



Session 1

Introduction to
ASP.NET and
ASP.NET Core

Session Overview

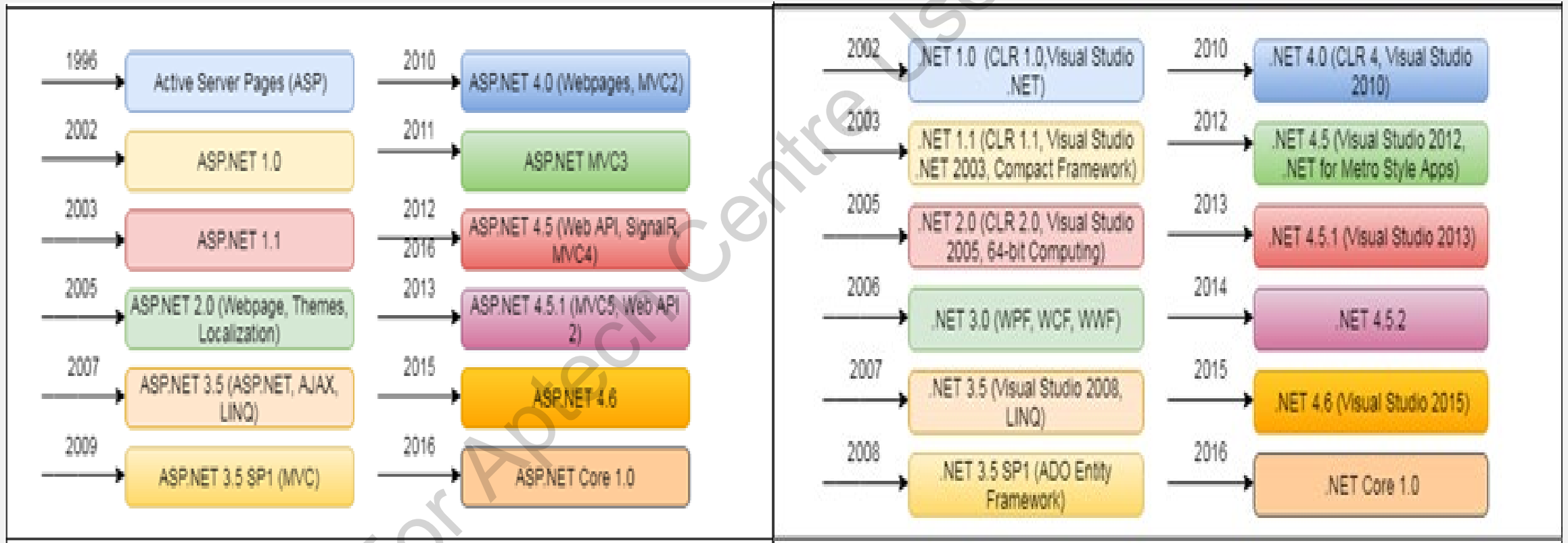
- Explain the ASP.NET Framework
- List the features of ASP.NET and its uses
- Describe the ASP.NET page lifecycle
- Explain ASP.NET Core
- List the advantages of ASP.NET Core
- Explain the difference between ASP.NET and ASP.NET Core

Introduction to ASP.NET

Microsoft developed Active Server Pages (ASP) to facilitate dynamic Web content.

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

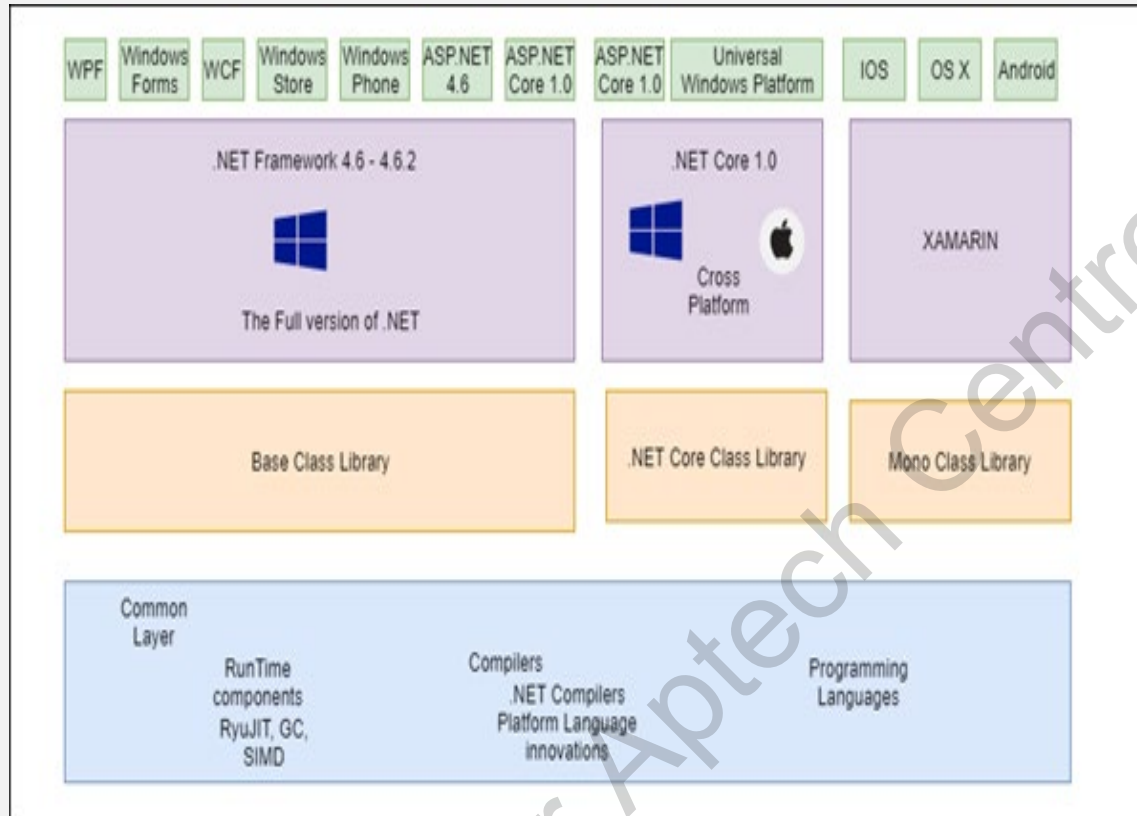
History of ASP.NET (1-2)



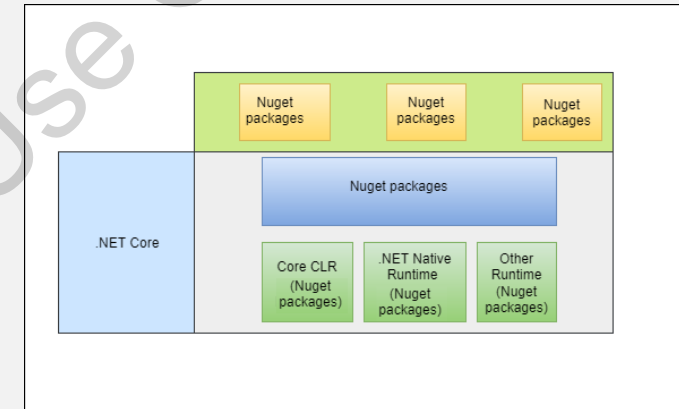
Release History of ASP.NET

Release History of .NET Framework

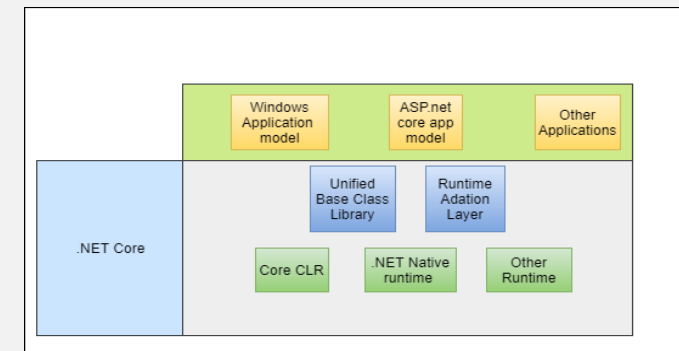
History of ASP.NET (2-2)



.NET Framework and .NET Core



Modular .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

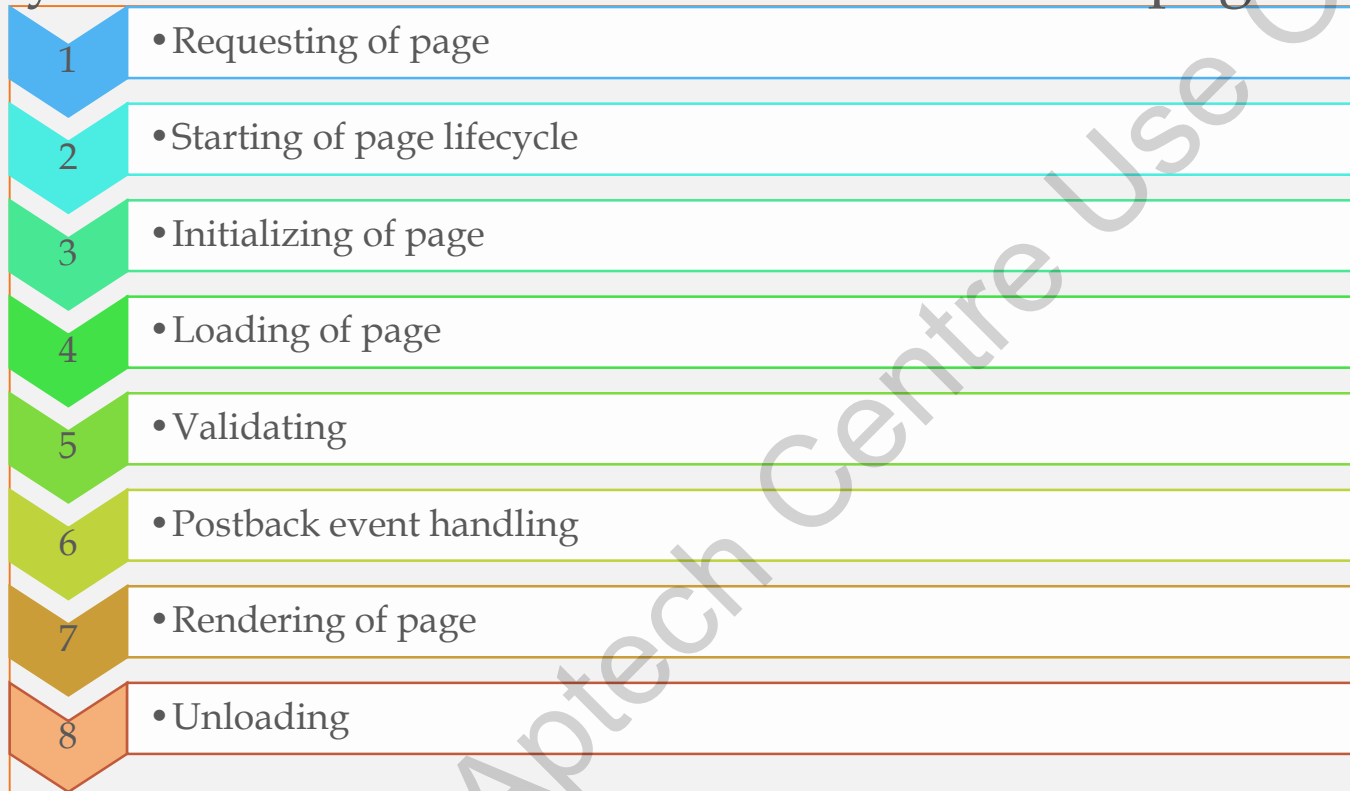
Interactive Data

HTTP Protocol

Visual Studio

ASP.NET Page Lifecycle (1-6)

Page Lifecycle involves how ASP.NET deals with pages to generate results.



Stages of an ASP.NET Page

The lifecycle can be categorized into two types:

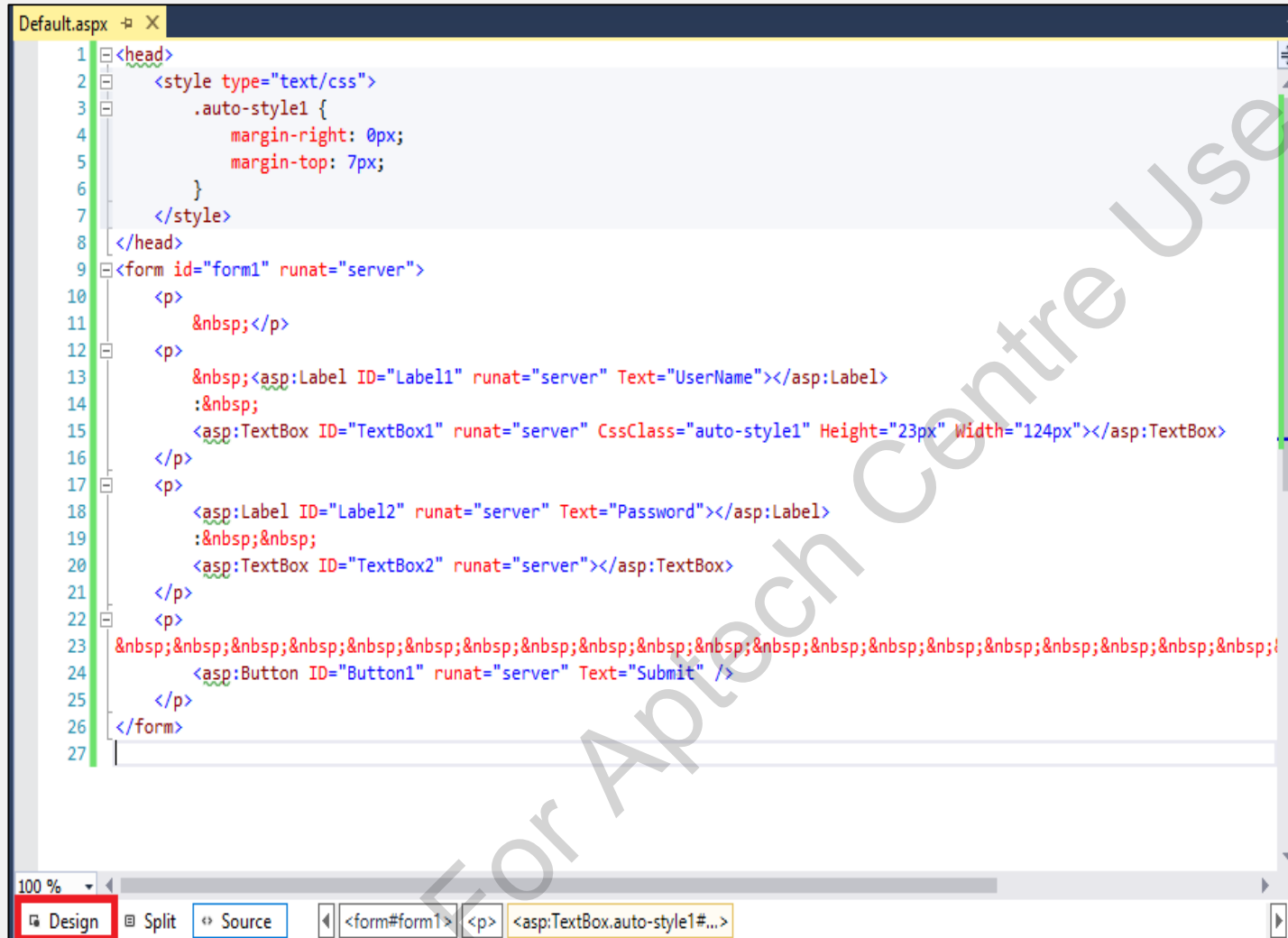
Application
Lifecycle

Page Lifecycle

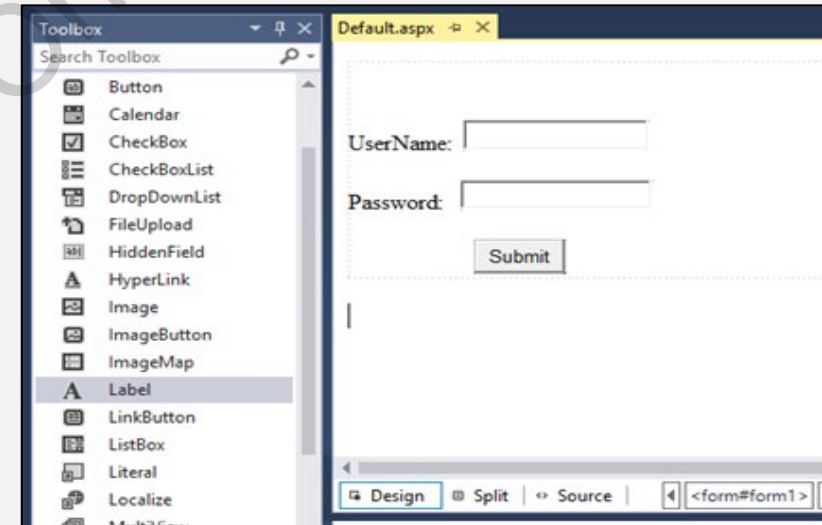
ASP.NET Page Lifecycle (2-6)

Code Snippet 1

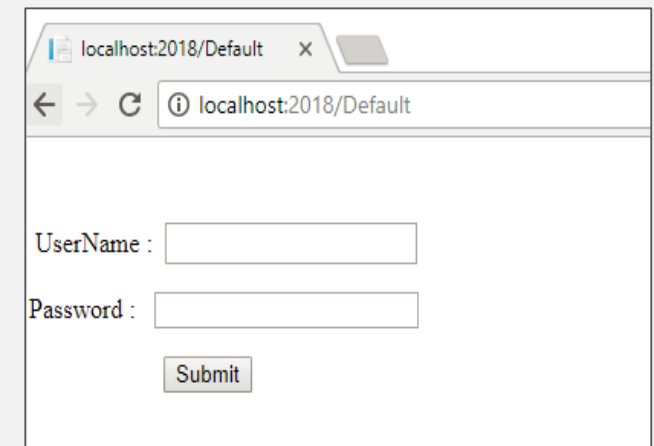
ASP.NET Page Lifecycle (3-6)



Page with Markup Showing the Design Tab



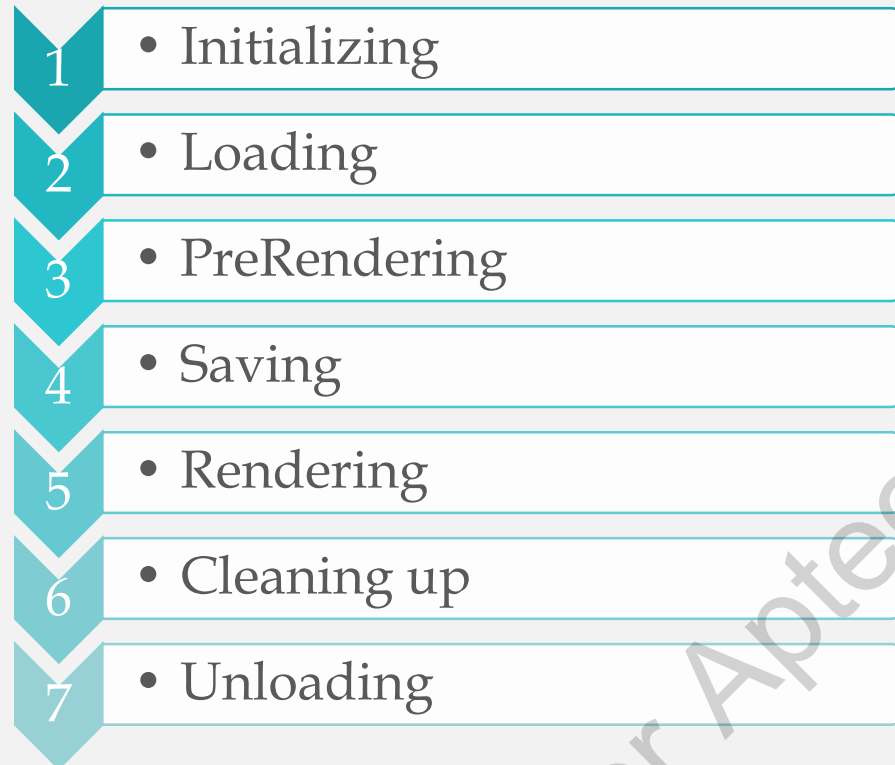
Page Design



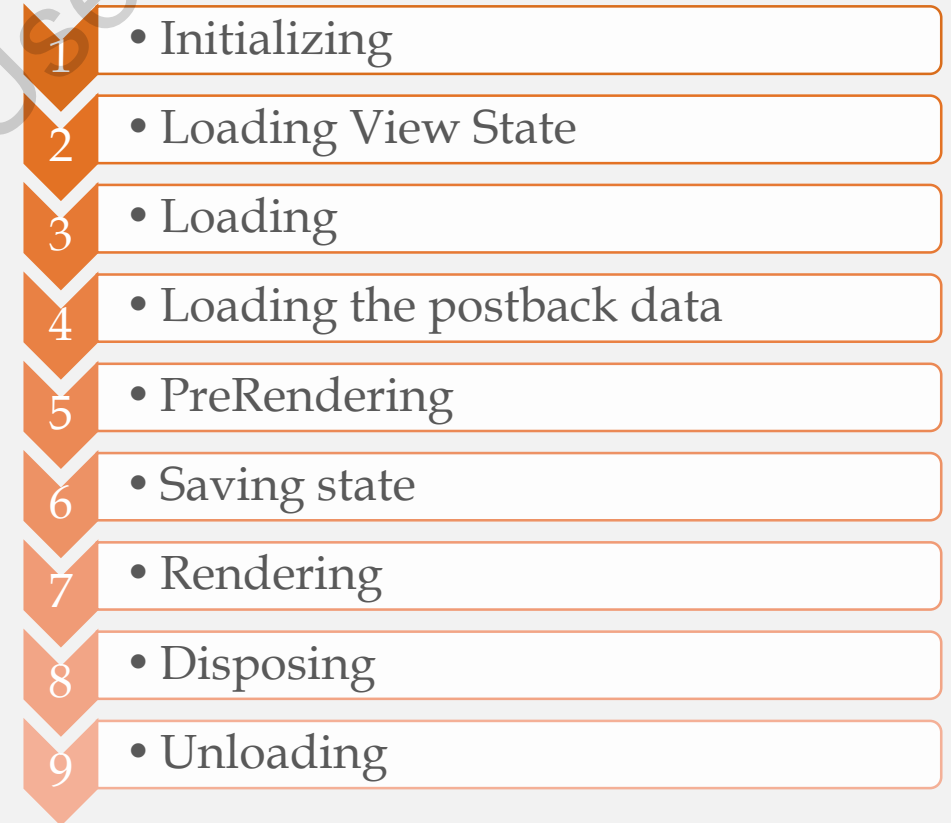
Sample Output

ASP.NET Page Lifecycle (4-6)

Lifecycle of a Page when a New Request is Placed



Lifecycle of a Page in case of a Postback Event



ASP.NET Page Lifecycle (5-6)

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

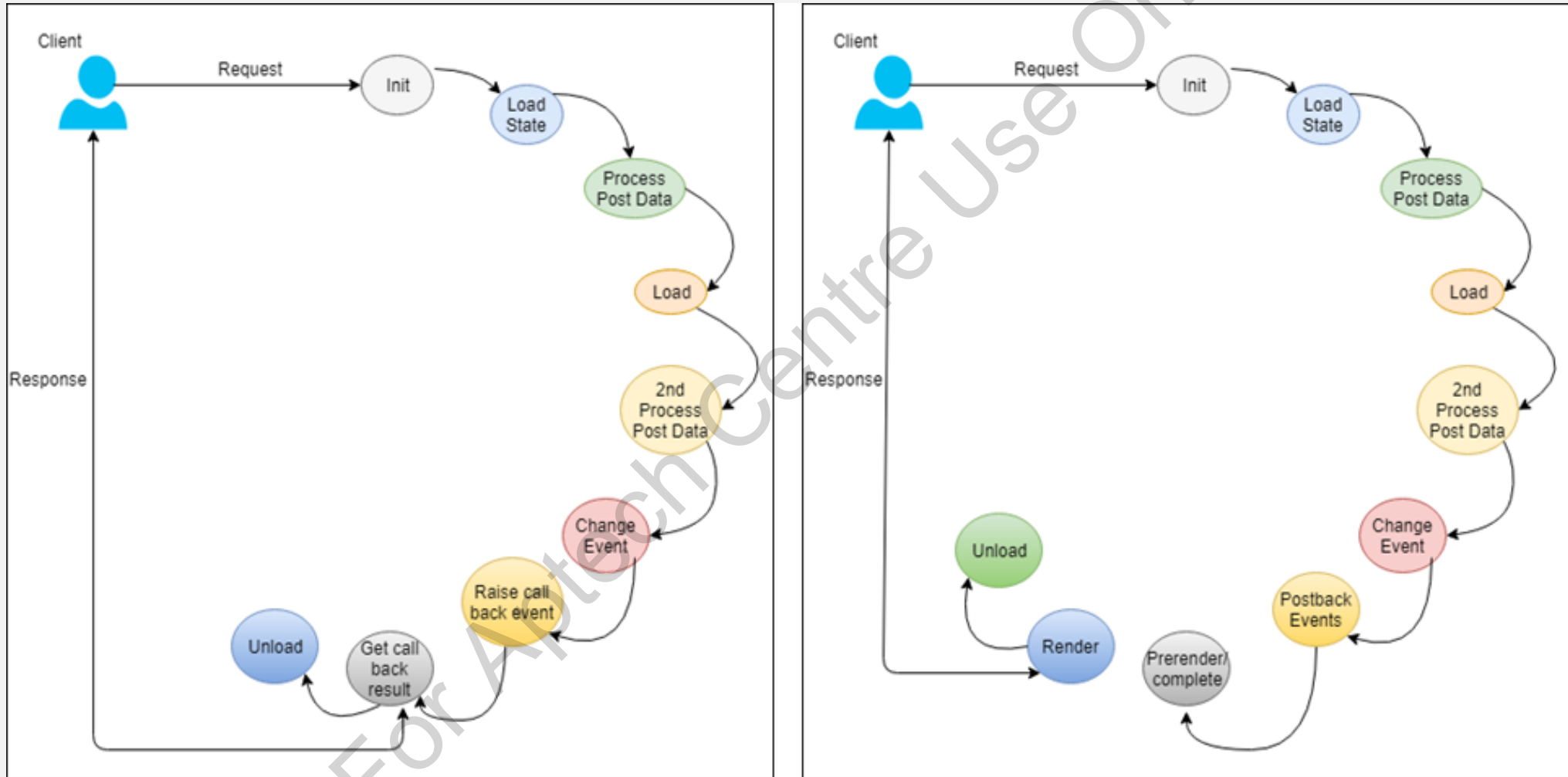
Page Rendering

Page_Render

Page Unloading

- Page_UnLoad

ASP.NET Page Lifecycle (6-6)



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.
- It 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

Summary

- Active Server Pages (ASP) was developed with an aim to generate Dynamic Web content.
- Today, there are two versions of .NET Framework: a logical extension of the present version .NET 4.5.2, officially called .NET 4.6 and .NET Core 1.0.
- 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 postback.
- ASP.NET Core is a open-source and cross-platform framework that helps to create novel cloud-based Internet associated applications.
- ASP.NET on the .NET Core 1.0 Framework offers the most adaptable version of ASP.NET.
- 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.