

## Bài 1

# Tổng quan về .NET Framework và C#

# Nội dung

- Giới thiệu
- Lịch sử phát triển
- Application Types and Technologies
- Công cụ phát triển
- Tạo ứng dụng Csharp đầu tiên

# Giới thiệu

- Những năm gần đây, .NET đã trở thành một hệ sinh thái mạnh để tạo ra bất kỳ loại ứng dụng nào trên nền tảng Windows. Với .NET bạn có thể tạo các loại ứng dụng như Windows apps, web services, web applications, Microsoft Phone....
- Phiên bản mới nhất của .NET có sự thay đổi lớn so với các phiên bản trước, có lẽ là sự thay đổi lớn nhất từ khi nó ra đời. Hầu hết mã .NET là open source và bạn có thể tạo các loại ứng dụng cho các nền tảng khác nhau. Phiên bản mới nhất của .NET là .NET Core, Microsoft cung cấp chu kỳ cập nhật nhanh chóng các tính năng mới thông qua hệ thống quản lý các gói “NuGet packages”.

# Lịch sử phát triển

.NET	CLR	C#	Visual Studio IDE
1.0	1.0	1.0	2002
1.1	1.1	1.2	2003
2.0	2.0	2.0	2005
3.0	2.0	2.0	2005+Extension
3.5	2.0	3.0	2008
4.0	4.0	4.0	2010
4.5	4.0	5.0	2012
4.5.1	4.0	5.0	2013
4.6	4.0	6	2015
.NET Core 1.0	CoreCLR	6	2015+Extension

# C# 1.0 Ngôn ngữ mới

- C# 1.0 là một ngôn ngữ hoàn toàn mới được thiết kế cho .NET Framework, tại thời điểm nó ra đời, .NET Framework gồm khoảng 3,000 lớp và CLR.
- C# được thiết kế bởi Anders Hejlsberg và chịu nhiều ảnh hưởng nhiều từ C++, Java và Pascal.
- Do C# tạo ra sau nên Microsoft đã phân tích những nhược điểm của các ngôn ngữ khác và đã có những cải tiến đáng kể khi tạo ra C# (*Tham khảo Page 4/Part I/ The C# Language / Professional C# 6 and .NET Core*)

# C# 2.0 và .NET 2.0 với Generics

- Với phiên bản này đã có sự thay đổi lớn về cả ngôn ngữ C# và mã IL, lý do là generics đã được đưa vào. Với Generics, nó có thể tạo các kiểu mà không cần biết các kiểu được sử dụng bên trong là gì? Các kiểu bên trong được định nghĩa tại thời điểm tạo đối tượng. Generics ra đời kéo theo hàng loạt các kiểu mới ra đời, đặc biệt là các lớp generics collection nằm trong namespace *System.Collections.Generic* và các collection cũ ít được sử dụng hơn.
- Chúng ta sẽ tìm hiểu kỹ hơn về Generics trong các chương sau

# .NET 3.0 với WPF

- Với .NET 3.0 ra đời, thì C# không có gì thay đổi, chỉ là bổ sung thêm các thư viện và nhiều kiểu mới, các namespaces với khoảng 12000 các kiểu. Trong đó có một số phần mới như:
- Windows Presentation Foundation (WPF) có lẽ là phần mới nhất bổ sung vào để tạo các ứng dụng Windows desktop.
- Windows Communication Foundation (WCF): hợp nhất tất cả các công nghệ Web Service, Remoting, Message Queue.... vào một, tuy nhiên chúng ta cần cấu hình khi sử dụng.
- Windows Workflow Foundation(WF): cho phép định nghĩa, thực thi và quản lý các luồng công việc.

# C# 3.0 và .NET 3.5 với LINQ

- .NET 3.5 ra đời kèm theo C# 3.0. Cải tiến chính trong phiên bản này là cú pháp truy vấn viết trực tiếp trong C# (LINQ), LINQ kết hợp với biểu thức lambda cho phép làm việc với các loại dữ liệu Collection, XML, and Database một cách dễ dàng hơn. Đặc biệt là không phải thay đổi lại mã IL.
- Để truy cập database và tạo các câu truy vấn, LINQ to SQL ra đời và phiên bản đầu tiên của Entity Framework cũng ra đời, cả hai đều đưa ra cơ chế ánh xạ với database quan hệ.



# C# 4.0 và .NET 4.0 với Dynamic and TPL

- Sự thay đổi trong phiên bản C# 4.0 là đưa ngôn ngữ tích hợp động giúp cho việc tích hợp các thành phần COM vào ứng dụng .NET một cách dễ dàng hơn, từ khóa sử dụng là `dynamic`.
- Một sự thay đổi bên trong .NET Framework là làm việc với multi-core CPUs, hỗ trợ lập trình song song với thư viện Task Parallel Library(TPL).
- Với những sự thay đổi của C# 4.0 dẫn đến .NET Runtime cũng được chuyển từ phiên bản 2.0 lên 4.0
- Với sự ra đời của visual studio 2010, một công nghệ lập trình web mới cũng được ra đời là ASP.NET MVC 2.0

# C# 5.0 và .NET 4.5 với Asynchronous Programming

- C# 5.0 chỉ có 2 từ khóa mới là await và async. Tuy nhiên nó giúp việc lập trình các phương thức bất đồng bộ dễ dàng hơn. Đặc biệt với sự ra đời của Windows 8 thì việc đưa lập trình bất đồng bộ vào để tránh việc khóa UI Thread khi sử dụng.
- Windows 8 cũng giới thiệu giao diện lập trình mới là Windows Store Apps kèm theo Windows Runtime. Các control WPF cũng cải tiến lại cho phù hợp với Runtime mới.
- Một phiên bản của giao tiếp độc lập trên nền tảng cũng được đưa ra là ASP.NET WEB API

# C# 6.0 và .NET Core

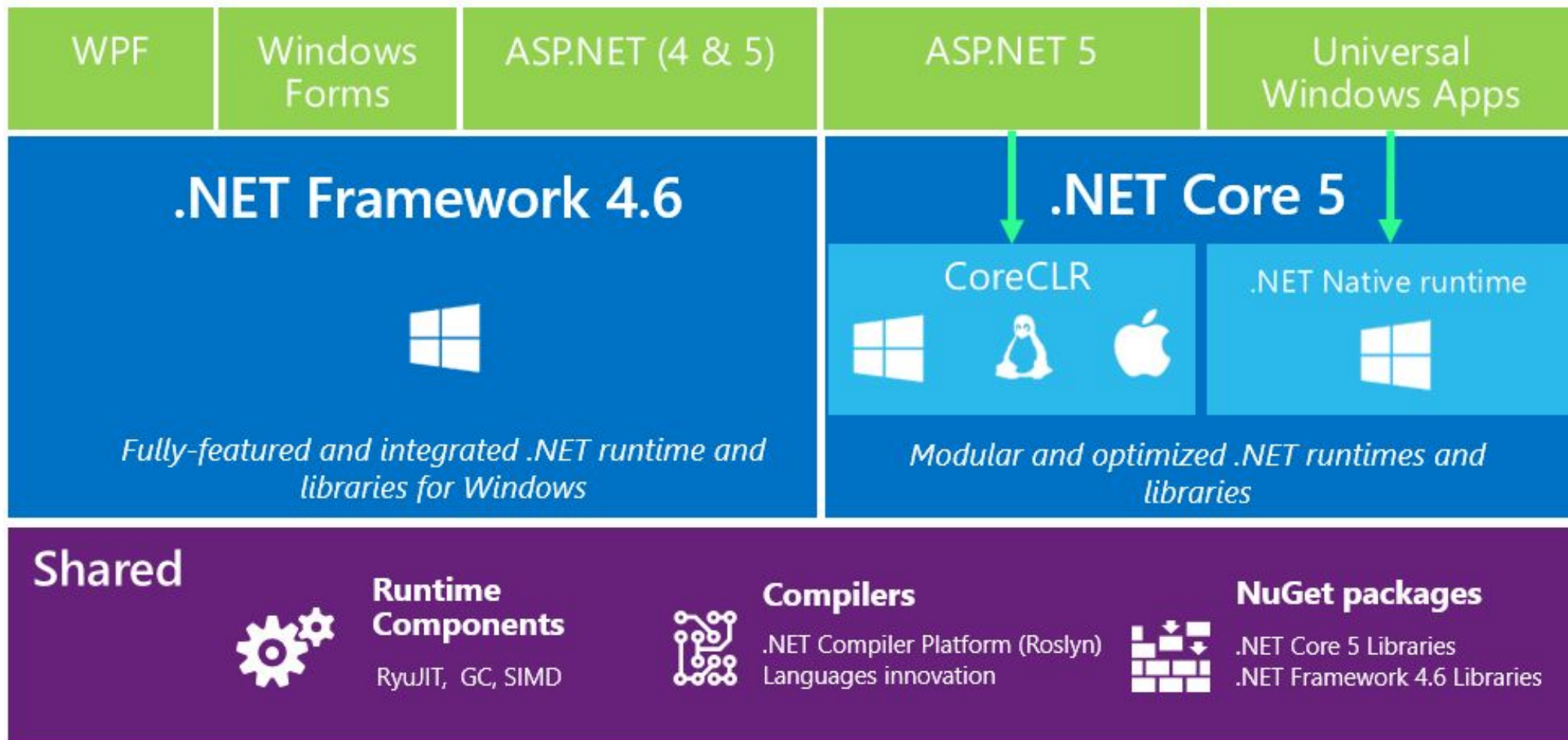
- C# 6.0 ra đời với một vài cải tiến nhỏ và thiết thực giúp giảm lượng code ở một vài nơi đồng thời trình compiler engine cũng được cải tiến với tên Roslyn.
- Với các phiên bản .NET trước bao gồm rất nhiều các Framework nhỏ theo từng mảng và kèm theo nhiều phiên bản nên việc quản lý sẽ trở lên khó khăn. Chính vì vậy việc đưa ra một version mới là cần thiết (.NET Core)
- .NET Core là một framework nhỏ hơn với các modular NuGet packages, và có một runtime được phân phối với mọi ứng dụng, nó là open source không chỉ có sẵn trên Windows mà còn cho nhiều thiết bị Windows cũng như cho Linux và OS X.

# C# 6.0 và .NET Core

- Để tạo các ứng dụng Web, ASP.NET Core 1.0 được viết lại hoàn toàn so với bản ASP.NET trước. Do vậy nó không tương thích hoàn toàn với các phiên bản cũ và ASP.NET MVC trước cũng được thay ASP.NET MVC 6. Với ASP.NET MVC 6 có overhead thấp hơn, performance tốt hơn, nó có thể chạy trên Linux. ASP.NET Web Forms không còn được hỗ trợ trên phiên bản này.

# .NET 2015

- .NET 2015 là một thuật ngữ bao hàm tất cả các công nghệ .NET



# .NET Framework 4.6

- NET Framework 4.6 là Framework liên tục được cải tiến trong 10 năm qua. Framework này được sử dụng để tạo các ứng dụng Windows Forms và WPF applications. ASP.NET 5 có thể chạy trên .NET Core và cũng có thể chạy trên .NET Framework 4.6. Nếu các bạn muốn tiếp tục làm việc với ASP.NET Web Forms trên .NET Framework 4.6, bạn hãy tìm hiểu các tính năng mới của ASP.NET 4.6 nhé. Chú ý bạn không thể chuyển từ .NET Core với ASP.NET Web Forms

# .NET Core 1.0

- .NET Core 1.0 là một phiên bản với tất cả các công nghệ mới, Framework này là *open source*. Runtime là *CoreCLR*. Không giống như .NET Framework bạn cần phải chỉ ra version cho ứng dụng cài trên hệ thống, với .NET Core 1.0, bao gồm runtime được phân phối kèm với ứng dụng. Với các phiên bản trước khi triển khai ứng dụng web bạn phải phụ thuộc vào phiên bản cài đặt trên Web Server. Tuy nhiên với .NET Core 1.0 nó được thiết kế ở dạng modular và được phân chia đặt trong NuGet packages, với ứng dụng bạn cần gì thì sẽ cài vào ứng dụng cái đó.

# Assemblies

- Assembly được biết đến là các thư viện ở dạng tệp tin (.DLL) hoặc tệp thực thi (.EXE).
- Trong Assembly chứa code ở dạng Intermediate Language (IL), code này đã được biên dịch tới phiên bản .NET Framework cụ thể.
- Assembly tự mô tả chính nó và có thể lưu trữ trên nhiều tệp tin, trường hợp lưu trữ trên nhiều tệp tin thì nó sẽ có một tệp chính chứa entry point và mô tả các tệp khác trong assembly.
- Cấu trúc assembly được sử dụng cho cả tệp thực thi và tệp thư viện, chỉ khác là assembly thực thi chứa đựng chương trình chính có entry point, trong đó assembly library thì không. Với .NET Framework 4.6, các assembly có hai kiểu: *private* và *shared* assemblies.
- Shared assemblies không áp dụng cho Universal Windows Platform bởi vì tất cả code được biên dịch thành một hình ảnh gốc (native image)



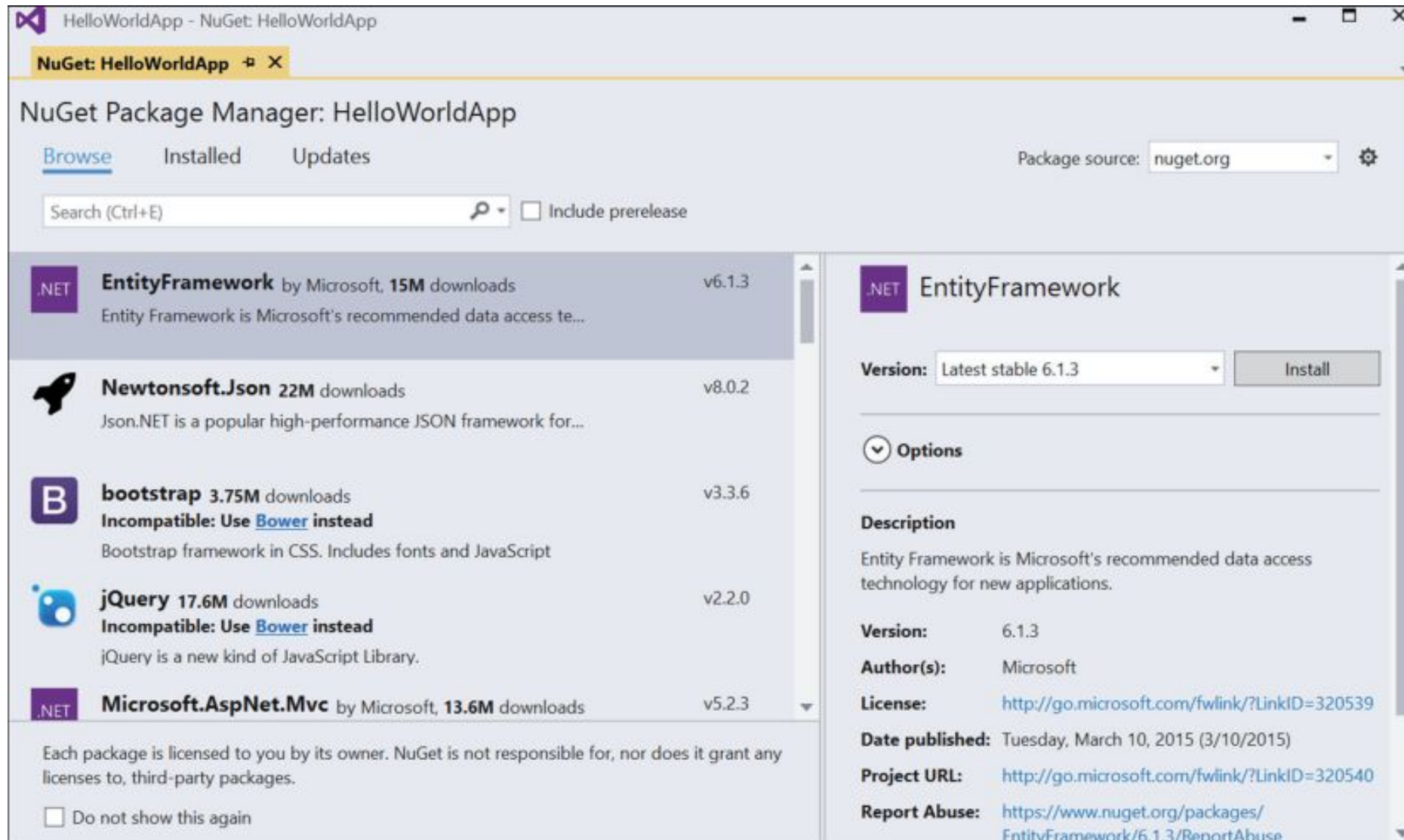
# Private và Shared Assemblies

- Private Assemblies là các assemblies được cung cấp kèm với phần mềm và chỉ sử dụng cho phần mềm đó.
- Shared Assemblies là các assemblies được cung cấp ở dạng thư viện chung, bất kỳ ứng dụng cũng có thể sử dụng nó

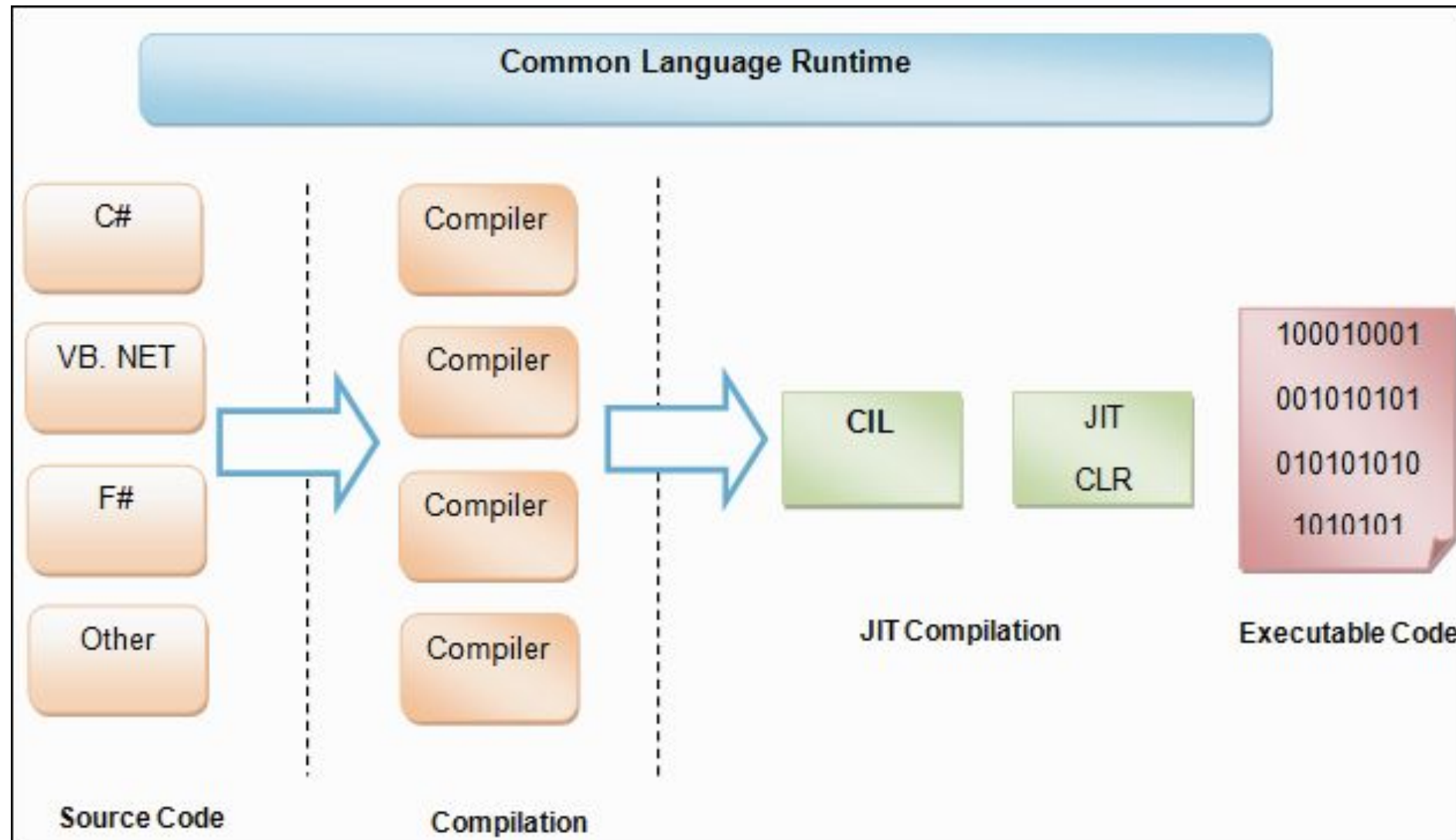
# NuGet packages

- Trước đây, để sử dụng lại các assemblies trong ứng dụng bạn cần phải add reference chúng vào ứng dụng, có thể có nhiều assembly rời rạc và khi đó có thể bạn cần thay đổi một số thông tin cấu hình bằng tay, dễ sai sót và mất thời gian và khó khăn, chính vì vậy việc đóng gói các assembly là cần thiết.
- NuGet packages ra đời giúp việc phân phối các assembly trở lên dễ dàng hơn. NuGet package là một tệp zip chứa 1 hoặc nhiều tệp assembly cũng như thông tin cấu hình và PowerShell scripts, nó được cung cấp bởi Microsoft và các hãng thứ 3, thể loại rất đa dạng...

# NuGet packages



# Common Language Runtime (CLR)



# .NET Native

- .NET Native là một tính năng mới của .NET 2015, nó sử dụng để biên dịch các Windows Store Apps thành Native code thời gian khởi động ứng dụng nhanh hơn 60% và giảm 15-20 việc sử dụng bộ nhớ. Hiện tại .NET Native cũng có trên cả các ứng dụng .NET Core, bạn có thể biên dịch các ứng dụng trên .NET Core để chạy trên cả Windows và Linux. Tất nhiên bạn cần những hình ảnh khác nhau cho mỗi nền tảng.

# Windows Runtime

- Bắt đầu với Windows 8, hệ điều hành Windows đưa ra một framework khác là Windows Runtime. Nó được sử dụng bởi Windows Universal Platform, version 1 đầu tiên cung cấp theo Windows 8, version 2 theo Windows 8.1, và version 3 với Windows 10. Không giống .NET Framework, Framework này được tạo sử dụng native code. Windows Runtime có thể được sử dụng với các ngôn ngữ như JavaScript, C++, và các ngôn ngữ .NET, Windows Runtime cung cấp các thư viện tổ chức thành các namespace bắt đầu với Windows.

# Application types and Technologies

- Data Access
  - Files and Streams
  - XML and JSON
  - ADO.NET
  - ADO.NET Entity Framework
- Windows Desktop Applications
  - Windows Forms
  - Windows Presentation Foundation

# Application types and Technologies

- Universal Windows Platform
  - Windows Store Apps
  - Windows Phone
- SOAP Services with WCF
- Web Services with the ASP.NET Web API
- Webhooks and SignalR
- Windows Services
- Web Applications
- Microsoft Azure



# Developers Tools

- Visual Studio Community
- Visual Studio Professional with MSDN
- Visual Studio Enterprise with MSDN
- Visual Studio Code

# Tạo ứng dụng C# đầu tiên

[XEM LIVE DEMO](#)

# HỎI ĐÁP





# TRẢI NGHIỆM THỰC HÀNH