Session 1

Introduction to ASP.NET and ASP.NET Core

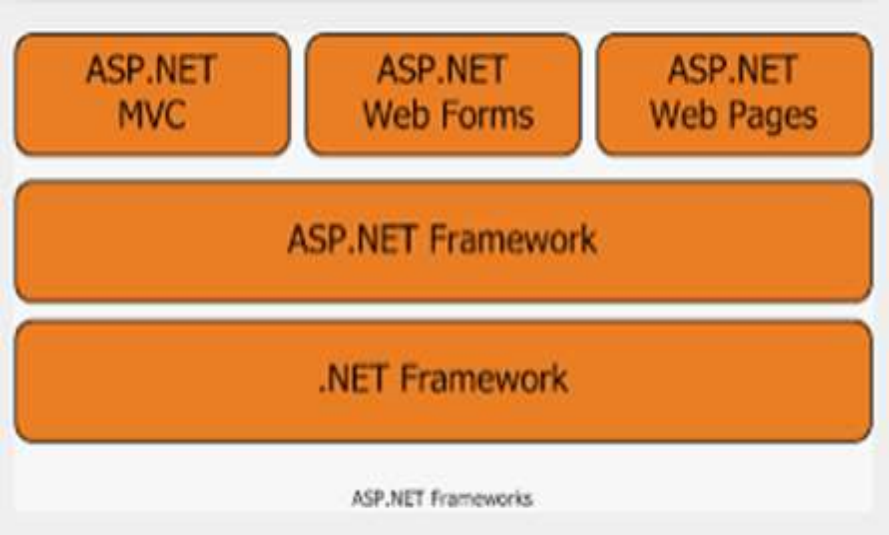
Session Overview

- Explain ASP.NET Framework and its history
- Explain ASP.NET Page Lifecycle and Lifecycle Events
- Explain ASP.NET Features and its Uses
- Explain ASP.NET Core
- List ASP.NET Core Advantages
- Describe ASP.NET Core Exceptions and Static Files
- Identify how to choose between ASP.NET and ASP.NET Core
- Describe the features of ASP.NET Core 5.0

Introduction to ASP.NET

Microsoft developed Active Server Pages (ASP) to build dynamic Web applications.

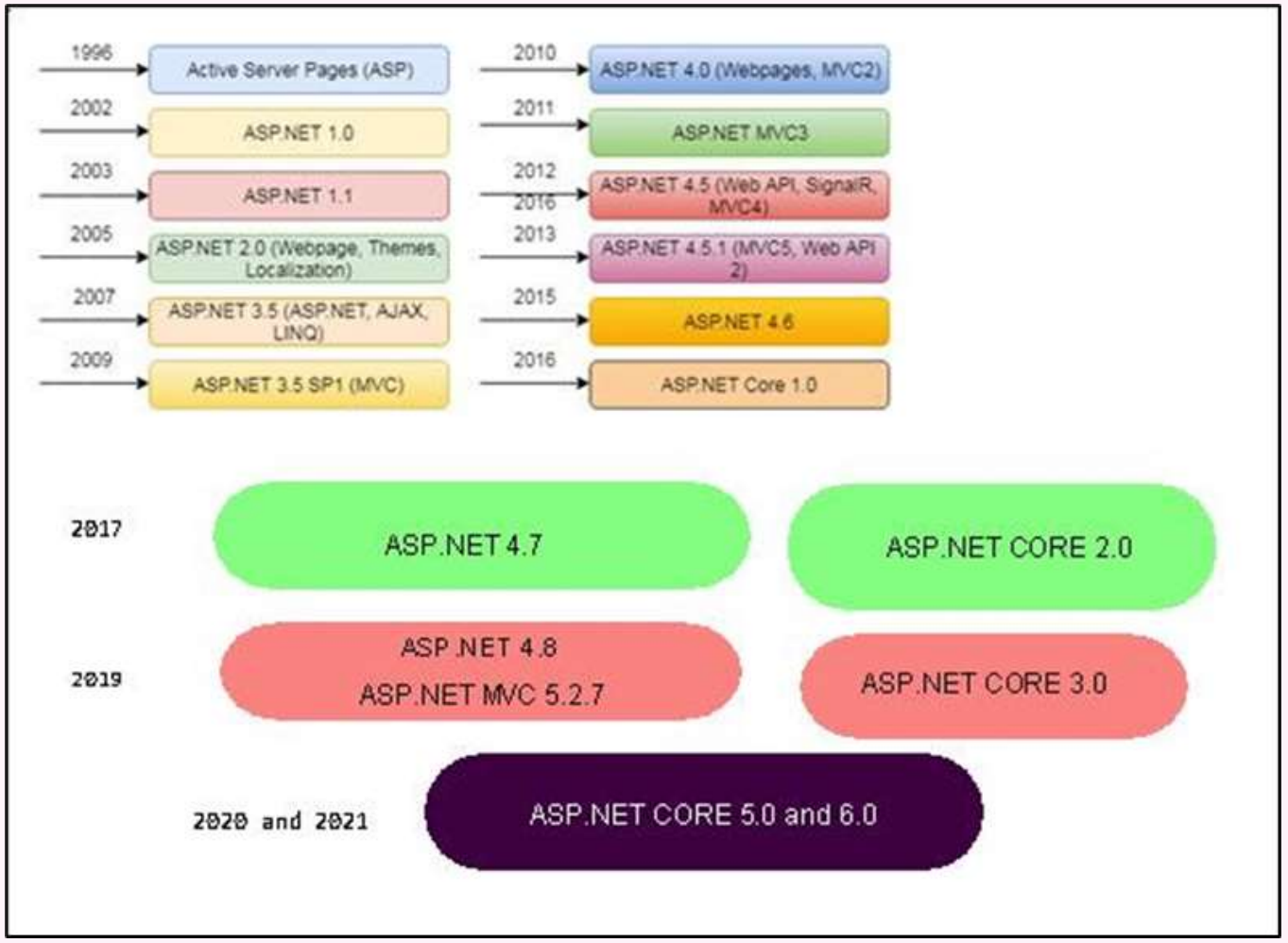
ASP was then replaced by ASP.NET, a technology working on the .NET platform to facilitate modern application development.



ASP.NET Frameworks

History of ASP.NET (1-2)

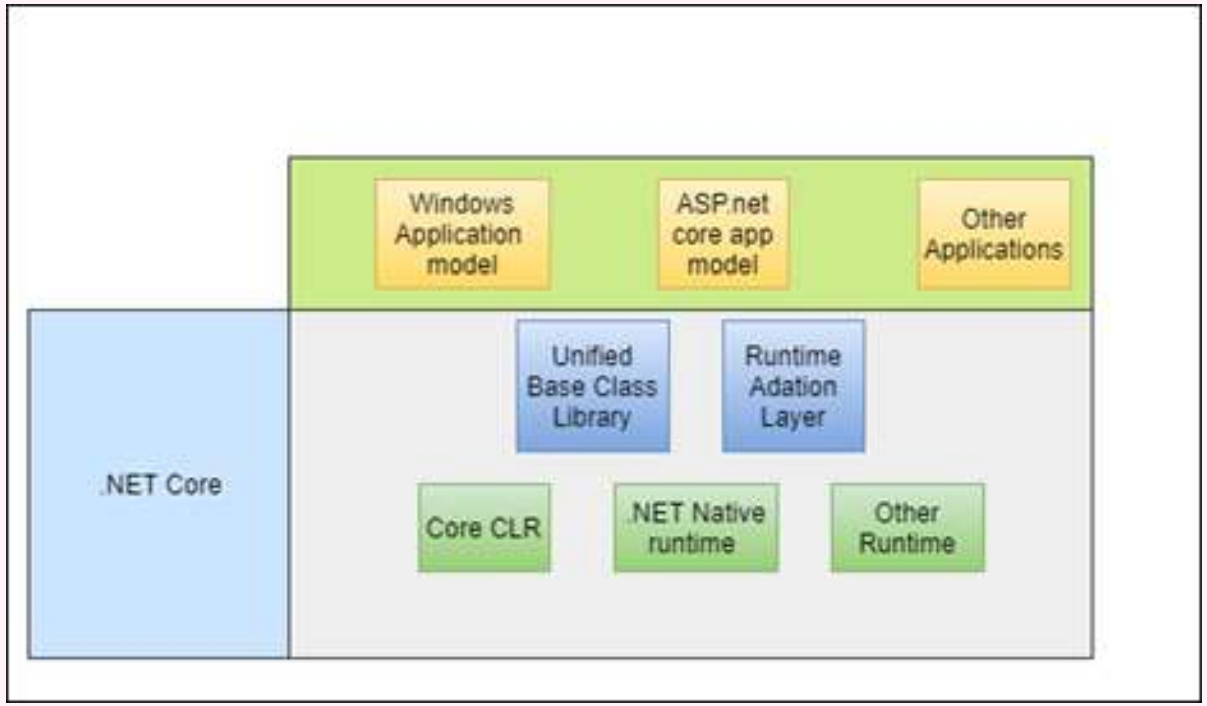
Release History of ASP.NET, ASP.NET MVC, and Core



History of ASP.NET (2-2)

Version	Release Date
.NET 6	November 8, 2021
.NET 5	November 10, 2020
.NET Core 3.1	December 3, 2019
.NET Core 3.0	September 23, 2019

Table 1.1: Versions of .NET and .NET Core



.NET Core Application Model

Uses and Features of ASP.NET (1-2)

Uses of ASP.NET

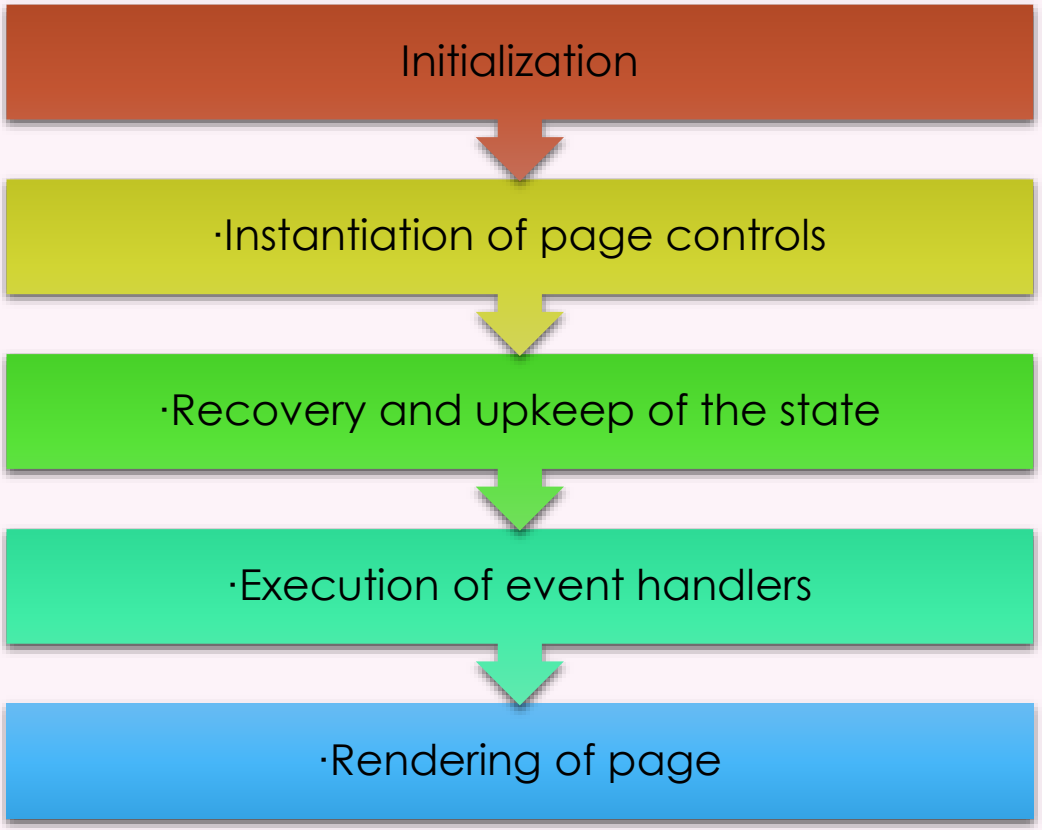
- Facilitates creating Complex applications easily
- Supports both Web-based and desktop-based applications
- Provides enhanced security due to Versatile and dynamic library
- Considerably reduces the code in large Web applications
- Provides What You See Is What You Get (WYSIWYG)
- Offers server controls and blueprints having drag-and-drop facility
- Allows easy modifications due to the separation of both HTML and source code

Uses and Features of ASP.NET (2-2)

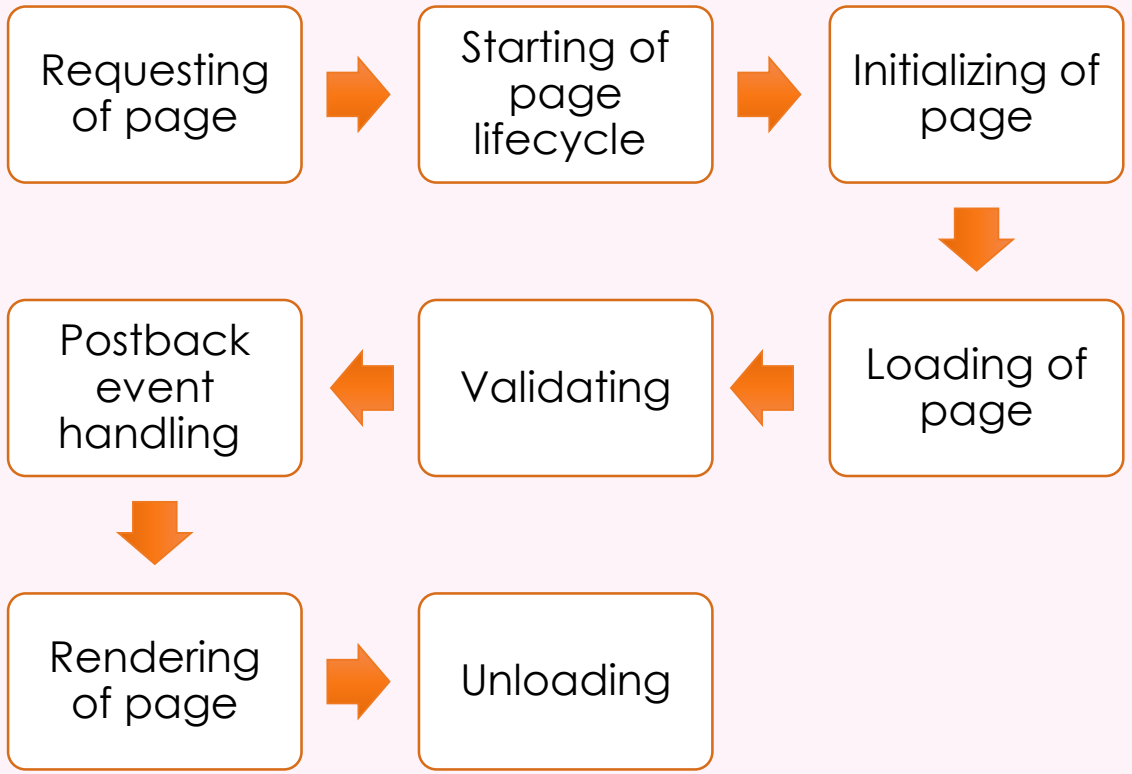
Features of ASP.NET

- **All-Inclusive Software Infrastructure**
- **Abstraction Layer**
- **Supports Multiple Languages**
- **Interactive Data**
- **HTTP Protocol**
- **Visual Studio**

ASP.NET Page Lifecycle (1-5)



Page lifecycle phases

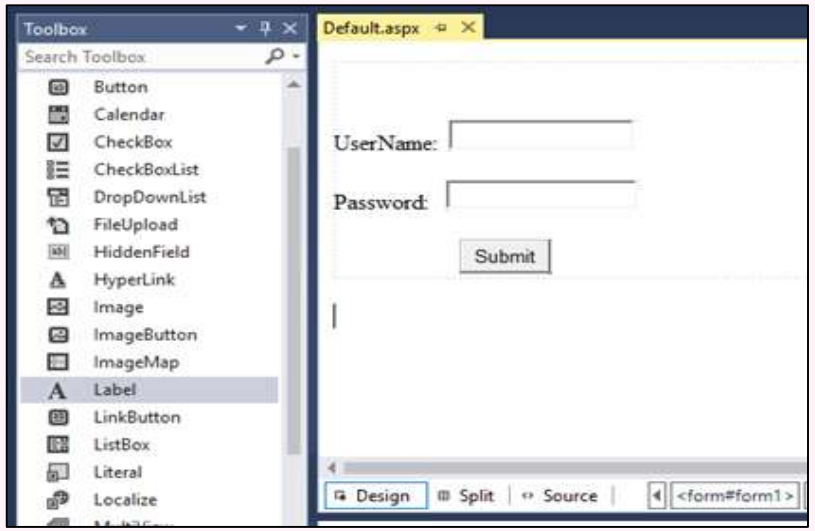


Different stages of an ASP.NET page

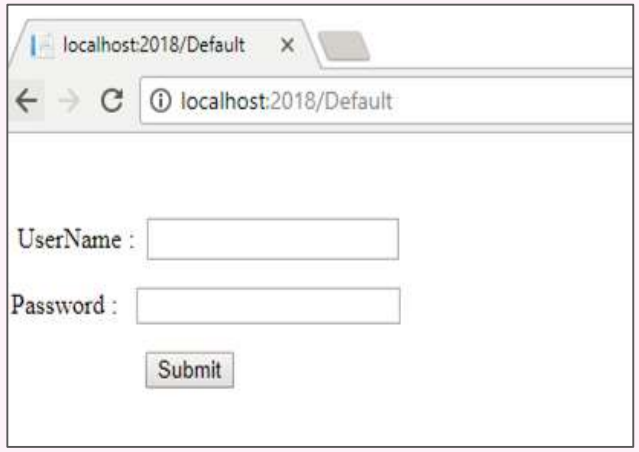
ASP.NET Page Lifecycle (2-5)

```
Default.aspx
1  <head>
2      <style type="text/css">
3          .auto-style1 {
4              margin-right: 0px;
5              margin-top: 7px;
6          }
7      </style>
8  </head>
9  <form id="form1" runat="server">
10     <p>
11          </p>
12     <p>
13          <asp:Label ID="Label1" runat="server" Text="UserName"></asp:Label>
14         : 
15         <asp:TextBox ID="TextBox1" runat="server" CssClass="auto-style1" Height="23px" Width="124px"></asp:TextBox>
16     </p>
17     <p>
18         <asp:Label ID="Label2" runat="server" Text="Password"></asp:Label>
19         :  
20         <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
21     </p>
22     <p>
23                              
24         <asp:Button ID="Button1" runat="server" Text="Submit" />
25     </p>
26 </form>
27
100 %
Design | Split | Source | <form#form1> | <p> | <asp:TextBox.auto-style1#...>
```

Page with Markup Showing the Design Tab



Page *Design*



Sample Output

ASP.NET Page Lifecycle (3-5)

1	Initializing
2	Loading
3	PreRendering
4	Saving
5	Rendering
6	Cleaning up
7	Unloading

Lifecycle of a Page when a New Request is Placed

1	Initializing
2	Loading View State
3	Loading
4	Loading the postback data
5	PreRendering
6	Saving state
7	Rendering
8	Disposing
9	Unloading

Lifecycle of a Page in case of a Postback Event

ASP.NET Page Lifecycle (4-5)

Events connected with the relevant page cycle phases:

Page Initialization
- Page_Init

View State Loading
- LoadViewState

Postback Data Processing
- LoadPostData

Page Loading
- Page_Load

PostBack Change Notification
- RaisePostDataChangedEvent

PostBack Event Handling
RaisePostBackEvent

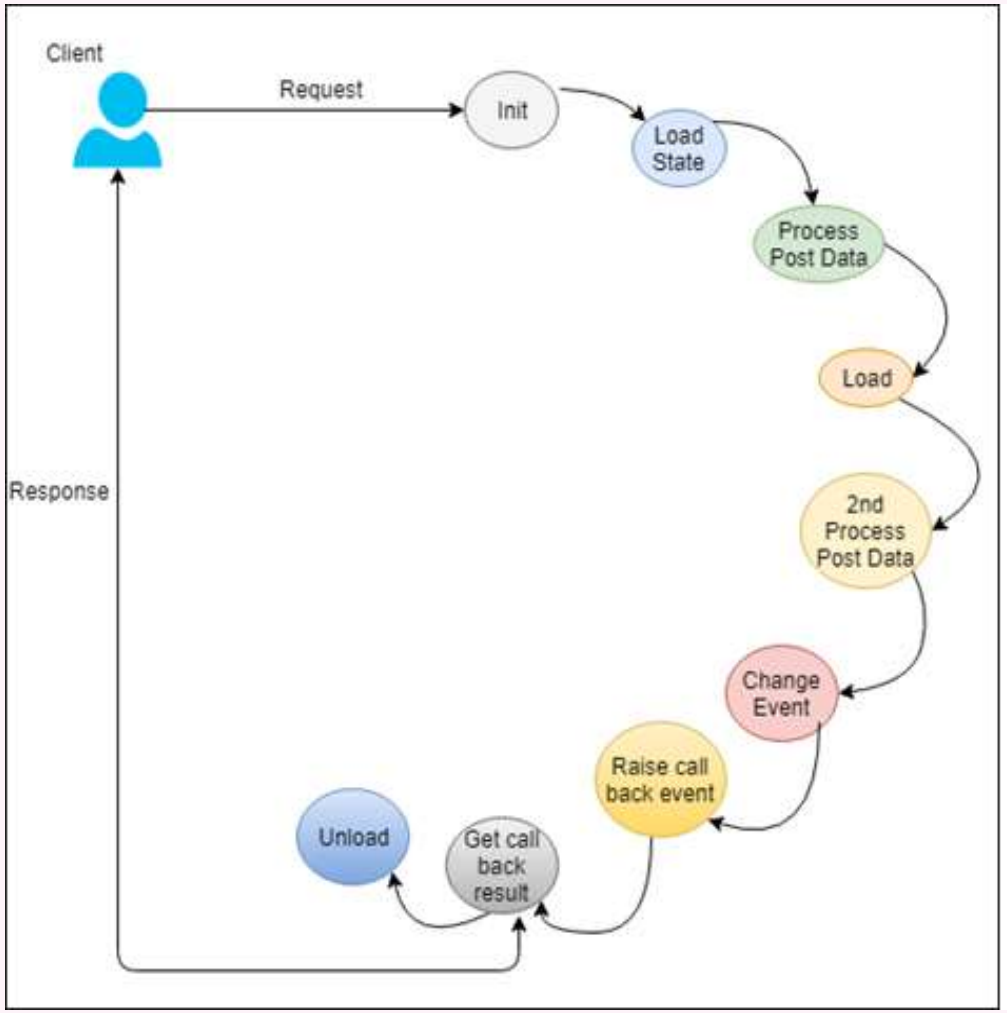
Page PreRendering Phase
Page_PreRender

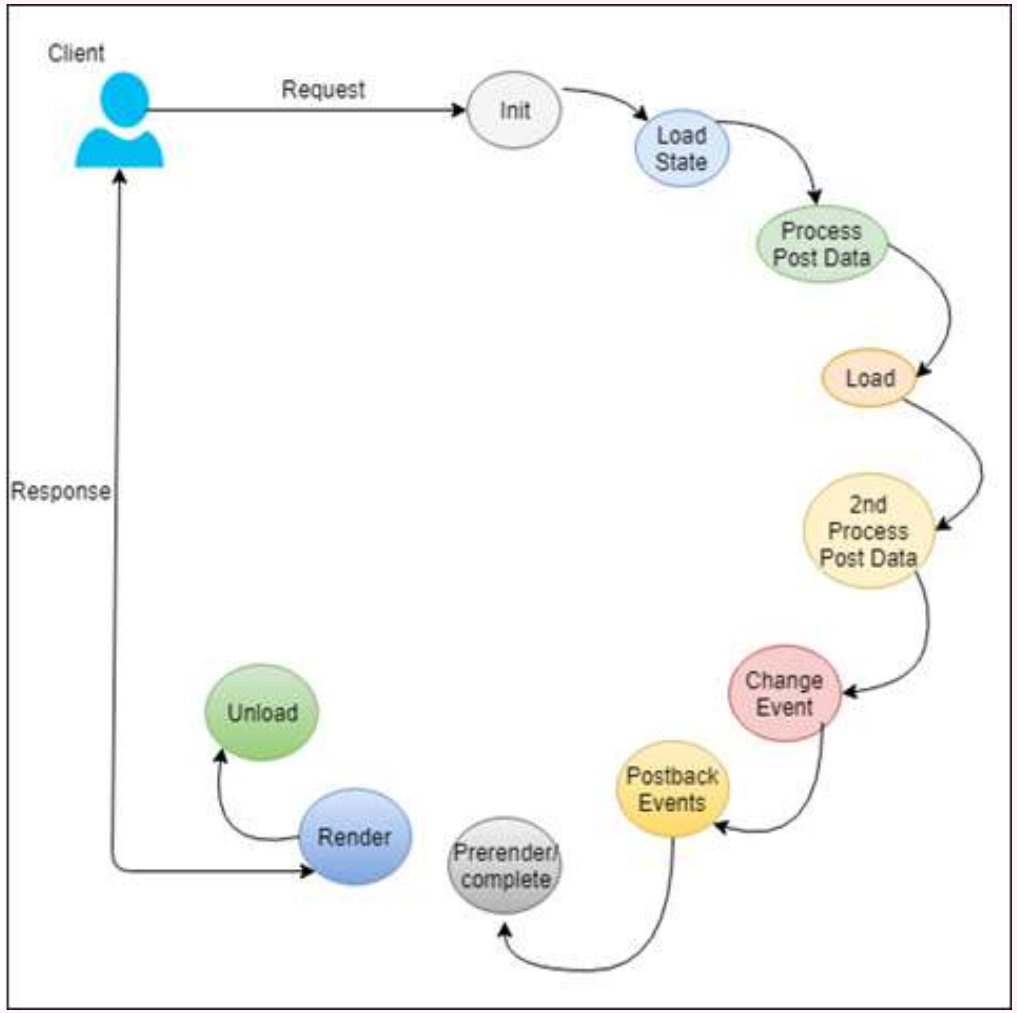
View State Saving
- SaveViewState

Page Rendering
Page_Render

Page Unloading
- Page_UnLoad

ASP.NET Page Lifecycle (5-5)





Processing of Controls (Postback and Callback)

ASP.NET Core Introduction (1-2)

- ASP.NET Core is an open-source and cross-platform framework.
- ASP.NET Core is supported by NuGet packages.
- Facilitates easier development of modern Web apps, including cloud-based applications.
- Apps can be executed on both .NET Core and .NET Framework.
- Due to modular components, developers can retain flexibility.
- They can create and execute apps on Windows, Mac OS, and Linux.

ASP.NET Core Introduction (2-2)

Improvements offered by ASP.NET Core

1. Streamlined Web development
2. A system that is set to work on cloud
3. Good community base
4. An integrated platform for creating a variety of Web applications and APIs
5. Assimilation of latest frameworks
6. Support for a flexible and lightweight HTTP request channel
7. Support for hosting itself in a targeted process or on different platforms
8. Simultaneous versioning of applications

ASP.NET Core Advantages

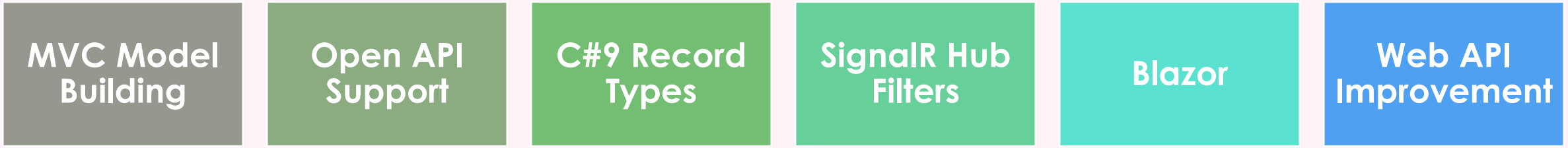
- Supports Dependency Injection (DI)
- Provides cross-platform compatibility
- Has simplified MVC and WEB.API development
- Focuses on increasing productivity
- Offers an open-source environment
- Promotes modularity

Choosing between ASP.NET and ASP.NET Core

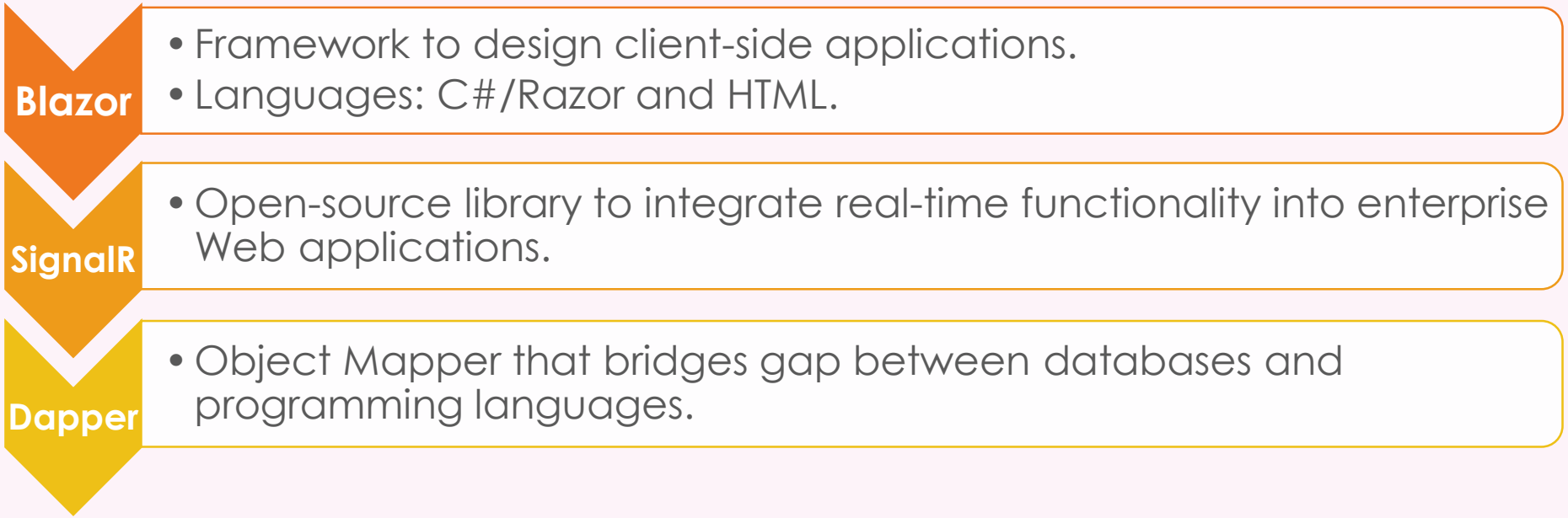
ASP.NET Core	ASP.NET
Apps can be built either for Windows, MacOS, or Linux.	Apps can be built only for Windows.
Recommended when developing a Web UI as of ASP.NET Core 2.x is Razor Pages.	Recommended when developing a Web UI use Web Forms, SignalR, MVC, Web API, or Web pages.
Many versions can be utilized per machine.	Only one version can be utilized per machine.
Apps can be created with the help of Visual Studio, Visual Studio for Mac, or Visual Studio Code using C# or F#.	Apps can be created with the help of Visual Studio using C#, VB, or F#.
Performance is better than ASP.NET.	Performance is less than ASP.NET Core.
Choose either .NET Framework or .NET Core runtime.	Developers must use .NET Framework runtime.

Difference between ASP.NET or ASP.NET Core

Features of ASP.NET Core 5.0



Features of ASP.NET Core 5.0 for performance improvement



Summary (1-2)

- Active Server Pages (ASP) was developed with an aim to generate Web content capable of changing based on the interaction with the user.
- With the help of ASP.NET, developers can integrate important elements of a business Website flawlessly and with a simple code.
- The ASP.NET lifecycle can be categorized into Application Lifecycle and Page Lifecycle.
- The lifecycle of an ASP.NET page is based on whether the page request is new or a post back.
- ASP.NET Core is a new open-source and cross-platform framework that helps developers to create novel cloud-based Internet associated applications.

Summary (2-2)

- ASP.NET is an established framework that offers all the elements necessary to develop enterprise-grade, server-based Web apps on Windows.
- ASP.NET Core helps in developing apps on not just Windows operating system, but also on MacOS and Linux.
- ASP.NET collaborates with popular JavaScript frameworks.
- Blazor is a new .NET Web framework used for developing client-side applications in C#/Razor and HTML.
- Dapper is a straightforward Object Mapper that performs Object-Relational Mapping (ORM).