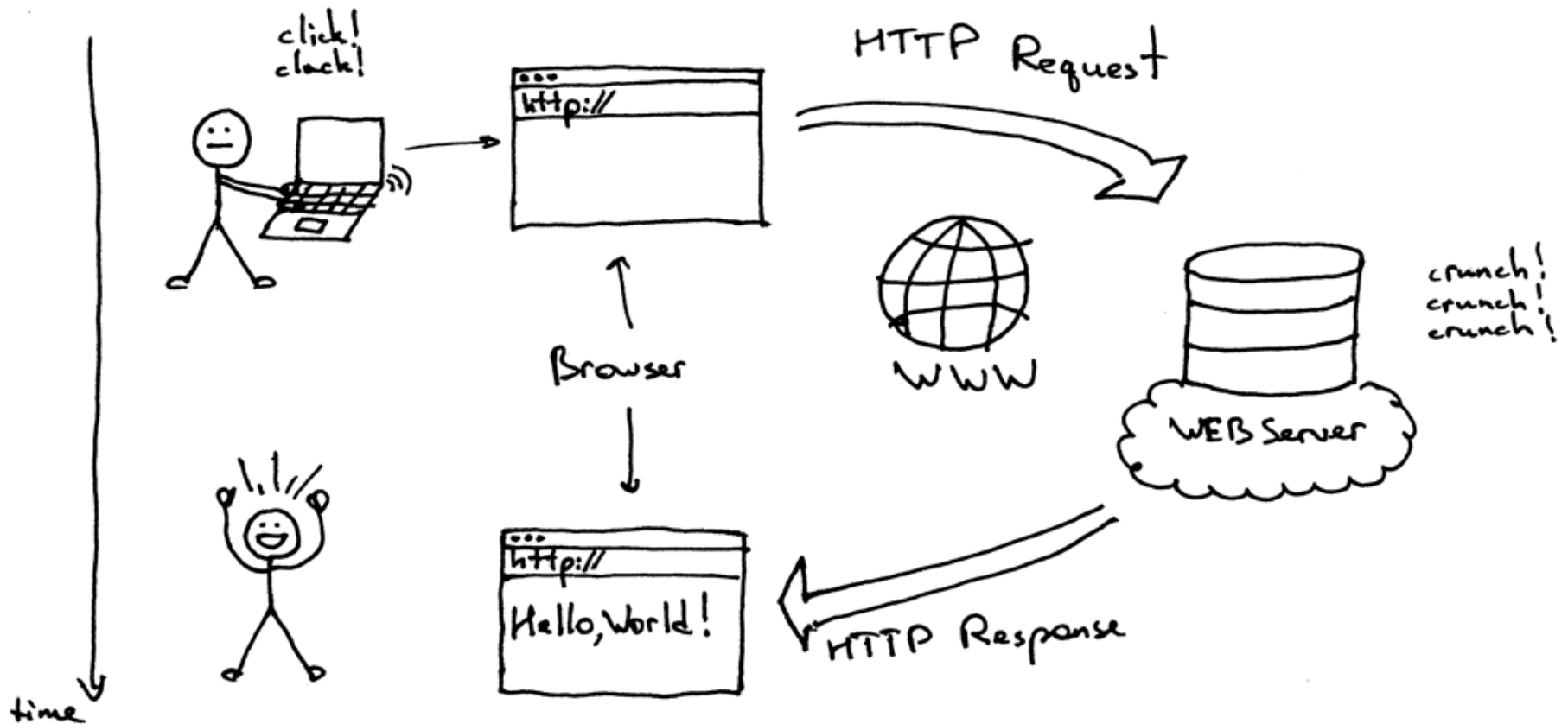


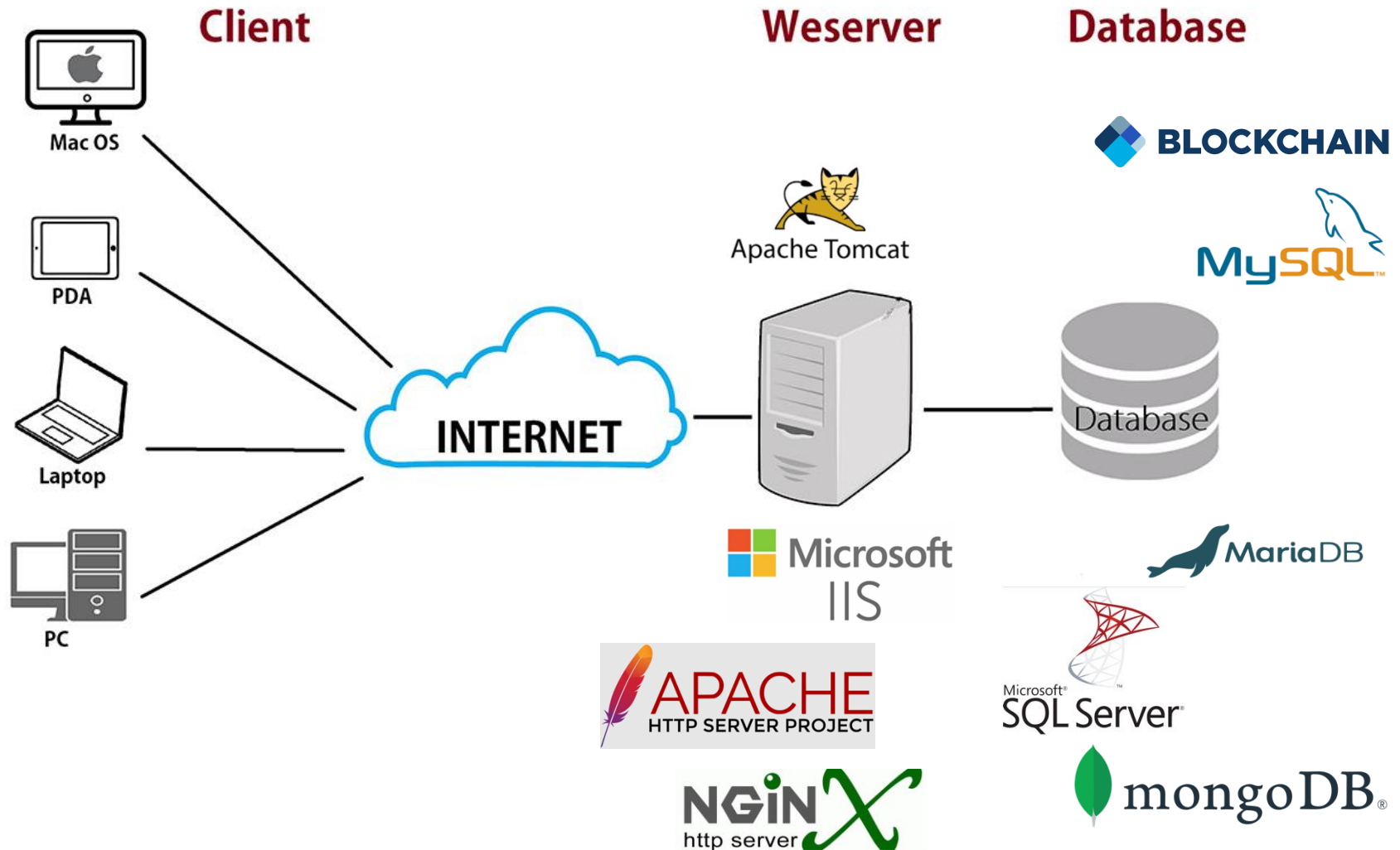


# Bảo mật web và ứng dụng

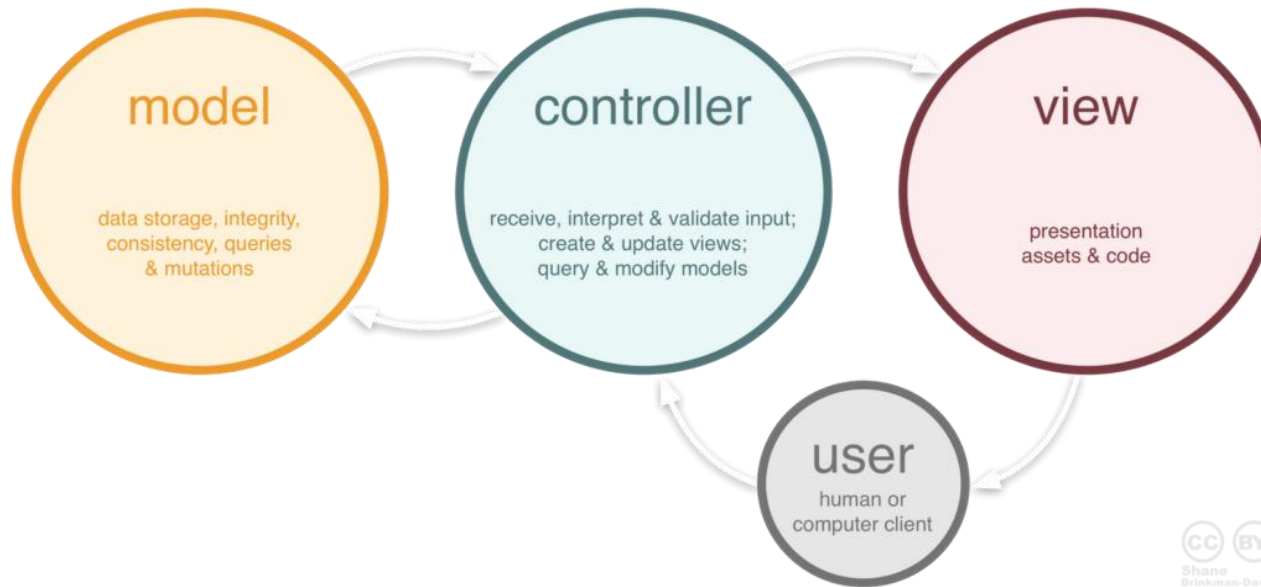
# Nguyên tắc hoạt động của web



# Nguyên tắc hoạt động của web



# Mô hình MVC



## Model

Model nghĩa là các dữ liệu cần thiết để hiển thị ở View.

## View

View đại diện cho các thành phần UI như XML, HTML. View sẽ hiển thị dữ liệu đã qua xử lý từ Controller.

## Controller

Controller có trách nhiệm xử lý các yêu cầu (request) được gửi đến. Nó sẽ xử lý các dữ liệu của người dùng qua Model và trả về kết quả ở View

# Hyper Text Markup Language



- Là standard markup language dùng **tạo ra trang web**
- Các thành phần HTML (**tag**) tạo nên **cấu trúc của trang web**
- Trình duyệt (browser) sẽ không hiển thị tag mà biểu diễn nội dung của chúng

[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)  
<https://www.w3.org/html/>

# Cấu trúc trang web



`<!DOCTYPE html>`

← Khai báo type **HTML**

`<html>`

`<head>`

`<title>Page title</title>`

`</head>`

Chứa  
**metadata**: tiêu  
đề trang, thư  
viện js, css,...

`<body>`

`<h1>This is a heading</h1>`

`<p>This is a paragraph.</p>`

`<p>This is another paragraph.</p>`

Chứa nội  
dung trang

`</body>`

`</html>`

# HTML Form



- Định nghĩa **form** để thu thập dữ liệu nhập vào từ người dùng

`<form>`

...

`</form>`

A visual representation of a login form. It consists of two input fields: the first is labeled "Username" with a person icon, and the second is labeled "Password" with a lock icon. Below these fields is a green button labeled "Login".

- Thuộc tính (**attributes**) cơ bản của form:

- Action
- Method
- Name
- Enctype

```
<form action="/action.php" method="post" enctype="multipart/form-data">  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

# HTML Form - Attributes



- **Action:** chỉ định URL sẽ gửi dữ liệu của form
- **Method:** chỉ định phương thức gửi dữ liệu
  - GET:
    - Thêm dữ liệu vào URL dạng name=value → **dữ liệu sẽ bị nhìn thấy** → **Không gửi dữ liệu nhạy cảm**
    - Độ dài URL: khoảng 3000 ký tự
    - Có thể lưu history, bookmark
  - POST:
    - Dữ liệu thêm vào Body của HTTP request
    - Không giới hạn kích thước
    - Không thể bookmark

```
/test/demo_form.php?name1=valu  
e1&name2=value2
```

```
POST /test/demo_form.php HTTP/1.1  
Host: w3schools.com  
name1=value1&name2=value2
```



# HTML Form - Attributes



- **Name:** tên của form, dùng để tham chiếu
- **Enctype**
  - **application/x-www-form-urlencoded:** *(mặc định)* tất cả các ký tự được mã hóa trước khi gửi
  - **multipart/form-data:** không mã hóa ký tự, bắt buộc khi sử dụng file upload
  - **text/plain:** khoảng trắng thành +, còn lại không mã hóa

# Thuộc tính enctype trong <form>

A screenshot of a web browser's developer tools, specifically the 'Net' panel. The top toolbar shows the 'XHR' icon highlighted with a red box. Below the toolbar, a table of network requests is shown. The first request, 'POST johnnyc...', is highlighted with a red box. The 'Response' tab is selected, showing '405 Not Allowed' status. The 'Request Headers' section is expanded, showing various headers. The 'Content-Type' header is highlighted with a red box. The 'User-Agent' header is also highlighted with a red box. The 'Request Headers From Upload Stream' section is expanded, showing 'Content-Length' and 'Content-Type' headers, both highlighted with a red box. Red arrows point from the 'Content-Type' header in the 'Request Headers' section to the 'Content-Type' header in the 'Request Headers From Upload Stream' section, and from the 'User-Agent' header in the 'Request Headers' section to the 'User-Agent' header in the 'Request Headers From Upload Stream' section.

405 Not Allowed

URL	Status	Protocol	Domain	Size	Remote IP	Timeline
POST johnnyc...	405 Not Allowed	http	johnnycode.com	663 B	108.162.199.69:80	37ms

Headers Post Response HTML Cookies

Response Headers

view source

CF-RAY c58981c7c7e0697

Connection keep-alive

Content-Type text/html

Date Wed, 30 Oct 2013 13:53:52 GMT

Server cloudflare-nginx

Transfer-Encoding chunked

Request Headers

view source

Accept text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Encoding gzip, deflate

Accept-Language en-US,en;q=0.5

Connection keep-alive

Cookie \_\_cfduid=d255dfd1a790039f449c046bff5ccfe141380633316102; \_\_utma=9499950.17:1383140986.23; \_\_utmz=9499950.1380633316.1.1.utmcsr=(direct)|utmccn=(direct)|utmc=9499950.1380633316; \_\_utmc=9499950; \_\_utmb=9499950.1.10.1383140986

Host johnnycode.com

User-Agent Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0

Request Headers From Upload Stream

Content-Length 67

Content-Type application/x-www-form-urlencoded

# HTML Form – Thẻ con



- `<input>` có các loại (type):
  - **Hidden**
  - **Text**
  - **Submit** button ↔ `<button>`
  - Radio
  - Checkbox
  - Button ↔ `<button>`
  - Reset
  - **Password**
  - Number
- `<select>` ↔ `<datalist>`
- `<textarea>`

[https://www.w3schools.com/html/html\\_form\\_elements.asp](https://www.w3schools.com/html/html_form_elements.asp)

[https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)

- Ngôn ngữ lập trình thông dịch cấp cao
- **JS** dùng lập trình **hành vi trang web**
  - HTML định nghĩa nội dung
  - CSS dùng để định dạng trang web
- Không chỉ dùng cho web
  - Nhiều chương trình server và desktop, như Node.js
  - CSDL: MongoDB, CouchDB
- Tham khảo:  
<https://www.w3schools.com/Js/default.asp>

# JavaScript có thể đặt ở đâu?



- Trong file HTML:
  - JS code đặt giữa **<script>** và **</script>**
  - JS code có thể đặt trong file **.js** riêng
    - Khai báo trong file HTML cần dùng với thuộc tính src (source) = "URL"
    - Ưu điểm:
      - Tách biệt code với HTML
      - Dùng lại code
      - Dễ đọc và bảo trì
      - Cache JS để tăng tốc độ load trang
  - Thẻ **<script>** có thể đặt tại **<head>** hoặc **<body>**

# JavaScript có thể làm gì?



- **Thay đổi nội dung của thẻ HTML**

`document.getElementById('demo').innerHTML = 'Hello JS';`

- **Thay đổi giá trị Attribute**

`document.getElementById("demo").style.fontSize = "35px";`

*or*

`document.getElementById('demo').style.fontSize = '35px';`

- **Ẩn và hiện thành phần HTML**

`document.getElementById("demo").style.display = "none";`

*or*

`document.getElementById('demo').style.display = 'none';`

`document.getElementById("demo").style.display = "block";`

# JavaScript – Ví dụ



```
<!DOCTYPE html>
<html>

<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>

<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

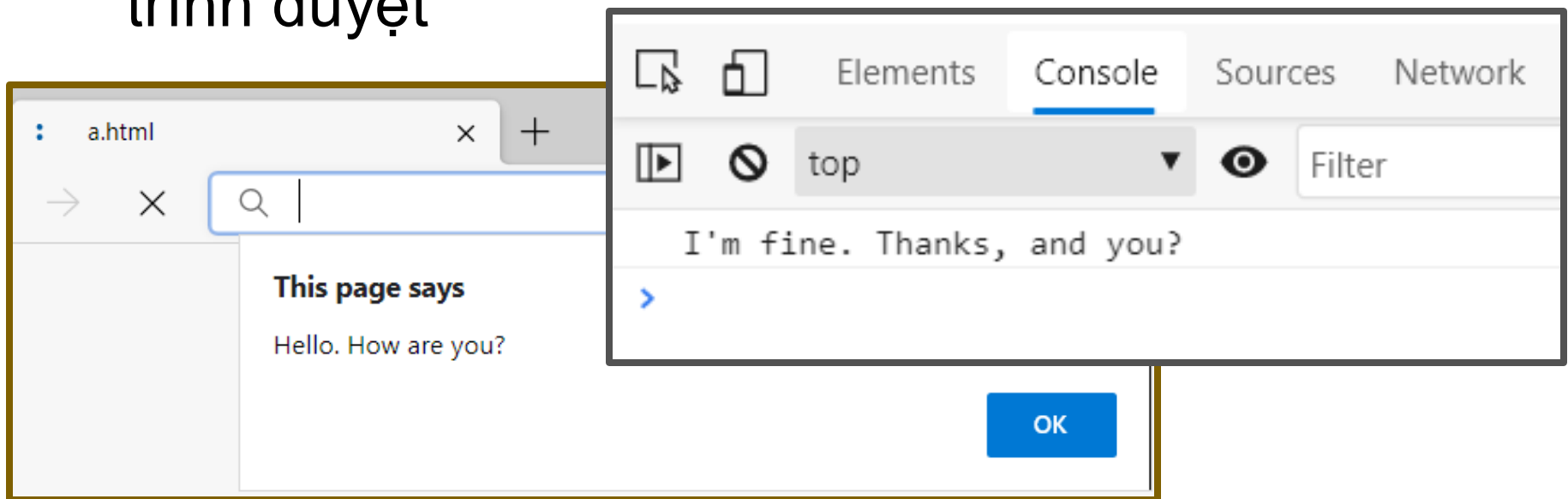
</body>
</html>
```



# JavaScript – Hiển thị dữ liệu



- `<element>.innerHTML`: hiện trong HTML element
- `document.write()`: hiện trong HTML
- **`window.alert()`**: hiện trong cửa sổ popup
- `window.console.log()`: hiện trong console của trình duyệt





# JavaScript Function



- Block code thực hiện nhiệm vụ cụ thể khi được gọi:
  - Khi có event (click vào nút)
  - Được gọi từ JS code
  - Tự động (tự gọi)

- Định nghĩa:

```
function <tên hàm>(<tham-số-1>, <tham số 2>,...) {  
    // Code  
}
```

```
<script>  
var x = sum(4, 3);  
document.getElementById("demo").innerHTML = x;  
  
function sum(a, b) {  
    return a + b;  
}  
</script>
```

# JavaScript - Object



- Định nghĩa:

```
var person = {  
  firstName: "John",  
  lastName : "Doe",  
  id       : 5566,  
  fullName : function() {  
    return this.firstName + " " + this.lastName;  
  }  
};
```

# JavaScript - Object



- Sử dụng:
  - Truy cập thuộc tính:
    - `objectName.propertyName`
    - `objectName.propertyName`
  - Truy cập phương thức:  
`objectName.methodName()`

```
// Display some data from the object:  
document.getElementById("demo").innerHTML =  
    "This is " + person.fullName;  
</script>
```

- Lưu ý: KHÔNG KHAI BÁO String, Number và Boolean như Object

# JavaScript Event



- HTML Event là thứ xảy ra với HTML elements:
  - Khi trang web load xong
  - Thay đổi giá trị input
  - Click vào button
  - ...
- **JS** cho phép **phản ứng lại event**: xử lý, xác thực input, hoạt động của người dùng và trình duyệt

# JavaScript Event



- Thêm trực tiếp vào thuộc tính event của HTML element  
`<element event='some JavaScript'>`
- Truyền hàm vào thuộc tính event của HTML element
  - `<element event="<tên_function()>">`

# JavaScript Event



Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

[https://www.w3schools.com/Js/js\\_events\\_examples.asp](https://www.w3schools.com/Js/js_events_examples.asp)

# JavaScript Form



- Kiểm tra thông tin form bằng JS
- Ví dụ: ngăn submit nếu input rỗng

```
<form name="myForm" action="/action_page.php" onsubmit="return validateForm()"
method="post">
```

```
Name: <input type="text" name="fname">
```

```
<input type="submit" value="Submit">
```

```
</form>
```

```
function validateForm() {
    var x = document.forms["myForm"]["fname"].value;
    if (x == "") {
        alert("Name must be filled out");
        return false;
    }
}
```

# JavaScript Form



- Kiểm tra tự động bằng thuộc tính **required**

```
<form action="/action_page.php" method="post">  
  <input type="text" name="fname" required>  
  <input type="submit" value="Submit">  
</form>
```

A screenshot of a web form with two input fields: "Username:" and "Password:". The "Username:" field is empty and has a yellow tooltip with an exclamation mark icon and the text "Please fill out this field." pointing to it. Below the "Password:" field is a "Submit" button. The entire form is enclosed in a black border.



# Kiểm tra dữ liệu



- Đảm bảo input: chính xác, rõ ràng và hữu dụng
- Một số loại:
  - Phải nhập giá trị
  - Ngày
  - Số
- Thực hiện:
  - Server side: sau khi gửi đến server
  - Client side: trước khi gửi

# Asynchronous JavaScript And XML



- Giới thiệu:
  - Không là ngôn ngữ lập trình
  - Là sự kết hợp: XHR, JS và HTML DOM
- Chức năng:
  - Đọc dữ liệu từ server sau khi trang đã load
  - Cập nhật trang web mà không cần reload
  - Gửi dữ liệu “ngầm” đến server

PROGRESS BAR



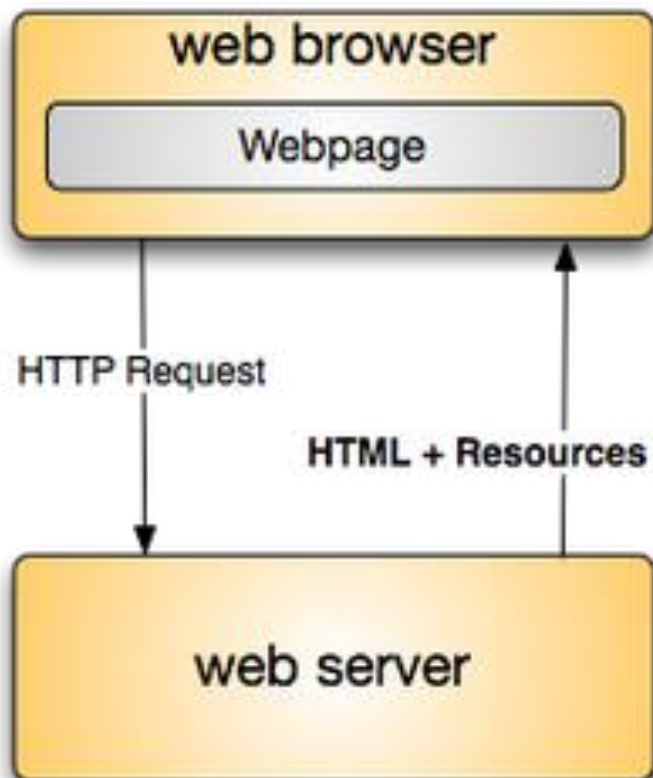
Curabitur sed ante non quam fermentum vehicula.

57%

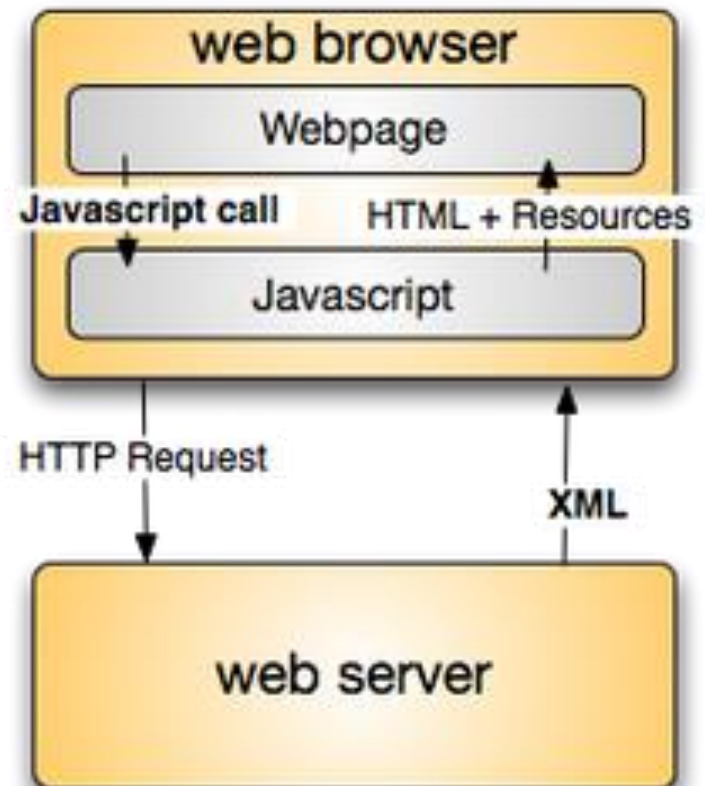
# Ajax



## Traditional web model



## AJAX web model



# Ví dụ: GET



```
<script>
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "demo_get2.asp?fname=Henry&lname=Ford", true);
    xhttp.send();
}
</script>
```

# Ví dụ: POST



```
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("POST", "demo_post2.asp", true);
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhttp.send("fname=Henry&lname=Ford");
}
</script>
```

- Thư viện JS, làm đơn giản việc lập trình JS và dễ dàng để học

**Write less, do more**

- Nhiều công ty lớn dùng:
  - Google
  - Microsoft
  - IBM
  - Netflix

## **\$(selector).action()**

- \$: định nghĩa, truy cập jQuery
- (selector): tìm HTML element
- action(): hành động muốn thực hiện
- Ví dụ:
  - \$(this).hide(): ẩn element hiện tại
  - \$("p").hide(): ẩn tất cả <p>
  - \$(".test").hide(): ẩn tất cả element có class="test"
  - \$("#test").hide(): ẩn element có id="test"

- Event Document đã load xong: ngăn thực thi code khi chưa load xong page

```
$(document).ready(function(){
```

```
    // jQuery methods go here...
```

```
});
```

```
$(document).ready(function(){  
    $("button").click(function(){  
        $(".test").hide();  
    });  
});
```

```
$(function(){
```

```
    // jQuery methods go here...
```

```
});
```



# jQuery Event



Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

- Cú pháp:

```
$("#p").click(function(){  
    // action goes here!!  
});
```

```
$("#p").on({  
    mouseenter: function(){  
        $(this).css("background-color", "lightgray");  
    },  
    mouseleave: function(){  
        $(this).css("background-color", "lightblue");  
    },  
    click: function(){  
        $(this).css("background-color", "yellow");  
    }  
});
```

# jQuery - AJAX get()



- Cú pháp: `$.get(URL, callback);`

- URL: chỉ URL muốn request
- Callback: tên hàm thực thi

- Ví dụ:

```
$("#button").click(function(){
    $.get("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

# jQuery - AJAX post()



- Cú pháp: `$.post(URL, data, callback);`
  - URL: chỉ định URL muốn request
  - Data: dữ liệu muốn gửi
  - Callback: tên hàm muốn thực thi
- Ví dụ:

```
$("#button").click(function(){
    $.post("demo_test_post.asp",
    {
        name: "Donald Duck",
        city: "Duckburg"
    },
    function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

- **Ngôn ngữ phía server** → tạo web động
- Được sử dụng rộng rãi, miễn phí, nguồn mở, hiệu quả
- Cạnh tranh với ASP
- Đủ mạnh:
  - Làm core cho hệ thống blog lớn (WP)
  - Chạy mạng xã hội lớn nhất (FB)
- Đủ dễ:
  - Cho người bắt đầu học

- Đa nền tảng: Windows, Linux, Unix, Mac OS,...
- Tương thích với hầu hết server: Apache, Nginx, IIS,...
- Hỗ trợ nhiều loại CSDL

- Đuôi file: **\*.php**

- Cú pháp:

**<?php <Mã nguồn PHP> ?>**

- Cặp thẻ **php** có thể đặt bất kỳ đâu
- Biến bắt đầu bằng: **\$**

- Lưu ý:

- Từ khóa (if, else,...), hàm, class: Không phân biệt Hoa thường
- Biến có phân biệt chữ Hoa và thường

# PHP – Ví dụ



```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $color = "red";
      echo "My car is " . $color . "<br>";
      echo "My house is " . $COLOR . "<br>";
      ECHO "My boat is " . $coLOR . "<br>";
    ?>
  </body>
</html>
```

My car is red  
My house is  
My boat is

# PHP – SuperGlobal



Truy cập bất kỳ đâu:

- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`
- `$_FILES`
- `$_ENV`
- `$_COOKIE`
- `$_SESSION`



# PHP – SuperGlobal



Element/Code	Description
<code>\$_SERVER['PHP_SELF']</code>	Returns the filename of the currently executing script
<code>\$_SERVER['GATEWAY_INTERFACE']</code>	Returns the version of the Common Gateway Interface (CGI) the server is using
<code>\$_SERVER['SERVER_ADDR']</code>	Returns the IP address of the host server
<code>\$_SERVER['SERVER_NAME']</code>	Returns the name of the host server (such as www.w3schools.com)
<code>\$_SERVER['SERVER_SOFTWARE']</code>	Returns the server identification string (such as Apache/2.2.24)
<code>\$_SERVER['SERVER_PROTOCOL']</code>	Returns the name and revision of the information protocol (such as HTTP/1.1)
<code>\$_SERVER['REQUEST_METHOD']</code>	Returns the request method used to access the page (such as POST)
<code>\$_SERVER['REQUEST_TIME']</code>	Returns the timestamp of the start of the request (such as 1377687496)
<code>\$_SERVER['QUERY_STRING']</code>	Returns the query string if the page is accessed via a query string
<code>\$_SERVER['HTTP_ACCEPT']</code>	Returns the Accept header from the current request
<code>\$_SERVER['HTTP_ACCEPT_CHARSET']</code>	Returns the Accept_Charset header from the current request (such as utf-8,ISO-8859-1)
<code>\$_SERVER['HTTP_HOST']</code>	Returns the Host header from the current request
<code>\$_SERVER['HTTP_REFERER']</code>	Returns the complete URL of the current page (not reliable because not all user-agents support it)
<code>\$_SERVER['HTTPS']</code>	Is the script queried through a secure HTTP protocol

# PHP – Form



```
<html>
  <body>
    <form action="welcome_get.php" method="get">
      Name: <input type="text" name="name">
      E-mail: <input type="text" name="email">
      <input type="submit">
    </form>
  </body>
</html>
```

## welcome\_get.php

```
<html>
  <body>
    Welcome <?php echo $_GET["name"]; ?><br>
    Your email address is: <?php echo $_GET["email"]; ?>
  </body>
</html>
```

# PHP – Xử lý dữ liệu



```
<?php
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    if (empty($_POST["name"])) {  
        $nameErr = "Name is required";  
    } else {  
        $name = test_input($_POST["name"]);  
    }  
  
    if (empty($_POST["email"])) {  
        $emailErr = "Email is required";  
    } else {  
        $email = test_input($_POST["email"]);  
    }  
}
```

```
?>
```

# PHP – Include và Require



- Thêm nội dung và file **php**
- Cú pháp:
  - **include** “tên\_file”; (phát cảnh báo và & tiếp tục thực thi)
  - **require** “tên\_file”; (phát sinh lỗi & dừng thực thi)

```
<html>
  <body>
    <div class="menu">
      <?php include 'menu.php' ;?>
    </div>

    <h1>Welcome to my home page!</h1>
    <p>Some text.</p>
  </body>
</html>
```

# PHP – Upload File



```
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        echo "File is not an image.";
        $uploadOk = 0;
    }
}
?>
```

# PHP – Upload File



```
// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
        echo "The file ". basename( $_FILES["fileToUpload"]["name"]). " has been uploaded.";
    } else {
        echo "Sorry, there was an error uploading your file.";
    }
}
```

# PHP – Thực thi chương trình bên ngoài



- **exec**

string **exec** ( string \$command [, array &\$output [, int &\$return\_var ] ] )

- **system**

string **system** ( string \$command [, int &\$return\_var ] )

- **passthru**

void **passthru** ( string \$command [, int &\$return\_var ] )

- **pcntl\_exec**

void **pcntl\_exec** ( string \$path [, array \$args [, array \$envs ] ] )

# MySQL



- Phổ biến nhất với hệ thống dùng PHP
- Nhanh, tin cậy và dễ sử dụng
- Sử dụng chuẩn SQL
- Đa nền tảng
- Được phát triển và hỗ trợ bởi Oracle



# MySQL – insert (Object-oriented)



```
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com)";

if ($conn->query($sql) !== TRUE) {
    $last_id = $conn->insert_id;
    echo "New record created successfully. Last inserted ID is: " . $last_id;
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
```

# MySQL – Select



```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
```

# MySQL – Delete



```
// sql to delete a record
```

```
$sql = "DELETE FROM MyGuests WHERE id=3";
```

```
if ($conn->query($sql) === TRUE) {
```

```
    echo "Record deleted successfully";
```

```
} else {
```

```
    echo "Error deleting record: " . $conn->error;
```

```
}
```

# MySQL – Update



```
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";  
  
if ($conn->query($sql) === TRUE) {  
    echo "Record updated successfully";  
} else {  
    echo "Error updating record: " . $conn->error;  
}
```

# MySQL – Comment



- MySQL, MSSQL, Oracle, PostgreSQL, SQLite
  - ' OR '1'='1' -- comment goes here
  - ' OR '1'='1' /\* comment goes here \*/
- MySQL
  - ' OR '1'='1' # comment goes here
- Access (using null characters)
  - ' OR '1'='1' %00
  - ' OR '1'='1' %16

# MySQL – Order by



```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

CustomerID	CustomerName
1	Alfreds Futterkiste
2	Ana Trujillo Emparedados y helados
3	Antonio Moreno Taquería
4	Around the Horn

CustomerID	CustomerName
91	Wolski
90	Wilman Kala
89	White Clover Markets
88	Wellington Importadora

# MySQL – Union

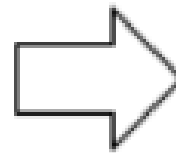


```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

id
1
2
3

UNION

id
2
3
4



id
1
2
3
4

# MySQL – TOP, LIMIT, OFFSET



```
SELECT TOP number | percent column_name(s)  
FROM table_name  
WHERE condition;
```

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
LIMIT number;
```

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
LIMIT number OFFSET offset;
```



# MySQL – Information\_Schema



Server: 127.0.0.1 » Database: information\_schema » View: SCHEMATA

Browse Structure SQL Search Export Tracking

```
SELECT * FROM `SCHEMATA`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

CATALOG_NAME	SCHEMA_NAME	DEFAULT_CHARACTER_SET_NAME	DEFAULT_COLLATION_NAME
def	aaa	utf8	utf8_unicode_ci
def	<u>banhang</u>	utf8	utf8_unicode_ci
def	baove	utf8	utf8_unicode_ci
def	blog_banhang	utf8	utf8_unicode_ci
def	cfvi	utf8mb4	utf8mb4_unicode_ci
def	champasack	utf8	utf8_unicode_ci
def	champasak	utf8	utf8_unicode_ci
def	congnghebachkhoa	utf8	utf8_unicode_ci

# MySQL – Information\_Schema



Server: 127.0.0.1 » Database: information\_schema » View: TABLES

Browse Structure SQL Search Export Tracking

```
SELECT * FROM `TABLES`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE	ENGINE	VERSION	ROW_FORMAT	TABLE_ROWS
def	aaa	aaa_commentmeta	BASE TABLE	InnoDB	10	Compact	0
def	aaa	aaa_comments	BASE TABLE	InnoDB	10	Compact	0
def	aaa	aaa_links	BASE TABLE	InnoDB	10	Compact	0
def	aaa	aaa_options	BASE TABLE	InnoDB	10	Compact	147
def	aaa	aaa_postmeta	BASE TABLE	InnoDB	10	Compact	0
def	aaa	aaa_posts	BASE TABLE	InnoDB	10	Compact	3
def	aaa	aaa_term_relationships	BASE TABLE	InnoDB	10	Compact	0
def	aaa	aaa_term_taxonomy	BASE TABLE	InnoDB	10	Compact	2
def	aaa	aaa_termmeta	BASE TABLE	InnoDB	10	Compact	0

# MySQL – Information\_Schema



Server: 127.0.0.1 » Database: information\_schema » View: COLUMNS

Browse Structure SQL Search Export Tracking

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE
def	aaa	aaa_commentmeta	meta_id	1	NULL	NO	bigint
def	aaa	aaa_commentmeta	comment_id	2	0	NO	bigint
def	aaa	aaa_commentmeta	meta_key	3	NULL	YES	varchar
def	aaa	aaa_commentmeta	meta_value	4	NULL	YES	longtext
def	aaa	aaa_comments	comment_ID	1	NULL	NO	bigint
def	aaa	aaa_comments	comment_post_ID	2	0	NO	bigint
def	aaa	aaa_comments	comment_author	3	NULL	NO	tinytext
def	aaa	aaa_comments	comment_author_email	4		NO	varchar
def	aaa	aaa_comments	comment_author_url	5		NO	varchar
def	aaa	aaa_comments	comment_author_IP	6		NO	varchar
def	aaa	aaa_comments	comment_date	7	0000-00-00 00:00:00	NO	datetime
def	aaa	aaa_comments	comment_date_gmt	8	0000-00-00 00:00:00	NO	datetime
def	aaa	aaa_comments	comment_content	9	NULL	NO	text

# Luyện tập thêm

---



- W3C
- Example exercise: <https://bit.ly/3mmT6EC>

# BT1: Thực hành 1 chút về HTML 😊



✍️ Viết một **trang html** đơn giản có **một form** điền thông tin **username/password**.

*Ví dụ*

**Simple demo form**


Username:

Password:


✍️ Đặt tên file: **BT1-MSSV1-MSSV2.html**

# BT2: Nâng cấp form đăng nhập nào :D



 **(Cá nhân)** Viết code **Javascript** kiểm tra điều kiện của **username/password** được nhập.

*Ví dụ*

A screenshot of a simple login form titled "Simple demo form". It contains two input fields: "Username:" and "Password:". Below the "Password:" field is a "Submit" button.

**Simple demo form**

Username:

Password:

1. Kiểm tra khi **submit form**
2. Điều kiện tham khảo:
  - a. Username và password không được để trống.
  - b. Username không chứa khoảng trắng, không chứa kí tự đặc biệt trừ - và \_
  - c. ...

# Bài tập



- **BT3:**
  - Sử dụng PHP/MySQL để hoàn thiện bài tập tạo form đăng nhập/ đăng ký đơn giản ở các buổi trước.

# Bảo mật web và ứng dụng



Trường ĐH CNTT TP. HCM