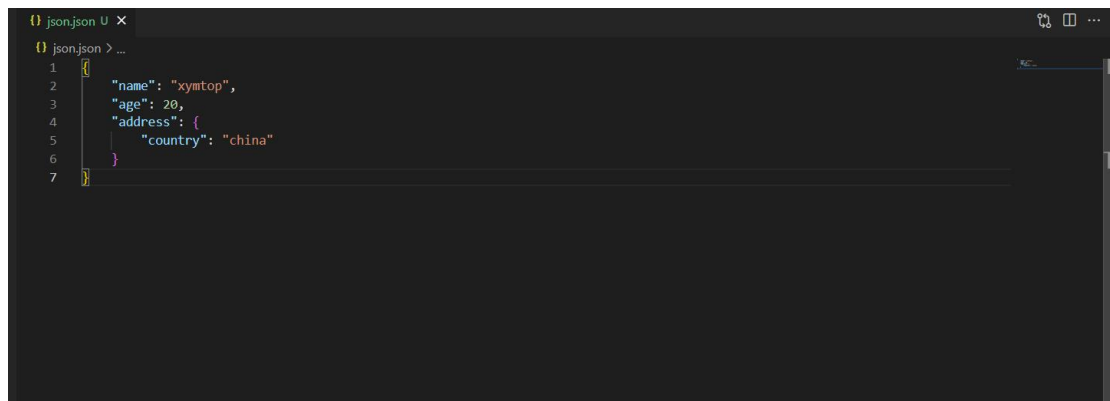


JSON 与 XML

JSON (JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (European Computer Manufacturers Association, 欧洲计算机协会制定的 js 规范) 的一个子集, 采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。易于人阅读和编写, 同时也易于机器解析和生成, 并有效地提升网络传输效率。

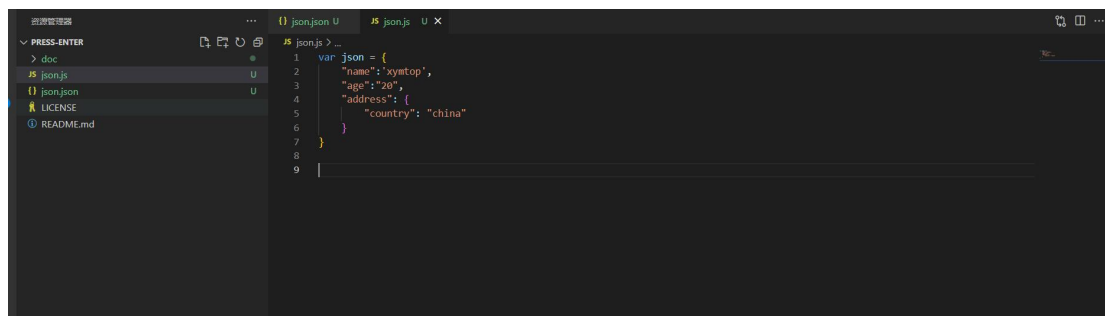
JSON (JavaScript Object Notation) 是一种轻量级的数据交换格式。易于人阅读和编写, 可以在多种语言之间进行数据交换。同时也易于机器解析和生成。它基于 JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999 的一个子集。



```
1 {  
2   "name": "xymtop",  
3   "age": 20,  
4   "address": {  
5     "country": "china"  
6   }  
7 }
```

JSON 是 Douglas Crockford 在 2001 年开始推广使用的数据格式, 在 2005 年-2006 年正式成为主流的数据格式, 雅虎和谷歌就在那时候开始广泛地使用 JSON 格式。任何支持的类型都可以通过 JSON 来表示, 例如字符串、数字、对象、数组等。但是对象和数组是比较特殊且常用的两种类型。

对象在 JS 中是使用花括号包裹 {} 起来的内容, 数据结构为 {key1: value1, key2: value2, ...} 的键值对结构。在面向对象的语言中, key 为对象的属性, value 为对应的值。键名可以使用整数和字符串来表示。值的类型可以是任意类型。

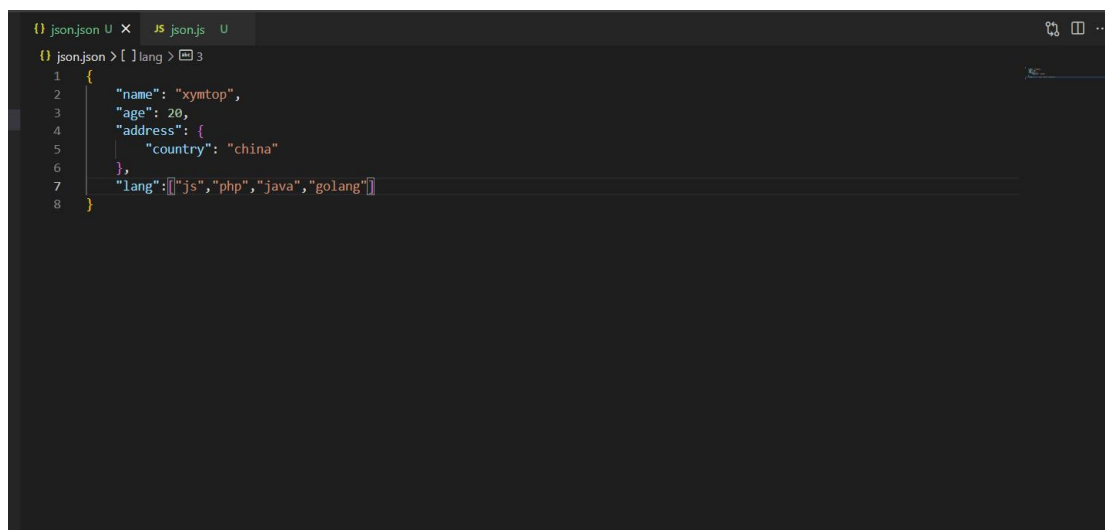


数组：数组在 JS 中是方括号 [] 包裹起来的内容，数据结构为 ["java", "javascript", "vb", ...] 的索引结构。在 JS 中，数组是一种比较特殊的数据类型，它也可以像对象那样使用键值对，但还是索引使用得多。同样，值的类型可以是任意类型。

简单地说，JSON 可以将 JavaScript 对象中表示的一组数据转换为字符串，然后就可以在网络或者程序之间轻松地传递这个字符串，并在需要的时候将它还原为各编程语言所支持的数据格式，例如在 PHP 中，可以将 JSON 还原为数组或者一个基本对象。在用到 AJAX 时，如果需要用到数组传值，这时就需要用 JSON 将数组转化为字符串。

对象是一个无序的“‘名称/值’对”集合。一个对象以 { 左括号开始，} 右括号结束。每个“名称”后跟一个: 冒号；“‘名称/值’对”之间使用逗号分隔。

和普通的 JS 数组一样，JSON 表示数组的方式也是使用方括号 []。



在处理 JSON 格式的数据时，没有需要遵守的预定义的约束。所以，在同样的数据结构中，可以改变表示数据的方式，也可以使用不同方式表示同一事物。如前面所说，除了对象和数组，你也可以简单地使用字符串或者数字等来存储简单的数据，但这样并没有多大意义。

JSON 和 XML 的可读性可谓不相上下，一边是简易的语法，一边是规范的标签形式，很难分出胜负。

XML 天生有很好的扩展性，JSON 当然也有，没有什么是 XML 可以扩展而 JSON 却不能扩展的。不过 JSON 在 Javascript 主场作战，可以存储 Javascript 复合对象，有着 xml 不可比拟的优势。

```
xml.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <country>
3    <name>中国</name>
4    <province>
5      <name>新疆维吾尔自治区</name>
6      <cities>
7        <school>新疆大学</school>
8        <school>新疆医科大学</school>
9      </cities>
10   </province>
11 </country>
```

XML 有丰富的编码工具，比如 Dom4j、Dom、SAX 等，JSON 也有提供的工具。无工具的情况下，相信熟练的开发人员一样能很快的写出想要的 xml 文档和 JSON 字符串，不过，xml 文档要多很多结构上的字符。

```
import java.net.URL;

import org.dom4j.Document;
import org.dom4j.DocumentException;
import org.dom4j.io.SAXReader;

public class Foo {

    public Document parse(URL url) throws DocumentException {
        SAXReader reader = new SAXReader();
        Document document = reader.read(url);
        return document;
    }
}
```

XML 的解析方式有两种：

一是通过文档模型解析，也就是通过父标签索引出一组标记。例如：
`xmlData.getElementsByTagName("tagName")`，但是这样是要在预先知道文档结构的情况下使用，无法进行通用的封装。

另外一种方法是遍历节点（document 以及 childNodes）。这个可以通过递归来实现，不过解析出来的数据仍旧是形式各异，往往也不能满足预先的要求。凡是这样可扩展的结构数据解析起来一定都很困难。

JSON 也同样如此。如果预先知道 JSON 结构的情况下，使用 JSON 进行数据传递简直是太美妙了，可以写出很实用美观可读性强的代码。如果你是纯粹的前台开发人员，一定会非常喜欢 JSON。但是如果你是一个应用开发人员，就不是那么喜欢了，毕竟 xml 才是真正的结构化标记语言，用于进行数据传递。

而如果不知道 JSON 的结构而去解析 JSON 的话，那简直是噩梦。费时费力不说，代码也会变得冗余拖沓，得到的结果也不尽人意。但是这样也不影响众多前台开发人员选择 JSON。因为 json.js 中的 toJSONString() 就可以看到 JSON 的字符串结构。当然对于不是经常使用这个字符串的人，这样做仍旧是噩梦。常用 JSON 的人看到这个字符串之后，就对 JSON 的结构很明了了，就更容易的操作 JSON。以上是在 Javascript 中仅对于数据传递的 xml 与 JSON 的解析。在 Javascript 地盘内，JSON 毕竟是主场作战，其优势当然要远远优越于 xml。如果 JSON 中存储 Javascript 复合对象，而且不知道其结构的话，我相信很多程序员也一样是哭着解析 JSON 的。

除了上述之外，JSON 和 XML 还有另外一个很大的区别在于有效数据率。JSON 作为数据包格式传输的时候具有更高的效率，这是因为 JSON 不像 XML 那样需要有严格的闭合标签，这就让有效数据量与总数据包比大大提升，从而减少同等数据流量的情况下，网络的传输压力。

可扩展标记语言 (Extensible Markup Language, XML)，标准通用标记语言的子集，可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。XML 是标准通用标记语言 可扩展性良好, 内容与形式分离, 遵循严格的语法要求, 保值性良好等优点。

在电子计算机中，标记指计算机所能理解的信息符号，通过此种标记，计算机之间可以处理包含各种的信息比如文章等。它可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。它非常适合万维网传输，提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据。是 Internet 环境中跨平台的、依赖于内容的技术，也是当今处理分布式结构信息的有效工具。早在 1998 年，W3C 就发布了 XML1.0 规范，使用它来简化 Internet 的文档信息传输。

可扩展标记语言与 Access, Oracle 和 SQL Server 等数据库不同，数据库提供了更强有力的数据存储和分析能力，例如：数据索引、排序、查找、相关一致性等，可扩展标记语言仅仅是存储数据。事实上它与其他数据表现形式最大的不同是：可扩展标记语言极其简单，这是一个看上去有点琐细的优点，但正是这点使它与众不同。

XML 的简单易于在任何应用程序中读/写数据，这使 XML 很快成为数据交换语言（此类语言主要包括 XML, JSON 等，常用于接口调用，配置文件，数据存储等场景），虽然不同的应用软件也支持其他的数据交换格式，但不久之后它们都将支持 XML，那就意味着程序可以更容易的与 Windows、Mac OS、Linux 以及其他平

台下产生的信息结合，然后可以很容易加载 XML 数据到程序中并分析它，并以 XML 格式输出结果。

XML 有两个先驱：SGML 和 HTML，这两个语言都是非常成功的标记语言，但是都有一些与生俱来的缺陷。XML 正是为了解决它们的不足而诞生的。

早在 Web 未发明之前，SGML(Standard Generalized Markup Language，标准通用标记语言)就已存在，正如它的名称所言，SGML 是国际上定义电子文件结构和内容描述的标准。SGML 具有非常复杂的文档结构，主要用于大量高度结构化数据的访问和其他各种工业领域，在分类和索引数据中非常有用。

虽然 SGML 的功能很强大，但是它不适用于 Web 数据描述，而且 SGML 软件的价格非常昂贵；另外，SGML 十分庞大，既不容易学，又不容易使用，在计算机上实现也十分困难：不仅如此，几个主要的浏览器厂商都明确拒绝支持 SGML，这无疑是 SGML 在网上传播遇到的最大障碍。鉴于这些因素，Web 的发明者——欧洲核子物理研究中心的研究人员，根据当时（1989 年）的计算机技术，发明并推出了 HTML。

1989 年，HTML 诞生，它抛弃了 SGML 复杂庞大的缺点，继承了 SGML 的很多优点。HTML 最大的特点是简单性和跨平台性。

HTML 是一种界面技术，它只使用了 SGML 中很少的一部分标记，例如 HTML 4.0 中只定义了 70 余种标记。为了便于在计算机上实现，HTML 规定的标记是固定的，即 HTML 语法是不可扩展的。HTML 这种固定的语法使它易学易用，在计算机上开发 HTML 的浏览器也十分容易。正是由于 HTML 的简单性，使得基于 HTML 的 Web 应用得到了极大的发展。

随着 Web 应用的不断发展，HTML 的局限性也越来越明显地显现了出来，如 HTML 无法描述数据、可读性差、搜索时间长等。人们又把目光转向 SGML，再次改造 SGML 使之适应现在的网络需求。随着先辈的努力，1998 年 2 月 10 日，W3C(World Wide Web Consortium，万维网联盟)公布 XML 1.0 标准，XML 诞生了。

XML 最初的设计目的是为了 EDI(Electronic Data Interchange，电子数据交换)，确切地说是为 EDI 提供一个标准数据格式。

当前的一些网站内容建设者们已经开始开发各种各样的 XML 扩展，比如数学标记语言 MathML、化学标记语言 CML 等。此外，一些著名的 IT 公司，如 Oracle、IBM 以及微软等都积极地投入人力与财力研发 XML 相关软件与服务支持，这无疑确定了 XML 在 IT 产业的重要地位。

XML 具有以下特点：

(1) XML 可以从 HTML 中分离数据。即能够在 HTML 文件之外将数据存储在 XML 文档中，这样可以使开发者集中精力使用 HTML 做好数据的显示和布局，并确保数据改动时不会导致 HTML 文件也需要改动，从而方便维护页面。XML 也能够将数据以“数据岛”的形式存储在 HTML 页面中，开发者依然可以把精力集中到使用 HTML 格式化和显示数据上。

(2) XML 可用于交换数据。基于 XML 可以在不兼容的系统之间交换数据，计算机系统和数据库系统所存储的数据有多种形式，对于开发者来说，最耗时间的工作就是在遍布网络的系统之间交换数据。把数据转换为 XML 格式存储将大大减少交换数据时的复杂性，还可以使这些数据能被不同的程序读取。

(3) XML 可应用于 B2B 中。例如在网络中交换金融信息，目前 XML 正成为遍布网络的商业系统之间交换信息所使用的主要语言，许多与 B2B 有关的完全基于 XML 的应用程序正在开发中。

(4) 利用 XML 可以共享数据。XML 数据以纯文本格式存储，这使得 XML 更易读、更便于记录、更便于调试，使不同系统、不同程序之间的数据共享变得更加简单。

(5) XML 可以充分利用数据。XML 是与软件、硬件和应用程序无关的，数据可以被更多的用户、设备所利用，而不仅仅限于基于 HTML 标准的浏览器。其他客户端和应用程序可以把 XML 文档作为数据源来处理，就像操作数据库一样，XML 的数据可以被各种各样的“阅读器”处理。

(6) XML 可以用于创建新的语言。比如，WAP 和 WML 语言都是由 XML 发展来的。WML(Wireless Markup Language, 无线标记语言)是用于标识运行于手持设备上（比如手机）的 Internet 程序的工具，它就采用了 XML 的标准。

总之，XML 使用一个简单而又灵活的标准格式，为基于 Web 的应用提供了一个描述数据和交换数据的有效手段。但是，XML 并非是用来取代 HTML 的。HTML 着重如何描述将文件显示在浏览器中，而 XML 与 SGML 相近，它着重描述如何将数据以结构化方式表示。

XML 文件格式是纯文本格式，在许多方面类似于 HTML，XML 由 XML 元素组成，每个 XML 元素包括一个开始标记(<)，一个结束标记(>)以及两个标记之间的内容，例如，可以将 XML 元素标记为价格、订单编号或名称。标记是对文档存储格式和逻辑结构的描述。在形式上，标记中可能包括注释、引用、字符数据段、起始标记、结束标记、空元素、文档类型声明(DTD)和序言。

具体规则如下：

1、必须有声明语句

XML 声明是 XML 文档的第一句，其格式如下：

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

2、注意大小写

在 XML 文档中，大小写是有区别的。“A”和“a”是不同的标记。注意在写元素时，前后标记的大小写要保持一致。最好养成一种习惯，或者全部大写，或者全部小写，或者大写第一个字母，这样可以减少因为大小写不匹配而产生的文档错误。

3、XML 文档有且只有一个根元素

良好格式的 XML 文档必须有一个根元素，就是紧接着声明后面建立的第一个元素，其他元素都是这个根元素的子元素，根元素完全包括文档中其他所有的元素。根元素的起始标记要放在所有其他元素的起始标记之前；根元素的结束标记要放在所有其他元素的结束标记之后。

4、属性值使用引号

在 HTML 代码里面，属性值可以加引号，也可以不加。但是 XML 规定，所有属性值必须加引号（可以是单引号，也可以是双引号，建议使用双引号），否则将被视为错误。

5、所有的标记必须有相应的结束标记

在 HTML 中，标记可以不成对出现，而在 XML 中，所有标记必须成对出现，有一个开始标记，就必须有一个结束标记，否则将被视为错误。

6、所有的空标记也必须被关闭

空标记是指标记对之间没有内容的标记，比如“”等标记。在 XML 中，规定所有的标记必须有结束标记。

(1)可扩展性方面：HTML 不允许用户自行定义他们自己的标识或属性，而在 XML 中，用户能够根据需要自行定义新的标识及属性名，以便更好地从语义上修饰数据。

(2)结构性方面：HTML 不支持深层的结构描述，XML 的文件结构嵌套可以复杂到任意程度，能表示面向对象的等级层次。

(3)可校验性方面：HTML 没有提供规范文件以支持应用软件对 HTML 文件进行结构校验，而 XML 文件可以包括一个语法描述，使应用程序可以对此文件进行结构校验。

虽然 XML 标准本身简单，但与 XML 相关的标准却种类繁多，W3C 制定的相关标准就有二十多个，采用 XML 制定的重要的电子商务标准就有十多个。这一方面说明 XML 确实是一种非常实用的结构化通用标记语言，并且已经得到广泛应用；另一方面，这又为了解这些标准带来一定的困难，除了标准种类繁多外，标准之间通常还互相引用，特别是应用标准，它们的制定不仅仅使用的是 XML 标准本身，还常常用到了其他很多标准。XML 标准的体系与 SGML 标准的体系非常相似，XML 相关标准也可分为元语言标准、基础标准、应用标准三个层次。

描述的是用来描述标准的元语言。在 XML 标准体系中就是 XML 标准，是整个体系的核心，其他 XML 相关标准都是用它制定的或为其服务的。

这一层次的标准是为 XML 的进一步实用化制定的标准，规定了采用 XML 制定标准时的一些公用特征、方法或规则。如 XML Schema 描述了更加严格地定义 XML 文档的方法，以便可以更自动化处理 XML 文档；XMLNamespace 用于保证 XML DTD 中名字的一致性，以便不同的 DTD 中的名字在需要时可以合并到一个文档中；XSL 是描述 XML 文档样式与转换的一种语言；XLink 用来描述 XML 文档中的超链接；XPointer 描述了定位到 XML 文档结构内部的方法；DOM 定义了与平台和语言无关的接口，以便程序和脚本动态访问和修改文档内容、结构及样式等。

XML 已开始被广泛接受，大量的应用标准，特别是针对因特网的应用标准，纷纷采用 XML 进行制定。有人甚至认为，XML 标准是因特网时代的 ASCII 标准。在这因特网时代，几乎所有的行业领域都与因特网有关。而它们一旦与因特网发生关系，都必然要有其行业标准，而这些标准往往采用 XML 来制定。