

1. Introduction

Introduction(簡介)

1.

Hi My name is Kevin. 我是一個技術移民到澳洲的軟體工程師。個人專注於各種技術和使用介面開發與研究。【C#(.NetFramework)(深入解析)】是一門銜接一門的「.Net Framework 學程」中其中一個課程。

C#是由微軟所開發的程式語言，它具備跨平台、物件導向的特性。能夠開發應用程式、Web Application、Mobile App。也是.Net Framework 的必修課程。

聽過.NET 技術，想成為「.NET Developer」，卻不知道從何學起嗎？

我的 C#課程將完整告訴您 C#的大小事。透過大量的範例，針對 C#各項特性，快速地給您更清晰的 C#觀念，蹲好馬步、打好底子，奠定未來的實戰基礎。明白「為何使用」，以及「為何不使用」。

想朝「.NET Developer」前進嗎？受夠了像是文字天書，艱深難懂的工具文件嗎？對自己的底子沒信心嗎？若你想扎實打底，本課程將讓你收穫滿滿！歡迎加入「IT 工具人學院，帶你學到會」，一起練功吧!!顆顆。

Tutorial Introduction(單元簡介)

1.

導讀

- (P1)導讀 C#(.NetFramework)

C#基礎介紹

- (T1)C#的 Type(型別)、NullableType(可空型別)
- (T2)討論 C#的 ObjectOriented(物件導向)、Interface、BaseClass、SubClass
- (T3)討論 Namespace。比較 StaticMethod、InstanceMethod。比較 Ref、Out、params。比較 MethodOverride，MethodHide
- (T4)比較 RAM 的 Stack、Heap。比較 Struct、Class。比較 ValueType、ReferenceType
- (T5)比較 public、protected、private。比較 AbstractClass、Interface
- (T6)比較 Public、Protected、Private。比較 Delegate、MulticastDelegates
- (T7)TryCatch、CustomException、SqlException、FormatException、OverflowException
- (T8)比較 Enum、EnumGetValue、EnumGetNames
- (T9)討論 AccessModifiers。比較 Public、Protected、Private、Internal、ProtectedInternal
- (T10)比較 SystemString、SystemTextStringBuilder。Ram 的 Stack、Heap
- (T11)比較 ConvertToString、ToString、OverrideToString
- (T12)比較 Contains、Equals、SequenceEqual、GetHashCode。比較 IEqualityComparer、AnonymousTypes(匿名型別)

Generic(泛型)、Reflection(反射)、PartialClass(部分類別)、PartialMethod(部分方法)

- (T13)Generic 泛型。比較 Array、Collection。比較 GenericCollection(泛型集合)、NonGenericCollection
- (T14)Reflection(反射)、讀取 DLL。比較 EarlyBinding、LateBinding
- (T15)CustomAttribute(客製化屬性)、Reflection(反射)
- (T16)PartialClass(部分類別)、PartialMethod(部分方法)

Collection(集合)

- (T17)比較 Indexer、Properties。比較 Dictionary、List
- (T18)討論 OptionalParameters 的 ParameterArrays、OverloadMethod、ParameterDefaultValue、OptionalAttribute
- (T19)討論 DictionaryTKey、TValue、KeyValuePairTKey、TValue
- (T20)討論 List、SimpleType(簡單型別)、Sort
- (T21)討論 List、ReferenceType(參考型別)
- (T22)討論 List、ReferenceType(參考型別)、Sort、Contains、Equals、SequenceEqual
 - (T20to22)討論 List 的 SimpleType、ReferenceType。討論 Sort、Contains、Equals、SequenceEqual
- (T23)比較 Stack、Queue

FuncDelegate(委派)、LambdaExpression(表達式)、AnonymousMethods(匿名方法)

- (T24)討論 FuncDelegate(委派)、LambdaExpression(表達式)、AnonymousMethod(匿名方法)
- (T25)討論 LambdaExpression(表達式)、AnonymousMethod(匿名方法)、RegisterEvent(註冊事件)

Thread(執行緒)、Async、Await

- (T26)討論 Thread(執行緒)
 - (T26-1)討論 Thread(執行緒)
 - (T26-2)討論 Thread(執行緒)、Join
- (T27)討論 MultiThread(多執行緒)、Monitor(監視)、Lock(鎖)
- (T28)討論 MultiThread(多執行緒)、DeadLock(死鎖定)、Mutex(互斥鎖)
 - (T28-1)討論 DeadLock(死鎖定)
 - (T28-2)討論 LockOrder，解決 DeadLock(死鎖定)
 - (T28-3)討論 Mutex(互斥鎖)，解決 DeadLock(死鎖定)解法 1
 - (T28-4)討論 Mutex(互斥鎖)，解決 DeadLock(死鎖定)解法 2
- (T29)討論 Thread(執行緒)、Async、Await
- (T30)比較 Thread(執行緒)、Async、Await

一些觀念

- (T31)討論 ExtendMethod(擴充方法)
- (T32)討論 YieldReturn 的 Filter、Total
 - (T32-1)討論 YieldReturn 的 Filter
 - (T32-2)討論 YieldReturn 的 Total

2. . Net Core 和 . Net Framework

. Net Core 和 . Net Framework

1.

在講 . Net Core 和 . Net Framework 之前，要先了解 . Net 的歷史。早期工程師在開發 Windows 軟體，都是使用 . Net Framework Library 中的 C# 程式語言。所以當你安裝 Windows 7、8、10、11...etc 的時候，其實也會內建安裝相對應的 . Net Framework 版本。後來，Microsoft 為了跨(統一)平台(世界)發展了 . Net Core Library，只要軟體工程師乖乖使用 . Net Core Library 中的 C# 程式語言寫軟體，那麼你就可以把同一份程式碼轉成各個平台可以執行的軟體，這些平台包括 Windows、Linux、Mac。Microsoft 為了跨(統一)手機平台(世界)，也發展了 Xamarin 允許軟體工程師使用 C# 來寫 APP，然後 Xamarin 就可以把你寫的同一份程式碼轉成各個手機平台可以執行的 APP，這些平台包括 Iphone、Android、Windows Phone。

接下來我來介紹 Web Application，就是一個可以用瀏覽器 Browser(瀏覽器)來操作的應用程式，Browser(瀏覽器)可以是 MicrosoftEdge、GoogleChrome、FireFox...etc。簡單來說，使用者不需要安裝特殊軟體，只要有 MicrosoftEdge，就可以使用 Web Application。通常 Web Application 是拿來開發 User Interface(UI)(使用者介面)，只要有上網經驗的 User(使用者)都可以很快上手。

Web Application 在 .Net Framework Library 技術底下又分兩種，早期只有 ASP. Net WebForm 結構，副檔名是 aspx。後來，主流變成另一個 Asp. Net MVC 結構，MVC 就是 Model、View、Controller。Web Application 在 .Net Core Library 技術底下，只有 Asp. Net MVC 結構。所以，如果你是新手，請直接學習目前主流的 Asp. Net MVC 結構。

我做一個簡單的總結。**.NET Framework Library** 是用來寫 Windows 作業系統的軟體。**.NET Core Library** 是用來寫跨平台的軟體。**Xamarin** 是用來寫跨手機平台的 App。而我們所使用的程式語言就是 **C#**。所以想成為 **.NET** 工程師，**C#** 程式語言必學!!

C# 這個名字是怎麼來的呢?

很久很久以前，高端程式語言只有 C，後來發展出 C++，後來微軟中的幾位大神抓取 C，C++ 的特性發展出 C# 程式語言。其實 C# 就是 C++ 的 ++，也就是 C++++。把兩個 + 放在上面，兩個 + 放在下面。也就是 ++ 在上排，然後另外 ++ 在下排，就可以組成 # 這個字。結論就是 # 就是 +++++。

本門課不帶你紙上談兵，全部用實戰範例真槍實彈的講解 C# 觀念給你聽。透過這些範例讓您打好紮實的底子，那我們直接來寫程式吧。
