XML:eXtensible Markup Language

特性1：可扩展

表现：可以自定义标签

特性2：标记语言

表现：还是标签

XML是独立于软件和硬件的信息传输工具？

宗旨：传输数据 而非显示数据

有自我描述性

作用：数据存储->

　　　配置文件：qq 迅雷等中的配置文件

　　　小型数据库

数据传输->Ajax：异步js和xml xml充当了数据交换的中介

（比如PHP和JS传输 需要一种东西 PHP和

JS都能读懂-> XML）

　　　web服务：比如航班查询 能查询的那个地方 并没有

　　　 数据库 而是传递了xml文档。

SVG：可伸缩的矢量图形？

XML结构：

树形结构

树形结构的特点：

有且仅有一个根元素

树中有3种节点：元素节点 属性节点 文本节点

开头必须有一句话：

<?xml version=‘1.0’ encoding=’utf-8’?>

XML语法：跟HTML一样 但是 元素区分大小写

字符实体： < &lt; > &gt; ‘ &apos; “ &quot; & &amp;

<![CDATA[数据]]> 这里的数据 就可以直接写 <>’”&

树中有3种节点：元素节点 属性节点 文本节点

元素的表现形式：仅包含元素的元素、仅包含文本的元素、包含元素和文本混合的元素

标签属性所代表的信息也可以被改成用子元素的形式来描述

元素和属性有何区别？

属性：表示的信息和内容本身没有太大关系时使用属性来描述这种信息

DTD验证：Document Type Definition，文档类型定义（Schema结构定义 趋势）

作用：验证xml文档是否符合规范（这个规范也是自定义的）

和css一样 内联外联

DTD定义元素 <!ELEMENT 元素名称 内容模型>

内容模型的常见类别：

EMPTY 元素为空 <!ELEMENT 元素名称 EMPTY>

ANY 有任何内容 <!ELEMENT 元素名称 ANY>

（） 包含子元素 <!ELEMENT 元素名称 (子元素1)>

, 子元素顺序 <!ELEMENT 元素名称 (子元素1，子元素2…)>

| 或 <!ELEMENT 元素名称 (子元素1|子元素2…)>

+ 至少出现1次 <!ELEMENT 元素名称 (子元素名称+)>

\* 0次或任意多次<!ELEMENT 元素名称 (子元素名称\*)>

　　　? 0次或1次 <!ELEMENT 元素名称 (子元素名称?)>

　　　(#PCDATA) 只含文本 <!ELEMENT 元素名称 (#PCDATA)>

在xml中，元素的表现形式有哪几种？

\* 仅包含文本的元素

\* 仅包含元素的元素

\* 包含元素和文本混合的元素

\* 空元素

\* 任意元素

（1）.仅包含文本的元素

<!ELEMENT 元素名 (#PCDATA)>

(2).仅包含元素的元素

这种元素的内容就是元素 不能出现文本

两种结构：

　　　顺序结构：元素必须按照声明次序显示

　　　<!ELEMENT note(元素1，元素2…)

　　　但是 机动性不够 太笨重 所以提供了三个数量词

　　　\* 任意个 +至少一个 ? 0或1个

　　　选择结构：多个元素之中选择其中一个 只能是一个！

　　　|表示或关系

　　　但是如果有了数量词 <!ELEMENT note(to|from)\*>

　　　就可以有多个

（3）.包含元素和文本的混合元素

<!ELEMENT body(#PCDATA|元素名)\*>

注：选择结构中 必须以#PCDATA开头

必须使用\*

DTD定义属性 <!ATTLIST 元素名称 属性名称 属性类型 取值方式>

属性类型：CDATA 值为字符的数据

(en1|en2..)选择列表中的一个值

ID 值为唯一的id

取值方式：值 属性的默认值

#REQUIRED 必须有这个属性

#IMPLIED 不一定非得有这个属性

#FIXED value 属性是固定的

注：条件最宽松的是#IMPLIED 因为这个选项表示 它不一定必须得有

　　选择类型中 en1 en2的位置不需要加’’ 但是默认值必须加’’

　　<!ATTLIST user id ID #REQUIRED>

　　user标签 id属性 属性类型：值唯一 必须有这个属性

　　<!ATTLIST user gender (man|woman)’man’>

　　user标签 gender属性 属性类型：可选值 默认值：man

　　<!ATTLIST name lang CDATA #IMPLIED>

　　name 标签 lang属性 属性类型：字符串 不一定必须有

　　<!ATTLIST user nationality CDATA #FIXED’usa’>

　　user 标签 nationality属性 属性类型：字符串 固定值 只能是usa

　　实体定义：

针对文本节点 当需要重复的使用某个固定的字符串时使用

定义实体：

在DTD中 分为两种实体：一般实体 参数实体

一般实体：在DTD中定义 在XML中引用

范围：内部/外部DTD皆可

定义格式：<!ENTITY 实体名称 值>

引用格式：&实体名称;

XML中引用外部DTD：<!DOCTYPE note SYSTEM ‘xxx.dtd’>

浏览器中会报错 但是没有问题 因为XML不是用来显示的 而是用来传输数据的

要是读取xml文档 肯定有这个实体内容

参数实体：在DTD中定义 在DTD中使用

范围：外部DTD

定义格式:<!ENTITY % 实体名称 值>

引用格式： %实体名称

引用实体

DOM方式操作XML

PHP操作XML的方式：DOM / SimpleXML 等12种 主要这两种

DOM方式 任何语言中都一样 通用

SimpleXML PHP专用操作XML的方式

DOM操作XML的步骤

* 将xml文档转成树模型
* 对树模型进行具体的curd操作
* 将操作的结果转成xml文档

需要用到的对象

DOMDocument

DOMElement

DOMNode

DOMNodeList

注意：对XML文档的操作 并不是在这个文档上直接以字符串方式完成操作

而是将这个文档读取到内存中 在内存中 对这个文档进行转换 操作的

是转换后的结果（复制一份再操作 还是原文件操作？）

需要操作两件事：节点、节点和节点之间的关系