

Cài đặt Visual Studio, tạo solution và project C#, C# interactive

[Hướng dẫn tự học lập trình C# toàn tập](#) > [Cài đặt Visual Studio, tạo solution và project C#, C# interact...](#)

Lập trình với C# nói riêng và các ngôn ngữ .NET khác nói chung hầu như đều sử dụng Visual Studio – môi trường phát triển ứng dụng tích hợp tốt nhất của Microsoft cho các công nghệ .NET. Bài học này sẽ hướng dẫn bạn cài đặt Visual studio, lựa chọn workload và các thành phần cần thiết cho việc học lập trình C# căn bản. Cuối cùng chúng ta sẽ xây dựng một project thử nghiệm đầu tiên cho chương trình Hello world với C#.

NỘI DUNG CỦA BÀI [Ấn]

1. Trình soạn thảo code cho C#
2. Thực hành 1: cài đặt Visual Studio
 - 2.1. Tải và chạy Visual Studio Installer
 - 2.2. Lựa chọn workload
3. Thực hành 2: tạo project C# đầu tiên – Hello World
 - 3.1. Tạo project trong Visual Studio
 - 3.2. Viết code cho chương trình đầu tiên "Hello world"
 - 3.3. Biên dịch và chạy debug
4. Thực hành 3: thêm project mới vào solution
5. Thực hành 4: tạo Blank Solution
6. Sử dụng C# interactive
7. Kết luận

Trình soạn thảo code cho C#

Việc lập trình C# có thể thực hiện trên một chương trình xử lý văn bản đơn giản như Notepad hay Notepad++. Để dịch mã nguồn C# thành chương trình có thể sử dụng trình biên dịch (C# compiler) qua giao diện dòng lệnh. Tuy nhiên, phương pháp này không phù hợp để phát triển các ứng dụng phức tạp.

Cách thức hiệu quả nhất là sử dụng một *môi trường phát triển ứng dụng tích hợp* (Integrated Development Environment, IDE).

Hiện nay có một số IDE hỗ trợ [các công nghệ .NET](#) như Visual Studio, Visual Studio Code, MonoDevelop, Morfik (cho phát triển ứng dụng web), SharpDevelop, hay Turbo C#.

Trong khóa học này chúng ta sẽ sử dụng Visual Studio.

Visual Studio của Microsoft là IDE hiệu quả nhất trên Windows cho C# (và các ngôn ngữ .NET khác). IDE này cung cấp tất cả những công cụ cần thiết để phát triển tất cả các loại [ứng dụng .NET](#) ở mọi cấp độ phức tạp.

Phiên bản mới nhất của Visual Studio tại thời điểm viết tài liệu này là Visual Studio 2017. Có 3 bản Visual Studio khác nhau để lựa chọn: Ultimate, Professional, Community (miễn phí). Để thực hiện dự án này chỉ cần sử dụng bản Community.

Các phiên bản của Visual Studio không có nhiều khác biệt đối với học lập trình C# căn bản. Tuy nhiên chúng tôi khuyến khích sử dụng phiên bản từ 2015 trở lên. Một số cú pháp/cấu trúc mới của C# không được hỗ trợ ở các phiên bản thấp hơn.

Edit: Hiện nay phiên bản Visual Studio 2019 đã được công bố. Bạn hoàn toàn có thể sử dụng. Tuy nhiên lưu ý một số giao diện (như tạo solution/project) có chút khác biệt so với bản 2017.

Thực hành 1: cài đặt Visual Studio

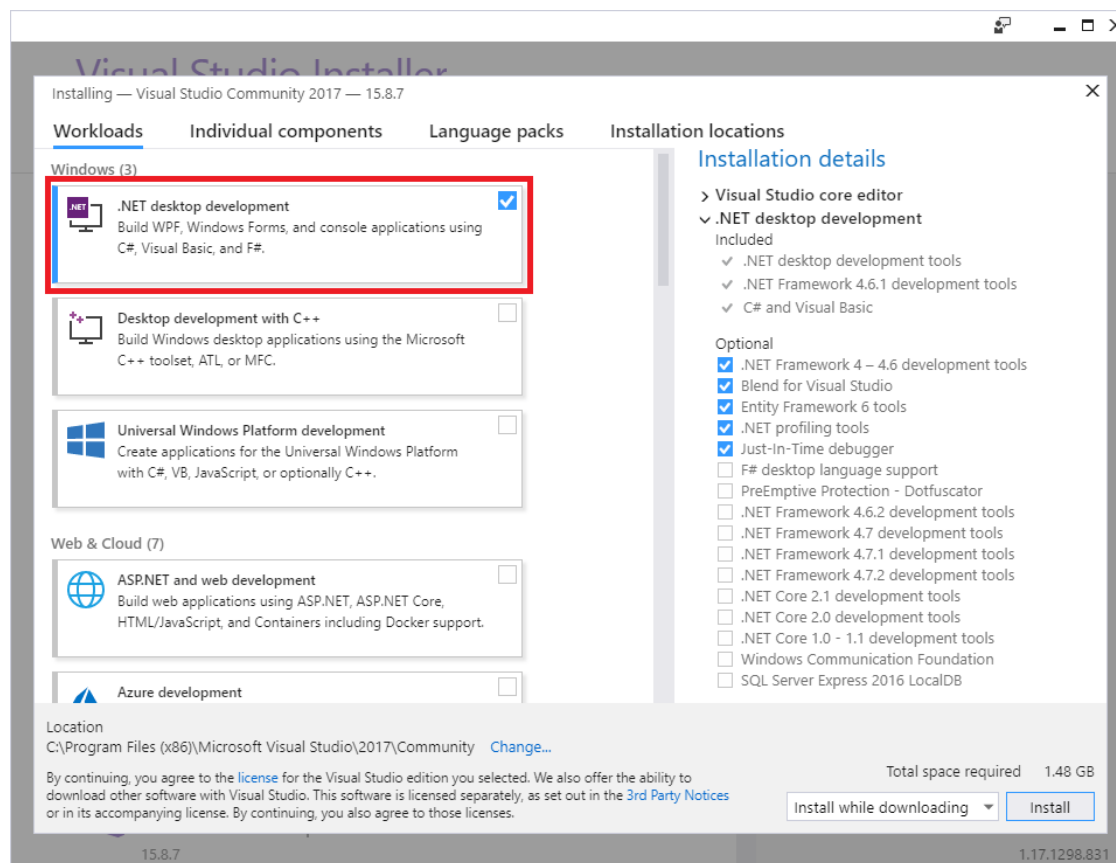
Tải và chạy Visual Studio Installer

Trước hết [tải Visual Studio Installer](#).

Visual Studio Installer là chương trình hỗ trợ cài đặt cho Visual Studio. Bắt đầu từ phiên bản 2017, việc cài đặt (và cập nhật/điều chỉnh) các thành phần của Visual Studio 2017 đều được thực hiện thông qua chương trình này.

Tất cả cài đặt đều yêu cầu kết nối Internet vì Visual Studio Installer sẽ tải các gói phần mềm từ server.

Khi chạy lần đầu, Visual Studio Installer có giao diện như sau:



Giao diện Visual Studio Installer

Giao diện chương trình này chia thành các tab:

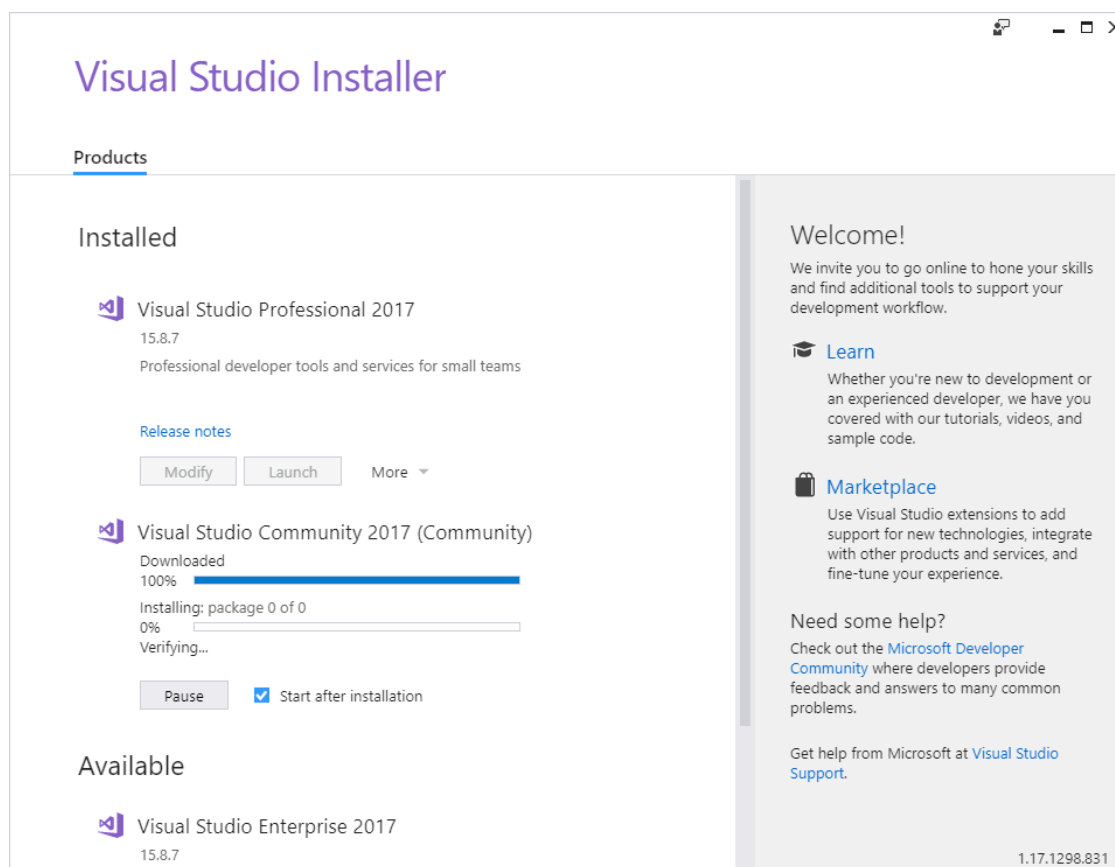
- Workloads: tiện lợi cho việc chọn nhóm các thành phần có liên quan cho từng mục đích làm việc.
- Individual components: cho phép chọn các thành phần riêng rẽ để cài đặt.
- Language packs: cho phép chọn các gói ngôn ngữ giao diện
- Installation locations: cho phép lựa chọn đường dẫn để cài đặt các thành phần.

Danh sách bên phải (Installation details) liệt kê các thành phần đang được lựa chọn.

Lựa chọn workload

Để thực hiện dự án này chúng ta chỉ cần chọn workload “.NET desktop development”, trong đó có hỗ trợ phát triển ứng dụng dòng lệnh Console, ứng dụng Windows Forms, ứng dụng Windows Presentation Foundation.

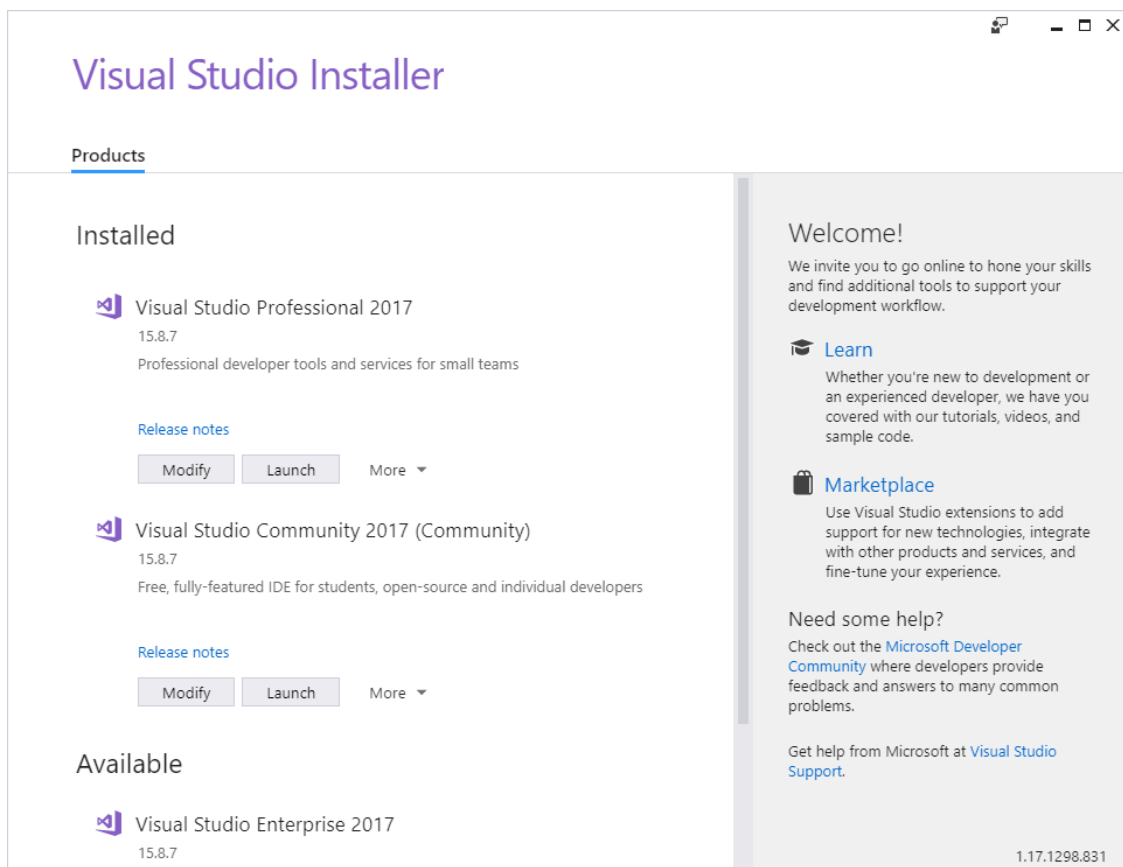
Bấm “Install” để bắt đầu download và cài đặt Visual Studio 2017 Community.



Quá trình cài đặt Visual Studio 2017 Community

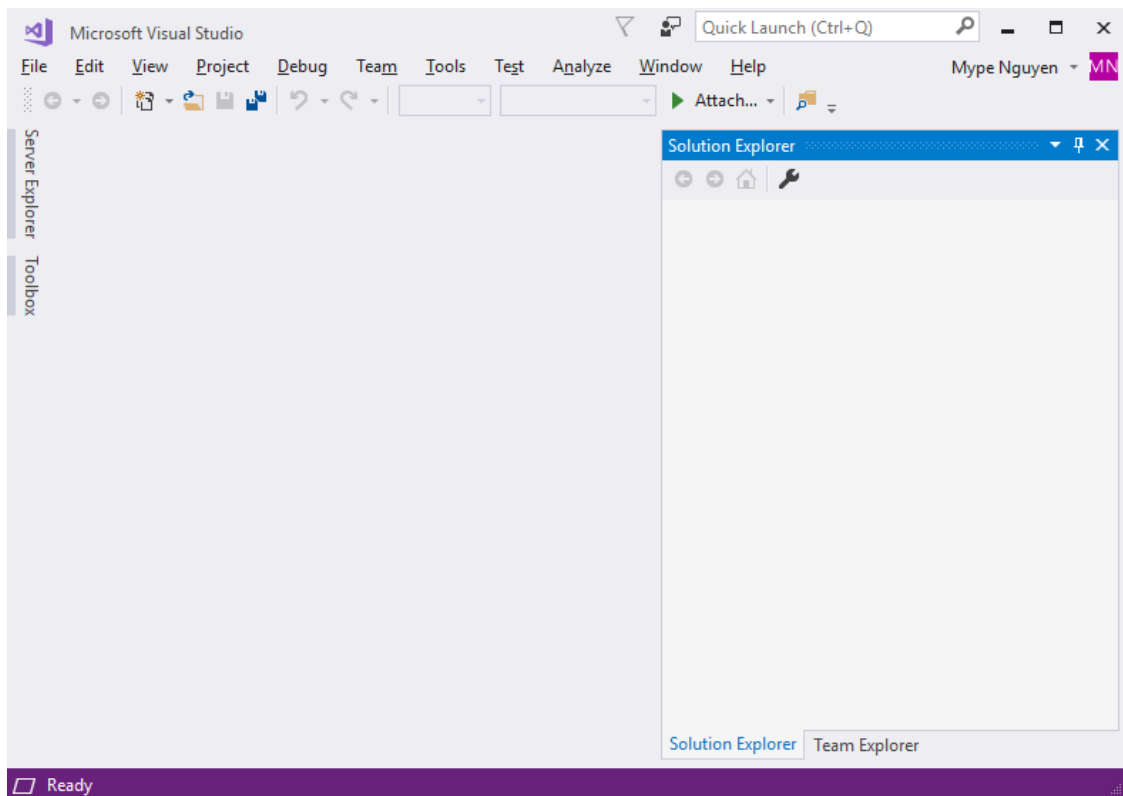
Thời gian cài đặt phụ thuộc nhiều vào tốc độ Internet. Hãy để máy đó và làm cốc cà phê trong lúc chờ đợi :).

Sau khi cài đặt xong, giao diện của Visual Studio Installer sẽ trở thành như sau:



Hoàn tất cài đặt Visual Studio 2017

Nếu cần thêm bớt các thành phần của Visual Studio thì bấm nút “Modify”, bạn sẽ được trả về màn hình lựa chọn thành phần cài đặt như lúc trước. Ấn nút “Launch” để khởi động Visual Studio.

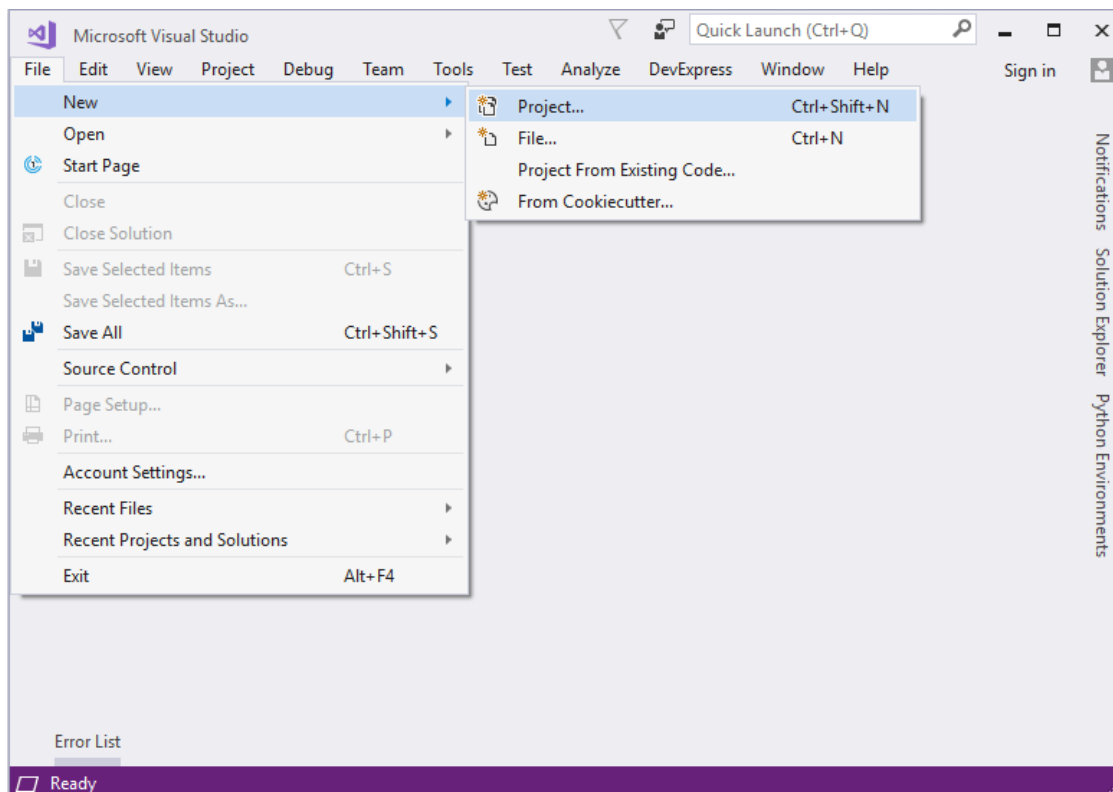


Giao diện chính của Visual Studio

Thực hành 2: tạo project C# đầu tiên – Hello World

Tạo project trong Visual Studio

Chọn File => New => Project hoặc bấm tổ hợp Ctrl + Shift + N



Tạo một project mới

Chọn loại project là Console App (.NET framework). Khi chọn loại project chúng ta có thể thiết lập các tham số sau:

- Name: tên dự án, đây cũng là tên mặc định của chương trình về sau;
- Location: thư mục chứa tất cả file của dự án;
- Solution name: solution cho phép quản lý nhiều dự án có liên quan (và sử dụng chung code với nhau);
- Framework: lựa chọn phiên bản của .NET framework; nếu lựa chọn phiên bản nào, khi triển khai ứng dụng đòi hỏi trên hệ thống của người dùng phải cài đặt .NET framework phiên bản tương đương hoặc cao hơn;
- Create directory for solution: tốt nhất luôn check mục này, đặt tất cả các dự án trong cùng một thư mục chung;
- Add to source control: chọn mục này nếu bạn sử dụng một chương trình kiểm soát mã nguồn nào đó (như Git). Mục này tạm thời không check.

Các khái niệm **solution** và **project** sẽ được xem xét chi tiết ở bài sau.

Điền thông tin như sau:

Điền thông tin cho dự án

Viết code cho chương trình đầu tiên “Hello world”

Click đúp vào file “Program.cs” để mở trong trình soạn thảo code.

Thêm code như sau vào thân phương thức `static void Main(string[] args)`

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6.
7. namespace ConsoleApp
8. {
9.     class Program
10.    {
11.        static void Main(string[] args)
12.        {
13.            Console.WriteLine("Hello world from C#");
14.            Console.WriteLine("Press any key to quit");
15.            Console.ReadKey();
16.        }
17.    }
18. }
```

Giao diện code editor của Visual Studio

Nếu để ý chúng ta sẽ thấy, khi gõ một vài ký tự, Visual Studio sẽ tự động liệt kê các code có chứa những chữ này. Chúng ta có thể trực tiếp lựa chọn bằng cách di chuyển tới mục mong muốn bằng các phím mũi tên, sau đó ấn phím tab mà không cần gõ hết câu.

Tính năng này của trình soạn thảo C# trong Visual Studio được gọi là *IntelliSense*.

IntelliSense giúp việc soạn thảo code C# đặc biệt nhanh chóng và tiện lợi, cũng như giúp giải phóng người lập trình khỏi việc phải ghi nhớ máy móc tất cả các tên gọi. Sau này chúng ta sẽ còn sử dụng nhiều tính năng khác của IntelliSense.

Biên dịch và chạy debug

- Dịch và chạy chương trình ở chế độ debug: F5
- Dịch và chạy chương trình (không debug): Ctrl + F5

Cũng có thể gọi các lệnh này qua menu Debug => Start Debugging hoặc Start Without Debugging

Chọn lệnh debug

Kết quả chạy chương trình

Kết quả chạy chương trình

Mặc dù chương trình của chúng ta chưa thực hiện được chức năng gì nhưng đến đây xin chúc mừng bạn đã viết được chương trình đầu tiên với C# và .NET framework.

Debug là chế độ hoạt động mà chương trình được gắn vào một chương trình đặc biệt gọi là debugger (ở đây là Visual Studio Debugger) để giúp chúng ta theo dõi được hoạt động của chương trình, như là giá trị của các biến.

Chương trình chạy ở chế độ debug có thể được dừng tại bất kỳ câu lệnh nào (bằng cách đánh dấu break ở câu lệnh đó) và tiếp tục chạy/dừng theo yêu cầu của người lập trình.

Debug giúp chúng ta phát hiện các lỗi logic của chương trình. Đối với lỗi cú pháp, trình soạn thảo của C# có thể trực tiếp hiển thị trong quá trình viết code mà không cần chạy thử.

Thực hành 3: thêm project mới vào solution

Trong bài thực hành 2, chúng ta mới tạo ra một project đầu tiên cho solution.

Trong Visual Studio, một solution có thể chứa nhiều project. Chúng ta sẽ thêm một project thứ hai vào solution. Cách thực hiện xem trong hình minh họa dưới đây.

Sau bước này chúng ta sẽ gặp lại giao diện chọn loại project như đã biết. Điền các thông tin cần thiết và ấn OK để thêm project mới vào solution.

Thực hành 4: tạo Blank Solution

Nếu một solution không chứa một project nào nó được gọi là solution trống (blank solution). Một solution mà không chứa project nào thì có giá trị gì? Bạn hẳn sẽ đặt câu hỏi đó.

Trên thực tế sẽ có nhiều lúc bạn muốn tạo một solution như vậy.

Ví dụ, bạn đã có sẵn một số project rời rạc. Bạn có thể tạo một blank solution sau đó lần lượt thêm project sẵn có vào solution này.

Hãy cùng tạo một blank solution.

Chọn File => New => Project như bạn đã thực hiện ở trên.

Trong hộp thoại New Project tìm đến nút Other Project Types\ Visual Studio Solutions.

Chọn Blank Solution. Đặt cho nó một cái tên và chọn đường dẫn.

Ấn OK, bạn sẽ thu được một blank solution trống trơn. Giờ bạn có thể thoải mái thêm project mới hoặc thêm project sẵn có vào đây.

Sử dụng C# interactive

Khi học lập trình C# cơ bản, rất nhiều khi bạn sẽ phải thử nghiệm các cú pháp, kiểu dữ liệu, cấu trúc điều khiển, v.v. Đôi khi cái cần thử nghiệm chỉ là một vài dòng code đơn giản. Nếu phải liên tục tạo project mới chỉ để thử nghiệm một vài dòng lệnh sẽ rất bất tiện và mất thời gian.

Visual Studio cung cấp một công cụ hỗ trợ trong trường hợp này: C# interactive. Đây là một giao diện dòng lệnh đặc biệt cho phép bạn viết và thực thi từng dòng, hoặc một nhóm lệnh C# mà không cần tạo project.

Để mở giao diện C# interactive, từ giao diện Visual Studio chọn View => Other windows => C# interactive.

Để tiện lợi bạn có thể đánh dấu ghim hoặc tách cửa sổ này ra khỏi Visual studio.

Từ dấu nhắc lệnh bạn có thể nhập bất kỳ lệnh nào của C#. Nếu ấn enter thì lệnh đó sẽ được thực thi ngay lập tức.

Nếu muốn viết cả một nhóm lệnh bao gồm nhiều dòng, khi kết thúc mỗi dòng bạn ấn tổ hợp Shift + Enter. Khi này dòng lệnh sẽ không được thực thi ngay. Bạn có thể tiếp tục viết thêm các dòng lệnh nữa.

Khi kết thúc dòng lệnh cuối cùng bạn ấn Enter. Khi đó cả nhóm lệnh sẽ được thực hiện.

Có một số lưu ý sau khi sử dụng C# interactive:

- C# interactive chỉ xuất ra được console (`Console.WriteLine`, `Console.Write`) nhưng không thể đọc được dữ liệu. Nếu viết các lệnh đọc (`Console.ReadLine`, `Console.ReadKey`), giao diện này sẽ bị đơ.
- Nếu giao diện bị đơ, bạn chỉ cần khởi động lại visual studio là xong.
- Giao diện này hỗ trợ intellisense như giao diện viết code bình thường.
- Bạn có thể sử dụng bất kỳ lệnh nào (trừ vài thứ gây treo giao diện như đã biết).
- Nếu muốn in kết quả của một biến ra màn hình, bạn có thể sử dụng `Console.Write()` hoặc `Console.WriteLine()` như đối với ứng dụng console,
- Hoặc đơn giản là viết tên biến và ấn Enter (**không** viết chấm phẩy sau tên biến).

Trong một số bài học tiếp theo chúng ta sẽ sử dụng C# interactive cho tiện lợi. Nếu muốn tìm hiểu kỹ hơn, bạn có thể đọc [bài viết này](#).

Kết luận

Trong bài học này chúng ta đã cài đặt xong Visual Studio – môi trường phát triển ứng dụng tích hợp (IDE) dành cho C# và các ngôn ngữ .NET khác. Chúng ta cũng đã tạo ra một

project Console App và viết được chương trình C# đầu tiên.

- + Nếu bạn thấy site hữu ích, trước khi rời đi hãy **giúp đỡ** site bằng một hành động nhỏ để site có thể phát triển và phục vụ bạn tốt hơn.
 - + Nếu bạn thấy bài viết hữu ích, hãy giúp **chia sẻ** tới mọi người.
 - + Nếu có thắc mắc hoặc cần trao đổi thêm, mời bạn viết trong phần **thảo luận** cuối trang.
- Cảm ơn bạn!