

项 目 名 称:	2015 最新.net 学习步骤		
项 目 经 理:		文 档 版 本 编 号:	1.1
起 草 人:	远征	起 草 日 期:	2015-06-08
复 审 人:		复 审 日 期:	
来 自 From	中国区视频市场	给 To	

2015 最新.net 学习步骤

撰 写 人 ： 远征

编写时间 ： 2015-6-8

审 核 人 ：

审核时间 ：

中国区视频市场

修订记录

修改标示说明：： *A - 增加 M - 修改 D - 删除

版本	日期	图 或 表 或 章节号	AMD	标题或简要描述	变 更 申 请号	修订者	批准者
1.0	2015-06-08		A	总体创建		远征	
1.1	2015-07-26		A	增加视频路径		远征	

目 录

第 1 章 引言	3
1.1 步骤介绍	3
第 2 章 .NET 学习步骤（传智播客）	4
2.1 传智播客最新学习路线	4
2.1.1 .NET 基础部分	4
2.1.2 C# 基础加强部分	4
2.1.3 数据库	5
2.1.4 三层项目-餐饮管理系统	6
2.1.5 WEB 开发基础加强	6
2.1.6 ASP.NET 部分	6
2.1.7 项目 1：B2C 电子商城部分	7
2.1.8 ASP.NET MVC 与 EF 部分	7
2.1.9 私有云 OA 项目	8
2.1.10 移动互联网应用	8
第 3 章 .NET 学习步骤（黑马）	10
3.1 传智播客最新学习路线图	10
第 4 章 学了.NET 能做什么开发	11
第 5 章 .NET 应该学什么怎么学	12
5.1 （一）	12
5.2 （二）	14
5.3 （三）	16
5.4 （四）	18

第1章 引言

1.1 步骤介绍

根据最新.Net 技术平台的发展趋势和.Net 整个课程体系得出如下完整的路线图,按照路线图学习完全可以达到非常牛逼的高手级别。

.Net 学习步骤分为 2 条路线: 1.传智播客路线 2.黑马路线。

条条大路通罗马,建议.net 学习者结合自己的实际情况选择学习路线。

第2章 .NET 学习步骤（传智播客）

2.1 传智播客最新学习路线

2.1.1 .net 基础部分



第二步：C#基础加强部分

第1阶段 面向对象

- >> 面向对象思想
- >> 类与对象
- >> 类的组成
- >> 封装及其应用
- >> 继承及其应用
- >> 多态及其应用
- >> 抽象类及其应用
- >> 接口及其应用
- >> 字段和属性及索引
- >> 简单工厂模式

第2阶段 字符串

- >> 字符串不可变性
- >> 字符串的缓存池特性
- >> 字符串的常用方法
- >> 字符串的属性
- >> StringBuilder
- >> 垃圾回收站
- >> ref和out

第3阶段 集合和文件流

- >> 泛型集合
- >> List<>和Dictionary
- >> List<>集合常用方法
- >> 装箱和拆箱
- >> 文件操作
- >> File类
- >> FileStream类
- >> 文件读写
- >> Path类
- >> StreamReader类
- >> StreamWriter类
- >> Directory类

第4阶段 正则表达式和xml文件操作

- >> 基本元字符和限定符
- >> 匹配邮箱正则表达式
- >> 案例网页提取职位信息
- >> 案例网页下载图片
- >> 敏感词过滤
- >> xml读写
- >> xml和html区别
- >> xml增删改查案例

第5阶段 委托和事件及反射

- >> 委托数据类型
- >> 委托本质
- >> 通过委托实现窗体传值
- >> 事件的本质
- >> 通过事件实现窗体传值
- >> 程序集
- >> Type类
- >> 反射介绍
- >> 通过反射实现记事本插件案例

网盘路径：传智播客.NET 视频全部打包(全) > 第二步：C#基础加强部分

2.1.2 数据库

第三步：数据库

第1阶段 数据库介绍

- >> 数据库的特点
- >> 数据库的种类
- >> 创建数据库、表
- >> 数据库中的类型
- >> 数据库的分离和附加
- >> 表的主键、外键及约束

第2阶段 sql

- >> 数据检索
- >> 一次性插入多条数据
- >> 修改表中数据
- >> 一次性删除多条数据
- >> 聚合函数、模糊查询
- >> 空值处理
- >> 排序、分组、分组后筛选数据
- >> 类型转换和联合结果集
- >> 字符串常用方法和日期函数

第3阶段 ADO.NET

- >> ADO.NET介绍
- >> 数据库连接对象 Connection和Command执行sql语句
- >> 连接池
- >> ExecuteNonQuery()方法
- >> ExecuteScalar()方法
- >> ExecuteReader()方法
- >> 通过C#操作数据库增删改查案例
- >> sql注入参数化
- >> App.config和SqlHelper封装
- >> Dataset使用
- >> 多条件查询

第4阶段 数据库高级

- >> case-end使用
- >> 子查询、分页及表连接
- >> 局部变量和全局变量及视图
- >> 事务、存储过程和触发器及索引
- >> 游标、数据库三大范式

网盘路径: 传智播客.NET 视频全部打包(全) > 第 3 步: 数据库

2.1.3 三层项目-餐饮管理系统



网盘路径: 传智播客.NET 视频全部打包(全) > 第 4 步: 三层项目-餐饮管理系统, +

送人事管理系统

2.1.4 Web 开发基础加强



网盘路径: 传智播客.NET 视频全部打包(全)>第 5 步: Web 开发基础加强 Asp.Net 部

分

第六步：Asp.Net部分

第1阶段 模拟IIS

- >> socket
- >> 委托
- >> 多线程
- >> http协议

第2阶段 一般处理程序

- >> 开发与部署
- >> asp.net生命周期
- >> 请求与响应+Razor模板引擎
- >> 调用ado.net
- >> 文件上传与下载
- >> 验证码、水印、缩略图

第3阶段 WebForm与数据绑定

- >> 页面类生命周期
- >> 服务器控件
- >> 数据源控件
- >> 数据绑定控件

第4阶段 状态保持

- >> 客户端实现
- >> 服务器端实现
- >> ViewState
- >> Cookie
- >> Session

第5阶段 异步Ajax

- >> XmlHttpRequest介绍
- >> JQuery的异步方法
- >> 实现省市联动
- >> Json, Ajax实现异步增删改查
- >> Ajax实现异步登陆
- >> Ajax无刷新分页

第6阶段 企业高级技术

- >> 全局应用程序类
- >> 母版页
- >> 错误处理
- >> 接口IHttpModule介绍
- >> 一般处理程序重用
- >> 缓存应用

网盘路径：传智播客.NET 视频全部打包(全)> 第6步：Asp.Net 部分

2.1.5 项目 1：B2C 电子商城部分（失联中……找到后会第一时间发给您）

第七步：项目1：B2C电子商城部分

第1阶段 设计

- >> 电子商务介绍
- >> 功能分析与数据库设计
- >> 三层代码生成工具
- >> 版本控制
- >> 框架搭建
- >> 简单工厂
- >> 抽象工厂

第2阶段 后台功能实现

- >> 主框架搭建
- >> 管理员信息维护、登录、验证
- >> 分类维护（树状控件）
- >> 商品维护
- >> 富文本编辑器
- >> 图片异步上传
- >> 分类三级异步联动

第3阶段 前台展示实现

- >> 用户注册、登录、验证
- >> 商品展示
- >> URL重定向
- >> 生成静态页

第4阶段 购买与支付

- >> 购物车
- >> 订单页面
- >> 移动支付

2.1.6 Asp.Net MVC 与 EF 部分

第八步：AspNetMVC与EF部分

- >> 复习委托,匿名方法与Lambda表达式
- >> 实体框架 (Entity Framework) 简介
- >> 使用EF完成数据添加, 删除, 更新与查询 (linq表达式)
- >> 追踪EF生成的SQL脚本
- >> 框架搭建
- >> EF映射原理分析
- >> EF延迟加载机制

- >> Model-First的创建
- >> Code-First的创建
- >> IQueryable接口与IEnumerable区别
- >> 导航属性应用
- >> 查询部分列, 排序与分页
- >> EF上下文对象创建问题
- >> EF跨数据库

- >> AspNetMVC请求模型
- >> 约定大于配置
- >> 深入理解Model,Controller, 以及View
- >> AspNet MVC 添加, 删除, 修改, 查询
- >> AspNet MVC分页
- >> 强类型视图
- >> HtmlHelper以及HtmlHelper扩展方法

- >> AspNetMVC 客户端校验
- >> JQuery Ajax
- >> AspNet MVC Ajax(Json)
- >> AspNet 路由规则 (Routing)
- >> 深入讲解ActionResult
- >> Razor 视图引擎
- >> AspNet MVC Areas介绍
- >> 过滤器
- >> AspNetMVC原理与框架源码分析

网盘路径: 传智播客.NET 视频全部打包(全)> 第 8 步: AspNet MVC 与 EF 部分

2.1.7 私有云 OA 项目（失联中……找到后会第一时间发给您）

第九步：私有云OA项目

- >> OA项目简介
- >> 云计算概念
- >> 云平台搭建
- >> 高可用性高扩展性框架搭建
- >> 面相对象高级应用
- >> Spring.Net框架应用
- >> T4模板自动化生成项目
- >> JQueryEasyUI搭建项目

- >> 日志处理系统 (Log4Net)
- >> Redis分布式队列集群应用
- >> 分布式缓存: Memcached
- >> Json.Net
- >> WebService
- >> WCF
- >> WebApi
- >> 权限分析设计

- >> 分布式文件集群搭建
- >> Nginx服务器集群
- >> 数据库集群
- >> Lucene.Net+盘古分词企业级
- >> 海量数据搜索
- >> 搜索热词统计
- >> 搜索热词统计
- >> MongoDB应用
- >> Bootstrap

- >> 工作流简介
- >> 状态机工作流
- >> BookMark
- >> 工作流持久化
- >> 财务审批

2.1.8 移动互联网应用（失联中……找到后会第一时间发给您）

第十步：移动互联网应用

第1阶段 HTML5

- >> HTML5 开发工具
- >> 新标签和属性的应用
- >> 本地存储的应用
- >> 智能表单的特点及应用
- >> 网页多媒体的实现及优势
- >> 本地文件操作 API 使用详解
- >> Canvas 绘图 API 及其注意事项
- >> 绘制基本元素
- >> Canvas 开发虐心游戏像素鸟
- >> 离线应用开发方式
- >> 跨文档通讯
- >> 摄像头及地理信息操作
- >> Web Socket 实时通讯

第2阶段 HybirdApp

- >> 移动应用市场以及发展趋势
- >> jQueryMobile 使用详解
- >> App Framework 使用
- >> Sencha Touch 框架使用
- >> 前端 MVC 思想及常用框架
- >> PhoneGap 环境搭建和配置
- >> Android SDK 环境配置详解
- >> 利用 PhoneGap 将 WebApp 打包成为可安装的应用
- >> PhoneGap 中间件的 API 详解
- >> Cordova 使用方式
- >> Node.js 简介
- >> 移动在线商城案例

第3阶段 Unity 3D

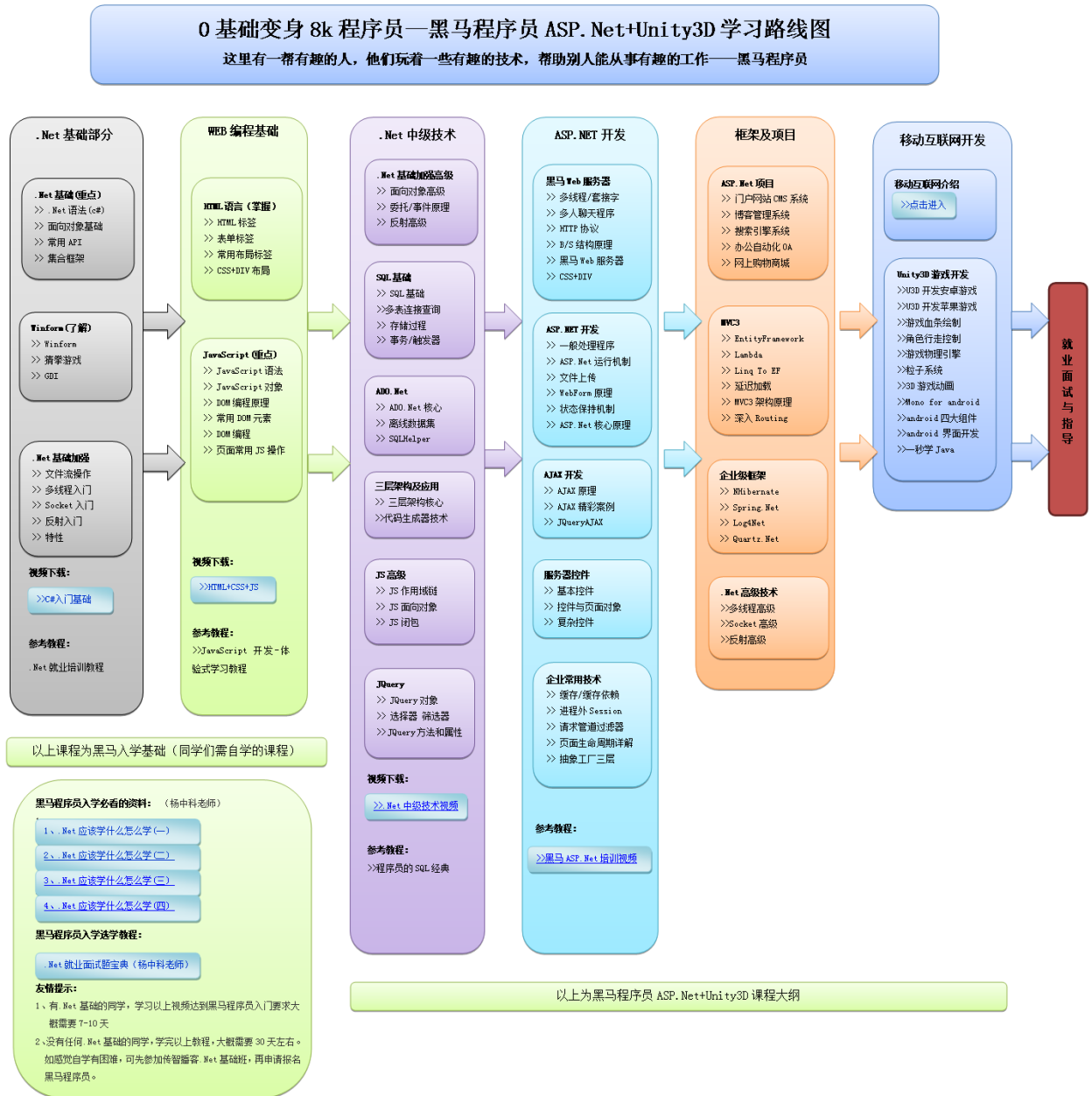
- >> Unity 编辑器的使用及注意事项
- >> Unity 开发游戏项目流程
- >> Native 2D 如何利用 Unity 做原生 2D 游戏开发
- >> 脚本引擎的使用及开发工具
- >> Unity 中的几种不同的坐标系以及使用场景
- >> 开发游戏中常见组成元素
- >> 物理引擎的使用方式与应用
- >> 碰撞检测和连续检测
- >> 游戏数据处理方式
- >> 游戏资源操作形式
- >> Shared SDK
- >> 多媒体在游戏中的应用
- >> 网络游戏的实现及通讯
- >> 2D 综合案例
- >> 3D 多人在线案例

第3阶段 Windows Phone

- >> WP 开发环境的搭建
- >> WP 模拟器的配置及使用方式
- >> XAML 语法和优势的介绍
- >> 框架和页面
- >> 常用控件与布局
- >> 通知中心 API 调用
- >> 外部存储操作 API
- >> 消息推送
- >> 蓝牙、近场通讯
- >> 传感器的操作形式
- >> 多任务操作
- >> 语音控制
- >> 多媒体操作
- >> 地理位置信息操作
- >> 网易新闻客户端
- >> 微博客户端
- >> 移动版在线商城客户端

第3章 .NET 学习步骤（黑马）

3.1 传智播客最新学习路线图



第4章 学了.Net 能做什么开发

如果你开始决定学习.Net 了，那么你的下一个问题很可能是“学了.Net 我能做什么？”。

.Net 目前主要的开发方向是：Web 开发、桌面系统开发、移动开发。

1) Web 开发通俗的说就是开发网站，包括类似于大众点评网等这样的互联网以及 OA 等内网系统，Web 开发是目前.Net 开发的主要方向，大部分招聘.Net 的职位都是做 Web 开发，以我教的传智播客.Net 班为例，2010 年几个就业班的学员 90%以上找到的都是 Web 开发方向的职位。

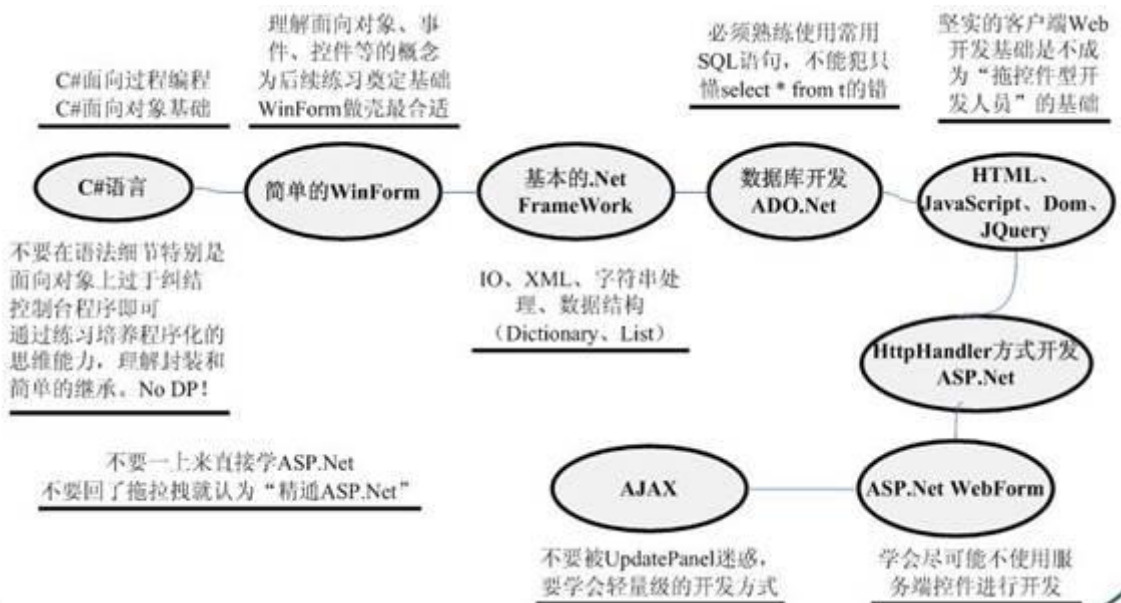
2) 桌面系统一般指的是使用 WinForm 技术开发 Windows 软件，比如以 Windows 软件形式运行的仓库管理系统、图书管理系统等，也就是一般说的 C/S 开发（不严谨的说法），目前很多公司都有 WinForm 开发的产品，不过由于大部分公司的产品都开始转向基于 Web 的版本，WinForm 版本都处于维护状态，因此基本不再招聘 WinForm 版本的新员工。

3) 移动开发主要指的是手机开发。现在移动开发最火的两个平台是 Android 和 IOS（Iphone、IPad），很多初学者误以为 Android、IOS 开发只能使用 Java、ObjectC 等语言进行开发，其实借助于 Mono 技术我们可以使用 C#/.Net 编写 Android、IOS 程序。由于 C#的高效开发和 Mono 的完美跨平台，这项技术目前已经得到了大量正式商业项目的应用，例如 Unity3D 就是一个市场占有率非常大的基于 Mono 技术的使用 C#进行跨平台游戏开发的技术。

第5章 .Net 应该学什么怎么学

5.1 （一）

目前.Net 开发主要方向是 Web 开发，因此在本篇中我将主要讲解做 Web 开发要学哪些东西，怎么学，由于内容比较多，我将分几次发。下图是我推荐的学习内容和学习顺序（除了这里列出的内容，要走向高手之路肯定还有更多技术需要学习，我这里只是对于初学者列出的主干学习内容，并不是全部）：



开发者口中的.Net 全称是.Net Framework，在中国大陆，一般将.Net 读作“道特耐特”（中文音标比英文音标更好读，呵呵）或者“点耐特”。.Net 是微软推出的一门技术，它是一个技术平台，基于这个平台进行开发可以简化很多工作。关于到底什么是.Net，不是一两句话能说清楚的，而且也没办法让初学者理解到底什么是.Net，因此你只要记住.Net 这个词，随着我们越学越深入，你就会理解.Net 到底是什么东西了。基于.Net 平台上有很多语言，比如 C#、VB.Net、IronPython、F#、Powershell 等，F#、Powershell 是满足特定用途的语言，IronPython、IronRuby 等并没有广泛应用，VB.Net 是仅次于 C#的第二大基于.Net 的语言，不过按照微软的战略是把主要精力放到 C#上，VB.Net 只是微软为了将 VB 开发人员平滑的转移到.Net 平台下的语言。因此对于语言来说大家学习 C#即可。C#在中国大陆的叫法一般是“C sharp(C 煞普)”，也有少数人读成“C 井”，我不推荐这种叫法。

初学者可能会弄不清 C#和.Net 的关系，.Net 是一个技术平台，而 C#是这个技术平台上的一门语言，就像 Windows 是一个操作系统，而 Word、QQ、FireFox 则是在 Windows 上运行的一个程序。初学者不必纠结于这个概念，不是很明白也没关系。

.Net 的开发工具是 VisualStudio，VisualStudio 现在有 VisualStudio2005、VisualStudio2008、VisualStudio2010 等版本，注意不要纠结于“到底装 VisualStudio2005 还是 VisualStudio2008，听说有的公司还在用 VisualStudio2005”、“到底是用新版本的 VisualStudio2010 还是用 VisualStudio2008”这样的问题，因为这些版本并没有本质的不同，使用方法都类似，就像你会用

WindowsXP 换了 Windows7 以后经过短暂的熟练会用 Windows7 了一样。因为 VisualStudio2010 比较消耗资源，如果你的计算机配置较好可以安装 VisualStudio2010，如果配置较差则推荐安装 VisualStudio2008。

要学习 Web 开发一定要按照正确的顺序进行学习，很多人被一些《21 天精通***》这样的垃圾书的影响，不学 C#、不学 .Net Framework、不学 Dom，直接上来就学 ASP.Net，拖两个控件、配两个属性、双击控件写一个事件响应方法，就开始自诩“精通 ASP.Net”了，照着书控件拖来拖去，可是连 C#、.Net、ASP.Net、HTML、JavaScript 之前的关系是什么都不清楚，闹出了很多笑话。欲速则不达，速成的结果就是以后浪费更多的时间，回想这么多年你走过的路，有多少次为了走快点就跳着走，最后又浪费时间返工的事情，按照正确的顺序才能更快、更好的掌握 .Net。

有的人可能认为“我不想成为高手，有一口饭吃就行了，会拖控件就行了呗”，千万不要以为会“不拖控件的开发”就是高手了，高手没有那么好当的。我没有奢求学编程的人个个都是以“成为世界顶尖级程序员”、“国内 .Net 开发者中的知名开发者”为目标，这个也不现实，大部分人学编程只是看做软件开发工资不低，想赚几年钱娶媳妇生娃，这种想法并不是多没出息的想法，毕竟“老婆孩子热炕头”是最朴实的幸福，你不需要背负“振兴中国软件产业”这种的重任，但是最起码应该找到一份不错的工作，工作几年以后能有钱交得起房子首付（对于大部分人来讲，能够交得起首付成为房奴都是很奢侈的事情）、每月供月供吧，如果只会拖控件的话就不是能不能成为高手的问题，而是能不能养家糊口的问题了。

下面来详细讲解学习的各个步骤：

一、C#面向过程编程

学习 C#语言分为 C#面向过程编程和 C#面向对象基础两部分。听到“C#面向过程编程”有人会问了“C#不是一门面向对象的语言吗，哪儿来的面向过程编程？”，其实面向对象不是代替面向过程的，面向对象是面向过程的一个补充和发展，面向过程就是代码逐行执行，面向过程就是 if/else，面向过程就是 for 循环，面向过程就是变量、函数，C#中最基础的也是 if、for、变量、函数等这些东西，这些东西就是 C#的面向过程部分。对于编程的初学者来说，最难的不是类、继承、控件、ASP.Net、AJAX 之类神乎其神的东西，而是“程序化的思维方式”，也就是怎么把一个问题的解决方式用计算机的思维方式描述。

我接到过很多想深入学习 .Net 学生的咨询，当我问到“你的 .Net 基础怎么样”的时候，很多人都回答“学过 C#、.Net、ASP.Net，感觉基础还行”，但是我给了他一个很简单的题，比如“用 for 循环累加的方式计算 1 到 100 之间整数的和”、“提示用户输入两个数字，然后程序打印两个数字的和”，很多人都做不出来。

对于初学者来说先不用管面向对象、ASP.Net 之类的东西迷惑，先学习用最朴实的 for、if 来编写程序，通过亲手编写练习题代码来逐渐形成程序化的思维方式，这个阶段也不要使用“看起来很酷”的 WinForm 或者 ASP.Net 页面来做，因为那样就会涉及到这个阶段不需要去了解的问题，先编写最简单的控制台程序就可以。推荐大家看我录制的视频教程《【传智播客.Net 视频】第 1 季 C#编程基础》，这个教程针对 C#零基础的同学进行讲解，并且在课程中给出了非常多的练习，一定要自己亲手把代码熟练的写出来，记住是亲手写出来，而不是照着我的代码抄，更不能是看懂了我的代码就以为自己掌握了，要知道“看懂了”和“会写了”之间差着十万八千里呢。对于初学者来说一开始写那些练习可能有点困难，先理解我写的代码，然后自己总结成“先***后***，再***，如果***则***”这

样的用语言描述的算法，然后照着算法描述自己写代码，发现有地方写起来遇到问题的话再参考我的代码，最后做到不用看我的代码、不用看算法描述也能把代码写出来，这样第一阶段的任务就完成了。下面列出我视频教程中的一些练习题，大家可以自测看这个阶段自己是否合格：

练习 1：提示用户输入用户名，然后再提示输入密码，如果用户名是“admin”并且密码是“888888”，则提示正确，否则提示错误，如果用户名不是 admin 还提示用户用户名不存在。

练习 2：提示用户输入年龄，如果大于等于 18，则告知用户可以查看，如果小于 10 岁，则告知不允许查看，如果大于等于 10 岁，则提示用户是否继续查看（yes、no），如果输入的是 yes 则提示用户可以查看，否则提示不可以查看。（给出测试用例。）

练习 3：从一个整数数组中取出最大的整数

练习 4：计算一个整数数组的所有元素的和。

练习 5：将一个字符串数组输出为分割的形式，比如{“浮云”，“神马”，“穿梭”}数组输出为“浮云|神马|穿梭”。不要使用 String.Join 等 .Net 内置方法。

练习 6：有一个整数数组，请声明一个字符串数组，将整数数组中的每一个元素的值转换为字符串保存到字符串数组中。

练习 7：将一个字符串数组的元素的顺序进行反转。{"3","a","8","haha"} 转换为 {"haha","8","a","3"}。提示：第 i 个和第 length-i-1 个进行交换。

学 C# 基础的时候的参考书推荐如下两本：

（1）《C#图解教程》，这本书讲的清晰易懂，不过部分内容安排顺序不合理，大家按照我视频中的顺序进行学习就可以。

（2）《C#入门经典》适合零基础的大部头，不过非常厚，采用的是老外的描述方式，中国人读起来有点别扭，而且中文版翻译的不是太好。

5.2 （二）

二、C#面向对象基础

初学者学面向对象的时候没必要（也做不到）把面向对象学的非常透、非常深，因为如果想深刻的理解面向对象，必须要有大量的编程经验才能真正体会面向对象到底是怎么回事、有什么好处。以前听到过有人建议初学者学习设计模式，我认为纯属误人子弟，**对于初学者来说没有大量的编程经验，根本看不懂设计模式在说什么**。这个阶段只要弄明白怎么声明类、什么是访问级别（主要是 public、private）、什么是属性、字段、方法、什么是继承、什么是 override 就可以。面向对象有三个特性：封装、继承、多态。对于初学者来说理解多态是非常困难的也是没太大必要的，这个阶段只要能掌握封装和继承就可以。

三、简单的 WinForm 编程

WinForm 是 .Net 中用来构建 Windows 窗体应用程序的技术，对于目前**国内市场来讲一般**也就是用来桌面版管理系统等的技术。有人会问“我们不是学 B/S 的 ASP.Net 开发吗？为什么还要学 C/S 的 WinForm？”，一定要注意我在这段的标题写的是“简单的”WinForm。和 ASP.Net 比较起来，学 WinForm 需要学习的前置知识非常少，可以非常快的入手开发，而学习 ASP.Net 则需要先学习 HTML、JavaScript、Dom 等知识。**这个阶段的学习重点不是 WinForm 本身，而是通过 WinForm**

在案例中理解类、对象、属性、方法、事件、索引器等概念，我们是“借壳学习”，因此不用对 WinForm 的深入技术（比如控件美化、布局、PInvoke、托盘编程、高级控件、ActiveX 等）浪费时间去研究。初学者很难把握 WinForm 学习的度，还是推荐大家参考看我录制的视频教程《[【传智播客.Net 视频】第 3 季 WinForm 基础](#)》。

四、常用 .Net Framework 类

.Net Framework 中的类是 .Net 中最丰富、最有价值的内容，.Net Framework 中的类是和表现层无关的，也就是无论是在控制台中、WinForm 中还是 ASP.Net 中都可以调用这些类，比如把一个字符串转换为整数都是使用 Convert.ToInt32() 这个方法，因此千万不能因为你是 在控制台程序中写这些类就认为自己在“编写 DOS 程序”。

.Net Framework 是非常庞大的，拥有上万个类，没必要把这几万个类掌握，在这个阶段把常用的 .Net Framework 类掌握了，会对后续阶段的学习非常有帮助。这些类包括：String 类常用方法、StringBuilder 类、IO 类（FileStream、File、Path）、XML 操作类、集合类（ArrayList、List<T>、Dictionary<K,V>）等。

五、SQL 语句

这个阶段主要学习 SQL 语法，能够掌握常见的 SQL 语法。数据库管理系统有 MSSQLServer、Oracle、MySQL、DB2、Access 等，这些不同的系统对于数据库管理员来说使用方法各异，但是对于开发人员来说差不多，因为都是用的统一的操作语言“SQL”，通过 SQL 这门通用的语言，你可以和不同的数据库进行沟通，所以只要掌握了 .Net 开发最常用的 MSSQLServer，以后根据工作的需要使用其他的数据库管理系统会非常简单。

现在企业招聘 .Net 开发人员的时候对于数据库考查非常多，甚至很多公司的笔试题中有 1/3 都是关于 SQL 语句、存储过程的，因此对于准备学习后找工作的同学来说必须把这部分内容学好。

这部分内容主要学习：MSSQLServer 的简单管理，Select、Insert、Update、Delete 语句，学完了这些内容就可以继续学习后续的 ADO.Net 了。学有余力的同学可以学习 Join、Union、子查询、存储过程等高级的内容。

这部分内容学习时大家可以参考我的《程序员的 SQL 金典》这本书（我将近期发布这本书的全部免费电子版）和《[【传智播客.Net 视频】第 4 季 SQL 从入门到提高](#)》。这套视频教程。

六、ADO.Net

ADO.Net 是 .Net 中用来访问数据库的技术，通过 ADO.Net 我们就能通过程序告诉数据库“请把年龄大于 20 岁的人员返回给我”。在学习 ADO.Net 之前一定要把常用的 SQL 语句掌握了，千万不能学了一句 Select * from Persons 就来学 ADO.Net，否则可能会出现下面这个真实的故事中的笑话了：

我以前碰到过一个人写一个“查询学号为 S001 的学生的姓名”的功能，他这么写程序（示例性代码，不严谨）：

```
SqlCommand cmd = conn.CreateCommand();  
  
cmd.CommandText = "Select * from Persons";  
  
SqlDataReader reader = cmd.ExecuteReader();  
  
while(reader.Read())
```



```
{  
    String no = (string)reader[ "No"];  
    if(no==" S001")  
    {  
        string name = (string)reader[ "Name"];  
        MessageBox.Show(name);  
    }  
}
```

看到这种写法相信所有人都要崩溃了，因为好像从来没学过 where 语句一样。因此必须在把常用的 SQL 语句掌握以后再学 ADO.Net。

这阶段的学习主要学习使用 SqlConnection、SqlCommand (ExecuteNonQuery、ExecuteReader、ExecuteScalar) 等类的使用、参数化查询等，掌握 DataSet 方式使用离线数据集，能够自己从头写一个 SqlHelper 类出来即可。

(注：这段我描述的是 ADO.Net 基于接口的编程，属于深入一点的知识，对于初学者来说如果看不懂也没关系，不用纠结于此) SqlConnection、SqlCommand 等类是 ADO.Net 中用来连接 MSSQLServer 的类，连接 Oracle 等数据库则使用 OracleConnection、OracleCommand 等类，虽然看似不同的类，但是由于 ADO.Net 通过 IDbConnection、IDbCommand 等接口统一了类的方法，所以 OracleConnection、OracleCommand 等类的用法和 SqlConnection、SqlCommand 基本一样，而且如果通过 DbProviderFactories 进行基于接口的 ADO.Net 编程，那么编程的时候就是使用 IDbConnection、IDbCommand 接口，完全不用管 SqlConnection、OracleConnection 等这些不同的类。学有余力的同学可以深入研究一下 ADO.Net 的接口编程，会对你理解设计模式有很大的帮助。

5.3 (三)

七、HTML、JavaScript、Dom

HTML 是对网页长什么样子的一个描述，也就是“这里显示一个超链接、那里显示一个按钮”都是通过 HTML 来进行描述的。HTML 做出来的页面是静态、不变化，如果来实现“点击【计算】按钮的时候让表格放大显示”这样的动态效果，则需要通过程序操作 Dom，一般使用 JavaScript 这种脚本语言来操作 Dom。

如果不想成为只会拖控件的开发人员的话，需要在学习 ASP.Net 之前把 HTML、JavaScript、Dom 这些好好学习。提到这一点有同学可能会提出异议“不是这样的呀，ASP.Net 已经封装的很好了，拖几个控件上来就可以了，ASP.Net 都帮我们生成 HTML 了，也不用学 JavaScript，动态效果写 C#代码就可以了”。

ASP.Net 中这种拖 ASP.Net 控件的开发技术就叫做 WebForm，微软的 WebForm 封装是非常伟大的技术，能够将复杂的 Web 开发简化到可以像开发 WinForm 程序一样点击，不得不佩服微软技术的强悍。使用 WebForm 我们就可以进行傻瓜化的开发，要是每天的工作、做每个项目的时候都

能一直这样“傻瓜”下去也是很美好的，但是很不幸有一些场合中是不能使用这种傻瓜化的开发方式的，举几个例子：

例子 1：在网站的注册页面中让用户所在的省、市，提供两个下拉列表框（DropDownList），一个为省的列表框，一个为市的列表框，当用户选择不同的省的时候，市列表框中就列出这个省下的所有市供用户选择。

对于这样一个功能点，如果只会拖控件的人很可能会这么写（假设省列表框 id 为 ddlProvince，市列表框 id 为 ddlCity），在 Page_Load 中加载省列表到 ddlProvince，然后在 ddlProvince 的 SelectedIndexChanged 事件中响应 ddlProvince 的选择变化，当 ddlProvince 选择变化后根据选中项来填充 ddlCity。用这种方法实现最大的缺点就是页面会频繁的刷新，每次选择省以后页面都会唰一下的刷新一次，这在网速比较慢或者服务器压力比较大的时候是很慢的，而且用户体验很差。这种情况用 JavaScript 操作 Dom 的方法来操作才会比较好。

例子 2：ASP.Net WebForm 中可以使用 UpdatePanel 控件很轻松的实现 AJAX 效果，不用懂 JavaScript、Dom、Json，把控件用 UpdatePanel 包裹起来 AJAX 效果就出来了。但是用 UpdatePanel 实现 AJAX 只能在对性能要求不高的场合使用，如果要开发互联网应用等对性能要求非常高的场合，UpdatePanel 控件是没法使用的。这时候就需要手写实现 AJAX，而手写 AJAX 则需要对 JavaScript、Dom 非常熟悉。

即使全部使用 ASP.Net WebForm 控件实现，也是需要对 HTML 熟悉才可以的，比如不可能项目中使用 ListView 控件默认生成的代码，都需要手动调整 ListView 的模板，而调整模板就需要手写 HTML，**不是什么都“拖一拖”就行的**。微软现在还推出了一个 WebForm 的兄弟产品：ASP.Net MVC，ASP.Net MVC 并不是替代 WebForm 的，各有所长，但是因为 ASP.Net MVC 的优点，以后肯定会有越来越多的项目使用 ASP.Net MVC。在 **ASP.Net MVC 中就没有 WebForm 中这么傻瓜化了，很多地方需要手写 HTML 代码，这就对 HTML 等要求更高。**

上面讲了 HTML、JavaScript、Dom 的重要性，下面来讲一下具体要学哪些东西、怎么学。需要提前说明：Web 开发中一个非常麻烦的一点就是 HTML、Dom 在不同浏览器有少量的差异性，也就是一段代码可能在 IE 中和 FireFox 中的效果不一样，**对于初学者来说暂时不需要管那些差异性，降低学习中不必要的难度，学习的时候以一个浏览器为主就可以了。**

刚接触 HTML 的人可能感觉比较难，但是一旦熟悉了这种标签式的用法，HTML 就变得非常简单了，HTML 中标签比较多，标签的可用属性更多，做为一个初学者没必要把所有标签、属性都学习了，而且也不可能都掌握，把最常用的标签以及它们的常用属性掌握了即可，主要的标签有：p、br、div、font、a、input、img、li、table、select、textarea、span。**学习 HTML 的时候一定要坚持手写 HTML，千万不要用 VisualStudio、Dreamweaver 等中的可视化的拖放方式来编写 HTML 页面**，如果只会在设计器可视化拖出 HTML 页面的话，后面的学习会困难重重。HTML 学习的一个目标就是能够完全手写一个很多网站都有的那种静态注册页面出来。

HTML 中还有一个知识点：CSS，CSS 是用来对页面进行美化的描述语言，通过 CSS 我们能够将按钮设置为红色或者给一个图片加上金色边框。CSS 中有非常多的样式名，不需要掌握这么多样式名，学习的时候只要过一遍，大概知道每个样式名是做什么的就可以；也不需要学习怎么将页面做的看好，不需要研究那些好看的 CSS 效果到底是什么做出来的，因为那些工作是美工做的事情，即使是没有美工的小公司开发人员只要学会怎么“偷”别的网站做好的特效就可以了。初学者只要掌握几种 CSS 选

择器和 Div+CSS 布局就可以。

和 ASP.Net 后台用 C# 写代码不一样，对于 Web 前台页面的动态效果一般使用 JavaScript（不用学已经在 Web 前端中被淘汰的 VBScript）来编写，因此需要首先掌握 JavaScript 语言，JavaScript 和 C# 的语法一样都是 C 语言发展过来，很多语法都和 C# 差不多，因此不用再从头学 JavaScript 语法，只要学习 JavaScript 和 C# 相比特有的语法就可以。JavaScript 语言也有很多可以很深入研究的东西，比如闭包、面向对象的实现等，但是除非做 Web 游戏开发、开发 JavaScript 库等对 JavaScript 要求非常高的领域，如果是开发互联网站、业务系统等（这占到公司项目中的大部分）对 JavaScript 并没有太高的要求，只要熟悉一下 JavaScript 的语法就可以，然后在后面的 Dom 中通过练习逐步熟悉 JavaScript。对于初学者来说 JavaScript 到底学到什么程度是够用了是很难界定了，大家可以参考《【传智播客.Net 视频】第 7 季 JavaScript》这套视频教程，然后学 Dom 的时候参考《【传智播客.Net 视频】第 8 季 Dom》进一步练习就可以。

很多初学者刚接触 Dom 的时候总是想弄明白“到底什么是 Dom”，劝大家没必要纠结于这些概念，只要知道它是干什么的就可以，“通过 JavaScript 调用 Dom 中的属性、方法就可以动态的操作 HTML 页面”、“getElementById 就是 Dom、btn1.value='abc' 就是 dom、btn1.onclick=myclick 就是 dom”，通过练习你就会明白这一点就可以了，咱们不是编教科书的，不用搞那些无聊的概念定义。

Dom 的东西非常多，还是那个原则，不用什么都记住，主要掌握下面几点：事件、动态设置事件、setInterval 定时器、event 对象、getElementById、动态创建 Dom 元素（createElement、innerHTML、innerText）。掌握这些东西还是要通过大量的练习来掌握的，只要能把第 8 季 Dom 这套视频中的练习熟练写出来这几个阶段的目标就达到了。

JQuery 目前是最火的一个 JavaScript 的库，千万不要以为是一个新的语言，听到过很多“jQuery 是 JavaScript 的终结者，jQuery 取代了 JavaScript”的说法，殊不知，jQuery 只是个 JavaScript 的一个封装库，**尽快 jQuery 语法看起来非常奇怪，但是 jQuery 确实是完全符合 JavaScript 语法的一个封装库。**jQuery 只是对 Dom 操作的一个封装，学之前需要 Dom 基础，否则学起来会很晕。我不建议初学者学完了 Dom 以后立即学 jQuery，还是用 Dom 多写一些程序以后再来学 jQuery 才会掌握的又牢靠又快速，我讲课的时候就在讲完了 Dom 以后隔了 8 天的 .Net 高级技术才讲 jQuery 也是基于这个考虑。参考书可以看《锋利的 JQuery》。

5.4 （四）

八、ASP.Net

ASP.Net 是 .Net 中用来开发动态 Web 应用程序的技术，ASP.Net 主要分 ASP.Net WebForm（以后简称 WebForm）和 ASP.Net MVC 两种技术，大部分人说 ASP.Net 都是指的 ASP.Net WebForm，而 ASP.Net MVC 则是微软最近推出的一门技术，前面已经提到过了 ASP.Net MVC 并不是用来代替 ASP.Net WebForm 的，因此不要认为“ASP.Net MVC 淘汰了 WebForm，学 WebForm 过时了”，对于初学者来说还是先学习 WebForm，等 WebForm 掌握的比较好以后根据自己的情况学习 ASP.Net MVC，现在很多公司招聘的时候都提出了“掌握 ASP.Net MVC 优先考虑”。下面我们还是主要来谈 WebForm 的学习。

很多人被那些“短平快” ASP.Net 的书误导了，开始学习就从 Button 控件、TextBox 控件、Repeater 控件、GridView 控件开始学习，为庞大的 ASP.Net 体系所折服，看着千八百页的书感叹“晕呢，这么多控件、什么多方法、这么多属性哪辈子学完呀”。当然也有人会拖几个控件以后就自诩“精通 ASP.Net”，“拖控件、设属性、写事件”靠此三招走天下，但是一旦遇到稍微复杂一点问题就懵了。

WebForm 是微软的一个伟大的创造，能够将复杂的 Web 开发进行了屏蔽，能够让我们进行傻瓜化的开发，但是微软再怎么封装也不可能让我们一直“快乐的傻下去”，不了解 Http、不了解“请求—处理—响应”的模型、不了解 ASP.Net 的内部原理，那么遇到一些稍微复杂一点的问题的话就会感觉很茫然。一旦了解了 Http、ASP.Net 的原理以后再去学 ASP.Net 你就能知道那些控件到底帮我们做了什么，这样就可以轻松的掌握那些控件，不是成为控件的奴隶，而是成为控件的主人，让控件为我们所用。正是基于这样的原则，我在传智播客.net 班讲 ASP.Net 基础的时候，我没有立即讲解那些控件怎么用，而是使用了 4 天的时间来讲 Http 协议、Web 开发基本原则、“请求—处理—响应”模型、基于 HttpHandler 开发 Web 应用、ViewState 原理、IsPostBack 原理、XSS 漏洞等基础的东西，前 4 天的内容是难度非常高的东西，一旦把这 4 天的内容掌握了，后面用几天时间学员们就能掌握 ASP.Net 的基础的东西，而且掌握的非常深入、牢固，如果对这个感兴趣可以参考我的《【传智播客.Net 视频】ASP.Net》上中下三季内容。

特别是以后会在 ASP.Net 开发中占有一席之地的 ASP.Net MVC 更是摒弃了 ASP.Net WebForm 中复杂的事件驱动、属性等非 Web 本源的东西，ASP.Net MVC 回归了 Web 开发的本质，把 Web 开发重新用“请求—处理—响应”这样一个最简单的模型进行描述，其实是比 ASP.Net WebForm 更简单，但是如果只会用 ASP.net WebForm 进行拖控件开发，那么如果需要用到 ASP.Net MVC 就会非常不适应。

学习 WebForm 的时候一定要先掌握不使用 WebForm 进行 Web 开发的技术，因为有的项目中如果完全使用 WebForm 中的服务端进行开发就会有性能等问题，因此这些项目中会不允许使用服务端控件或者不允许使用服务端控件的某些特性。服务端控件必须学，因为很多功能点中用服务端控件会“多快好省”，但是不要离了服务端控件就活不了。

WebForm 中并不是所有控件都是常用的，初学者的目的是用最短的时间学最有用的内容。工具箱中“标准”中常用的控件如下：Label、Button、LinkButton、ImageButton、HyperLink、DropDownList、ListBox、CheckBox、RadioButton、Literal、FileUpload。很多书上讲 WebForm 的时候讲把大量的篇幅放到 GridView、DataList、DetailsView、Repeater、ListView 等这些数据绑定控件上，其实这些控件大同小异，只要掌握了一个控件另外一个控件很容易掌握，推荐大家学习 ListView 和 Repeater 就够了。

上面我谈了学习 .Net Web 开发的的八步，这八步看似漫长，其实如果能把所有精力投入上面的话大约两个月就能学下来。当然学完了这些只是掌握了 .Net 开发的基础知识，还需要在项目开发中将这些基础知识进一步综合应用。