }else {  
        let age = parseInt($('#numAge').val());  
        if (age < $('#numAge').attr('min')) {  
            $('#errorMsg').html(`Tuoi phai >= ${$('#numAge').attr('min')}`);  
        }  
        if (age > $('#numAge').attr('max')) {  
            $('#errorMsg').html(`Tuoi phai <= ${$('#numAge').attr('max')}`);  
        }  
    }  
}  
  
$.ready(() => {  
    $('input[type="button"]').click(jqValid);  
});
```

## Nội dung

- *HTML Form*
- *Regular Expressions*
- *Javascript Validation*
- *JQuery*
- **Bài tập**



## Bài tập

- Tạo trang HTML Đăng ký và Đăng nhập cho user với các thông tin như sau:
  - **Họ tên** (yêu cầu viết hoa chữ đầu).
  - **Username** (không được có khoảng trắng, chỉ gồm các ký tự, ký số và dấu \_, không được bắt đầu bởi ký số).
  - **Email** (theo chuẩn email thông thường).
  - **Số phone** (theo chuẩn 10 số, bắt đầu bởi số 0).
  - **Ngày sinh** (theo định dạng dd/mm/yyyy và tuổi nằm trong khoảng [15, 55]).
- Yêu cầu:
  - Bố cục ngay giữa màn hình và các label, input,... phải ngay hàng thẳng lối.
  - Khi nhấn submit thì phải validate dữ liệu và thông báo chính xác lỗi cho người sử dụng.