

## LÀM VIỆC VỚI FORM

Dữ liệu của người dùng từ trình duyệt sẽ được gửi lên máy chủ dưới dạng từng cặp **biến=gia\_trị** và có thể đi theo 2 kiểu khác nhau là GET hoặc POST.

### 1. Truyền dữ liệu thông qua phương thức GET

Dữ liệu gửi từ trình duyệt lên qua phương thức GET là phần dữ liệu được nhập trực tiếp theo sau địa chỉ URL do trình duyệt gửi lên, được phân biệt với tên file script bằng dấu hỏi chấm (?). Ví dụ địa chỉ URL sau:

<http://nhatnghe.com/?id=lichkhaigiang>

Khi đó, trình duyệt sẽ gửi theo địa chỉ trên một cặp “**biến = giá trị**”, trong đó biến có tên là **id** và giá trị là **lichkhaigiang** ( **id=lichkhaigiang** ).

Chúng ta cũng có thể đưa lên nhiều cặp **biến=gia\_trị** bằng cách phân cách chúng bởi dấu **&**:

<http://nhatnghe.com/forum/newreply.php?do=newreply&p=266060>

Với địa chỉ URL trên, chúng ta sẽ gửi lên 2 cặp **biến=gia\_trị** theo phương thức GET, đó là: **do=newreply**, **p=266060**. Khi trình duyệt gửi các thông tin này lên máy chủ, PHP sẽ tự động sinh ra một mảng có tên là **\$\_GET[]** để nắm giữ tất cả các cặp biến và giá trị đó.

Ví dụ với địa chỉ URL sau:

<http://nhatnghe.com/forum/newreply.php?do=newreply&p=266060>

Thì PHP sẽ tự động sinh ra một mảng **\$\_GET** có nội dung sau:

```
$_GET["do"] = "newreply" // tương ứng với cặp do=newreply
$_GET["p"] = 266060 // tương ứng với cặp p=266060
```

Sau đó, trong trang web của các bạn có thể sử dụng các biến này.

Ví dụ một đoạn chương trình sau để khi người dùng nhập vào biến **user=hoangth** thì cho hiển thị "Hello, my Boss", còn nếu biến **user** khác sinh thì "Hello " + giá trị của biến:

Code:

```
<?
// Hàm isset được sử dụng để kiểm tra xem một biến đã được thiết lập hay chưa
if (isset($_GET["user"]))
{
    if ($_GET["user"]=="hoangth")
    {
        echo "Hello, my boss. Good morning!";
    }
    else
    {
        echo "Hello, " . $_GET["user"] . ". Good morning!";
    }
}
```

```
}  
else  
{  
    echo "Hello, guest. How do you do?";  
}  
?>
```

Save lại, lưu với tên là welcome.php trong thư mục www. Mở trình duyệt lên, gõ vào ô Address dòng chữ sau:

<http://localhost/welcome.php?user=hoangth>

Hãy xem kết quả hiển thị trên màn hình, sau đó thay chữ hoangth bằng một cái tên gì đó xem kết quả ra sao.

## 2. Phương thức POST

Post là phần dữ liệu được gửi qua các form HTML có **method="POST"**

Để lấy các biến theo kiểu POST, PHP sẽ tự động sinh ra mảng có tên là `$_POST[]`. Mảng này có chỉ số chính là tên của các phần tử trong form (các thẻ input, select... có thuộc tính name) và giá trị là nội dung giá trị do người sử dụng nhập vào các phần tử có tên tương ứng. Chẳng hạn với mẫu biểu HTML sau:

Code:

```
<form method="POST">  
<p>  
    User Name:<input type="text" name="T1" size="20"> </p>  
<p>  
    Password:  
    <input type="password" name="T2" size="20"></p>  
<p>Sex: <Select name="sex">  
<option value=1>Male </option>  
<option value=0>Female </option>  
</select>  
</p>  
<input type="submit" value="Gui đi" name="B1">  
</form>
```

Khi người dùng nhập user name (giả sử là hoangth), password (giả sử là 123456) và chọn sex là Male, khi đó, mảng `$_POST` sẽ có các phần tử sau:

```
$_POST["T1"] = hoangth  
$_POST["T2"] = 123456  
$_POST["sex"] = 1
```

Sau khi lấy được các giá trị này rồi, các bạn có thể thoải mái sử dụng.

*Đây là ví dụ một chương trình giải phương trình bậc nhất*

Code:

```
<form method="POST">
<p style="margin-top: 0; margin-bottom: 0">
  Nhập a:<input type="text" name="a" size="20"></p>
<p style="margin-top: 0; margin-bottom: 0">Nhập b:<input type="text" name="b"
size="20"></p>
<p style="margin-top: 0; margin-bottom: 0">
  <input type="submit" value="Tính" name="B1"></p>
</form>
<?
    $a=0;
    $b=0;
    if (isset ($_POST["a"]))
    {
        $a=$_POST["a"];
    }
    if (isset ($_POST["b"]))
    {
        $b=$_POST["b"];
    }
    if ($a<0)
    {
        echo "<BR>Nghiem la: " . $b/$a;Chỗ này là -$b/$a
    }
    else
    {
        if ($b==0)
        {
            echo "<BR>Vo so nghiem";
        }
        else
        {
            echo "<BR>Vo nghiem";
        }
    }
?>
```

# EXTENSIBLE MARKUP LANGUAGE – XML

## 1. Khái niệm

Dữ liệu HTML

```
Ông Nguyễn Văn Tèo<br>
105 Bà Huyện Thanh Quan <br>
HCM
```

Dữ liệu XML

```
<address>
  <name>
    <title> Ông </title>
    <first-name> Tèo </first-name>
    <last-name> Nguyễn Văn </last-name>
  </name>
  <street> 105 Bà Huyện Thanh Quan </street>
  <city> HCM </city>
</address>
```

## 2. Ví dụ về tài liệu XML:

books1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<books>

  <book isbn="978-1594489501">
    <title> Lưới trời ai dệt </title>
    <author> Nguyễn Tường Bách </author>
    <publisher> NXB Lao động </publisher>
    <price>24.27</price>
  </book>

  <book isbn="978-1594489587">
    <title> Tôi tự học </title>
    <author> Nguyễn Duy Cần </author>
    <publisher> NXB Thanh niên </publisher>
    <price>14.97</price>
  </book>

  <book isbn="978-0545010221">
    <title> Tâm vấn phóng dao </title>
    <author> Mạc can </author>
    <publisher> NXB Văn hoá </publisher>
    <price>9.24</price>
  </book>
</books>
```

books2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<books>
  <book isbn="978-1594489501"
    title="Lưới trời ai dệt"
    author="Nguyễn Tường Bách"
    publisher="NXB Lao động"
    price="24.27">
  </book>
  <book isbn="978-1594489587"
    title="Tôi tự học"
    author="Nguyễn Duy Cần"
    publisher="NXB Thanh niên"
    price="14.97">
  </book>
  <book isbn="978-0545010221"
    title="Tâm vấn phóng dao"
    author="Mạc can"
    publisher="NXB Văn hoá"
    price="9.24">
  </book>
</books>
```

### 3. Các quy luật khi tạo file XML

- Nên có duy nhất 1 tag gốc (Root Element) bao chứa mọi tag bên trong. Tài liệu XML không có root element được gọi là một XML fragment (mảnh). Để làm cho nó well-formed, bạn cần thêm một root Element.
- Mỗi tag phải có đóng. Mở đầu và đóng phải giống hoàn toàn (phân biệt hoa thường)
- Phần tử con phải nằm hoàn toàn trong cha
- Mỗi phần tử có thể có 1 hoặc nhiều thuộc tính. Giá trị của 1 thuộc tính phải đặt trong dấu nháy.
- Nên có các Processing Instruction nằm trong cặp tags `<? và ?>`. Thường nó cho biết version của XML mà parser cần làm theo, nó cũng cho biết data trong XML dùng encoding nào, thí dụ như utf-8. Còn một Attribute nữa là standalone

### 4. Các khái niệm

- Element: Phần tử (name, title, book)
- Root Element: Phần tử gốc (books)
- Attribute: Thuộc tính ( author, publisher)
- Parser:
- Xpath: Là cách thức để định vị hay lựa chọn phần tử trong 1 tài liệu XML

Ví dụ:

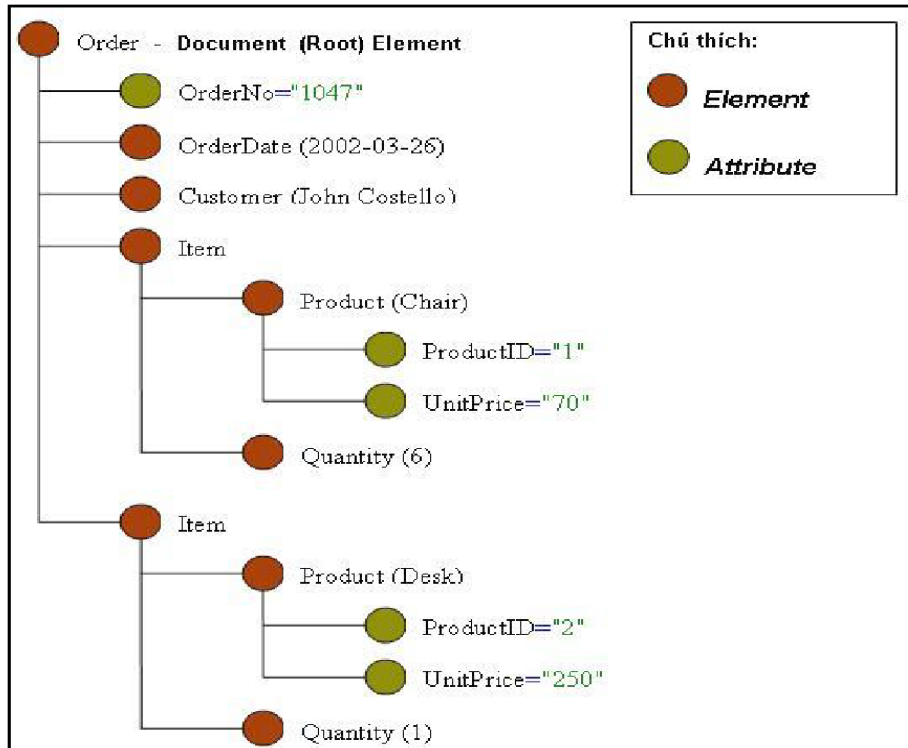
```
<order orderno="1047">
  <orderdate>2002-03-26</orderdate>
  <customer>John Costello</customer>
  <item>
    <product productid="1" unitprice="70">Chair</product>
    <quantity>6</quantity>
  </item>
  <item>
    <product productid="2" unitprice="250">Desk</product>
    <quantity>1</quantity>
  </item>
</order>
```

Xpath là `/order` để chọn phần tử order

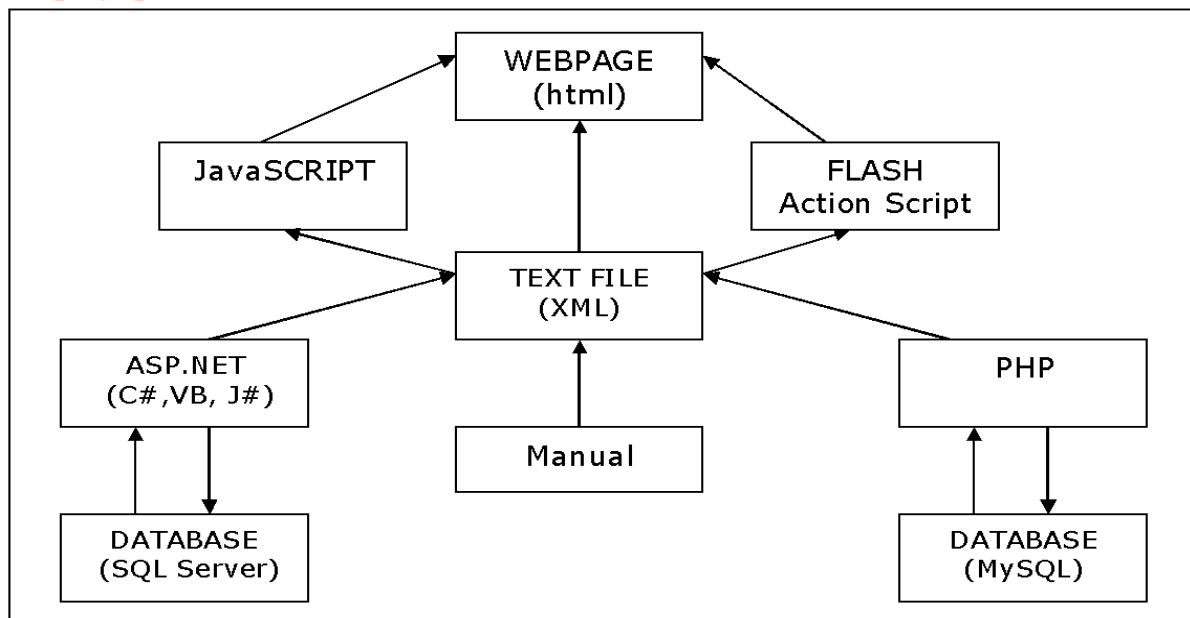
Xpath `/order/Customer` để chọn phần tử Customer

XPath `/order/@orderno` để chọn thuộc tính orderno của order

Xpath `/order//product` để chọn các node Product



## 5. Ứng dụng của XML



### a. Lưu trữ dữ liệu

← T →	idAlbum	TenBH	Nhacsi	Casi	Hinhcasi	idBaihat
<input type="checkbox"/>	1	Diễm xưa	Trịnh Công Sơn	Trần Thu Hà	thuha.jpg	1
<input type="checkbox"/>	2	Em ơi Hà Nội phố	Trần Tiến	Quang Dũng	quangdung.jpg	2
<input type="checkbox"/>	3	Anh Cho Em Mùa Xuân	Nguyễn Hiền	Đàm Vĩnh Hưng	mrdam.jpg	4
<input type="checkbox"/>	4	Bài Ca Không Quên	Phạm Minh Tuấn	Cầm Vân	camvan.jpg	8
<input type="checkbox"/> Check All / <input type="checkbox"/> Uncheck All With selected:						

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<Album>
  <Items><idAlbum>1</idAlbum>
    <TenBH>Diễm xưa</TenBH>
    <Nhacsi>Trịnh Công Sơn</Nhacsi>
    <Casi>Trần Thu Hà</Casi>
    <Hinhcasi>thuha.jpg</Hinhcasi>
    <idBaihat>1</idBaihat>
  </Items>
  <Items><idAlbum>2</idAlbum>
    <TenBH>Em ơi Hà Nội phố</TenBH>
    <Nhacsi>Trần Tiến</Nhacsi>
    <Casi>Quang Dũng</Casi>
    <Hinhcasi>quangdung.jpg</Hinhcasi>
    <idBaihat>2</idBaihat>
  </Items>
  <Items><idAlbum>3</idAlbum>
    < TenBH>Anh Cho Em Mùa Xuân</TenBH>
    <Nhacsi>Nguyễn Hiền</Nhacsi>
    <Casi>Đàm Vĩnh Hưng</Casi>
    <Hinhcasi>mrdam.jpg</Hinhcasi>
    <idBaihat>4</idBaihat>
  </Items>
  <Items><idAlbum>4</idAlbum>
    <TenBH>Bài Ca Không Quên</TenBH>
    <Nhacsi>Phạm Minh Tuấn</Nhacsi>
    <Casi>Cẩm Vân</Casi>
    <Hinhcasi>camvan.jpg</Hinhcasi>
    <idBaihat>8</idBaihat>
  </Items>
</Album>
```

- b. XML thường dùng làm database cho các file Flash
- c. XML được dùng để lưu dữ liệu trong công nghệ RSS

## 6. Đọc file XML dùng JQuery

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<script type="text/javascript" src="jquery.js"></script>
<title>Reading XML with jQuery</title>
<script>
$(document).ready(function(){
$.ajax({
  type: "GET",
  url: "websites.xml",
  dataType: "xml",
  success: function(xml) {
    $(xml).find('site').each(function(){
      var id = $(this).attr('id');
      var title = $(this).find('title').text();
      var url = $(this).find('url').text();
      var brief = $(this).find('brief').text();
```

```
var long = $(this).find('long').text();
var data = "";
data += "<div id=link_" + id + ">";
data += "<a href='" + url + "'>" + title + "</a>";
data += "<p>" + brief + "</p>";
data += "<p>" + long + "</p>";
data += "</div>";
$(data).appendTo('#wrap');
});
}
});
});
</script>
</head>
<body>
<div id="wrap"> <h2>ĐỌC FILE DÙNG JQUERY</h2> </div>
</body>
```

## 7. Đọc file XML dùng PHP với thư viện DOM

DOMXML là thư viện PHP giúp bạn làm việc với các tài liệu XML (đọc, ghi,...). Thư viện DOMXML có một số hàm thường dùng sau đây:

[DOMDocument->createAttribute\(\)](#)

[DOMDocument->createElement\(\)](#)

[DOMDocument->load\(\)](#)

[DOMDocument->createAttribute\(\)](#)

[DOMDocument->createTextNode\(\)](#)

[DOMDocument->save\(\)](#)

[DOMNode->appendChild\(\)](#)

Ví dụ : đọc file XML dùng DOMXML

```
<table border="1" width="450" cellspacing="0" cellpadding="4">
<?php
    $doc = new DOMDocument();
    $doc->load('books1.xml');
    // $doc->save('books10.xml');
    $books = $doc -> getElementsByTagName( "book" );
    foreach( $books as $b ) {
        $authors = $b -> getElementsByTagName( "author" );
        $author = $authors -> item(0) -> nodeValue ;
        $publishers = $b -> getElementsByTagName( "publisher" );
        $publisher = $publishers -> item(0) -> nodeValue;
        $titles = $b -> getElementsByTagName( "title" );
        $title = $titles -> item(0) -> nodeValue;
    }
?>
<tr>
<td> <?=$title?> </td>
<td> <?=$author?> </td>
<td> <?=$publisher?> </td>
</tr>
<?php } //foreach ?>
```



</table>

## 8. Đọc file XML dùng PHP với thư viện SimpleXML

Thư viện SimpleXML giúp bạn làm việc với các tài liệu XML rất đơn giản. Thư viện SimpleXML có một số hàm thường dùng sau đây:

[SimpleXMLElement::addAttribute](#)

[SimpleXMLElement::addChild](#)

[SimpleXMLElement::asXML](#)

[SimpleXMLElement::attributes](#)

[SimpleXMLElement::children](#)

[SimpleXMLElement::count](#)

[SimpleXMLElement::xpath](#)

**Ví dụ 1 : về đọc file XML dùng SimpleXML**

```
<table border="1" width="450" cellpadding="4">
<?php
$xml= simplexml_load_file("books1.xml"); // Load file
$xml->getName(); // Đọc thẻ root
?>
<?php foreach($xml->children() as $b) { // Lặp in hết nội dung ?>
<tr>
<td> <? echo $b -> title ?> </td>
<td> <? echo $b -> author ?> </td>
<td> <? echo $b -> publisher ?> </td>
<td> <? echo $b -> price ?> </td>
<td> <? echo $b['isbn'] ?> </td>
</tr>
<?php }?>
</table>
```

**Ví dụ 2:**

```
<table border=1 width=600 cellpadding=0 align=center>
<tr>
<th>Title</th>
<th>Author</th>
<th>Publisher</th>
<th>Price</th>
<th>ISBN</th>
</tr>
<?php
$books= new SimpleXMLElement("books1.xml",null,true); // Load from file (true)
foreach($books as $b){
echo <<<EOF
<tr>
<td> {$b -> title} </td>
<td> {$b -> author} </td>
<td> {$b -> publisher} </td>
<td> {$b -> price} </td>
<td> {$b['isbn']} </td>
</tr>
EOF;
}
```

?>  
</table>