Before dot net (1990-1998) the software platform are ------

1. **VB:** Visual basic also known as Visual Basic Classic is introduced in 1991; it is considered the third generation of event-driven programming languages and integrated development environment (IDE). VB is derived from the BASIC programming language and is considered to be event-driven and object-oriented.

Visual Basic use for ----

1. Rapid application development (RAD) of graphical user interface (GUI) applications.
2. Access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects.
3. Creation of ActiveX controls and objects.
4. **VC++:** Visual C++ was Microsoft's implementation of a professional Windows hosted IDE for developing Windows software. Visual C++ 1.0 was the first release of Visual C++, released in 1993 for 16-bit development.
5. **ASP:** ASP stands for Active Server Pages which is known as classic ASP. Classic ASP is a server-side scripting environment that you can use to create and run dynamic web applications. With ASP, you can combine HTML pages, script commands, and COM components to create interactive web pages that are easy to develop and modify. Classic ASP is the predecessor to ASP.NET, but it is still in wide use today.

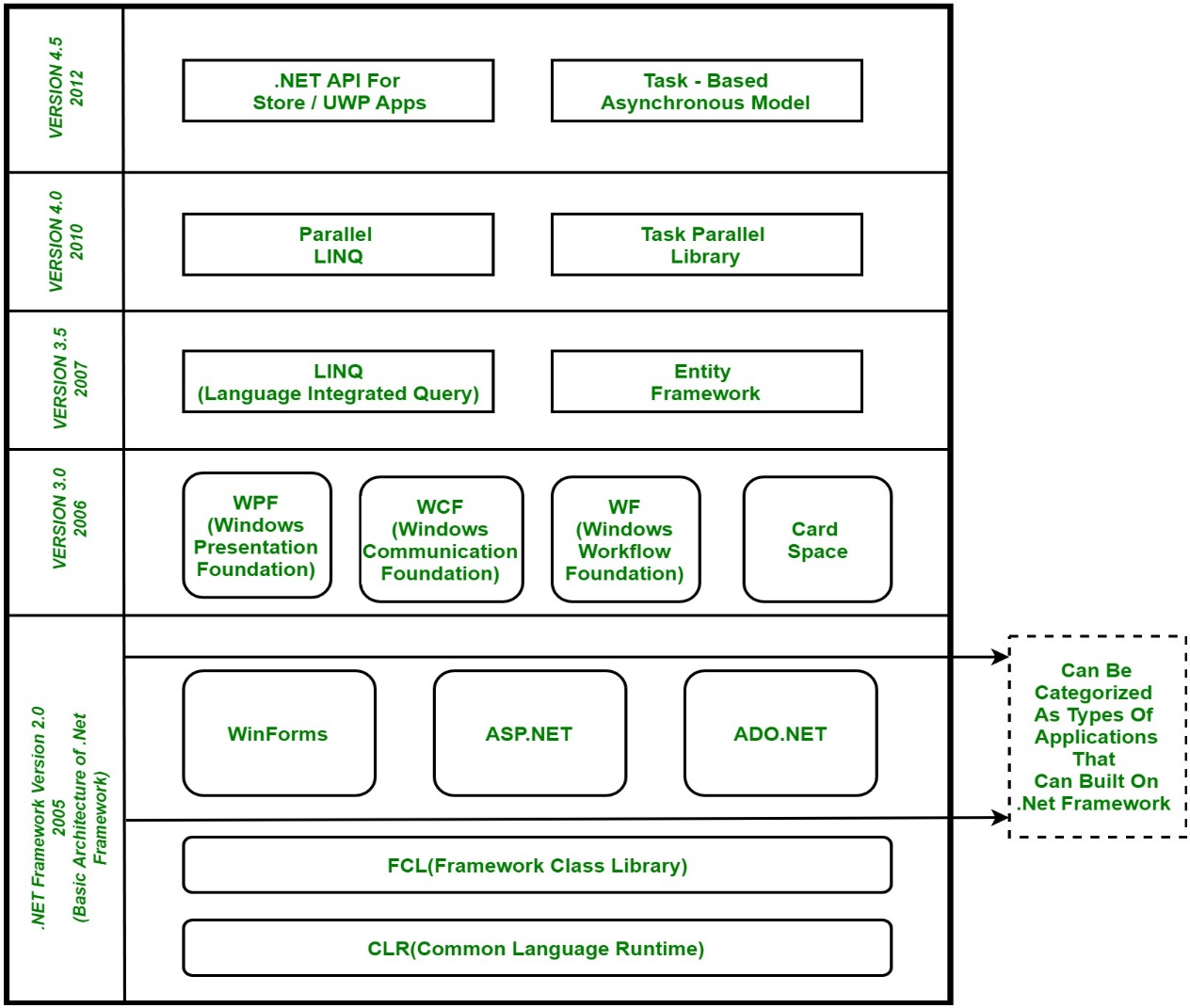
# .NET

1. After coming java as platform independent Microsoft coming with .Net framework.
2. .Net will support for multiple programming language.
3. .net framework can be written as VB, C#, Jscript various language.

## **Components of .NET Framework**

1. CLR (Common Language Runtime)
2. CTS (Common Type System)
3. BCL (Base Class Library)
4. CLS (Common Language Specification)
5. FCL (Framework Class Library)
6. .NET Assemblies
7. XML Web Services
8. Window Services

## **Version of .NET:**



1. Version- 1.0 (2002):

* Initial release.
* Introduced CLR 1.0
* Use of DLLs as class libraries.
* OOP support for Web development.

1. Version- 1.1 (2003):

* Added **ASP.NET** Control support for Mobile device development.
* Added support for **ADO.NET** classes for Oracle database and ODBC database connectivity.
* IP6 and fixed issue to Code Access Security for ASP.NET.

1. Version- 2.0 (2005):

* New CLR 2.0
* Enhancement of **ASP.NET & ADO.NET,**
* Generics Types, Partial Types, Anonymous methods, Nullable Types, Iterators,
* Covariance and Contravariance
* 64bit support we added.

1. Version- 3.0 (2006):

* Windows Presentation Foundation (WPF),
* Windows Communication Foundation (WCF),
* Windows Workflow Foundation (WWF).

1. Version- 3.5 (2007):

* Build-in-Support **for AJAX,**
* **Language Integrated Query (LINQ),**
* Expression trees, HashSet collections, WCF and WF integration
* Peer-to-Peer networking
* **ASP.NET MVC**

1. Version- 4.0 (2010):

* New CLR 4.0,
* Task Parallel Library (TPL),
* Managed Extensibility Framework (MEF),
* Dynamic Language Runtime (DLR) were a major addition of this release.

1. Version- 4.5 (2012):

* Async support,
* Support for Windows Store app,
* Enhancements of WPF, WCF, WF, MEF, and ASP.NET, and base framework classes, such as support for arrays larger than 2GB on 64-bit platform.

1. Version- 4.5.1 (2013):

* Performance and debugging improvements,
* support for automatic binding redirection, and
* expanded support for Windows Store application.

1. Version- 4.5.2 (2014) :

* ASP.NET APIs enhancements,
* System DPI support for Windows Forms controls,
* Event Tracing for Windows (ETW) and New Workflow features and
* Debugging improvements were added in this release.

1. Version- 4.6 (2015):

* ASP.NET enhancements,
* ADO.NET always an encrypted feature for SQL Server 2016,
* new **64-bit JIT compiler,**
* Assembly Loader improvements,
* enhancements to Garbage Collector

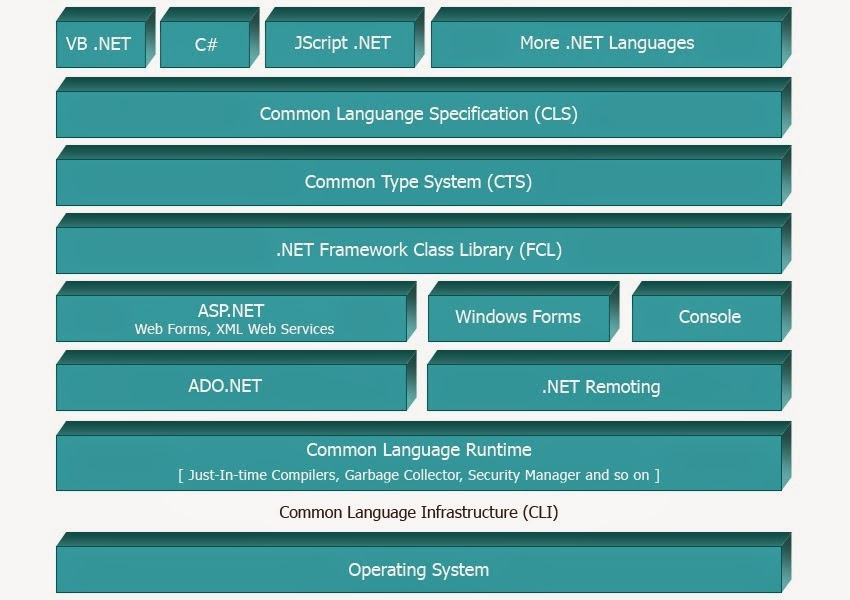
1. Version- 4.7 (2017):

* High DPI support for Windows Forms controls,
* Touch support for WPF in Windows 10,

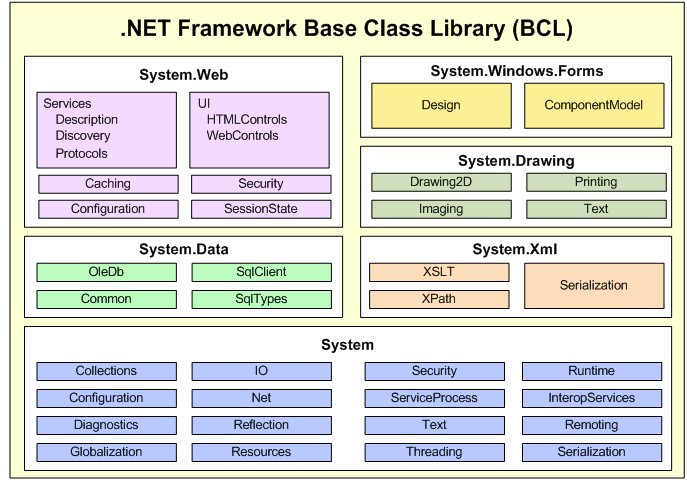
1. Version- 4.8 (2019):

* JIT improvements, Updated ZLib,
* FIPS improvements,
* Malware scanning for Assemblies and Accessibility Enhancements

## **.NET framework architecture**



## **Base Class Library .net Framework**

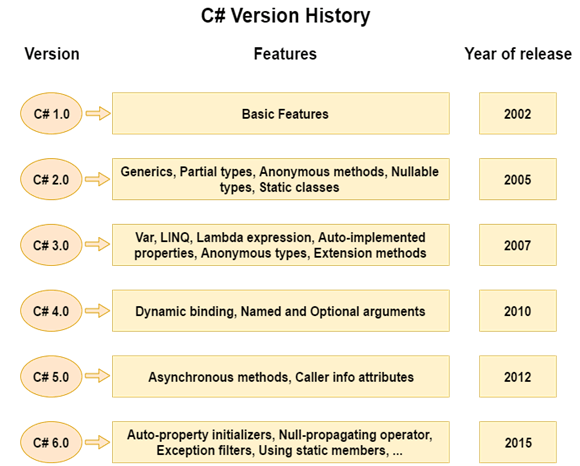


# C#.NET

C# is pronounced as "C-Sharp". It is an object-oriented programming language provided by Microsoft that runs on .Net Framework. By the help of C# programming language, we can develop different types of secured and robust applications:

* Window applications
* Web applications
* Distributed applications
* Web service applications
* Database applications etc.

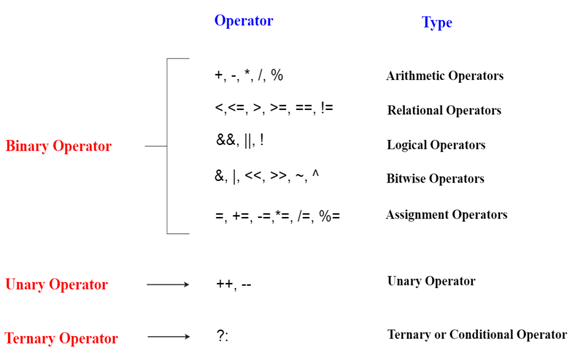
**Version:**



#### Data Types:

[](https://www.tutorialsteacher.com/Content/images/csharp/datatypes.png)

#### Operator:



#### Output / input:

Output: Console.WriteLine(“ hello world ”);

Input: Console.ReadLine();

#### Control Statement :

* If-else:
* Switch:
* For loop
* While loop
* Do while loop
* Break
* Continue
* Goto :

#### Function:

#### Array

#### Class

#### Properties:

1. **Sealed keyword:**
2. Sealed class: sealed class is a class that cannot inheritance by other class. But sealed class can be extending other class. Sealed class declared by sealed keyword.

Code:

sealed class Person{

string name;

}

Class User : Person {

// cannot extends Person class

}

1. Sealed method: Sealed keyword is preventing the further override the child override method.

We cannot override the sealed method. Sealed keyword always use in child class method.

Code:

Class A{

Int x = 10;

public void show(){

Console.WriteLine(x);

}

}

Class B : A{

Public sealed override void show(){

Console.WriteLine(x);

}

}

Class C : B{

public override void show(){

Console.WriteLine(x);

}

}

#### Abstraction

#### Namespace

#### File I/O

#### Collection:

1. **Generic:**

#### Delegates:

#### Reflection:

#### Multithread:

#### Synchronization:

#### Web service

# XML

* Xml stands for Extensible Markup Language.
* XML is a makeup language much like HTML.
* XML was designed to store and transport data.
* XML was designed to be self-descriptive.
* XML was designed to be both human and machine readable.
* XML is used to structure data which are created by user.
* Xml case sensitive.

Code:

<user>

<name> shuvo </name>

<age> 25 </age>

</user>

**XML Structure:**

<root>  
  <child>  
    <subchild>.....</subchild>  
  </child>  
</root>

### Feature of XML:

1. XML Element:

<user>

<name> shuvo </name>

<age> 25 </age>

</user>

1. **XML attribute:**

* XML elements can have attributes, just like HTML. Attribute values must always be quoted.
* attributes cannot contain multiple values (elements can)
* attributes cannot contain tree structures (elements can)
* attributes are not easily expandable (for future changes)

<person gender="female">

<name> shuvo </name>

</person>

1. **XML Namespace:** XML Namespaces provide a method to avoid element name conflicts.
2. Name conflicts in XML can easily be avoided using a name prefix.

<h:table>  
  <h:tr>  
    <h:td>Apples</h:td>  
  </h:tr>  
</h:table>  
  
<f:table>  
  <f:name>African Coffee Table</f:name>  
</f:table>

1. **xmlns Attribute:** When using prefixes in XML, a namespace for the prefix must be defined. The namespace can be defined by an xmlns attribute in the start tag of an element.

<h:table xmlns:h="http://www.w3.org/TR/html4/">  
  <h:tr>  
    <h:td>Apples</h:td>  
    <h:td>Bananas</h:td>  
  </h:tr>  
</h:table>

# .NET WINFORM

# ADO.NET

**Configure .net framework.**

1. Connect the data base from “Data Sources” Menu.
2. Configure the “App.config” xml file.

<connectionStrings>

<add name =”dbcs” connectionString=”” providerName =”System.Data.SqlClient”>

</connectionStrings>

1. Now create an object of configurationManager class.

string cs = ConfigurationManager.ConnectionStrings[“dbcs”].connectionString;

SqlConnection con = new SqlConnection(cs);

**Select :**

1. Sql query for select:

String sql = “select \* from customer where cid=@cid”;

1. Create command object :

SqlCommand cmd = new SqlCommand(query, con);

1. Value add in insert query:

cmd.Parameters.AddWithValue(“@cid”,1364);

1. Connection open.

Con.Open();

1. Execute query.

SqlDataReader rd = Cmd.ExecuteReader();

1. Close the connection

Con.close;

**Inset/update/delete:**

1. Sql query:

String query = “insert into customer values(@cid, @cname);

1. Create Command object of the query.

SqlCommand cmd = new SqlCommand(query, con);

1. Value add in insert query:

Cmd.Parameters.AddWithValue(“@cid”,1364);

Cmd.Parameters.AddWIthValue(“@cname”,”shuvo”);

1. Open the connection

Con.Open();

1. Execute the query.

Int a = Cmd.ExecuteNonQuery();

1. Close the connection

Con.close();

# LINQ

# Entity Framework

# ASP

Asp are creating for server slide application. With duration it is develop in two different platforms.

1. Classic ASP
2. ASP.NET

### Classic ASP:

1. ASP stands for Active Server Pages which is known as classic ASP.
2. Classic ASP is old but make a server-side scripting environment which create dynamic web applications.
3. With ASP, you can combine HTML pages, script commands, and COM components to create interactive web pages that are easy to develop and modify.
4. Classic ASP is the predecessor to ASP.NET, but it is still in wide use today.
5. An ASP file has the file extension ".asp"
6. An ASP file is just the same as an HTML file
7. An ASP file can contain server scripts in addition to HTML
8. Server scripts in an ASP file are executed on the server
9. Edit, change, add content, or customize any web page
10. Respond to user queries or data submitted from HTML forms
11. Provide web security since ASP code cannot be viewed in a browser

**Language:** classis asp scripting language is VBScript.

**Output**: The Response.Write() method is used by ASP to write output to HTML.

<% Response.Write("Hello World!") %>

**Variables**: <% Dim x(2,2) %>

### ASP.NET

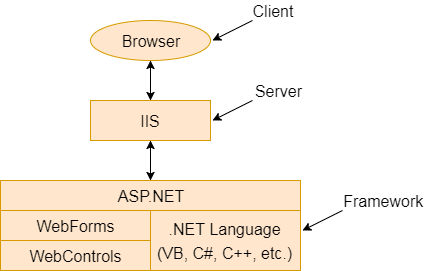
1. Asp.net is a web development platform, which provides a programming model, a comprehensive software infrastructure for web application. It is a server-side Scripting language. It browser independent.
2. Asp.net was released in 2002 as successor to classic ASP.

3 programming models for creating ASP.NET web sites and web applications.

1. ASP.net web form
2. ASP.NET MVC
3. ASP.NET Web Pages

# ASP.net Web Forms:

1. Web Forms is the oldest ASP.NET programming model, with event driven web pages written as a combination of HTML, server controls, and server code.
2. Web Forms are compiled and executed on the server, which generates the HTML that displays the web pages.
3. In Web form we can use VB or C# as programming language.
4. ASP.NET web forms contain various web pages and GUI applications such as text box, data grid, label, checkbox, hyperlink, etc.
5. It provides flexibility to web pages at run time as well as design time. It also provides a feature to write code in a separate file from the controls
6. Web Forms are made up of two components: the visual portion (the ASPX file), and the code behind the form, which resides in a separate class file.



**Features of Web Forms:**

1. **Server Controls:**

It provides a vast set of server controls. These controls are like objects, and they run when they are requested and rendered to the browser. Some web pages are similar to HTML elements like text-box, button, checkbox, and hyperlink.

Label: < asp:LabelID="Label\_1" runat="server" Text="Label" > </asp:Label>

Text Box: < asp:TextBoxID="Text\_Box" runat="server" > </asp:TextBox>

1. **Default page:**

Default page is initial state of asp.net web form where server is start.

1. **Mater Pages:**

Mater Pages is responsible for the consistent layout of our web applications. It gives a proper appearance and standard to different pages.

1. **Working with data:**
2. **Ispostback:**
3. **QueryString:**
4. **Membership:**
5. **Client Script and Client Frameworks**
6. **State Management:**
7. ViewState:

ViewState[“user”] = UserTextBox.Text;

1. ApplicationState:
2. sessionState:
3. **Reapter Control:**
4. **Error Handling**
5. **Routing:**

URL routing can be configured to a web application. A request URL is a URL that a user enters in a browser to browse in a specific place.

1. **Security:**

Security always plays a crucial role in software development. ASP.NET provides different configuration options and extensibility points to make our systems more secure.

**Code :**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="W.aspx.cs"   Inherits=" " %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<style type="text/css">

.auto-style1 {

     width: 100%;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<table class="auto-style1">

 <tr>

   <td> <asp:Label ID="Label1" runat="server" Text="Name"></asp:Label>  </td>

  <td> <asp:TextBox ID="name" runat="server" required="true"></asp:TextBox> </td>

 </tr>

 <tr>

   <td> <asp:Label ID="Label2" runat="server" Text="Password"></asp:Label></td>

   <td><asp:TextBox ID="T2" runat="server" TextMode="Password"></asp:TextBox></td>

 </tr>

 <tr>

   <td> <asp:Label ID="Label5" runat="server" Text="Select Course"></asp:Label>s</td>

    <td>

  <asp:CheckBox ID="CheckBox1" runat="server" Text="J2SEE" />

  <asp:CheckBox ID="CheckBox2" runat="server" Text="J2EE" />

   </td>

  </tr>

<tr>

 <td><asp:Button ID="Bnt" runat="server"  CssClass="btn" OnClick="Click"/> </td>

  </tr>

</table>

</form>

</body>

</html>

# ASP.NET MVC:

**The MVC (Model-View-Controller)** is an application development pattern or design pattern which separates an application into three main components:

1. Model
2. View
3. Controller
4. Model: Model is a part of the application which implements the logic for the data domain of the application. It is used to retrieve and store model state in a database such as SQL Server database. It also used for business logic separation from the data in the application.
5. View: View is a component that forms the application's user interface. It is uses to create web pages for the application. An example would be an edit view of a Products table that displays text boxes, drop-down lists and check boxes based on the current state of a Product object.
6. Controller: Controller is the component which handles user interaction. It works with the model and selects the view to render the web page. In an MVC application, the view only displays information whereas the controller handles and responds to the user input and requests.

# ASP.NET WEB PAGE:

ASP.NET Web Pages is a framework that you can use to create dynamic web pages. It provides fast and lightweight way to combine server code with HTML. It helps to add video, links to the social sites. It also provides other features like you can create beautiful sites that conform to the latest web standards.

**Razor Markup:**

1. Razor is a simple markup syntax for embedding server code (C# or VB) into ASP.NET web pages.
2. C# code blocks are enclosed in @{ ... }
3. Inline expressions (variables or functions) start with @

<!DOCTYPE html>  
<html lang="en">  
<head>  
     <meta charset="utf-8" />  
     <title>Web Pages Demo</title>  
</head>  
<body>  
     <h1>Hello Web Pages</h1>  
     <p>The time is @DateTime.Now</p>  
</body>  
</html>

**Rezor Engine**: Rezor engine convert the rezor syntax in html format. Then

**Webpage layout:** In web page we can break the code, in block of content and we can reuse this block separate file. like headers and footers, in separate files.

1. @RenderBody() :

Folder:

Form :

Object:

# .NET CORE

.

NET Core is a new version of .NET Framework, which is a free, open-source, general-purpose development platform maintained by Microsoft. It is a cross-platform framework that runs on Windows, macOS, and Linux operating systems.

.NET Core Framework can be used to build different types of applications such as mobile, desktop, web, cloud, IoT, machine learning, micro services, game, etc.

.NET Core is written from scratch to make it modular, lightweight, fast, and cross-platform Framework. It includes the core features that are required to run a basic .NET Core app. Other features are provided as NuGet packages, which you can add it in your application as needed. In this way, the .NET Core application speed up the performance, reduce the memory footprint and becomes easy to maintain.

**.NET CORE feature:**

* 1. ASP.NET Core
  2. Universal Windows app

**Version of .NET Core:**

1. **.NET 6:**

* 2021
* Visual Studio 2022
* Hot Reload improvements, RyuJIT compiler, and runtime performance boost,and early builds of MAUI,the multi-platform UI support based on Xamarin

1. **.NET 5**

* 2020
* Visual Studio 2019 V 16.8
* Uniform runtime behavior with a single.NET runtime that can be used everywhere.
* A development platform for everything from Linux and Windows to iOS and Android to TV and watches and Web Assembly.

1. **.NET Core 3.1**

* 2019
* Visual Studio 2019 V 16.4
* Issue fixes &  improvements of .Net Core 3.0
* Long Term Support (LTS) for three years.

1. **.NET Core 3.0**

* 2019
* Visual Studio 2019 V 16.3
* Improved overall Performance.
* Improved API performance

1. **.NET Core 2.0**

* 2017
* Implements.NET Standard 2.0,
* Supports 6 new distros,
* RyuJIT x86 JIT in Core 2.0,
* Dotnet restore is an implicit command.

1. **.NET Core 1.0**

* 2016
* Visual Studio 2015
* First time release,
* open-source, Cross-platform, has flexible deployment,
* uses ASP.NET Core and UWP

# ASP.NET CORE MVC