1. **HTML4.01 + CSS2 + JavaScript**

备注:缺点是数据缺少结构无法在不同系统中重用.

DHTML不是一个标准或协议,它通指HTML+CSS+JS.

AJAX不是一种新语言,它只是一种编程模式,通指异步JS+XML.

ECMAScript = Javascript.

---------------------------.html---------------------------

*<html>*

*<head>*

*<title> Example</title>*

*</head>*

*<body>Content</body>*

*</html>*

---------------------------.css---------------------------

*a:active{color: #7b0099;}*

*img {border:none;}*

---------------------------.js---------------------------

*function Test() {*

*alert(“Example!”);*

*}*

1. **XML:** 可扩展的语言,主要用于推动结构化数据在不同系统中的共享交换.

---------------------------.xml---------------------------

*<?xml version="1.0"?>*

*<note>*

*<to>Cyndi</to>*

*<from>Snow</from>*

*<heading>Reminder</heading>*

*<body>Don't forget me this weekend!</body>*

*</note>*

1. **DTD or XML Schema:** 用于定义XML文档的结构,常用于检验XML文档的合法性或对不同XML数据文件之间格式的统一定义, 区别在于Schema同时还支持数据类型和命名空间的定义,并且Schema是用XML格式编写的,它是做为DTD的继承者产生的.

---------------------------.dtd---------------------------

*<!ELEMENT note (to, from, heading, body)>*

*<!ELEMENT to (#PCDATA)>*

*<!ELEMENT from (#PCDATA)>*

*<!ELEMENT heading (#PCDATA)>*

*<!ELEMENT body (#PCDATA)>*

---------------------------.xsd---------------------------

*<?xml version="1.0"?>*

*<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">*

*<xs:element name="note">*

*<xs:complexType>*

*<xs:sequence>*

*<xs:element name="to" type="xs:string"/>*

*<xs:element name="from" type="xs:string"/>*

*<xs:element name="heading" type="xs:string"/>*

*<xs:element name="body" type="xs:string"/>*

*</xs:sequence>*

*</xs:complexType>*

*</xs:element>*

*</xs:schema>*

1. **XHTML1.0:** 从HTML4.01开始形成一个XHTML的分支,将HTML改用严格的XML进行定义,同时配有相应的DTD文件用于格式验证和解析.

备注: HTML5加入了语义标签的定义,如article, header, footer等.

---------------------------.xhtml---------------------------

*<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"*

*"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">*

*<html xmlns="http://www.w3.org/1999/xhtml” xml:lang="en" lang="en">*

*<head>*

*<title>Example</title>*

*</head>*

*<body>*

*<img id="validation-icon" src="http://www.w3.org/Icons/valid-xhtml10" />*

*<br />*

*</body>*

*</html>*

1. **XLink + XPointer + XForm:** 分别定义了XML中的链接和表单,做为XHTML的补充

---------------------------.xml (xlink, xpointer)---------------------------

*<?xml version="1.0"?>*

*<homepages xmlns:xlink="http://www.w3.org/1999/xlink">*

*<homepage xlink:type="simple" xlink:href="http://www.w3school.com.cn">Visit W3School</homepage>*

*<homepage xlink:type="simple” xlink:href="http://dog.com/dogbreeds.xml#xpointer(id('Rottweiler'))">Visit W3School</homepage>*

*</homepages>*

---------------------------.xml (xform)---------------------------

*<xforms>*

*<model>*

*<instance>*

*<person>*

*<fname/>*

*<lname/>*

*</person>*

*</instance>*

*<submission id="form1" action="submit.asp" method="get"/>*

*</model>*

*<input ref="fname"><label>First Name</label></input>*

*<input ref="lname"><label>Last Name</label></input>*

*<submit submission="form1"><label>Submit</label></submit>*

*</xforms>*

1. **XPath2.0:** XPath使用路径形式来查询 XML 文档中的节点.

备注: XPath功能越强大XSLT和XQuery就越简单.

------------------------------------------------------

*/bookstore/book*

1. **XQuery1.0:** XQuery是用于查询XML数据的脚本语言, 类同于SQL, 同时是XPath的一种宿主语言.

备注: XQueryX将 XQuery 语法表达的查询转变成用 XML 表达的查询, XQuery方便人类阅读, XML方便机器阅读.

--------------------------.xql----------------------------

*for $x in doc("books.xml")/bookstore/book*

*return if ($x/@category="CHILDREN")*

*then <child>{data($x/title)}</child>*

*else <adult>{data($x/title)}</adult>*

1. **XSL1.1(XSLT2.0 + XSL-FO):** XSL是一个概念模型,用于对XML进行样式表达,包含XML result的构造(transform)及格式化样式(formatting),其中transform由XSLT完成,Formatting由XSL-FO完成. XSLT主要用于查询XML数据,相较XQuery而言更侧重于发布查询后的数据,常用于将XML数据二次转化成另一种XML结果集的,它可以被嵌入到非XML的代码中,如HTML,使其成为带样式的XSL文件.同时也是XPath的一种宿主语言.

--------------------------.xsl----------------------------

*<?xml version="1.0" encoding="ISO-8859-1"?>*

*<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">*

*<xsl:template match="header">*

*<fo:block font-size="14pt" font-family="verdana" color="red" space-before="5mm" space-after="5mm">*

*<xsl:apply-templates/>*

*</fo:block>*

*</xsl:template>*

*</xsl:stylesheet>*

Web Service: 服务提供者通过UDDI向注册中心发布服务,服务请求者通过WSDL从注册中心获取服务描述,然后服务请求者和服务提供者通过SOAP进行通讯.

1. **SOAP:** SOAP是一种基于 XML 的通讯协议,约定了通过http在应用程序之间进行信息交换的规则.

备注:web service是基于组件的平台,是面向服务的网络应用,不仅面向人(web网站是一种向人提供的服务)同时面向机器,它主要由SOAP + WSDL + UDDI构成.(以天气预报为例)

SOA是面向服务的架构缩写,web service是实现SOA的一种技术标准,或者理解为SOA的一种具体实现.

------------------------SOAP请求------------------------------

*POST /InStock HTTP/1.1*

*Host: www.example.org*

*Content-Type: application/soap+xml; charset=utf-8*

*Content-Length: nnn*

*<?xml version="1.0"?>*

*<soap:Envelope*

*xmlns:soap="http://www.w3.org/2001/12/soap-envelope"*

*soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">*

*<soap:Body xmlns:m="http://www.example.org/stock">*

*<m:GetStockPrice>*

*<m:StockName>IBM</m:StockName>*

*</m:GetStockPrice>*

*</soap:Body>*

*</soap:Envelope>*

------------------------SOAP响应------------------------------

*HTTP/1.1 200 OK*

*Content-Type: application/soap+xml; charset=utf-8*

*Content-Length: nnn*

*<?xml version="1.0"?>*

*<soap:Envelope*

*xmlns:soap="http://www.w3.org/2001/12/soap-envelope"*

*soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">*

*<soap:Body xmlns:m="http://www.example.org/stock">*

*<m:GetStockPriceResponse>*

*<m:Price>34.5</m:Price>*

*</m:GetStockPriceResponse>*

*</soap:Body>*

*</soap:Envelope>*

1. **WSDL:** WSDL是一种用于WEB服务的描述语言,它规定了web服务的位置和服务所提供的操作,这种语言也可理解为RDF的一种继承.

---------------------------.wsdl---------------------------

*<wsdl:types>*

*<s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">*

*<s:element name="HelloWorld">*

*<s:complexType>*

*<s:sequence>*

*<s:element minOccurs="0" maxOccurs="1" name="param" type="s:string" />*

*</s:sequence>*

*</s:complexType>*

*</s:element>*

*<s:element name="HelloWorldResponse">*

*<s:complexType>*

*<s:sequence>*

*<s:element minOccurs="1" maxOccurs="1" name="HelloWorldResult" type="s:int" />*

*</s:sequence>*

*</s:complexType>*

*</s:element>*

*</s:schema>*

*</wsdl:types>*

*<wsdl:message name="HelloWorldSoapIn">*

*<wsdl:part name="parameters" element="tns:HelloWorld" />*

*</wsdl:message>*

*<wsdl:message name="HelloWorldSoapOut">*

*<wsdl:part name="parameters" element="tns:HelloWorldResponse" />*

*</wsdl:message>*

1. **UDDI:** 定义了发现和发布web service的方式. 它本身也相当于是一种web服务,由UDDI注册中心提供.
2. **DOM3 + SAX:** DOM是文档对象模型,分为Core DOM/XML DOM/HTML DOM,定义了访问和操作结构化文档的标准,通过JavaScript可以实现和HTML页面的交互.SAX是基于事件的分析器,如expat,但它不是W3C定义的.
3. **WAP:** WAP是一套让移动设备显示WWW信息的协议,包含WML+WMLScript.

--------------------------.wml----------------------------

*<?xml version="1.0"?>*

*<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"*

*"http://www.wapforum.org/DTD/wml\_1.1.xml">*

*<wml>*

*<card id="HTML" title="HTML Tutorial">*

*<p>Sample Text1</p>*

*</card>*

*<card id="XML" title="XML Tutorial">*

*<p>Sample Text2</p>*

*</card>*

*</wml>*

1. **SMIL2.1:** SMIL可用于组合视频、音频、文本以及图形进行幻灯片模式播放,是一种多媒体呈现的描述, 常见于彩信的格式定义.

--------------------------.smil----------------------------

*<smil>*

*<body>*

*<seq repeatCount="indefinite">*

*<img src="image1.jpg" dur="3s" />*

*<img src="image2.jpg" dur="3s" />*

*</seq>*

*</body>*

*</smil>*

1. **SVG:** SVG是一种用XML格式存储的图片,具备矢量特征,它拥有对应的编辑器.

--------------------------.svg(这是一个圆)----------------------------

*<?xml version="1.0" standalone="no"?>*

*<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">*

*<svg width="100%" height="100%" version="1.1" xmlns="http://www.w3.org/2000/svg">*

*<circle cx="100" cy="50" r="40" stroke="black" stroke-width="2" fill="red"/>*

*</svg>*

语义网(也称web3.0)的七层架构:统一的资源定位器(URI) -> 句法XML -> 语法RDF -> 词汇RDFS/OWL -> 由1~4层形成的逻辑推理 -> 验证 -> 信任

1. **RDF(URI):** RDF是描述网络资源的标准(它可以用XML描述,也可以用其它语法描述)，通常资源、属性和属性值的三元组合可形成一个描述(主谓宾)*.* 比如网页的标题、作者、修改日期、内容以及版权信息,也可以将RDF理解为元数据语言。RDF供机器阅读,它只定义了一些框架性的东西,具体的语义由使用者二次定义.

备注:XML不是RDF的唯一表达语言,RDF也可以使用框图来描述. URI全称统一资源标识符,其中主谓必须为URI.RDF通过使用URI标识实现了分布.

--------------------------.rdf----------------------------

*<?xml version="1.0"?>*

*<rdf:RDF*

*xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"*

*xmlns:si="http://www.recshop.fake/siteinfo#">*

*<rdf:Description rdf:about="http://www.w3schools.com/RDF">*

*<si:author>Jan Egil Refsnes</si:author>*

*<si:homepage>http://www.w3schools.com</si:homepage>*

*</rdf:Description>*

*</rdf:RDF>*

1. **RDF Schema + OWL:** 它是一种词汇描述语言.RDF与RDF Schema的关系类似于XML与XML Schema的关系.RDF在不同组织之间交换时可以通过定义RDF Schema来约定共同的语义,但它与XML Schema最大区别在于它不是约束XML数据的格式,而是描述RDF数据中词汇的含义,它也可看做是RDF的扩展,用于定义应用程序专业的类和属性的方法,OWL是RDFS的一种扩展,它具备更强大的描述能力,如对称性等,通过RDFS/OWL实现了语义网的推理.

--------------------------.rdfs----------------------------

*<?xml version="1.0"?>*

*<rdf:RDF*

*xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#"*

*xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"*

*xml:base= "http://www.animals.fake/animals#">*

*<rdfs:Class rdf:ID="animal" />*

*<rdfs:Class rdf:ID="horse">*

*<rdfs:subClassOf rdf:resource="#animal"/>*

*</rdfs:Class>*

*</rdf:RDF>*

1. **SPARQL:** 用于查询RDF数据的标准语言.

------------------------------------------------------

*PREFIX foaf: <http://xmlns.com/foaf/0.1/>*

*SELECT ?url*

*FROM <bloggers.rdf>*

*WHERE {*

*?contributor foaf:name "Jon Foobar" .*

*?contributor foaf:weblog ?url .*

*}*

1. **RSS2:** RSS是一种使用XML格式来定义一组数据展示的协议,它被约定能自动更新数据.RSS1.0可理解为RDF的一种应用.

备注:ATOM是另一种feed协议.

------------------------------------------------------

*<?xml version="1.0" encoding="ISO-8859-1" ?>*

*<rss version="2.0">*

*<channel>*

*<title>W3Schools Home Page</title>*

*<link>http://www.w3schools.com</link>*

*<description>Free web building tutorials</description>*

*<item>*

*<title>RSS Tutorial</title>*

*<link>http://www.w3schools.com/rss</link>*

*<description>New RSS tutorial on W3Schools</description>*

*</item>*

*<item>*

*<title>XML Tutorial</title>*

*<link>http://www.w3schools.com/xml</link>*

*<description>New XML tutorial on W3Schools</description>*

*</item>*

*</channel>*

*</rss>*