**.NET入门教程 学习.NET的几个基本步骤**

**.NET** 是 Microsoft XML Web services 平台。XML Web services 允许应用程序通过 Internet 进行通讯和共享数据，而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建 XML Web services 并将这些服务集成在一起之所需。对个人用户的好处是无缝的、吸引人的体验。

如果你已经有较多的面向对象开发经验，跳过以下这两步：

第一步掌握一门.NET面向对象语言，C#或VB.NET我强烈反对在没系统学过一门面向对象(OO)语言的前提下去学ASP.NET。ASP.NET是一个全面向对象的技术，不懂OO，那绝对学不下去!

第二步对.NETFramework类库有一定的了解可以通过开发WindowsForm应用程序来学习.NETFramework。ASP.NET是建构在.NETFramework之上的技术，你对.NETFramework了解得越深，学习ASP.NET就越快。

举个例子：下面简单列出几个对掌握ASP.NET非常重要的概念：对象的内存模型，委托，事件，多线程，程序集和应用程序域，安全模型.NETFramework当然还有许多其它的东西，但不理解与把握清楚上述这些东西，是很难真正把握ASP.NET的。出于急迫的心情与现实的考虑，不少人没有扎实的面向对象基础就想直接学习ASP.NET，其结果只能是欲速则不达。在具备了OO基础之后，以下是具体的学习ASP.NET技术步骤。

**第一步：学习HTML与CSS**

这并不需要去学一大堆的诸如Dreamweaver，Firework之类的各种网页设计工具，关键是理解HTML网页嵌套的block结构与CSS的box模型。许多ASP.NET控件最后都必须转化为HTML。而且，div+CSS是当前主流的网页布局模型。学习这部分时，关键在于理解概念，而不需要将精力花在美化页面的许多技巧上，那是网站美工的工作，不是程序员的工作。

**第二步：学习JavaScript**

JavaScript不是Java，它主要运行于浏览器端。可以完成许多工作，功能也很强大：比如它将客户端网页中的HTML元素看成一棵树，可以编写代码访问并修改树节点，动态生成新的HTML代码，从而达到动态修改网页显示特性的目的。

JavaScript在目前的网站开发中用得很多，非常重要。

另外，它也是目前非常流行的AJAX技术的基础。

**第三步：学习计算机网络原理**

找一本大学<计算机网络>教材，重点看看它的有关互联网的部分，了解一些域名解析和HTTP协议等知识。这是进行互联网开发的理论基础。

**第四步：学习ASP.NET表示层技术，会设计Web页面**

需要结合先前打好的面向对象技术的基础，至少掌握以下内容：

(1)各种Web控件的使用方法，

(2)理解信息在网页中的传送方式，比如Cookie，ViewState，Session等的使用。

(3)ASP.NET应用程序与网页的生命周期，以及相关对象(比如httpcontext，response，request)的用途。

(4)ASP.NET实现事件驱动的内幕

(5)自定义用户控件

再次强调一下，没有OO基础，很难掌握上述技术，就只能被这些东东牵着鼻子走了，会很被动。

**第五步掌握数据库技术**

具体地说，要学习以下内容：

(1)学会使用SQLServer2005：不要求精通它的各种工具与管理配置技术，但至少知道如何连接，如何建表，如何创建存储过程

(2)学习ADO.NET，掌握使用代码人工访问数据库(别用VS2005的向导)的方法

(3)学习数据绑定控件的使用

**第六步理解多层架构**

这时，先前在OO学习阶段涉及到的程序集与应用程序域等就派上用场了，现在，网站架构大多采用多层架构：表示层、业务逻辑层、数据存取层以及数据库本身。

可以先上网找一此多层架构的资料看，再找一个现成的比较复杂的开源ASP.NET项目分析一下其架构就差不多了。基本上都是一个套路，到处应用。

有的朋友问：学习架构是不是必须学习设计模式。

我的看法是：不必!当然，你如果学习过设计模式，那当然更好。但在实际开发中，如果只想着机械地套用某种模式，反而起不到好的结果。我的观点：在学习设计模式时要多思多悟，其思想就会渐渐地融入你的大脑，在真实的设计实践中，忘掉所有的写在书上的模式，一切从实际出发，相信你的直觉，只要达到设计要求的方案就是可行的方案，事实上，你这样做了之后，回过头来再看，会发现你的设计往往暗合设计模式的理论。

**第七步学习XML与WebService**

先了解XML的基础知识，找本讲XML的书看一下就差不多了，然后，再学习WebService。WebService其实可类比为远程方法调用(以XML格式表达的调用信息)。

学到了这里，如果你还有兴趣，不妨再去看看SOA，不过SOA的资料都是理论与概念，看起来比较郁闷，而且离实际开发可能比较远。所以，这是可选项。

**第八步学习AJAX**

学习AJAX的主要目的是创建具有更丰富特性的Web表示层，而经过前面七步的学习，到此再学习AJAX已水到渠成，所有的基础都已具备，不会有太大的问题了。

在基础未具备的前提下直接学AJAX，就象“浮沙之上筑高台”，所以，将AJAX的学习次序安排在这里。第九步学习RIA技术

RIA：Richinternetapplication，可看成是将C/S与B/S优点合二为一的尝试。就具体技术来说，主要指微软的Silverlight(WPF\_E)，毕竟前头一路行来都是微软路线。