自我介绍：

我是陈超超，本次是进击吧！Blazor！系列课程的第二节课，在上一节课中，张善友老师带我们认识了Blazor。

从这节课开始我们以实战的方式带大家进一步了解和掌握Blazor的开发，这次系列课程属于零基础入门课程，我们目标是能让一个从未接触过Blazor的程序员通过本系列课程就能开发出自己的Blazor应用。

既然是实战，我们先把PPT晾边上，直接从创建项目开始

* **（打开VS，创建项目）**

对于两种模式上节课有详细的介绍，这里我不赘述了

（选择WebAssembly创建项目）

Host选项是使用Asp.net core托管blazor wasm，提供文件压缩、集成identity验证、还有Server Prerendering等功能，用了之后，就可以直接发布asp.net core，跟普通服务一样部署运行，不再需要静态文件服务托管

* **（项目结构）**

**Client：**Blazor项目，本身就是一个Asp.Net Core项目，主要是Main中的Builder使用了WebAssemblyHostBuilder，如果只做前端，就可以直接拿这个项目进行发布和部署。

（演示一下发布，并展示文件）

**Server：**服务端程序，也是一个Asp.Net Core项目，可以用他给前端提供WebAPI，并且承载了Client编译的静态文件，如果我们前后端部署在一起，那么直接发布他即可，这也是结构最简单部署方式。

（演示一下发布，并展示文件）

**Shared：**共享项目，这就是Blazor相对于其他前端开发解决方案的最大的优势，Blazor的前后端使用了相同的语言、相同的框架和语言规范。因此我们可以共享Dto，共享常用组件库，甚至共享相同的业务代码。

（分别打开前端与后端的项目引用，看到引用了相同项目）

我们聚焦前端

**wwwroot：**静态文件存放位置，用途和Aps.net core相同，里面的index.html文件就是主页

**Pages、Shared：**页面文件存放位置

**\_Imports.razor：**全局导入配置

**App.razor：**应用根组件

**Program.cs：**项目入口，这里构建了WebAssemblyHostBuilder对象，然后把APP加到根组件映射，这样就能在index.html文件中的<app>标签载入App.razor组件了

现在我们看看默认项目执行效果

（点击页面切换）

做过前端的同学应该发现了，他的页面切换直接使用了路由，那么我们一起来看一下他的路由是怎么实现的

* （路由）

**App.razor:**首先我们看看App.razor文件，里面就一个Route 组件，我们把当前项目的程序集赋值给了**AppAssembly**属性。

我们再看看页面文件，每个文件顶部有一个@page标记当前页的路由信息。

到这里他的路由原理就比较清晰了

为了验证，我们把Counter的@page改成RouteAttribute试一下。

（修改@page并测试，顺带介绍局部页的概念）

我们还可以通过AdditionalAssemblies属性支持多个程序集，此处我们就不演示了。

Route里面有两个模板属性，分别是路由命中和未命中显示的内容

这里我们关注一下RouteView 组件，从 Router 接收 RouteData 以及任何所需的参数。其次通过**DefaultLayout**参数定义了默认布局。

**NavMenu.razor:**有了路由必定有导航组件，我们一起看一下NavMenu.razor文件，这里的NavLink组件就是超链接。

这里我们不用超链接元素<a>的原因是它能根据其 href 是否与当前 URL 匹配来切换 active CSS 类，这有助于在导航菜单中显示那个页面是活动页。

他是通过当前路由地址与href属性的值进行匹配，匹配分为全匹配和任何前缀匹配

Match 属性：

NavLinkMatch.All：NavLink 在与当前整个 URL 匹配的情况下处于活动状态。

NavLinkMatch.Prefix（默认）：NavLink 在与当前 URL 的任何前缀匹配的情况下处于活动状态。

当然我们也能通过注入NavigationManager来事项代码控制路由

（编写一个按钮，让他能进行导航到某页）

* **（项目初始化）**

接下来我们一起通过开发一个待办事项应用来学习Blazor。

（打开准备好的ToDo Demo项目）

这是之前准备的项目，这里删除了默认模板中的一些样式，添加了几个服务端接口用于模拟数据返回，这部分内容我们将在以后的课程中进行介绍，这里就不展开了

项目开始前，我们先引入一个UI轮子，这可以提高我们开发效率，减少大量样式编写任务，这里使用我正在参与开发的Ant Design Blazor作为基础UI库

（UI库的引入）

首先我们从布局中选择一个觉得适合的布局开始

（打开antdesign文档页面）

开发过程中要善于使用文档，没有人能记住所有开发手册

（按照事先准备好的代码编写模板页面，首页）

（复制样式文件）

为了节约时间，项目中的样式文件我就直接使用事先准备好的文件

我们也可以使用scss等，这里主要样式Blazor相关，因此就不用这些了

（运行看到程序）

程序的基本布局和菜单已经完成

* **（我的一天）**

（添加页头）-PageHeader

（注入TaskSvr，获取数据）

（foreach的用法，@的用法介绍）

（输出没有格式的Title列表）

（使用卡片，编辑所有字段，复制css代码）

（重要）

事件的使用

（日期）

计划时间需要三种颜色显示，先用if来实现，再用switch实现

（标题）

抽屉的使用

（新增）

数据绑定

按键事件使用

（删除）

确认框的使用

（完成）

样式控制

ClassMapper的使用

（重要）

组件的创建（更多使用方法请关注后续课程）

（全部）

Table的使用

（分页，服务端排序等将在下一节课中介绍）