# 概论

1、因为 DOM 要构造整个文档树，要花费大量的内存和处理器时间。因此，性能问题决定了使用 DOM 很难处理大型文档。

创建XML

# 使用方式

## 1.创建XML

详细见手册

//创建XML字符串或XML文档

$dom = new DomDocument('1.0');

/\*\*

\* 创建节点（有2个参数）

\*/

$root = $dom->createElement('books');

/\*\*

\*添加节点

\*/

$books = $dom->appendChild($root);

// set the formatOutput attribute of domDocument to true

$dom->formatOutput = true;

/\*\*

\* 保存，save XML as string or file

\*/

$test1 = $dom->saveXML(); // put string in test1

$dom -> save('test1.xml'); // save as file

## 2.将document转换为SimpleXML文件

$dom = new domDocument;

$dom->loadXML($path);

if (!$dom) {

echo 'Error while parsing the document';

exit;

}

/\*\*

\* 将document文件转化为simpleXML对象

\*/

$s = simplexml\_import\_dom($dom);

// Great American Novel

echo $s->book[0]->title;

## 3.使用xpath

$path = APPLICATION\_PATH."/../public/xml/firstStairMenu.xml";

$dom = new DOMDocument();

$dom -> load($path);

$xpath = new DOMXPath($dom);

//获取所有节点数,用于获得一共有多少节点

$query = "/root/\*";

$entries = $xpath -> query($query);

//查找某一节点

XmlNode node = xmlDoc.SelectSingleNode("root/goods[@name='"+name+"']");

查询语句一般为 //\*

用于选择查到的节点

Item(@parent)是必有的对于多个查询结果其参数是对应的不同的选项，对于单个选项其值为0即可

Item()

选值是从0开始

//查找节点属性的值 使用item标识第几个 使用nodeValue等取值

attributes->item(2)

所有节点

/root/mark[1]是从1开始，不是0开始

nodeValue选中节点的值

nodeName选中节点的名字