**WEB开发技术对比**

**JAVA**

* **简介：**

Java是由Sun Microsystems公司于1995年5月推出的Java程序设计语言（以下简称Java语言）和Java平台的总称。用Java实现的HotJava浏览器（支持Java applet）显示了Java的魅力：跨平台、动态的Web、Internet计算。从此，Java被广泛接受并推动了Web的迅速发展，常用的浏览器现在均支持Java applet。另一方面，Java技术也不断更新。

* **IDE：**

Eclipse、Myeclipse、Jbuilder2008、Jdeveloper、Netbeans等，很多！

* **版本历史：**

1995年5月23日，Java语言诞生

1996年1月，第一个JDK-JDK1.0诞生  
1996年4月，10个最主要的操作系统供应商申明将在其产品中嵌入JAVA技术  
1996年9月，约8.3万个网页应用了JAVA技术来制作

1997年2月18日，JDK1.1发布  
1997年4月2日，JavaOne会议召开，参与者逾一万人，创当时全球同类会议规模之纪录

1998年12月8日，JAVA2企业平台J2EE发布

1999年6月，SUN公司发布Java的三个版本：标准版（J2SE）、企业版（J2EE）和微型版（J2ME）

2000年5月8日，JDK1.3发布  
2000年5月29日，JDK1.4发布

2001年6月5日，NOKIA宣布，到2003年将出售1亿部支持Java的手机  
2001年9月24日，J2EE1.3发布

2002年2月26日，J2SE1.4发布，自此Java的计算能力有了大幅提升

2004年9月30日18:00PM，J2SE1.5发布，成为Java语言发展史上的又一里程碑。为了表示该版本的重要性，J2SE1.5更名为Java SE 5.0  
2005年6月，JavaOne大会召开，SUN公司公开Java SE 6。此时，Java的各种版本已经更名，以取消其中的数字“2”：J2EE更名为Java EE，J2SE更名为Java SE，J2ME更名为Java ME

2006年12月，SUN公司发布JRE6.0  
目前JDK7.0已经在研发中，已经有测试版在 https://jdk7.dev.java.net/ 可以下载使用

* **应用方面：**

可以从事JSP网站开发、Java编程、Java游戏开发、Java桌面程序设计，以及其他与Java语言编程相关的工作。

**PHP**

* **简介**

PHP，一个嵌套的缩写名称，是英文超级文本预处理语言（PHP:Hypertext Preprocessor）的缩写。PHP 是一种 HTML 内嵌式的语言，是一种在服务器端执行的嵌入HTML文档的脚本语言，语言的风格有类似于C语言，现在被很多的网站编程人员广泛的运用。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创新的语法。它可以比 CGI 或者 Perl 更快速的执行动态网页。

* **IDE**

Eclipse、Komodo、PHP 设计器、PhpED、PHPEdit、Zend Studio等。

* **版本历史**

在1997年，任职于 Technion IIT 公司的两个以色列程序设计师：Zeev Suraski 和 Andi Gutmans，重写了 PHP 的剖析器，成为 PHP 3 的基础，而 PHP 也在这个时候改称为PHP: Hypertext Preprocessor.[5]。经过几个月测试，开发团队在1997年11月释出了 PHP/FI 2，随后就开始 PHP 3 的开放测试，最后在1998年6月正式释出 PHP 3。Zeev Suraski 和 Andi Gutmans 在 PHP 3 释出后开始改写 PHP 的核心，这个在1999年释出的剖析器称为 Zend Engine[7]，他们也在以色列的 Ramat Gan 成立了 Zend Technologies 来管理 PHP 的开发。

在2000年5月22日，以Zend Engine 1.0为基础的PHP 4正式释出，2004年7月13日则释出了PHP 5，PHP 5则使用了第二代的Zend Engine[5]。PHP包含了许多新特色，像是强化的面向对象功能、引入PDO（PHP Data Objects，一个存取数据库的延伸函数库）、以及许多效能上的增强。目前PHP 4已经不会继续更新，以鼓励用户转移到PHP 5。

2008年PHP 5成为了PHP唯一的有在开发的PHP版本。将来的PHP 5.3将会加入Late static binding和一些其他的功能强化。PHP 6 的开发也正在进行中，主要的改进有移除register\_globals、magic quotes 和 Safe mode的功能。

* **应用方面**

**PHP常见的模板引擎**  
Smarty  
Smarty的特点是将模板编译成PHP脚本，然后执行这些脚本。很快，非常灵活。  
Heyes Template Class  
一个非常容易使用，但功能强大并且快速的模板引擎，它帮助你把页面布局和设计从代码中分离。  
FastTemplate  
一个简单的变量插值模板类，它分析你的模板，把变量的值从HTML代码中分离处理。  
ShellPage  
一个简单易用的类，可以让你的整个网站布局基于模板文件，修改模板就能改变整个站点。  
STP Simple Template Parser  
一个简单、轻量级并且易于使用的模板分析类。它可以从多个模板中组装一个页面，把结果页面输出到浏览器或者文件系统。  
OO Template Class  
一个你可以用在自己程序中的面向兑现的模板类。  
SimpleTemplate  
一个可以创建和结构化网站的模板引擎。它可以解析和编译模板。  
bTemplate  
短小但是快速的模板类，允许你把PHP逻辑代码从HTML修饰代码中分离。  
Savant  
一个强大且轻量级的PEAR兼容模板系统。它是非编译型的，使用PHP语言本身做为它的模板语言。  
ETS – easy template system  
可以使用完全相同数据重组模板的模板系统。  
EasyTemplatePHP  
适用于你的站点的一个简单但是强大的模板系统。  
vlibTemplate  
一个快速、全能的模板系统，它包含一个缓存和调试类。  
AvanTemplate  
多字节安全的模板引擎，占用很少系统资源。它支持变量替换，内容块可以设置显示或隐藏。  
Grafx Software’s Fast Template  
一个修改版本的Fast Template系统，它包括缓存功能，调试控制台以及沉默去除为赋值块。  
TemplatePower  
一个快速、简单、功能强大的模板类。主要功能有嵌套的动态块支持，块/文件包含支持以及显示/隐藏未赋值的变量。  
TagTemplate  
这个库的功能被设计来使用模板文件，同时允许你从HTML文件检索信息。  
htmltmpl: templating engine  
一个适用于Python和PHP的模板引擎。它面向希望在项目中分离代码和设计的web应用开发人员。  
PHP Class for Parsing Dreamweaver templates  
一个分析Dreamweaver模板的简单类，被用于Gallery 2 和WordPress的自定义模块中。  
MiniTemplator (Template Engine)  
针对HTML文件的一个紧凑型模板引擎。对于模板变量和块定义它具有简单的语法。其中块可以嵌套。  
Layout Solution  
简化网站开发和维护。它拥有常用的变量和页面元素使你不需要重复做页面布局工作。  
Cached Fast Template  
它已经纳入 FastTemplate ，允许你缓存模板文件，甚至可以在分离的块内容上缓存不同的规格。  
TinyButStrong  
一个支持MySQL, Odbc, Sql-Server和ADODB的模板引擎。它包含7个方法和两个属性。  
Brian Lozier’s php based template engine  
只有2K大小，非常快并且是面向对象设计。  
WACT  
一个从设计中分离代码的模板引擎。  
PHPTAL  
一个PHP下面的XML/XHTML模板库。

**ASP.NET**

* **简介**

ASP.net构架是可以用Microsoft（R）公司最新的产品 Visual Studio.net开发环境进行开发，WYSIWYG（What You See Is What You Get所见即为所得）的编辑。这些仅是ASP.net强大化软件支持的一小部分。

* **IDE**

Visual Studio

* **应用**

WEB、应用程

///////////////////////////////

要说WEB开发的语言，数不胜数，cgi/perl/ruby所代表的脚本语言，php/java/.net所代表的类c语言，asp所代表的basic语言......经过这么多年互联网应用中的大浪淘沙，当前主流的也就属php/java/.net三大阵营了。   
　　为什么是这三种语言处于WEB应用的主流地位？个人认为，它们有一个共性，就是类c语法。应用最广泛的语言就是c和basic，但basic本身不是完全OO的，而类c语言既拥有最广大的应用群体，又是面向对象语言，有basic之长而无basic之短。因此从学习曲线、IDE支持、编码规范等方面，都有着很大的优势。   
　　在php/java/.net这三者中，我看好java在企业级应用、php在WEB应用方面的前景，一方面二者都可以跨平台部署，另一方面比起.net来更轻巧和精简。一个php安装包，加上apache，也就几十兆；java更是只需jdk和appserver即可，加一起也就一二百兆，还可自由选择appserver。相比而言，.net安装包动辙上G，只能部署在windows下面，并且需要.net framework的支持。另一方面又分为vb.net/asp.net/c#.net，不仅分散了资源，也给应用者带来了困惑。   
　　为什么说php比java更合适web应用？一方面web应用要求更丰富的客户体验、更快捷的客户响应，另一方面要求更低的开发门槛和开发成本。在这方面java就显得庞大而臃肿，越来越多的框架更降低了灵活性，纯jsp+bean的模式至今仍有很大的市场也印证了这一点。相比而言，php有两个主要的缺点：一是缺乏IDE支持/无法重构、测试，二是处理不同数据库的语句各不相同，在更换数据库时需要修改大量的代码（到现在为止php仍然无法对mysql使用事务操作）。后者可以通过封装数据库操作来解决，前者暂时还没有好的解决办法。但尽管如此，作为一门快速开发语言，在WEB开发上PHP还是有着相当大的优势的，如果也出现一个类似rails的框架并解决跨数据库的问题，那更是如虎添翼。   
　　补充：现在已经有了php for rails。即cakephp。www.cakephp.org。   
以php的用户群来看，如果cakephp确实够优秀，确实是web开发的福音。   
  
再解释一下我不看好asp/ruby的理由：   
■asp：已经被微软自己和市场两方面淘汰。虽然ASP也可以使用javascript作为编程语言，但绝大多数人还是使用basic语言。弱解释的basic语言使它入门虽容易，但仅仅捆绑在IIS下，以及没有一个良好的框架支撑，使得它渐趋势微，在微软的产品线中又被asp.net所代替。   
  
■ruby：当前一个红得发热的新贵，初初试用了一下，语法更OO，更简洁，应该适合WEB开发。但它的缺点也很明显，第一是缺乏良好的IDE支持，诸如重构/语法检查方面都很弱；第二它比起asp/php/jsp来讲，它更像是一种脚本语言。学习曲线大，语法本身过于取巧，太过灵活

/////

PHP的主要语言开发者之一、Zend公司的创始人之一Andi Gutmans最近在blog中直言不讳地[批评了](http://andigutmans.blogspot.com/2008/03/java-is-losing-battle-for-modern-web.html)Java语言。他指出，目前Java厂商试图在JVM上提供动态语言实现的路子根本不对，Java已经输掉了Web开发语言的战争，PHP是事实上的标准和胜利者。

　　Gutmans的这篇文章在技术界引发了强烈争议，很对人对此有不同的意见，我在这里就阐述一下我对PHP和Java两种语言的看法。

　　我早先曾经在《[Perl、PHP、ASP、JSP技术比较](http://www.williamlong.info/archives/460.html)》一文中对比过PHP和Java的异同，简而言之，PHP是一种解释执行的脚本语言，语法和C语言类似，易学易用，不懂电脑的非专业人员稍经学习也能使用PHP.而Java要先编译成Class文件，然后在Java虚拟机上执行，Java开发需要熟悉Java语法以及一些核心的架构，从而实现一种可复用的、跨平台的软件，Java比PHP要难学的多。

　　实际上，从技术架构来说，Java语言相比PHP有明显的优势，Java使用的是面向对象的系统设计方法，而PHP还是采用面向过程的开发方法。PHP只能实现简单的分布式两层或三层的架构，而JAVA可以实现多层架构。数据库层（持久化层）、应用（业务）逻辑层、表示逻辑层彼此分开，而且现在不同的层都已经有一些成熟的开发框架的支持。例如Struts就是利用Java的web开发技术实现了MVC的设计模式，而在业务逻辑层也有Spring框架，数据库持久化层有Hibernate等框架。这些框架可以方便开发者高效、合理、科学得架构多层的商业应用。从数学运算和数据库访问速度来讲，Java的性能也优于PHP.实际上，对于跨平台的大型的企业应用系统来讲，Java几乎已经成为唯一的选择（微软.NET不支持跨平台），但是在于Web网站应用开发来讲，Java却面临着被PHP边缘化的危险，几乎所有的虚拟主机都支持PHP+MySQL，而支持Java的却少之又少，在资源上，网上有不计其数的PHP资源，很多著名的大型网站（例如Facebook、Mediawiki等）都是基于PHP的，而成功的Java网站却寥寥无几，这又是什么原因呢？

　　Java的理念是“一次编写，到处运行”，Java在应用框架底下的架构是无与伦比的，远胜过其他任何语言，Java的框架利于大型的协同编程开发，系统易维护、可复用性较好。而PHP很容易自学，让热备能快速简洁地编写代码，适合于快速开发，中小型应用系统，开发成本低。在调试、发布上，PHP也较Java简单。

　　理念上的不同导致了Java和PHP在Web应用开发上显示了不同的结果，尽管Java的数学计算和数据库访问都有优势，架构也相当完美，但是PHP却可以简单轻松地支持高强度Web访问，能够快速开发应用，支持PHP的虚拟主机多如牛毛，使得用PHP开发一个网站比用Java开发一个网站要快得多，容易的多。Java所拥有的优点只适合传统的以软件项目为核心的开发模式，而PHP更适合于以客户为核心的[SaaS](http://www.williamlong.info/?tags=SaaS)的开发模式，因此，PHP目前在Web网站开发的优势完全是因为Web网站开发的特殊性而导致的，并非编程语言特性所决定。

　　因此，PHP在Web网站开发语言的战争中已经慢慢将Java抛在了后面，Java要想在Web网站开发上奋起直追，的确应该多听听Andi Gutmans的意见，彻底改变一下思路才行，否则两者的差距只能是越来越远。

　　另外，虽然在国际上LAMP（Linux+Apache+Mysql+Php）架构已经占领了统治地位，但是在中国国内还有一个怪胎，就是基于微软的Windows Server+IIS+SqlServer+ASP/ASP.NET的架构，从理论上说，微软的这一套架构也能实现不错的性能，只可惜Windows和SqlServer价格不菲，整个架构在性能上比起LAMP不仅没有什么优势，反而还有不少劣势，因此微软的这一套家伙在国际市场上吃不开。而国内使用微软盗版则可以忽略成本，而ASP的易学易用性不逊于PHP，甚至连IT外行都能够使用，因此在国内也有不小的市场。但是随着中国在保护知识产权方面的日趋完善，免费的LAMP依旧是未来的趋势。

摘自<http://www.williamlong.info/archives/1314.html>

///

ASP全名Active Server Pages，是一个WEB服务器端的开发环境，利用它可以产生和执行动态的、互动的、高性能的WEB服务应用程序。ASP采用脚本语言VBScript（Java script）作为自己的开发语言。   
PHP是一种跨平台的服务器端的嵌入式脚本语言。它大量地借用C,Java和Perl语言的语法, 并耦合PHP自己的特性,使WEB开发者能够快速地写出动态产生页面。它支持目前绝大多数数据库。还有一点，PHP是完全免费的，不用花钱，你可以从PHP官方站点(http: //www.php.net)自由下载。而且你可以不受限制地获得源码，甚至可以从中加进你自己需要的特色。   
JSP是Sun公司推出的新一代网站开发语言，Sun公司借助自己在Java上的不凡造诣，将Java从Java应用程序和Java Applet之外，又有新的硕果，就是JSP，Java Server Page。JSP可以在Serverlet和JavaBean的支持下，完成功能强大的站点程序。   
三者都提供在 HTML代码中混合某种程序代码、由语言引擎解释执行程序代码的能力。但JSP代码被编译成 Servlet并由Java虚拟机解释执行，这种编译操作仅在对JSP页面的第一次请求时发生。在ASP 、PHP、JSP环境下，HTML代码主要负责描述信息的显示样式，而程序代码则用来描述处理逻辑。普通的 HTML页面只依赖于Web服务器，而ASP 、PHP、JSP页面需要附加的语言引擎分析和执行程序代码。程序代码的执行结果被重新嵌入到HTML代码中，然后一起发送给浏览器。ASP 、PHP、JSP三者都是面向Web服务器的技术，客户端浏览器不需要任何附加的软件支持。   
技术特点:   
ASP:   
1. 使用VBScript 、 JScript等简单易懂的脚本语言，结合HTML代码，即可快速地完成网站的应用程序。   
2. 无须compile编译，容易编写，可在服务器端直接执行。   
3. 使用普通的文本编辑器，如Windows的记事本，即可进行编辑设计。   
4. 与浏览器无关(Browser Independence), 客户端只要使用可执行HTML码的浏览器，即可浏览Active Server Pages所设计的网页内容。Active ServerPages 所使用的脚本语言(VBScript 、 Jscript)均在WEB服务器端执行，客户端的浏览器不需要能够执行这些脚本语言。   
5.Active Server Pages能与任何ActiveX scripting语言兼容。除了可使用VB Script或JScript语言来设计外，还通过plug－in的方式，使用由第三方所提供的其它脚本语言，譬如REXX 、Perl 、Tcl等。脚本引擎是处理脚本程序的COM(Component Object Model) 对象。   
6. 可使用服务器端的脚本来产生客户端的脚本。   
7. ActiveX Server Components(ActiveX 服务器组件 )具有无限可扩充性。可以使用Visual Basic 、Java 、Visual C++ 、COBOL等程序设计语言来编写你所需要的ActiveX Server Component 。   
PHP:   
1 数据库连接   
PHP可以编译成具有与许多数据库相连接的函数。PHP与MySQL是现在绝佳的群组合。你还可以自己编写外围的函数去间接存取数据库。通过这样的途径当你更换使用的数据库时，可以轻松地修改编码以适应这样的变化。PHPLIB就是最常用的可以提供一般事务需要的一系列基库。但PHP提供的数据库接口支持彼此不统一，比如对Oracle, MySQL，Sybase的接口，彼此都不一样。这也是PHP的一个弱点。   
JSP:   
1.将内容的产生和显示进行分离   
使用JSP技术，Web页面开发人员可以使用HTML或者XML标识来设计和格式化最终页面。使用JSP标识或者小脚本来产生页面上的动态内容。产生内容的逻辑被封装在标识和JavaBeans群组件中，并且捆绑在小脚本中，所有的脚本在服务器端执行。如果核心逻辑被封装在标识和Beans中，那么其它人，如Web管理人员和页面设计者，能够编辑和使用JSP页面，而不影响内容的产生。在服务器端，JSP引擎解释JSP标识，产生所请求的内容（例如，通过存取JavaBeans群组件，使用JDBC技术存取数据库），并且将结果以HTML（或者XML）页面的形式发送回浏览器。这有助于作者保护自己的代码，而又保证任何基于HTML的Web浏览器的完全可用性。   
2.强调可重用的群组件   
绝大多数JSP页面依赖于可重用且跨平台的组件（如：JavaBeans或者Enterprise JavaBeans）来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使得这些组件为更多的使用者或者用户团体所使用。基于组件的方法加速了总体开发过程，并且使得各种群组织在他们现有的技能和优化结果的开发努力中得到平衡。   
3.采用标识简化页面开发   
Web页面开发人员不会都是熟悉脚本语言的程序设计人员。JavaServer Page技术封装了许多功能，这些功能是在易用的、与JSP相关的XML标识中进行动态内容产生所需要的。标准的JSP标识能够存取和实例化JavaBeans组件，设定或者检索群组件属性，下载Applet，以及执行用其它方法更难于编码和耗时的功能。通过开发定制化标识库，JSP技术是可以扩展的。今后，第三方开发人员和其它人员可以为常用功能建立自己的标识库。这使得Web页面开发人员能够使用熟悉的工具和如同标识一样的执行特定功能的构件来工作。 JSP技术很容易整合到多种应用体系结构中，以利用现存的工具和技巧，并且扩展到能够支持企业级的分布式应用。作为采用Java技术家族的一部分，以及Java 2EE的一个成员，JSP技术能够支持高度复杂的基于Web的应用。由于JSP页面的内置脚本语言是基于Java程序设计语言的，而且所有的JSP页面都被编译成为Java Servlet，JSP页面就具有Java技术的所有好处，包括健壮的存储管理和安全性。作为Java平台的一部分，JSP拥有Java程序设计语言“一次编写，各处执行”的特点。随着越来越多的供货商将JSP支持加入到他们的产品中，您可以使用自己所选择的服务器和工具，修改工具或服务器并不影响目前的应用。   
应用范围：   
ASP是Microsoft开发的动态网页语言，也继承了微软产品的一贯传统，只能执行于微软的服务器产品,IIS(Internet Information Server) (windows NT)和PWS(Personal Web Server)(windows 98)上。Unix下也有ChiliSoft的组件来支持ASP，但是ASP本身的功能有限，必须通过ASP＋COM的群组合来扩充，Unix下的COM实现起来非常困难。   
结论：ASP简单而易于维护，是小型网站应用的最佳选择，通过DCOM和MTS技术，ASP甚至还可以完成中等规模的企业应用。  
PHP可在Windows,Unix,Linux的Web服务器上正常执行,还支持IIS,Apache等一般的Web服务器,用户更换平台时,无需变换PHP代码,可即拿即用。   
结论：PHP因为结构上的缺陷，使的其只适合编写小型的网站系统。  
JSP同PHP类似，几乎可以执行于所有平台。如Win NT,Linux,Unix。在NT下IIS通过一个外加服务器，例如JRUN或者ServletExec，就能支持JSP。知名的Web服务器Apache已经能够支持JSP。由于Apache广泛应用在NT、Unix和Linux上，因此JSP有更广泛的执行平台。虽然现在NT操作系统占了很大的市场份额，但是在服务器方面Unix的优势仍然很大，而新崛起的Linux更是来势不小。从一个平台移植到另外一个平台，JSP和JavaBean甚至不用重新编译，因为Java字节码都是标准的与平台无关的。   
结论：对于脚本语言来讲，JSP还是拥有相当大的优势的，虽然其配置和部署相对其他脚本语言来说要复杂一些，但对于跨平台的中大型网站系统来讲，基于JAVA技术的JSP（结合JavaBean和EJB）几乎成为唯一的选择。  
性能比较：   
有人做过试验，对这三种语言分别做回圈性能测试及存取Oracle数据库测试。在循环性能测试中，JSP只用了令人吃惊的四秒钟就结束了20000＊20000的回圈。而ASP、PHP测试的是2000＊2000循环（少一个数量级），却分别用了63秒和84秒。（参考PHPLIB）。 数据库测试中，三者分别对 Oracle 8 进行 1000 次 Insert,Update,Select和Delete： JSP 需要 13 秒，PHP 需要 69 秒，ASP则 需要 73 秒。   
前景分析：   
目前在国内PHP与ASP应用最为广泛。而JSP由于是一种较新的技术，国内采用的较少。但在国外，JSP已经是比较流行的一种技术，尤其是电子商务类的网站，多采用JSP。采用PHP的网站如新浪网（sina）、中国人（Chinaren）等，但由于PHP本身存在的一些缺点，使得它不适合应用于大型电子商务站点，而更适合一些小型的商业站点。首先，PHP缺乏规模支持。其次，缺乏多层结构支持。对于大负荷站点，解决方法只有一个：分布计算。数据库、应用逻辑层、表示逻辑层彼此分开，而且同层也可以根据流量分开，群组成二维数组。而PHP则缺乏这种支持。还有上面提到过的一点，PHP提供的数据库接口支持不统一，这就使得它不适合运用在电子商务中。   
ASP和JSP则没有以上缺陷，ASP可以通过Microsoft Windowsd的COM/DCOM获得ActiveX规模支持，通过DCOM和Transcation Server获得结构支持；JSP可以通过SUN Java的Java Class和EJB获得规模支持，通过EJB/CORBA以及众多厂商的Application Server获得结构支持。三者中，JSP应该是未来发展的趋势。世界上一些大的电子商务解决方案提供商都采用JSP/Servlet。比较出名的如IBM的E－business，它的核心是采用JSP/Servlet的Web Sphere。它们都是通过CGI来提供支持的。但去年10月后它推出了Enfinity，一个采用JSP/Servlet的电子商务Application Server，而且声言不再开发传统软件。   
　Perl  
  
　　Perl（Practical Extraction and Report Language）是一种很古老的脚本语言。最初的Web应用大多是用Perl编写的，Perl很像C语言，使用非常灵活，对于文件操作和处理具有和C语言一样的方便快捷。  
  
　　也正是因为Perl的灵活性和“过度”的冗余语法，也因此导致许多Perl程序的代码令人难以阅读和维护，因此使用的人在逐渐减少，并且目前有被Python替代的可能。  
  
　　另外Perl对于CPU的消耗似乎较高，效率似乎有一些不足。  
  
　　Perl开发的成功案例：  
  
　　Movable Type — 世界上使用最多的Blog系统之一，功能强大的足以和一个CMS系统相比拟。  
  
　　Awstats — 非常经典的日志系统，速度极快，功能也非常强大。  
  
结论：Perl在部分应用中能发挥很大优势，但其维护性差使得其普及变得很困难。  
  
.NET  
  
.NET的优势：企业信息系统部门可自行编程  
.NET是Microsoft贴在现有产品和未来产品上的一个新式行销标签。在Microsoft服务器端产品中，如基于Windows DNA 2000技术的izTalk Server 2000和Application Center 2000，.NET标签将随处可见。然  
而，.NET最令人感兴趣的特色却在于它的开发平台、语言和协议。   
  
　　 通过.NET，Microsoft为我们提供了一种全新的开发平台，这个平台将推动以新体系为基础的协同Web应用开发。根据Microsoft的说法，.NET至少将成为一种“对Web编程”的方法。当前，第一版Visual Studio .NET已经出现，由此出发，我们或许可以比较精确地勾画出.NET平台最终的全貌。   
：.NET开发工具--Visual Studo.NET的易用性使得它在Java很难实现的“用户企业（信息系统部门）也可编程的系统”上超人一头。笔者认为可能还存在其他这样的领域。当然，尽管.NET的标准语言VB.NET和C#都与Java一样复杂，但是Visual Studo.NET却很好地消除了这种复杂性。   
　　比如，日本兰碧儿（NOEVIR）只靠自己的信息系统部门就完成了Web代理店支援系统的从企划到编程的全部工作。日本一家系统集成商曾表示，此前导入Java系统的用户企业中“选用.NET的信息系统部门就有望自行开发或对系统进行修改”。   
　　笔者认为，由于Java本身较为复杂，且不易学习掌握，因此“用户企业只负责企划，而将编程工作完全委托给集成商的现象非常普遍”。   
　　在COBOL和VB（客户商/服务器）的全盛时期，信息系统部门自己编程、对系统进行部分修改和功能追加是很正常的事情。如果连细小的系统修改都要外包，那么系统的维护成本自然会很高，因此就将失去灵活性。甚至于还有可能削弱信息系统部门的业务能力。说到企划，听起来好听，但如果连大体的情况也不了解，恐怕与开发商交涉都很困难。这也许就是Java的一个“弊端”吧。   
　　此外，“成本比UNIX低”以及“可以充分利用Windows技术资源”等原因也在加速.NET Framework的普及。作为客户端/服务器系统的开发环境，VB的后续语言--.NET中的“Windows Form”最具竞争力。越来越多的系统集成商开始选择这一工具构筑系统。   
.NET的劣势：应用成果少、开放性差   
　　反过来说，.NET的劣势则在于在大型系统中的应用成果少、缺乏“开放性”。   
　　对于应用成果来说，今后其数量将会不断增加。据悉，日本Kabu.com证券将在2003年度内利用.NET Framework全面重新构筑该公司目前利用ASP构筑的系统。   
　　在“开放性”方面又将会如何呢？Java普及的主要因素很多，但笔者认为其中最关键的在于“开放性”。由于可各种OS上运行，而且很多开发商还提供了Web应用服务器和开发工具，因此用户可以随意选择。而开发商则通过在功能和性能方面相互竞争来不断提高产品水平。   
　　而.NET Framework怎么样呢？只要想应用于业务系统中，其OS实际上则只能限于Windows。尽管微软已经提供了FreeBSD版.NET Framework，但其目的仅限于学术研究。   
　　今后人们所能期待的是Mono Project。Mono Project是指完全独立于微软、正在开发开放源码版.NET Framework的计划。主导该计划的是一家名为Ximian的Linux企业，这家公司成功地开发了Linux标准桌面环境GNOME，目前已受到业界的广泛关注。   
　　2003年1月该计划还发布了运行Web应用程序ASP.NET的Web服务软件XSP。尽管很多功能仍在开发之中，但目前开发工作进展顺利。目前正面向Linux和Windows进行开发。据Mono Project称，由于该软件依赖于Linux的部分不多，因此能够轻松地向其他UNIX OS移植。   
　　能否形成像Mono一样的开放环境将是.NET普及的关键所在。微软目前已经向业界标准化机构ECMA（欧洲计算机制造商协会）提交了.NET标准，此举表明微软将采用开放标准。但目前尚不清楚此举只是单纯的一种姿态，还是真的要开放.NET。这就要看微软今后是否能够继续公开足够的相关标准和信息了。  
可以说，无论是在技术上还是在战略上，Microsoft都对.NET寄予了厚望。当前的DNA 2000技术并未包含新的.NET平台；相反，.NET将是一种全新的技术，有望揉合许多根本性的、深层次的创意。   
  
　　 作为一种全新的平台和技术，.NET带来了许多新的产品，但这些产品和现有技术的兼容性未能获得百分之百的保证。.NET支持27种编程语言，它们将共享一组提供基本服务的类。.NET应用不再直接在裸机上运行，由于看好一种运行在虚拟机CLR（Common Language Runtime）上的称为MSIL的中间语言，Microsoft放弃了Intel x86代码。  
.net就远来说可能更有前途，毕竟是microsoft统治时期  
就JAVA来说，最近比较热，J2EE等都是市场比较抢手的。  
  
XML   
Extensible Markup Language,可扩展标识语言。它是一种由W3C发展，可由网页设计者自行定义标签（TAG）HTML延伸规格，主要目的是让使用者能够设计结构性资料及文件格式，并对资料如何存储，接受与应用提供标准描述方法。简单地说，XML就是一种用来在互联网上创建电子文档的标准。  
  
CGI   
Common Gateway Interface,公共网关接口。网页服务器与应用程序之间传递资料的接口规范，CGI程序客用冷和程序语言产生，通常为Perl，使用CGI程序可以读取使用者的输入并产生动态的HTML网页。CGI每次响应一个请求都会建立一个进程（Process），对于服务器来说是一个不校的负担哦。不过，现在的Fast－cgi以及对此有很大的改善，所以就不用怕使用CGI编的论坛会占有你过多的系统资源了.  
DELPHI  
Delphi是著名的Borland（现在已和Inprise合并）公司开发的可视化软件开发工具。“真正的程序员用c，聪明的程序员用Delphi”，这句话是对Delphi最经典、最实在的描述。Delphi被称为第四代编程语言，它具有简单、高效、功能强大的特点。和VC相比，Delphi更简单、更易于掌握，而在功能上却丝毫不逊色；和VB相比，Delphi则功能更强大、更实用。可以说Delphi同时兼备了VC功能强大和VB简单易学的特点。它一直是程序员至爱的编程工具。  
　　Delphi具有以下的特性：基于窗体和面向对象的方法，高速的编译器，强大的数据库支持，与Windows编程紧密结合，强大而成熟的组件技术。但最重要的还是Object Pascal语言，它才是一切的根本。 Object Pascal语言是在Pascal语言的基础上发展起来的，简单易学。  
　　Delphi提供了各种开发工具，包括集成环境、图像编辑（Image Editor），以及各种开发数据库的应用程序，如DesktopDataBase Expert等。除此之外，还允许用户挂接其它的应用程序开发工具，如Borland公司的资源编辑器（Resourse Workshop）。  
　　在Delphi众多的优势当中，它在数据库方面的特长显得尤为突出：适应于多种数据库结构，从客户机／服务机模式到多层数据结构模式；高效率的数据库管理系统和新一代更先进的数据库引擎；最新的数据分析手段和提供大量的企业组件。  
　　Delphi发展至今，从Delphi１、Delphi２到现在的Delphi5，不断添加和改进各种特性，功能越来越强大。Delphi５添加了对IDE（集成开发环境）的很多改进新特性，扩展了数据库支持（ADO和InterBase数据库），带有Internet支持的MIDAS改进版，TeamSouse版本控制工具，转换功能，框架概念以及很多的新组件与新特性。  
JavaScript+Html  
JavaScript 是一种描述语言，此一语言可以被箝入 HTML 的文件之中。   
透过 JavaScript 可以做到回应使用者的需求事件 (如： form 的输入) 而不用任何的网路   
来回传输资料，所以当一位使用者输入一项资料时，它不用经过传给伺服端 (server)   
处理，再传回来的过程，而直接可以被客户端 (client) 的应用程式所处理。你也可   
以想像成有一个可执行程式在你的客端上执行一样！目前已有一些写好的程式在   
Internet 上你可以连过去看看，以下有一些计算器的例子，在 Nescape 上。   
  
JavaScript 和 Java 很类似，但到底并不一样！ Java 是一种比 JavaScript 更复杂   
许多的程式语言，而 JavaScript 则是相当容易了解的语言。JavaScript 创作者   
可以不那么注重程式技巧，所以许多 Java 的特性在 Java Script 中并不支援  
Javascript的语法基本上与Java一致，但是由于Javascript是一个弱类型的脚本语言，在程序编写的过程中会有一些不同。同时由于Javascript是基于对象的语言，注意不是面向对象的语言，所以它在对对象的支持上有一定缺陷，大家所熟悉的诸如对象继承，多态等面向对象语言所具有的基本特性在Javascript中只能通过一些变通手段来实现（通常比较复杂）。然而，弱类型语言也有其优点，那就是简单性，Javascript中类型转化是非常方便的（弱类型语言在代码中基本上表现为无类型），一个String通过一个简单的加减操作就可以转化为Int（相当于调用了Integer.ParseInt(String)），而且不会抛异常。Javascript作为一种解释性语言，还能使用在编译性语言C/C++、JAVA难以支持的eval语句。由于运行在沙箱中，Javascript运行时有很多的安全性限制。它不允许访问本地的硬盘，并不能将数据存入到服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互，从而有效地防止数据的丢失。  
  
  
Ajax是Asynchronous JavaScript and XML的简称，它表现出一个Web开发上的根本转变，那就是，Web上可能做些什么.   
Ajax的定义  
　　Ajax不是一个技术，它实际上是几种技术，每种技术都有其独特这处，合在一起就成了一个功能强大的新技术。Ajax包括：   
XHTML和CSS   
使用文档对象模型(Document Object Model)作动态显示和交互   
使用XML和XSLT做数据交互和操作   
使用XMLHttpRequest进行异步数据接收   
使用JavaScript将它们绑定在一起   
Ajax理念的出现，揭开了无刷新更新页面时代的序幕，并有代替传统web开发中采用form(表单)递交方式更新web页面的趋势，可以算是一个里程碑。但Ajax都不是适用于所有地方的，它的适用范围是由它的特性所决定的。它的特性主要有：   
　　1、按需取数据。页面不读取无用的冗余数据，而是在用户操作过程中的某项交互需要某部分数据时才会向服务器发送请求。   
　　2、无刷新更新页面。客户端利用XMLHTTP发送请求得到服务端应答数据，在不重新载入整个页面的情况下用JavaScript操作DOM最终更新页面。  
　　3、预读功能也可以通过Ajax实现，但并不是Ajax的优势所在，它的主要优势还是在交互方面。   
  
　　Ajax带来的正面效果：  
1、减轻服务器的负担。因为Ajax的根本理念是“按需取数据”，所以最大可能在减少了冗余请求和响影对服务器造成的负担。  
2、减少用户实际和心理等待时间。  
　　传统的web应用模型工作起来就象这样：大部分界面上的用户动作触发一个连接到Web服务器的HTTP请求。服务器完成一些处理---接收数据，处理计算，再访问其它的数据库系统，最后返回一个HTML页面到客户端。这是一个老套的模式，自采用超文本作为web使用以来，一直都这样用, 但看过《The Elements of User Experience》的读者一定知道，是什么限制了Web界面没有桌面软件那么好用。  
　　这种旧的途径让我们认识到了许多技术，但它不会产生很好的用户体验。当服务器正在处理自己的事情的时候，用户在做什么？没错，等待。每一个动作，用户都要等待。   
  
　　很明显，如果我们按桌面程序的思维设计Web应用，我们不愿意让用户总是等待。当界面加载后，为什么还要让用户每次再花一半的时间从服务取数据？实际上，为什么老是让用户看到程序去服务器取数据呢？   
  
　　Ajax如何不同凡响  
　　通过在用户和服务器之间引入一个Ajax引擎，可以消除Web的开始－停止－开始－停止这样的交互过程. 它就像增加了一层机制到程序中，使它响应更灵敏，而它的确做到了这一点。   
  
　　不像加载一个页面一样，在会话的开始，浏览器加载了一个Ajax引擎---采用JavaScript编写并且通常在一个隐藏frame中。这个引擎负责绘制用户界面以及与服务器端通讯。Ajax引擎允许用异步的方式实现用户与程序的交互－－不用等待服务器的通讯。所以用户再不不用打开一个空白窗口，看到等待光标不断的转，等待服务器完成后再响应。  
　　通常要产生一个HTTP请求的用户动作现在通过JavaScript调用Ajax引擎来代替. 任何用户动作的响应不再要求直接传到服务器---例如简单的数据校验，内存中的数据编辑，甚至一些页面导航---引擎自己就可以处理它. 如果引擎需要从服务器取数据来响应用户动作---假设它提交需要处理的数据，载入另外的界面代码，或者接收新的数据---引擎让这些工作异步进行，通常使用XML, 不用再担误用户界面的交互。   
  
　　谁在使用Ajax  
　　在采用Ajax的开发上面，Google做了巨大的投资。去年Google所有主要的产品都用了这项技术---Orkut, Gmail, 以及最近的beta版的Google Groups, Google Suggest和Google Maps---它们全是Ajax的应用。(要想了解更多这些Ajax实际的技术细节，请看它们的分析文章:Gmail, Google Suggest, Google Maps). 其它的像：Flickr, 采用许多人们喜欢的Ajax特性，　还有Amazon的A9.com搜索引擎也采用类似的技术。   
  
　　这些项目证明了Ajax不只是学术上的，也有许多真实世界成功应用。这不是什么实验室里的技术。Ajax的应用可大可小，从非常简单的，像单一功能的Google Suggest到非常复杂的Google Maps。