1. Cấu trúc

<?xml version="1.0" encoding="utf-8" ?> // dòng đầu tiên chỉ thị đây là tài liệu XML

- Có cấu trúc thẻ giống HTML(bao gồm cặp thẻ và thẻ rỗng)

- Cấu trúc thẻ trong XML ta hoàn toàn có thể tự đặt theo ý nghĩa của ứng dụng.

- Cấu trúc thẻ phải hợp lý(Có thẻ mở, thẻ đóng)

- XML Phân biệt chữ hoa chữ thường nên cần đồng bộ khi sử dụng.

- Chỉ có duy nhất 1 thẻ gốc bọc toàn bộ nôi dung XML

- Giá trị thuộc tính của thẻ phải đặt trong dấu '', "";

- Không được sử dụng &

& => &amp;

< => &lt;

> => &gt;

' => &apos;

" => &quot;

- <!-- note -->

XML không phải cơ sở dữ liệu, nó là kết quả của các truy vấn

VD:

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

<info>

<name>Nguyen Huu Dat</name>

<age>27</age>

<skill>

<lang id=”001”>PHP</lang>

<lang id=”002”>Codeingiter</lang>

<lang id=”003”>Zend framework</lang>

<lang id=”004”>Laravel</lang>

</skill>

<country>Hung Yen &amp; Van Giang</country>

<!--hello-->

</info>

2. Phân đoạn dữ liệu: Biến dữ liệu từ không hợp lệ => hợp lệ. Biến dữ liệu thành dạng text thuần

<![CDATA[

// dữ liệu

]]>

3. DOM // Document Object Model: Mô hình đối tượng dữ liệu trên nền tàng javascript

dom = new ActiveXObject("Microsoft.XMLDOM") // Đối tượng trên IE

dom.async=false; // đồng bộ

dom.load('file.xml')

document.getElementsByTagName('tag\_name').item("vị\_trí-0,1,2").firstChild.nodeValue // Lấy dữ liệu của cặp thẻ

attr = document.getElementsByTagName('tag\_name').item('vi\_tri') // Tìm vị trí của thẻ chứa thuộc tính

atttr.getAttribute('attr\_tag') // Lấy giá trị của thuộc tính attr\_tag

4. PHP DOM

$dom = new DOMDocument();

$dom→load('file.xml');

$title = $dom→getElementsByTagName('tag\_name')→item('vi\_tri')→nodeValue;

$vitri = $dom→getElementsByTagName('tag\_name')→item('vi\_tri'); // Tìm vị trí của thẻ chứa thuộc tính

$gia\_tri = $vi\_tri→getAttribute('ten\_thuoc\_tinh'); // Lấy giá trị thuộc tính

Vòng lặp

$obj = $dom→getElementsByTagName('parent\_tag\_name');

foreach($obj as $tmp){

$title = $tmp→getElementsByTagName('son\_tag\_name')→item('vi\_tri')→nodeValue;

echo $title

}

5. Create file XML with DOM

$dom = new DOMDocument(“1.0”, “utf-8”); // Khởi tạo đối tượng

createElement(“name”,”value”) // Tạo 1 thẻ

createTextNode(“value”) // Tạo giấ trị trên thẻ, thuộc tính

createAttribute(“name”) // Tạo thuộc tính

createCDATASection(“Phan\_doan\_du\_lieu”) // Khởi tạo phân toạn dữ liệu

appendChild(???) // Thiết lập mối quan hệ của các thẻ

save(‘nameFile’)

Ví dụ:

$dom = new DOMDocument("1.0","utf-8");

$info = $dom->createElement("info");

$dom->appendChild($info);

$name = $dom->createElement("name", "Nguyen Huu Dat");

$info->appendChild($name);

$age = $dom->createElement("age","27");

$info->appendChild($age);

$skill = $dom->createElement("skill");

$info->appendChild($skill);

$lang = $dom->createElement("lang","PHP");

$skill->appendChild($lang);

$id1 = $dom->createAttribute("id");

$value1 = $dom->createTextNode("001");

$id1->appendChild($value1);

$lang->appendChild($id1);

$contry = $dom->createElement("country", "Hung Yen &amp; Van Giang");

$info->appendChild($contry);

$comment = $dom->createComment("Hello");

$info->appendChild($comment);

$dom->save('info.xml');

6. SimpleXML đi 1 đường thẳng từ trên xuống và bỏ qua phần tử gốc

$dom = simplexml\_load\_file('file.xml');

Ví dụ: $lang\_laravel = $dom->skill->lang[3]; // Lấy giá trị của thẻ

$attr\_4 = $dom->skill->lang[3]['id'] // Lấy giá trị của thuộc tính id

$langs = $dom->skill->lang;

foreach($langs as $lang){

echo $lang . " - " . $lang['id']

}