



- **单元1 - ASP.NET介绍及开发环境：**提供一些有关ASP.NET的相关常识，使得学员能较为了解ASP.NET的技术概况。进一步对微软的开发工具有些了解并进行选择及安装设置。完成后，开始我们的第一个Hello Web！项目，以了解Web Forms项目的来龙去脉。最后提供了一些开发资料供大家参考。



1. ASP.NET开发技术简介

LearnASP.NET-rev1.1 <http://www.coding24.org>

- **ASP.NET开发技术简介：**内容包含了：Web开发技术、ASP.NET开发技术、ASP.NET技术的演进、ASP.NET Core与ASP.NET 4.X的比较、以及ASP.NET创建Web应用的3种方式。



Web开发技术简介

- **前端技术：** 如视觉设计、平面设计、**HTML**、**CSS**、JavaScript及其上的一些前端框架，如**Bootstrap**等，与编程相关较低。脚本语言由浏览器运行，一般也可称为静态网页技术。
- **后端技术：** 包含了在Web服务器编写应用程序的相关技术，如：**ASP.NET (C#)**、JSP/Servlet (Java)、PHP及一些后端框架或模式，如MVC (Model-View-Controller)等。与编程有关，代码由Web服务器运行，被称简单称为动态网页技术。
- **数据库技术：** 如**SQL Server** / Oracle / MySQL数据库管理系统，或**ORM技术**（如微软的**Entity Framework** / LINQtoSQL）。

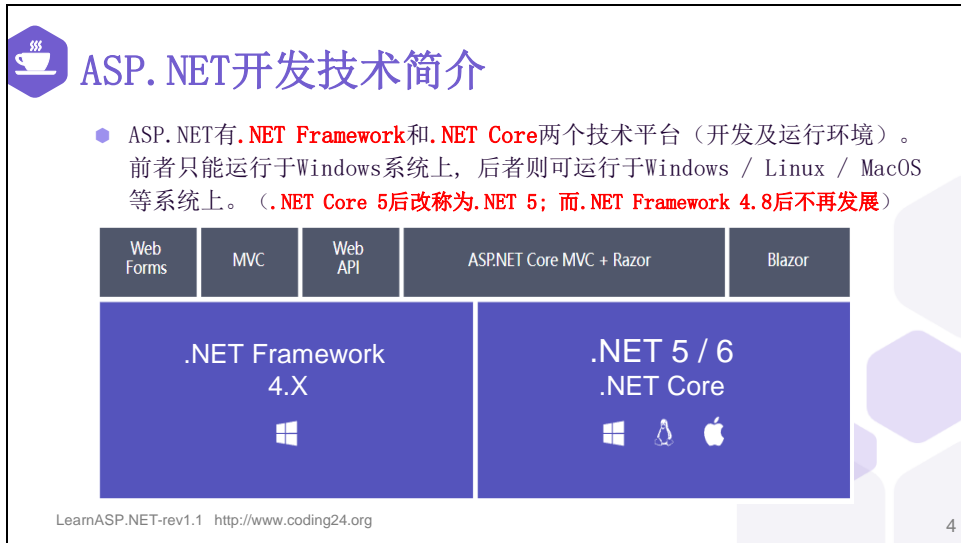
LearnASP.NET-rev1.1 <http://www.coding24.org>

3

Web开发技术简介

- 动静之分：
 - 静态页面：只包含HTML元素和CSS样式，有时也包含了Javascript或jQuery等前端脚本，一般以扩展名.htm或.html存储
 - 动态页面：可以包含HTML元素和CSS样式及前述各项静态页面有的技术，还可以包含一些服需要在Web服务器端编译执行的代码，如微软的C#，Visual Basic等，在Java Web里面自然就是JSP/Servlet（Java代码）。
- 前后之分：
 - 前端技术： 如视觉设计、平面设计及HTML、CSS、JavaScript及其上的一些前端框架，如jQuery、Bootstrap、Node.js、Angular，React、Vue等，与编程相关较低（当然这里年来前端愈发多元了，这点愈来愈模糊了），脚本语言由浏览器运行，一般也可称为静态网页技术。
 - 后端技术： 包含了在Web服务器编写应用程序的相关技术，如：ASP.NET（C#，VB）、JSP/Servlet（Java）、PHP、Ruby、Python等，以及一些后端框架或模式，如MVC（Model-View-Controller）、SSH（Struts-Spring-Hibernate）、SSM（Spring-SpringMVC-MyBatis）等。与编程有关，代码由Web服务器运行，有时也被称简单称为动态网页技术。
 - 全栈（全端）开发：前后端都具备的工程师在当下的流行语言称：全栈工程师。当然，登高必自卑，行远必自迩，还是要步步踏实去学去做，否则也意味著都不熟练的意思。
- 数据库技术： 如SQL Server / Oracle / MySQL数据库管理系统，或ORM技术（如微软

的Entity Framework / LINQtoSQL），以及数据库的管理、备份、复原以及效能调校等技术。有时这项技术也被当下称为后端技术的一项。



The diagram illustrates the ASP.NET development technology overview. At the top, a purple hexagon icon with a coffee cup is next to the title "ASP.NET开发技术简介". Below the title, a bullet point states: "ASP.NET有.NET Framework和.NET Core两个技术平台（开发及运行环境）。前者只能运行于Windows系统上，后者则可运行于Windows / Linux / MacOS等系统上。（.NET Core 5后改称为.NET 5；而.NET Framework 4.8后不再发展）". Below this text is a table with five columns: "Web Forms", "MVC", "Web API", "ASP.NET Core MVC + Razor", and "Blazor". Under "Web Forms" and "MVC" is a blue box labeled ".NET Framework 4.X" with a Windows logo. Under "Web API", "ASP.NET Core MVC + Razor", and "Blazor" is a blue box labeled ".NET 5 / 6 .NET Core" with logos for Windows, Linux, and macOS. At the bottom left of the diagram is the text "LearnASP.NET-rev1.1 <http://www.coding24.org>" and at the bottom right is the number "4".

ASP.NET开发技术简介

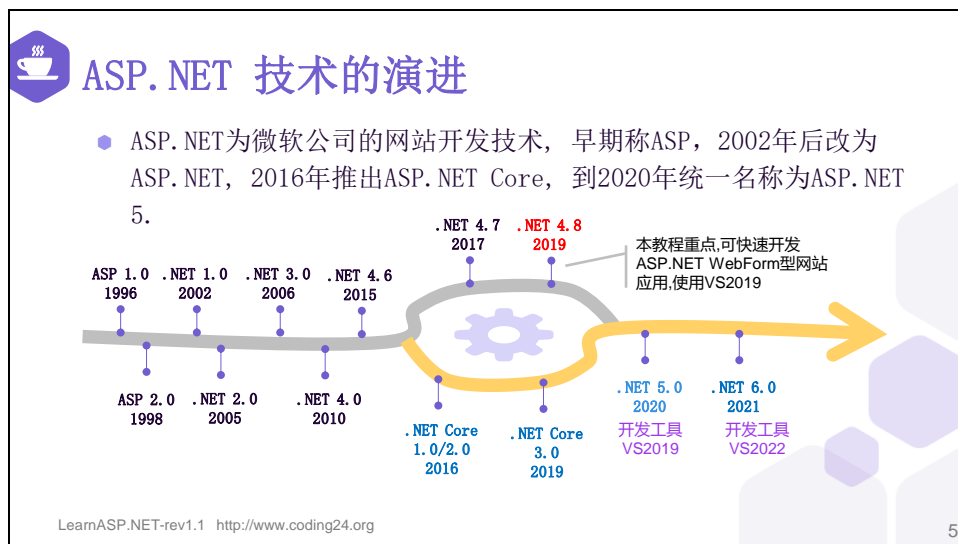
- ASP.NET有.NET Framework和.NET Core两个技术平台（开发及运行环境）。前者只能运行于Windows系统上，后者则可运行于Windows / Linux / MacOS等系统上。（.NET Core 5后改称为.NET 5；而.NET Framework 4.8后不再发展）

Web Forms	MVC	Web API	ASP.NET Core MVC + Razor	Blazor
.NET Framework 4.X		.NET 5 / 6 .NET Core		

LearnASP.NET-rev1.1 <http://www.coding24.org> 4

ASP.NET开发技术简介：

- .NET、ASP.NET及编程语言C#三者关系：
 - .NET是微软公司的一个开发平台，有.NET Framework及.NET Core两者。
 - .NET是一个应用程序开发和运行的环境，在这平台上可开发Windows视窗应用、Web应用、手机应用及其它类型的应用（如Web Service）。
 - 而ASP.NET就是.NET平台上的一个组件（模块），专用来开发Web应用的。
 - 微软的.NET平台可用多种语言进行应用开发，例如我们在开发ASP.NET Web Forms应用时可选择C#或是VB（Visual Basic）编程语言。
- .NET Framework和.NET Core是两个微软的技术平台（开发及运行环境）
 - .NET Core（第5版后称.NET 5，自然也没.NET Core 6，而称.NET 6）是一个新的开源并且跨平台的用来构建可以所有操作系统（包括Windows, Mac, and Linux）上面运行的应用的的框架。
 - .NET Framework 4.8后不再推出新的版本。.NET Framework 有两种安装方式：随着Visual Studio工具一起安装（常见方式），另一種是单独安装。



ASP.NET 技术的演进：

- 1996 年微软公司提出了 ASP 1.0（Active Server Pages）技术，为一种新的 Web 开发技术，降低了动态网页开发的难度。而 ASP 使用简单的脚本语言 JavaScript 或 VBScript，能够将代码直接嵌入 HTML，使设计 Web 页面变得更简单。
- 2002 年再提出 ASP.NET 技术，最初的名字为 ASP+，后来改为 ASP.NET。ASP.NET 是一个建立在 .NET 之上的 Web 运行环境，它不再使用脚本语言，而是使用 C# 或 VB 语言进行开发；著名的 WebForm 项目开发具有传承 WinForm 开发的优势，同时具后置代码特性（Code-Behind），让网页设计与编程能够分开，不再混在一个文档中，是一个新的里程。
- 在 2016 年为跨平台并轻量化而提出了 ASP.NET Core（第 5 版后称 ASP.NET 5）放弃了 WebForm 的开发模式，前端采用了 Razor 语法（一种新的前端脚本语言），又回到过去将 HTML 与代码混合的时代。
- .NET Framework 目前的版本：

<https://dotnet.microsoft.com/zh-cn/download/dotnet-framework>


版本	发布日期	终止支持
.NET Framework 4.8 (推荐)	2019 年 4 月 18 日	
.NET Framework 4.7.2	2018 年 4 月 30 日	
.NET Framework 4.7.1	2017 年 10 月 17 日	

- LTS 版本（长期支持版本）：.NET 6 是自 2019 年 12 月发布的 .NET Core 3.1 以

来的第一个 LTS 版本。LTS 版本计划每两年发布一次。技术更新时要稍加注意。
在2022年2月，.NET 7预览版也释出了。

<https://dotnet.microsoft.com/zh-cn/download/dotnet>

版本	状态	最新发布	最新发布日期	终止支持
.NET 7.0	预览 ⓘ	7.0.0-preview.1	2022年2月17日	
.NET 6.0 (推荐)	LTS ⓘ	6.0.2	2022年2月8日	十一月 08, 2024
.NET 5.0	当前 ⓘ	5.0.14	2022年2月8日	五月 08, 2022
.NET Core 3.1	LTS ⓘ	3.1.22	2021年12月14日	十二月 03, 2022

 ASP.NET Core与ASP.NET 4.X的比较		
技术平台	ASP.NET Core (ASP.NET 5 / 6)	ASP.NET 4.X (最终版4.8)
操作系统	适用于Windows、macOS 或 Linux	适用于Windows
使用技术	Razor (建立UI)、MVC、Web API、SignalR	Web Form、SignalR、MVC、Web API、Webhook或Web Pages
编程语言	C#、F	C#、VB、F
开发工具	Visual Studio、Visual Studio Code	Visual Studio
运行环境	.NET Core Runtime	.NET Framework Runtime


资料来源: 微软MSDN

LearnASP.NET-rev1.1 <http://www.coding24.org>

6

ASP.NET Core与ASP.NET 4.X的比较:

- ASP.NET Core是 ASP.NET 4.x 的重新设计，.NET Core自第5版开始称为.NET 5，所以在网站开发组件称为ASP.NET 5或ASP.NET 6。
- ASP.NET Core组件是一个跨平台的开源框架，可将开发完成的Web项目布署（发布）在 Windows、macOS 或 Linux平台上。
- ASP.NET 4.x 是一个成熟的框架，提供在 Windows 开发及布署Web 应用程序。
- 两者都可以使用C#作为其开发语言（后端的开发语言），但其前端却有不同：在ASP.NET是采用服务器控件，而ASP.NET Core其前端是采用Razor+HTML的网页。



ASP.NET 创建 Web 应用的3种方式

模式	开发平台	开发方式
Web Forms 模式 (Web窗体)	适用于.NET Framework平台	在原有HTML标签中加上服务器控件，且可采用后置代码方式（可选C#或VB），适用于快速开发
MVC模式	适用于.NET Framework及.NET Core平台	将网站应用划分为MVC三模块，其中的View采用Razor语法混编于HTML标签中，Model及Controller则采用C#编程语言开发。
Razor Page 模式 (Web Page)	适用于.NET Core平台 (或.NET5后)	Razor Page模式可视为简化版的MVC，一个Page包含了两个部份，一个是.cshtml网页文件，以HTML及Razor语法完成；另一个是.cs文件，主要用于处理业务逻辑。

资料来源: 微软MSDN

LearnASP.NET-rev1.1 <http://www.coding24.org>

7

ASP.NET 创建 Web 应用的3种方式:

- **Web Form模式(Web窗体):** 在原有HTML标签中加上服务器控件，且可采用后置代码方式（可选C#或VB），适用于快速开发。
- **MVC模式:** 将Web项目划分为MVC三模块进行开发，其中的View采用Razor语法混编于HTML标签中，Model及Controller则采用C#编程语言开发。不管在.NET Framework或.NET Core平台均有MVC的开发模式，当然若要开发MVC Web项目最好在.NET Core平台是较好的选择，因它可布署于不同的平台上。
- **Razor Page模式(Web Page):** Razor Page项目与JSP或PHP或者1996年的ASP技术较为相似。可视为简化版的MVC，一个Page包含了两个部份，一个是.cshtml网页文件，以HTML及Razor语法完成；另一个是.cs文件，主要用于处理业务逻辑。Razor其实有点像是Web Form技术，也有自己的.cs，但相较于Web Form可能较轻量，能应付大型网站开发，但也少了Web Form的方便性。



ASP.NET WebForm/Core如何选择

- **ASP.NET Web Forms技术的优势:**
 - 接轨Window Forms的开发经验 / 可快速开发并部署一个 .NET Web 应用 / 维护并升级现有的 .NET Web应用程序 / 为了找一份工作? 目前.NET Core可能没很多 / 非软件专业人员踏入动态网站开发。
- **ASP.NET Core (.NET 5/6) 的优势:**
 - 开发一个新的Web应用应选.NET Core / .NET 6 (.NET Core), 是一个长期稳定版, 切入.NET Core的不错时间点 / 可跨台与Java网站开发技术一较高下。
- **MVC框架的应用:**
 - 不管在.NET Framework或Core平台都支持本模式的开发。但前端需采Razor语法, 不能再使用Web Forms技术了。

LearnASP.NET-rev1.1 <http://www.coding24.org>

8

ASP.NET Web Forms/Core如何选择:

- **ASP.NET Web Forms技术的优势:**
 - ASP.NET Web Forms有接轨WinForm的开发经验。
 - 若快速开发并部署一个 .NET Web 应用并且你没有充足的时间去学习 .NET Core (.NET5/6), 那么你应该选 ASP.NET Web Forms。因为 .NET Core有一定的学习曲线(至少要再去熟悉Razor语法)。
 - 若要维护并升级现有的 .NET Web应用程序, 那么应该选择.NET Framework。将原有Web应用程序移植到.NET Core(.NET5后) 有一定的工作量需考虑。
 - 为了找一份工作? 在2022年初的现在, 可能没有很多的关于.NET Core Web应用开发的职位可选择。
 - 若你是非软件专业人员踏入动态网站开发或是一位非计算机相关专业初学者踏入动态网站开发, 则ASP.NET Web Forms技术是一个不错选择。
- **ASP.NET Core (.NET 5/6) 的优势:**
 - 若要开发一个新的Web应用并且.NET Framework 和.NET Core (.NET5/6) 框架可供选择, 那么应该选.NET Core。
 - 2021年底发的.NET 6 (.NET Core), 这是一个长期稳定版, 是一个由.NET Framework切入.NET Core的时间点。
 - ASP.NET Core是微软网站技术的未来, 可跨台, 或可与Java网站开发技术一较高下, 但相较于ASP.NET Web Forms, 开发难度提高。
- **MVC框架的应用**
 - 至于MVC模式开发, 不管在.NET Framework或Core平台都支持本模式的开发。但

前端需采Razor, 不能再使用Web Forms技术了。



- **Visual Studio工具及IIS服务器:** 内容包含: 工具安装-Visual Studio 2019 、Web服务器以及微软的IIS服务器。在教程中使用Visual Studio自带的IIS Express, 已非常好用, 不一定要在自己的系统上构建单独的IIS, 因为完成后甚至可直接发布到远程的服务器, 如微软的Azure。



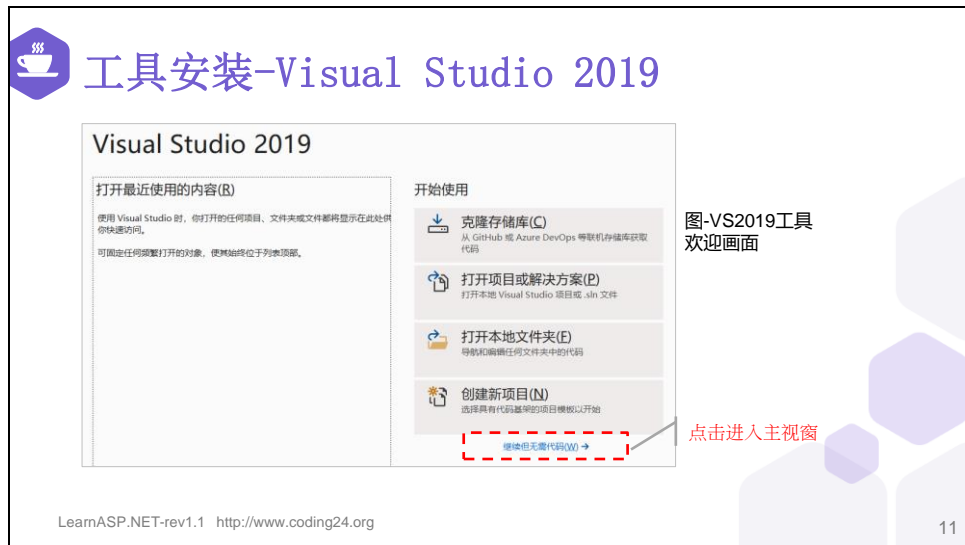
工具安装-Visual Studio 2019

- 在.NET应用开发中，Visual Studio是主要的开发工具。至目前（2022年初），已发展至VS2022版，**在本教程采用VS2019。**
- 开发ASP.NET WebForm项目不能使用VS2022（不支援.NET 4.X），**最多只能安装VS2019**，VS2017亦可。
- 有3种发行：企业版（Enterprise）、专业版（Professional）及**社区版（Community）**。**建议用社区版**，可免费使用且功能相近于专业版。
- 又划分为适用于Windows或Mac的版本可供下载使用。

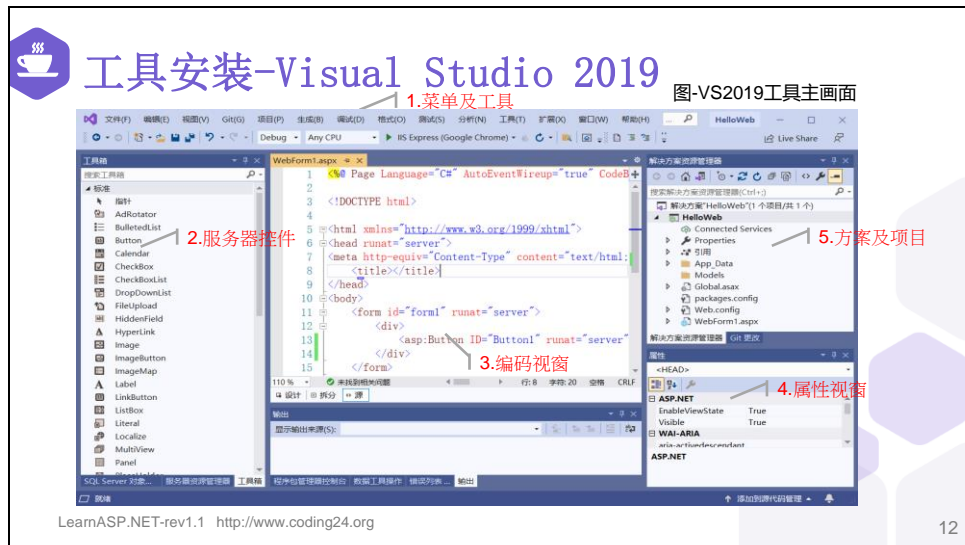
LearnASP.NET-rev1.1 <http://www.coding24.org> 10

工具安装-Visual Studio 2019:

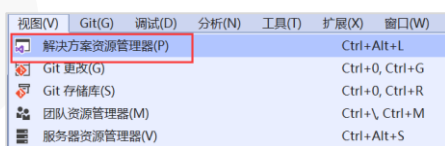
- 以功能而言可划分为三版：企业版（Enterprise）、专业版（Professional）及社区版（Community）。社区版功能相近于专业版，且免费使用。本教程建议使用。
- 以发行的年份而言：近期的版本有VS2017, VS2019及VS2022。若要开发Web Forms项目可选2017/2019版本（支持.NET Framework），不能选2022版（不支持.NET Framework）；若要开发ASP.NET Core Web项目，则Core5（.NET5）以前（不含）可用2017，Core5要选2019或2022，Core6（.NET6）后只能选2022版。估计在这两三年每位.NET的开发人员都都会装好装满几个版本的开发工具了！不过，还好的是在一台电脑上同时装上VS2022/2019/2017没什么问题，都能兼容，只是SSD磁盘要买大些。
- 若选VS2019则可开发ASP.NET Web Forms及ASP.NET Core 5项目。
- 每个版本又划分为适用于Windows或Mac的版本可供下载使用。
- Visual Studio Code（VS Code）的情况：Microsoft在2015年发布的，可运行于 Mac OS X、Windows和 Linux 之上的开发工具（毕竟现VS只有PC/Mac两种版本），针对于编写现代Web和云应用的跨平台源代码编辑器，支持多种编程语言，例如C++、C#、Java、Python、PHP、Go等。目前发展快，看不明，可能最近几年“前端”太火红，想用VS去吃下这块市场，估计这些前端工程师绝不买单。推出较轻量级的VS Code，又能再跨到Linux这一块，或许机会多些，再观察，各位不用担心，至少我们的ASP.NET Web Forms及ASP.NET Core MVC型应用在VS Code应无法开发，暂时看起来与我们较没有相关。



- 在这欢迎画面中，可以点击「继续但无需代码」进入主开发视窗，亦可直接选上面的四项功能处理相应的事件。
- 在开发一段时间后，左侧窗格中会出现近期的解决方案名称，也可以点选直接打开它。



- Visual Studio不管在那个版本其主开发主界面都差不多，有下面数个主要分区：
 - 菜单及工具栏：在上方，可点击以操作相应的功能，如创建Web项目等。其中常要先选定Web服务器（如IIS）及运行的浏览器（如Chrome）以方便后续的开发及调试。
 - 工具箱区（服务器控件）：在左侧，有各式的服务器控件或前端标签可用。有标准控件，数据控件，验证控件等类别。
 - 编码视窗区：在中间，当在编辑.aspx文档时，在此区的左下角有「设计 / 拆分 / 源」三模式可选，在初期，一般建议选「拆分」模式可同时进行页面布置及代码撰写，在熟悉开发后，大都在「源」开发模式中编写代码。
 - 属性视窗：在右下方，当点击控件时，属性视窗会列出这控件的所有属性，改变其值可以进行属性设定。
 - 方案及项目视窗：在右上方，方案是一个较大的组成，可以把它想成一个文件夹，项目是一个较小的模块，在创建时会挂上相应的类库以支持开发相应的软件项目。通常一个方案会包含数个相关的项目。在初期学习阶段一个方案一个项目也没问题。
 - 某个视窗不见了如何处理？在主菜单中，选视图 -> 找到被你关闭的那个，点击它。





幻灯片内容：标题为“IIS服务器”，左侧有一个咖啡杯图标。正文包含两个要点：1. 微软的Web服务器组件称为IIS（Internet Information Services），可运行ASP.NET网站应用程序。2. 本教程使用Visual Studio工具内置的IIS Express，不用另行安装。右侧是Microsoft IIS的标志。底部有页码13和网址http://www.coding24.org/。

IIS服务器

- 微软的Web服务器组件称为IIS（Internet Information Services），可运行ASP.NET网站应用程序。
- 本教程使用Visual Studio工具内置的IIS Express，不用另行安装。


LearnASP.NET-rev1.1 <http://www.coding24.org> 13

Web服务器:

- 微软的Web服务器组件称为IIS（Internet Information Services），可运行ASP.NET网站应用程序。
- 在Windows系统内置有IIS组件，可透过操作系统的控制台进行IIS服务的打开或关闭其功能（默认是不启用）。手动开启IIS及管理步骤大略如下：
 - 开启IIS：选择“开始”→“控制面板”→“程序”→“打开或关闭Windows功能”命令，在对话框中选中“Internet信息服务”复选框。
 - 管理IIS：选择“开始”→“控制面板”→“系统和安全”→“管理工具”→“Internet信息服务(IIS)管理器”命令
- 网站项目在完成后可部署到前述的IIS上或微软的Azure服务器上，以提供用户使用网站功能。若只在VS自带的IIS Express中，无法提供远程用户浏览网站。
- 本教程使用Visual Studio工具内置的IIS Express，不用另行安装。
- **Microsoft Azure:** 是微软公司提供的云服务器及数据中心平台，我们所开发的ASP.NET Web应用及所使用的SQL Server数据库都可发布至Azure平台上使用。



- **实作1-Visual Studio开发工具安装：**工欲善其事必先利其器，开发工作首先自然是要安装及设置开发环境了。讲义中建议使用Visual Studio 2019较佳。它能开发Web Forms型应用，而且.NET Framework能自动装至4.7，若在安装时选4.8也能上到这最成熟的版本。最重要，同时也能开发.NET Core 5（.NET 5）。



实作1-Visual Studio开发工具安装

实作说明：下载及安装Visual Studio工具，以进行后续开发。

实作步骤：

1. 使用浏览器连至Visual Studio 2019的下载网页，下载「社区版」(Community版)：

VS2019: <https://docs.microsoft.com/zh-cn/visualstudio/releases/2019/release-notes>

VS2017: <https://docs.microsoft.com/zh-cn/visualstudio/releases/2017/release-notes>

单击按钮即可下载最新版 Visual Studio 2019。有关如何安装和更新 Visual Studio 2019 的说明，请参阅[Visual Studio 2019 更新到最新版本](#)。另请参阅[如何脱机安装](#)的说明。

Download Community 2019 ↓ Download Professional 2019 ↓ Download Enterprise 2019 ↓

LearnASP.NET-1.1 <http://www.coding24.org> 15

实作1-Visual Studio开发工具安装：

- 建议下载VS2019（要开发Web Forms）可依下述网址下载，并选择下载社区版（Download Community）。
 - VS2019: <https://docs.microsoft.com/zh-cn/visualstudio/releases/2019/release-notes>
 - VS2017: <https://docs.microsoft.com/zh-cn/visualstudio/releases/2017/release-notes>
 - 本课程建议使用VS2019，亦可使用VS2017，但不能使用VS2022，因这个版本不支持Web Forms项目开发。
- 直接连到Visual Studio的官网则会下载最新版，官网下载页如下（简中）：
 - 在2022年初，这个网址会下载VS2022版，网址如下。
 - VS2022: <https://visualstudio.microsoft.com/zh-hans/downloads/>
- 注意：VS2022不支持ASP.NET Web Forms项目的开发，若要开发要下载VS2019或VS2017版。
- 注意：VS2019若是早期下载安装的，那不支持.NET5的开发，启动它的安装程序即可进行更新，不用重安装。
- 微软也是有佛心的，社区版功能相近于专业版，但免费，是一个不错的学习选择。



实作1-Visual Studio开发工具安装

2. 运行安装程序，在”工作负荷”页，选择3个工具：

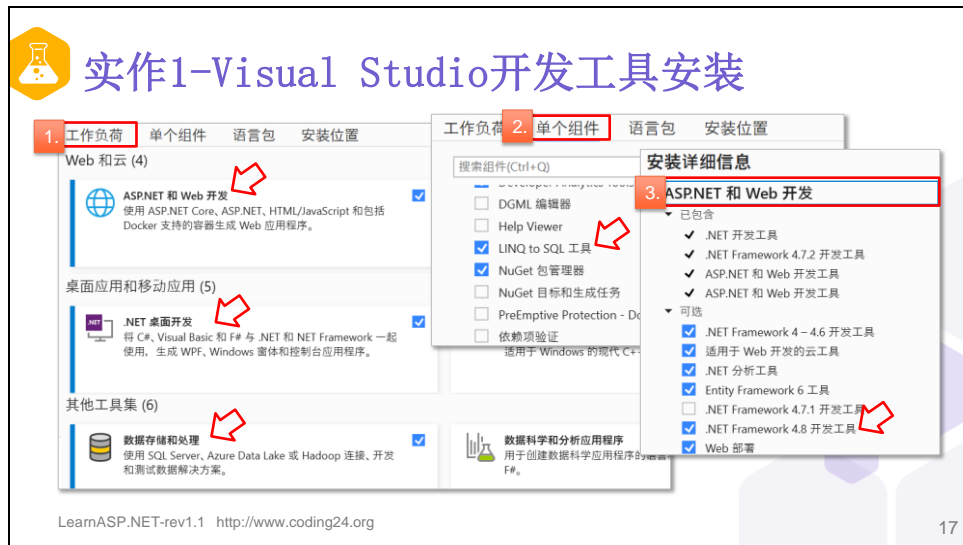
- 「**ASP.NET和Web开发**」(记得要加选.NET Framework4.8开发工具)
- 「**.NET桌面开发**」(不是必要但建议选取)
- 「**数据存储和处理**」(不是必要但建议选取)

3. 在”单个组件”页，加选1个工具：**LINQ to SQL工具**。
(本教程中用到)

LearnASP.NET-rev1.1 <http://www.coding24.org>

16

- 工作负荷页即是选择要安装那些开发模块，若只是要开发Web应用，那只要选取「ASP.NET和Web开发」即可；但在基础.NET应用开发时，一般还会用到「.NET桌面开发」及「数据存储和处理」，因此教程中选了3个模块。您可按实际需求进行选取及安装，若有不足再另行安装亦可。
- LINQ to SQL是简化版的Entity Framework，一个ORM框架，若开发中有用到，那选它。



- 在模块及单个组件选完后，在右侧「安装详细信息」中（ASP.NET和Web开发），要再找到「.NET Framework 4.8开发工具」项，这项要装上，毕竟是最成熟的一个版本。



实作1-Visual Studio开发工具安装

4. 在安装页面下方选择「全部下载后再安装」项，并点击「安装」，进行工具的安装，按网络及电脑速度可能需要10-15分钟完成安装。
5. 启动Visual Studio: 在Windows开始 -> 找到「Visual Studio 2019」。第一次启动需要一个微软帐号，如果没有那注册一个，若不使用帐号登入则只能用30天。
6. 完成！若要增减功能可运行「Visual Studio Installer」。

LearnASP.NET-rev1.1 <http://www.coding24.org>

18

- 使用VS工具需要一个微软的email帐号，格式通常为xxx@outlook.com或xxx@hotmail.com，若没有申请一个。此外也可以使用github.com的帐号亦可。
- 若已安装过VS2017，不用移除它，可装VS2019或VS2022，可兼容没问题，但不建议装了较新版本后再去装旧的版本，因为一般都是向前兼容的。
- 在安装VS2019时默认可能只装上.NET Framework 4.7，可使用没问题，安装时若忽略选取4.8版若要再装，可透过(1)Visual Studio Installer安装,或(2)启动VS工具后在主菜单选取「工具」->「获取工具程功能」。在安装程序中选取「ASP.NET和Web开发」->「.NET Framework 4.8开发工具」，进行安装。



- **实作2-开发HelloWeb项目：**第一个ASP.NET Web Forms项目，动手做是了解一个项目开发来龙去脉的最佳方式。动手实作吧！

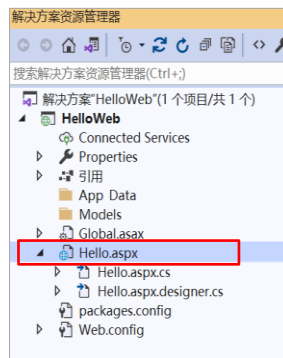


实作2-开发HelloWeb项目

● 实作说明：创建第一个Web Forms网站并添加Hello.aspx页面。

● 实作步骤：


1. 使用Visual Studio, 主菜单「文件」->「新建」->「项目」, 找到「ASP.NET Web应用程序(.NET Framework)」, 项目/解决方案名为HelloWeb, 放置在工作目录中。框架: .NET Framework 4.8 (选4.X亦可), 完成输入后点击「创建」按钮。
2. 在画面中选「空」, 在添核心引用选「Web窗体」, 取消「为HTTPS配置」。
3. 在HelloWeb项目中添加「Web窗体」->命名为Hello.aspx



LearnASP.NET-1.1 <http://www.coding24.org>

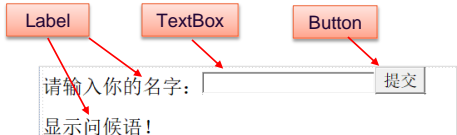
20

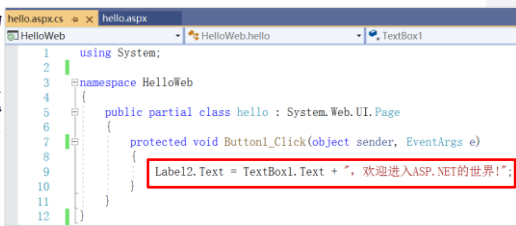
- 工作目录（文件夹）建议在C或D磁盘创建一个文件夹，命名如aspcode或是用你的英文名字亦可，建议不要放在桌面上。文件夹建议最好也不要使用中文，主要是中文文件夹名在不同的语系编码时可能要再进行处理，不然可能会有问题。
- HTTPS（Hyper Text Transfer Protocol over SecureSocket Layer），HTTPS在HTTP的基础上利用SSL/TLS建立安全通道，加密数据包，使传输内容获得加密、传输身份获得验证以及数据完整性获得了保护。但SSL证书需要申请及购买，功能越强大的证书费用越高。在Web项目开发前期暂可不用理会这项功能要求。



实作2-开发HelloWeb项目

4. 在页面Hello.aspx上布置文字及控件如右
5. 在Button1（按钮控件）后置代码中输入所需代码
6. 运行程序：鼠标右击 **Hello.aspx** -> 「在浏览器中查看」，运行结果如下。





LearnASP.NET-rev1.1 <http://www.coding24.org> 21

- 运行网页（网站）的方式：
 - 一般运行（生成运行）：
 - （1）打开要运行的页面（.aspx），点击工具栏中的「IIS Express (Chrome)」运行。一个方案中若有多个项目时，要先将此项目设为启动项目。
 - （2）工具栏中，点击「IIS Express (Chrome)」，并在浏览器中打入网址。一个方案中若有多个项目时，要先将此项目设为启动项目。
 - 轻量运行（局部生成）：
 - （1）在项目视窗选取欲运行的页面（.aspx），鼠标右击->「在浏览器中查看」，查看运行结果。
 - （2）在代码编辑视窗中，在.aspx文档上，鼠标右击->「在浏览器中查看」，查看运行结果。若目前重新编辑的页面有需额外的新添资源，如图片，或有需要额外刚编写的类，则不能采用此种运行方式。
- 可透过工具栏上的「IIS Express (Chrome)」项更换浏览器。一般建议采用Chrome浏览器使用。
- Hello.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="HelloWeb.aspx.cs"
Inherits="LearnASPNET.Labs.Unit01.HelloWeb" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Hello Web</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="Label1" runat="server" Text="请输入你的名字: "></asp:Label>
```



```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="提交" />
<br />
<br />
<asp:Label ID="Label2" runat="server" Text="显示问候语! "></asp:Label>
</div>
</form>
</body>
</html>
```

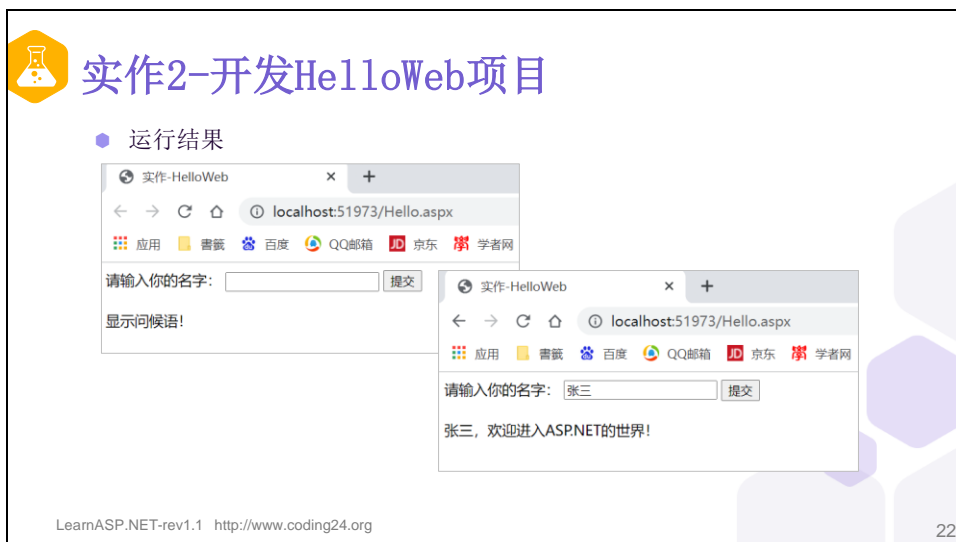
- Hello.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace LearnASPNET.Labs.Unit01
{
    public partial class HelloWeb : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Label2.Text = TextBox1.Text + ", 欢迎进入ASP.NET的世界! ";
        }
    }
}
```



- 本页为HelloWeb的运行结果，在TextBox中打入“张三”，并点击提交可查看运行结果。
- ASP.NET C#代码放置可选两处：
 - Inline Code（代码内嵌）：代码与HTML标签混在一处（在Web Forms开发不建议此法），如下代码可放在.aspx档中，但前后要放在<%...%>中：

```
<%Response.Write("Hello World!"); %>
```

这种写法类似于早期的ASP写法，只是ASP是用JavaScript或VBScript脚本语言并非现在的C#或Visual Basic (VB)编程语言。
 - Code Behind（代码后置）：代码（.cs或.vb）与页面（.aspx）分离。本教程建议使用这种方式，可以让网页设计与编程分离，在设计期及维护期都有较为简洁的好处，不致于太过于混乱。



3. 范例代码及资源介绍

LearnASP.NET-rev1.1 <http://www.coding24.org>

- 范例代码及资源介绍

教程中提供了一套学习与开发两用的Visual Studio 2019版项目—LearnASP.NET，请下载最新版本学习ASP.NET项目开发。也外也提供了一些重要的ASP.NET开发资源。

☕ 范例代码介绍及使用

● 请先取得压缩包
LearnASPNETv11.zip

解压

LearnASPNET

所需材料

packages

LearnASPNET.sln

用VS2019打开

images

Booksdb.sql

bootstrap-5.1.3-dist.zip



LearnASP.NET-rev1.1 <http://www.coding24.org>

24

- 课程中建议使用Visual Studio 2019版，.NET Framework升到4.8版。



ASP.NET开发资源

- 微软ASP.NET 4.x (中文)
<https://docs.microsoft.com/zh-cn/aspnet/overview>
- 微软ASP.NET Core (中文)
<https://docs.microsoft.com/zh-cn/aspnet/core>
- 微软MSDN技术文档 (中文)
<https://docs.microsoft.com/zh-cn/documentation/>
- 菜鸟教程
<https://www.runoob.com/aspnet/aspnet-tutorial.html>
- W3School
<https://www.w3school.com.cn/aspnet/aspnet.asp>
- Microsoft asp.net blog (英文)
<https://weblogs.asp.net/>
- ASP.NET软件GitHub
<https://github.com/aspnet/>

LearnASP.NET-rev1.1 <http://www.coding24.org> 25

ASP.NET 开发资源

- 微软ASP.NET 4.x (中文)
<https://docs.microsoft.com/zh-cn/aspnet/overview>
- 微软ASP.NET Core (中文)
<https://docs.microsoft.com/zh-cn/aspnet/core>
- 微软MSDN技术文档 (中文)
<https://docs.microsoft.com/zh-cn/documentation/>
- 菜鸟教程
<https://www.runoob.com/aspnet/aspnet-tutorial.html>
- W3School
<https://www.w3school.com.cn/aspnet/aspnet.asp>
- Microsoft asp.net blog (英文)
<https://weblogs.asp.net/>
- ASP.NET软件GitHub
<https://github.com/aspnet/>



单元小结

- 本单元介绍ASP.NET开发技术，使开发人员了解ASP.NET Web Forms项目在快速开发以及大量已存在的系统维护等方面仍是一个不可或缺的开发技术，这现象可能持续数年以上。
- 介绍了Visual Studio 2019社区版工具，并透过实作Visual Studio开发工具安装以让开发人员做好环境准备。
- 开发了第一个HelloWeb（Web项目）以了解页面（.aspx）及后置代码（.cs）的区别及其一体性关系。