ASP.NET MVC

MVC 5

MVC Core

Eng. Basma Hussien

Why we need web Applications?

Early Web Development

1. **Static pages (Already-made)**
   - **HTML pages.**
   - **Describes layout.**
   - **Interactivity achieved mainly using hyperlinks.**

2. **Dynamic pages (Interactive)**
   - **HTML & scripts (ex: javascript).**
   - **Pages contained client-side scripts.**
   - **Can be used to validate user input.**

But….

Why we need server-side programming ?!

But… we need more interactivity, using client-side apps we can't process data, Like:

- Connecting to db
- Accessing files on hard disk(write, modify, …)
- Works with devices (flash memory, …)

So, we need server-side programming…

Differences between client-side and server-side

- **Client-Side Apps (Websites):**
  - Website runs on client, run on "browser"
  - Browser interprets each line and each tag and then translate it "render it" to the HTML page

- **Server-Side Apps (Web Applications):**
  - Application runs on server; any user can connect to it using URL.
  - It has 2 parts (2 layers):
    - Client: HTML, CSS, JavaScript, bootstrap
    - Server: Programming logic

ASP uses C# as a backend, HTML & Css & javaScript as a FrontEnd

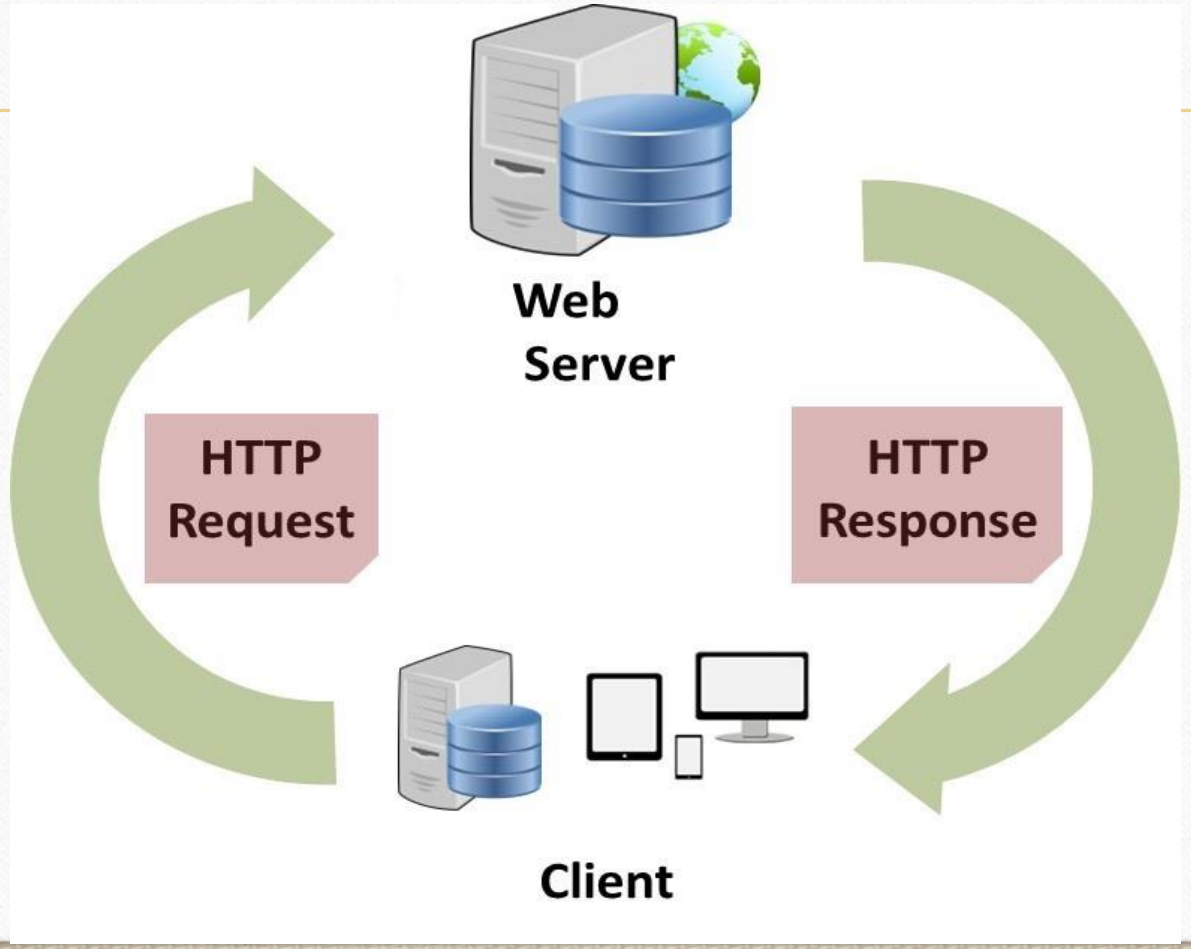
Below is a table of differences between Web Application and Website:

WEB APPLICATION	WEBSITE
Web application is designed for interaction with end users.	Website basically contains static content.
The user of web application can read the content of web application and also manipulate the data.	The user of website only can read the content of website but not manipulate .
The web application site should be precompiled before deployment.	The website does not need to be precompiled .
The function of web application is quite complex .	The function of website is simple.
Web application is interactive for users.	Web site is not inetractive for users.
The browser capabilities involved with web application is high.	The browser capabilities involved with web site is high.
Integration is complex for web application because of its complex functionality.	Integration is simpler for web site.
Web application mostly requires authentication	In wenb site authentication is not necessary.
EXAMPLE :- Amazon, Facebook, etc.	EXAMPLE :- Breaking News, Aktu website, etc.

IIS

- IIS is a web server offered by Microsoft that runs on Windows machine.
- IIS has its own ASP.NET process engine to handle the ASP.NET requests.
- Note: When we run our ASP.NET web application from Visual Studio IDE, it integrates the ASP.NET Engine which is a part of IIS Express built into VS and responsible for executing all kinds of ASP.NET requests and responses.

How we browse the websites?



ASP

- ASP stands for **A**ctive **S**erver **P**ages
- ASP is a development framework for building web pages.

ASP Technology

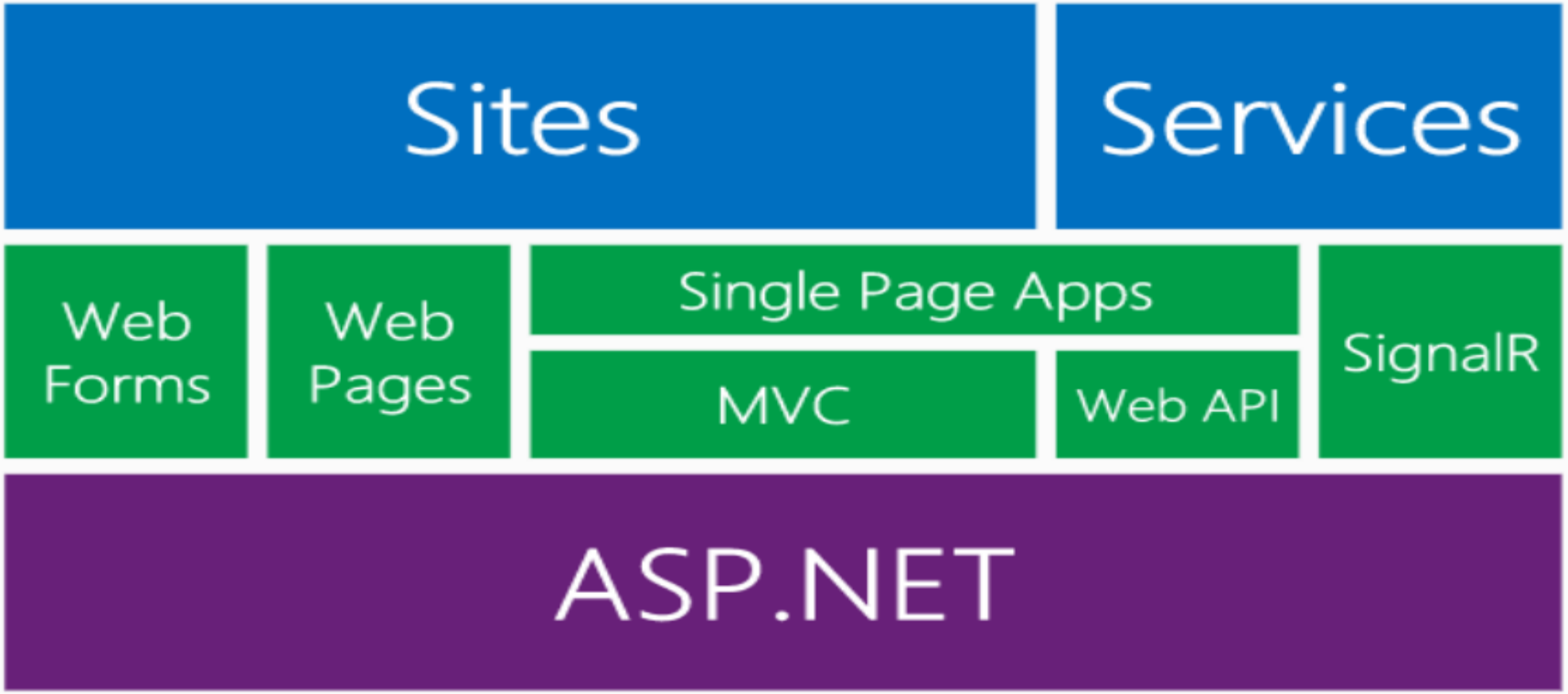
- ASP and ASP.NET are server side technologies.
- Both technologies enable computer code to be executed by an Internet server.
- When a browser requests an ASP or ASP.NET file, the ASP engine reads the file, executes any code in the file, and returns the result to the browser.

ASP different development models

- Classic ASP
- ASP.NET Web Forms
- ASP.NET MVC
- ASP.NET Web Pages
- ASP.NET API
- ASP.NET Core

ASP.NET Family

One ASP.net : A framework for us all



Web Development Using VS 2022

Visual Studio Installer

Installed Available

All installations are up to date.

Visual Studio Community 2022

17.5.0

Powerful IDE, free for students, open-source contributors, and individuals

Release notes

Modify

Launch

More

Developer News

The fastest way to add files and folders

Your Visual Studio team is excited to announce th...

Monday, February 27, 2023

Reference GitHub Issues and Pull Requests in Visual Studio

We're excited to announce the preview release of...

Thursday, February 23, 2023

Visual Studio 2022 can now Always Update on Close!

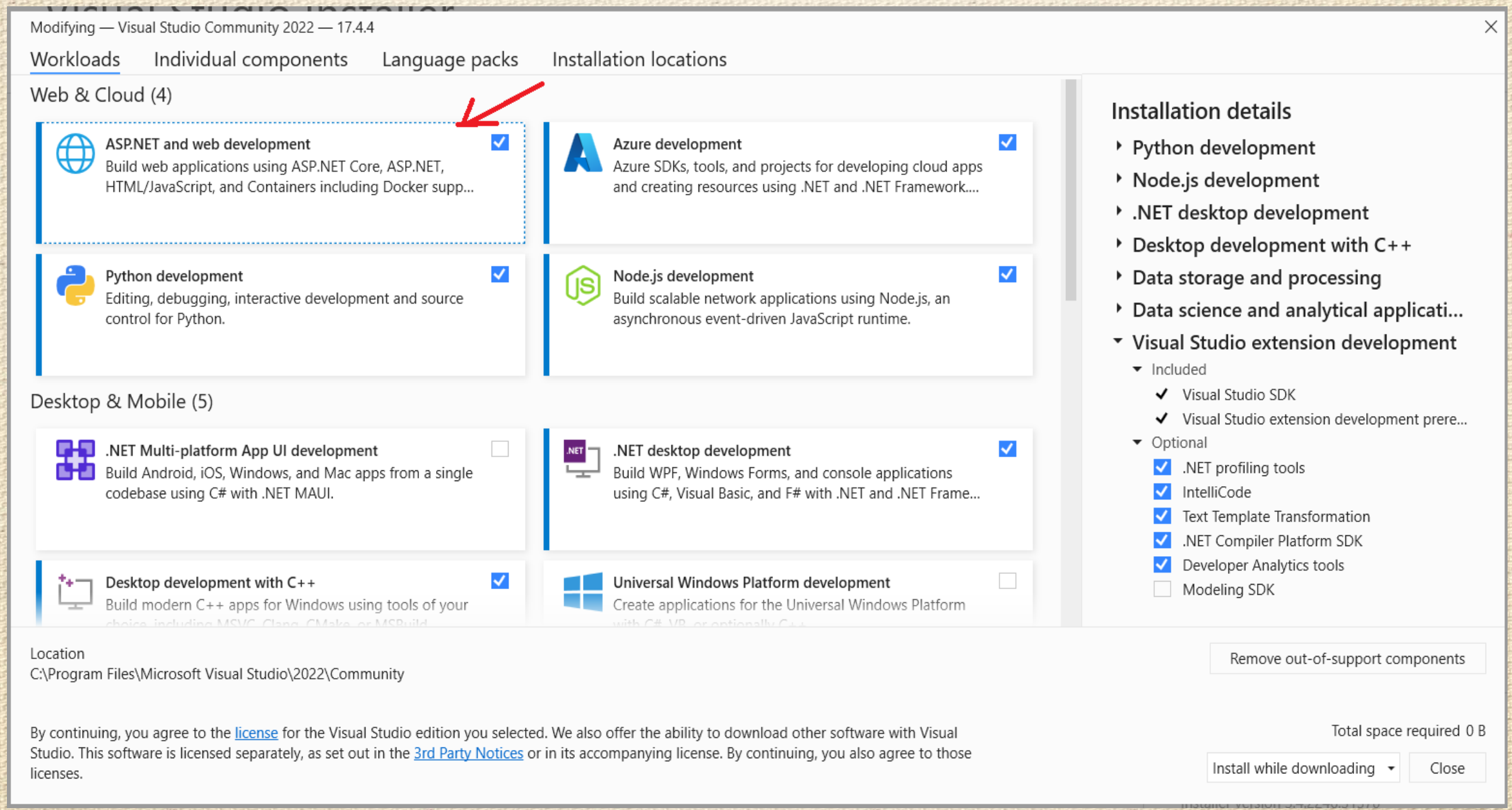
Do you want Visual Studio 2022 to always stay up...

Wednesday, February 22, 2023

View more Microsoft developer news...

Need help? Check out the Microsoft Developer Community or reach us via Visual Studio Support.

Installer version 3.5.2145.59678


Modifying — Visual Studio Community 2022 — 17.4.4
Workloads
Individual components
Language packs
Installation locations
Web & Cloud (4)
ASP.NET and web development
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
Azure development
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....
Python development
Editing, debugging, interactive development and source control for Python.
Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.
Desktop & Mobile (5)
.NET Multi-platform App UI development
Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.
.NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...
Desktop development with C++
Build modern C++ apps for Windows using tools of your
Universal Windows Platform development
Create applications for the Universal Windows Platform
Installation details
Python development
Node.js development
.NET desktop development
Desktop development with C++
Data storage and processing
Data science and analytical applicati...
Visual Studio extension development
Included
Visual Studio SDK
Visual Studio extension development prere...
Optional
.NET profiling tools
IntelliCode
Text Template Transformation
.NET Compiler Platform SDK
Developer Analytics tools
Modeling SDK
Location
C:\Program Files\Microsoft Visual Studio\2022\Community
Remove out-of-support components
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.
Total space required 0 B
Install while downloading
Close

Modifying — Visual Studio Community 2022 — 17.4.4

Workloads | Individual components | Language packs | Installation locations

Search components (Ctrl+Q)

- [ ] .NET Framework 4.6.1 targeting pack
- [ ] .NET Framework 4.6.2 SDK
- [ ] .NET Framework 4.6.2 targeting pack
- [ ] .NET Framework 4.7 SDK
- [ ] .NET Framework 4.7 targeting pack
- [ ] .NET Framework 4.7.1 SDK
- [ ] .NET Framework 4.7.1 targeting pack
- [ ] .NET Framework 4.7.2 SDK
- [x] .NET Framework 4.7.2 targeting pack
- [x] .NET Framework 4.8 SDK
- [x] .NET Framework 4.8 targeting pack
- [ ] .NET Framework 4.8.1 SDK
- [ ] .NET Framework 4.8.1 targeting pack
- [x] .NET Framework project and item templates
- [ ] .NET MAUI SDK for Android
- [ ] .NET MAUI SDK for iOS
- [ ] .NET MAUI SDK for Mac Catalyst
- [ ] .NET MAUI SDK for Windows

Installation details

- Visual Studio core editor
- ASP.NET and web development
- Azure development
- Python development
- Node.js development
- .NET desktop development
- Desktop development with C++
- Data storage and processing
- Data science and analytical applicati...
- Visual Studio extension development
- Individual components
  - [x] Advanced ASP.NET features
  - [x] .NET Framework project and item templates

Location
C:\Program Files\Microsoft Visual Studio\2022\Community

Remove out-of-support components

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 577 MB

Install while downloading | Modify

Razor

What is Razor ?

- Razor is a simple **markup syntax** that lets you embed server-based code (Visual Basic and C#) into web pages.
- Razor is not a programming language. It's a server-side markup language.
- Razor is based on ASP.NET, and designed for creating web applications. It has the power of traditional ASP.NET markup, but it is easier to use, and easier to learn.

Main Razor Syntax Rules for C#

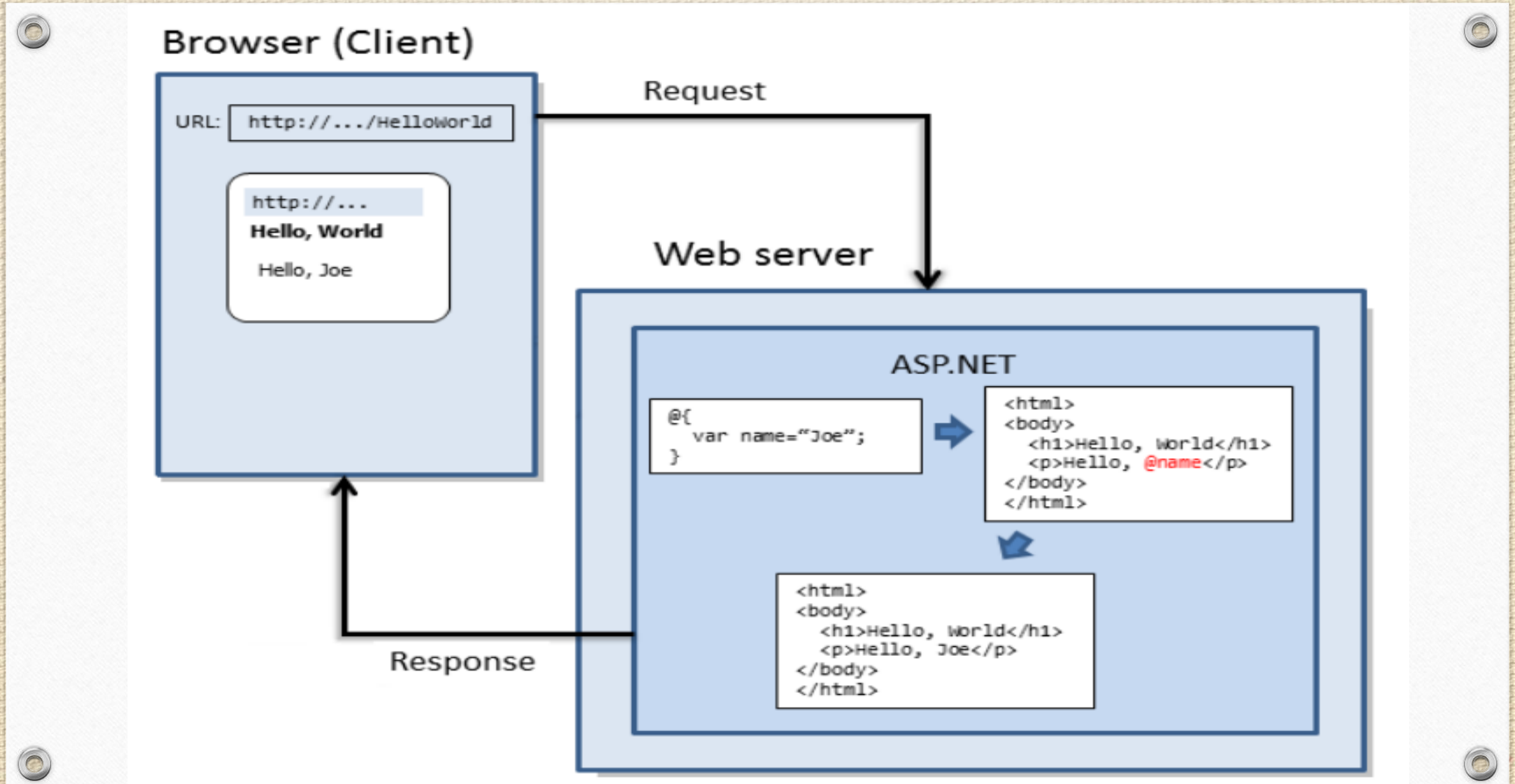
- Razor code blocks are enclosed in **@{ ... }**
- Inline expressions (variables and functions) start with **@**
- Code statements end with semicolon **;**
- Variables are declared with the **var** keyword
- Variables must be initialized before used
- Strings are enclosed with quotation marks **“ ”**
- **if statements** should be enclosed in **{}** even if it holds single line code
- C# code is case sensitive
- C# files have the extension **.cshtml**

Razor Example

```
<!-- Single statement block -->
@{ var myMessage = "Hello World"; }


<!-- Inline expression or variable -->
<p>The value of myMessage is: @myMessage</p>


<!-- Multi-statement block -->
@{
var greeting = "Welcome to our site!";
var weekDay = DateTime.Now.DayOfWeek;
var greetingMessage = greeting + " Today is: " + weekDay;
}
<p>The greeting is: @greetingMessage</p>
```


Browser (Client)
URL:
http://.../HelloWorld
http://...
Hello, World
Hello, Joe
Request
Web server
ASP.NET
@{
var name="Joe";
}
<html>
<body>
<h1>Hello, World</h1>
Hello, @name
</body>
</html>
<html>
<body>
<h1>Hello, World</h1>
Hello, Joe
</body>
</html>
Response

Request Object

The **Request** object contains all the information that the browser sends to your application when a page is requested or submitted.

- **Request.Form** gives you values from elements inside the submitted <form> element if the request is a **POST** request.

- **Request.QueryString**

Reading User Input

- An important feature of dynamic

web pages is that you can read user input.

- Input is read by the **Request[]**,

and posting (input) is tested by

the IsPost condition:

```
@{
var totalMessage = "";
if(IsPost)
    {
    var num1 = Request.Form["SalaryTXT"];
    var total = 10 + num1.AsInt();
    totalMessage = "Total = " + total;
    }
}
<html>
<body >
<form action="" method="post">
     <p><label for="SalaryTXT">Salary:</label><br>
     <input type="text" name="SalaryTXT" /></p>
     <p><input type="submit" value="Add " /></p>
</form>
<p>@totalMessage</p>
</body>
</html>
```

Page Layout

- You can use the **@RenderPage()** method to import content Block from separate files:

```
@RenderPage("header.cshtml")
```

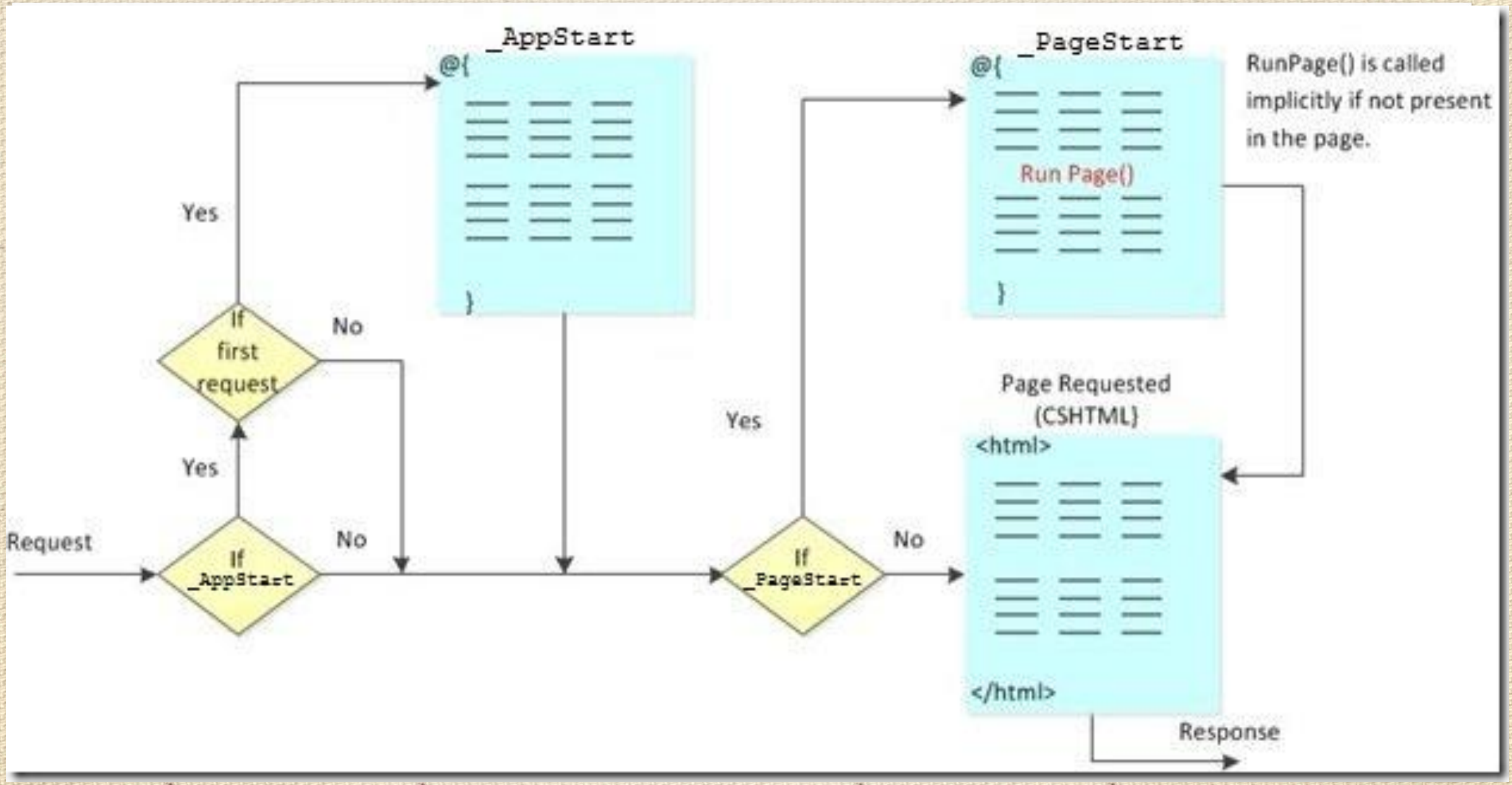
```
@RenderSection(“mysection“, required : false)
```

Page Layout (Cont.)

- You can define a consistent layout for all your pages, using a layout **template** (layout file).
- A layout page (template) contains the structure, but not the content, of a web page.
- The layout page is just like a normal web page, except from a call to the **@RenderBody()** method where the content page will be included.

Preventing Files from Being Browsed

- With ASP.NET, files with a name that **starts with an underscore** cannot be browsed from the web.
  - _Layout.cshtml
  - _PageStart.cshtml


_AppStart
@{
}
_PageStart
@{
Run Page()
}
RunPage() is called implicitly if not present in the page.
Page Requested (CSHTML)
<html>
</html>
Request
If _AppStart
Yes
No
If first request
Yes
No
If _PageStart
Yes
No
Response

The "Page" Object

Some Page Object Methods:

Method	Description
href	Builds a URL using the specified parameters
RenderBody()	Renders the portion of a content page that is not within a named section (In layout pages)
RenderPage(*page*)	Renders the content of one page within another page
RenderSection(*section*)	Renders the content of a named section (In layout pages)
Write(*object*)	Writes the object as an HTML-encoded string
WriteLiteral	Writes an object without HTML-encoding it first.

The "Page" Object

Some Page Object Properties:

Property	Description
IsPost	Returns true if the HTTP data transfer method used by the client is a POST request
Layout	Gets or sets the path of a layout page
Page	Provides property-like access to data shared between pages and layout pages
Request	Gets the HttpRequest object for the current HTTP request
Server	Gets the HttpServerUtility object that provides web-page processing methods

ASP.NET Helpers

- ASP.NET helpers are components that can be accessed by single lines of Razor code.

- Web Helpers greatly simplifies web development and common programming tasks.

ASP.NET Helpers Examples

- WebGrid Helper
- Chart Helper
- FileUpload Helper
- Json Helper
- WebImage Helper
- Recaptcha Helper
- WebMail Helper

Thank You ☺