燕十八XML笔记

1:XMl

重要性: 非常重要,

因为计算机世界的通用语言., 是我们学习Ajax,Webservice的基础

是服务器之间交换数据常用的语言.

2:XML出现的历史背景

要替换掉html

3:为什么要替换掉html?

答: 因为html中的信息,既有显示效果,如h1,h2显示效果不同,

又有数据信息,

给数据的交换,解析,带来极大不变

4:有没有替换掉html?

答:没有,因为网页已经太多了,无法实际完成替换工作.

5:xhtml是什么?

答:就是xml没能替换成功html,

但是,把html的语法,按照xml的标准,严格了一点,形成了xhtml

1:显示/数据分离 [最重要的区别]

2:标签自描述性

解释: html中,h1,h2,p标签,都是规范定好的,

而xml的标签,是你自己命名的,因此名称就可以说明数据的性质

所以叫"自描述性"

<td>刘备</td><td>28</td>

<姓名>刘备</姓名><年龄>28</年龄>

3:xml的语法严格,错误无法运行

二:XML语法规范

1: 如何声明一个xml文件

答:由2部分组成

声明部分,声明xml的版本与编码(版本只有1.0)

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

文档节点部分

<节点名></节点名>

注意:在xml里,节点/元素/标签,一个意思.

2: 整个文档有且只有一个根节点

如<root><a></a></root>正确

<root></root><a></a>,错误

3.1:标签的写法

答:标签必须成对出现,进行闭合.

如<a></a>,而且标签区分大小写.

(在XML标签属性全小写)

或者作为单标签自闭合

3.2: 标签可以嵌套,但不能相互嵌套

<root><a></root></a>

4.1 属性的写法

属性必有值,推荐小写,属性值必用引号引起来.

5.1 注释的写法

<!-- 我是注释 -->

6.1 特殊字符要用实体来代替

' -> &apos;

" -> &quot;

> -> &gt;

< -> &lt;

7.1 如果有大段的文本,比如一篇文章,从数据库取来的.

里面有大量的',",>,<,&,那我岂不是还要用实体替换一遍?

答: CDATA节点来表示

CDATA节用来放大段文件,不用考虑实体

表单中可以使用 disable 使表单失效

<![CDATA[]]>

在XML中一切都是节点 包括空格 换行 注释也是

一般的小偷程序:

1:file\_get\_contents/curl -->页面的html

2:正则分析视频地址

XML的解释是很复杂的，需要用到dom知识

在PHP中可以通过basename()方法得出一个路径的最后部分，也就是最后一个斜线之后的部分

在PHP中字符串操作是很强大的，用到最多的是substr()

strpos() strstr():从指定位置截取到最后

str\_shuffle()把一个自付出打乱

<?php

//产生一个随机姓名 由3-6个任意数字字母组成

$str = 'abcdefghijklmnopqrstuvwxyz0123456789' ;

$shuffled = str\_shuffle ( $str );

$name = substr($shuffled,0,rand(3,6));//这里也可以使用两个随机函数实现起始位置的随机 $name = substr($str,rand(0,strlen($str)),rand(3,strlen($str)))

// 输出类似于: bfdaec

echo $name ,"<br>";

echo strlen($name);

?>

XML DOM解释

解释顺序：文档对象->节点列表（所以要以数组形式来取出节点对象）->节点对象

domdocument

在PHP中创建一个DOM文档对象 new domdocument();

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

1.把要分析的XML文档在如内存 $dom->load($path);返回一个dom树

2.$dom->getElementsByTagname('title') 返回的是domlist对象，利用他的item(index)方法可以真正取到domelement对象

在PHP中所有的XML节点都被封装成了一个对象，即使找到了唯一的标签，然而这个标签又是一个节点列表，...最后才是得到文本节点 DomText 对象

<?php

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

//载入XML文档

$dom->load('./demo2.xml');

$titles = $dom->getElementsByTagname('title');

//DOMNodeList Object( [length] => 2)

$title0 = $titles->item(0);

print\_r($title0->childNodes);//又是一个节点列表

echo $title0->childNodes->length;

print\_r($text=$title0->childNodes->item(0));//这里返回的是一个DOMText对象，不能转化成字符串，所以不可以用echo输出

echo $text->wholeText;

?>

由于对象是可以连贯操作的，所以可以写成以下形式

$dom->getElementsByTagname('title')->item(0)->childNodes->item(0)->wholeText;

真是相当的复杂！！！

解释XML的步骤：面->串->点

整个文档是一个面，通过getElementsByTagName()得到的是一个串，通过item()的出来的是一个点 而这个点下面又有很多childNodes 再通过item()得到最具体的点

总结：获取对象的顺序

DOMDocument->DOMNodeList->DOMElement->DOMNodeList->DomText

如果一个节点下只有一个子节点,比如<title>天龙八部</title>

当得到title节点时，无需再通过 $title->childNodes->item(0)->wholeText;只需要$title->nodeValue即可取出“天龙八部”

无论是哪种编程语言，都必须会有XML解释的

DOM创建XML 用程序生成一个XML文档

在XML中一切都是节点，包括看得见的和看不见的

创建方法和js中的步骤完全一样，就是先创建一个父node节点，子node节点，然后把子节点加到父节点中

比如 要创建 <book><name>红楼梦</name><price>￥100</price></book>

<?php

//header("Content-Type:text/xml");

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

$name = $dom->createElement('name');

$text = $dom->createTextNode('红楼梦');

$name->appendChild($text);

$book = $dom->createElement('book');

$book->appendChild($name);

// $cdata = $dom->createCDATASection('天龙八部<><<');

$attr = $dom->createAttribute('id');

$attr->value = 'sn001';

$book->appendChild($attr);

$bookstore = $dom->createElement('bookstore');

$bookstore->appendChild($book);

$dom->appendChild($bookstore);//刚才出错是因为没有此句，因为XML的根节点必须append在document对象上

echo $dom->saveXML();//输出XML到浏览器，以纯文本的格式

echo $dom->save('create.xml')?'ok':'fail';

?>

创建技巧：由内而外 先子后父 父.appendChild(子) 严格按照这个顺序来就不会出错

// header("Content-Type:text/xml");//特别注意：如果单纯是在浏览器输出，必须要有此句才能原样输出，如果知识性保存成XML文档，则不能有这句

接口就是一个约定 规范

比如：请求一个URL地址（路由），我返回一个XML（通过PHP的DOMDocument对象创建）文档给你，通过操作这个XML文档就可以实现数据共享交互。

可以动态查数据库，创建XML文档，也就是生成API接口，给第三方开发者使用

节点的删与改

节点的删除：与js一样，都是先找到父节点，然后通过父节点的 removeChild($child)方法删除子节点

节点的修改：先找到父节点，然后通过父节点的 replaceChild($newnode,$oldnode)方法替换原来的子节点 其实不是修改，而是创建了一个新的，然后用新的来替换了旧的

制作Rss订阅，当订阅的内容有更新时自动推送给用户而无需用户自己去找

作业：自己动手写一个 feed类 实现通过 PHP生成一个Rss订阅源

从商品表中导出信息，放在Rss中

过程分析：由于要不断创建类似 <name>朱明浩</name> 这样的节点，所以可以封装一个方法

function createEle($name,$value) {

$text = $this->dom->createTextNode($value);

$name = $this->dom->createElement($name);

$ele = $name->appendChild($text);

return $ele;

}

封装一个createItem()方法，可以方便生成多个item代码块

function createItem($arr) {

$item = $dom->createElement('item');

foreach($arr as $k=>$v) {

$item->appendChild($this->createEle($k,$v);

}

return $item;

}

封装一个 createchannel()生成channel部分

function createchannel() {

$channel = $this->dom->createElement('channel');

$channel->appendChild($this->createEle('title',$this->title);

$channel->appendChild($this->createEle('link',$this->link);

$channel->appendChild($this->createEle('description',$this->description);

$this->rss->appendChild($channel);

}

protected function addItem($list) {

foreach($list as $goods) {

$this->rss->appendChild($this->createItem($goods));

}

}

simplexml快速解释文档

$simplexml = simplexml\_load\_file('');

$simplexml->count();计算子元素个数

$simplexml->children();得出所有的子元素

$s->getName();//得出当前元素的名称

添加子元素

思路也是：先找到父元素，通过addChild();

$sim->addChild('movie','电影');//要求两个参数都是字符串

xml 与 数组 的相互转换

由于对象与数组在simplexml对象中是交替出现的，所以必须要用递归才能事项将所有对象转换成数组

到底是数组好 还是 XML 好呢？

答：数组有利于操作，XML有利于网络传输，信息交互

<?php

$sim = simplexml\_load\_file('demo2.xml');

// print\_r((array)$sim);//要想实现全部转换，必须要用到递归算法 //

//使用递归把simplexml对象转成数组，思路：循环遍历，如果是对象就转为数组

//

function toArray($sim){

$arr = (array) $sim;//把最外层对象转成数组

foreach ($arr as $key => $value) {//对于数组中的每一个元素，无论他的值是simplexml对象或者是数组都转数组，因为数组下面藏着的是对象，如果没有跳过数组则无法进入到对象内部

# code...

if($value instanceof simplexmlelement || is\_array($value

)) {

$arr[$key] = toArray($value);

}

}

return $arr;

}

print\_r($sim\_arr = toArray($sim));

// echo toArray($sim)['book'][0]['title'];//此句为何会报错,因为PHP中不能直接操作函数的返回值 也就是不能在括号后进行操作

echo $sim\_arr['book'][0]['title'];

?>

arraytoXML案例 从数据库中取出数组，转成XML ，也就是做了一个接口

思路：

递归与if判断是密不可分的，递归中的各个if语句部分都会执行到

bool is\_scalar ( mixed $var )

标量变量是指那些包含了 integer 、 float 、 string 或 boolean 的变量，而 array 、 object 和 resource 则不是标量。

$arr = array(

'name'=>'张三',

'age'=>20,

'job'=>array(

'title'=>'经理',

'name'=>'小明',

'salary'=>3000

)

);

function array2xml($arr,$node=null){

if($node == null) {

$sim = new simplexmlelement('<?xml version="1.0" encoding="UTF-8"?><root></root>');

}else{

$sim=$node;

}

foreach ($arr as $key => $value) {

# code...

if(is\_array($value)){

array2xml($value,$sim->addchild($key));//创建一个父节点，并回头重新调用array2xml(),此时$sim就是新创建的节点，$value是一个数组，当$value再次来到这个循环时，传给$arr,自此，$value就已经是一个标量的 is\_scalar($value),走的是else部分！！！

}else{

$sim->addchild($key,$value);//把子节点挂在$sim中，这个$sim是会在根节点和自定义节点之间变得，主要是看是否传$node参数

}

}

return $sim->saveXML();

}

header('content-type:text/xml');

echo array2xml($arr);

注意：XML中的标签名不能是以数字开头的纯数字

所以还要判断 is\_numeric()

实战：完成对XML的查询，要求在线词典功能

isset($\_POST['name']);判断有没有赋值，没赋值则返回false

如果 var 存在并且值不是 NULL 则返回 TRUE ，否则返回 FALSE 。

<?php

//要求，根据表单提交过来的单词，查询dict.xml,查询出该单词的解释，例句

//

$world = isset($\_POST['world'])?$\_POST['world']:'';

if(empty($world)){

exit('你想查什么？');

}

//接下来创建dom对象，循环查询，判断name节点值是否等于$world

//

$dom = new domdocument('1.0','utf-8');

// print\_r($dom);

$dom->load('dict.xml');

$dom->getElementsByTagname('dict');

$namelist = $dom->getElementsByTagname('name');//取出所有name节点集合

foreach ($namelist as $key => $value) {

# code...

if ($value->nodeValue==$world) {

# code...

echo $world.'<br>';

echo '意思：'.$value->nextSibling->nodeValue.'<br>';

echo '例句：'.str\_replace('/r/n', '<br>', $value->nextSibling->nextSibling->nodeValue).'<br>';

break;//查到了，必须退出！！！

}

}

?>

实战案例：把dict.xml导入到mysql数据库中

思路：先转换成数组,问题是，转换出来的是多维数组，而关系型数据表就是一张二维表，所以要想办法转成一个二维数组形式才可以

XPath：必有一条路从根节点到指定节点，这个路径就是从根节点如何走过来

先用DOMDocument对象载入XML文档，然后new XPath($dom)对象,利用这个对象的query方法

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

$dom->load('');

$xpath = new XPath($dom);

$sql = './root/book/title';

$xpath->query($sql);

调试：出现T\_String 说明是字符串出错，很可能是单双引号的配对问题，也有可能是变量没有加$，于是理解成了字符串

$sql = "/bookstore/book[title='诗经']/price";

$sql = '/bookstore/book[price>40]/title';

$sql = '/bookstore/book[2]/title';//注意 下标从1开始，调试技巧，出现DOMList错误，只需加上 item(0)即可

$rs = $xpath->query($sql);//返回的是一个节点列表,也就是一个串，相当于getElemtsByTagName()

echo $rs->item(0)->nodeValue;

/a/b,这说明,a,b就是父子关系,而如果用/a//b,这样说明a只是b的祖先就行,忽略了层次

// 不分层次,查出所有的title

/\*

$sql = '//title';

foreach($xpath->query($sql) as $v) {

echo $v->nodeValue,'<br />';

}

\*/

xpath是根据DOM标准来查询,

html也是DOM,

也能查,岂只是xml

$html = new DOMDocument('1.0','utf-8');

$html->loadhtmlfile('dict.html');

$xpath = new DOMXPATH($html);

$sql = '/html/body/h2';

echo $xpath->query($sql)->item(0)->nodeValue,'<br />';

// 查询id="abc"的div节点

$sql = '//div[@id="abc"]';

echo $xpath->query($sql)->item(0)->nodeValue;

// 接下来用xpath来查询词典，优点是：不用对整个XML文档进行循环遍历，大大提高查询速度，无论查询那个单词，速度几乎一样

$xpath = new DOMXpath($xml);

// 查询/dict下的word,且name=$word的节点下面的/name节点

$sql = '/dict/word[name="' . $word . '"]/name';

//echo $sql;

$names = $xpath->query($sql);

if($names->length == 0) {

echo 'sorry';

exit;

}

// 查到了

$name = $names->item(0);

echo $word,'<br />';

echo '意思:',$name->nextSibling->nodeValue,'<br />';

echo '例句:',$name->nextSibling->nextSibling->nodeValue,'<br />';

xml对象转为数组并放回原来数组的位置中 也就是 $arr[$k] = toArray($sim);