**Where we are:**

|  |  |
| --- | --- |
| Intro to MVC | Midterm, Navigation |
| Controllers and Views, Publishing to Azure | Forms and Tag Helpers |
| Unit Testing, Dependency Injection | Annotation and validation |
| Models, Entity Framework and Scaffolding | Authorization and Authentication |
| Customizing Scaffolding | API Controllers, Final |

**Announcement**

* Instructions for signing up for Azure are on Moodle

**Review**

* Advantages of using the MVC architecture
  + Reduces dependencies
  + Separation of concerns – easier to work with modules that do one thing.

**Views**

* Razor Syntax
  + @Model for definition (declaration?)
    - @Model Book
  + @model for access
    - <h2> @model.Title </h2>
  + Conditional expressions
    - @switch (myInt) {  
       case 0:  
       @:<h3>Show something</h3>  
       break;  
       …  
       default:  
       break;  
      }
    - @if (logicalExpression) {  
       // do stuff  
      }
  + foreach expressions
    - We need to have a a collection in our model
    - @foreach (var a in Model.Students) {  
      // do something  
      }
* layout
  + Will wrap the views that use it at the location specified by  
    @RenderBody
  + Gets a title from @ViewBag
  + Named with an underscore: *\_MyLayout.cshtml*
  + used by a view with a C# code block
    - @{  
       Layout = "\_MyLayout";  
       ViewBag.Title = "My Title";  
       }
  + Default layout using a view start file: *\_ViewStart.cshtml*
    - Added to the Views folder
    - Has a C# block containing: Layout = “\_MyLayout”;
      * Remove this line from the views
* View bag (already covered on Tuesday)
* Continue the MVC Movie Tutorial with Adding a Controller:  
  <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-view>
* Add a ViewBag
* Try creating an Empty MVC site (like Freeman does)