# Visual Studio Code - Preparando entorno

.NET Core ofrece una plataforma modular y rápida para crear aplicaciones que se ejecutan en Windows, Linux y macOS. Use Visual Studio Code con la extensión de C# para disfrutar de una sólida experiencia de edición con compatibilidad total para C# IntelliSense(completado de código inteligente) y para depuración.

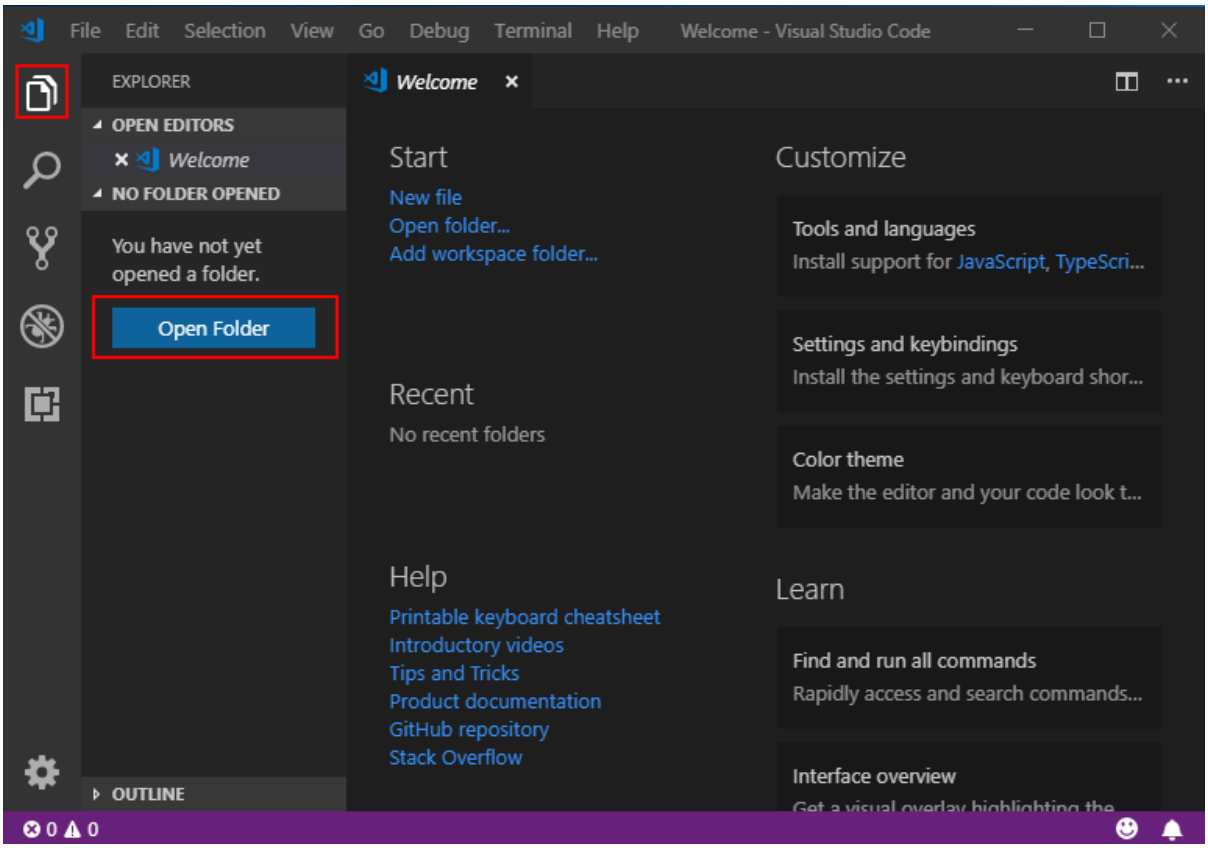
## Requisitos previos

1. Instale Visual Studio Code.
2. Instale el SDK de .NET Core.
3. Instale la extensión de C# para Visual Studio Code. Para más información sobre cómo instalar extensiones en Visual Studio Code, vea el Marketplace de extensiones de VS Code.

## Hello World

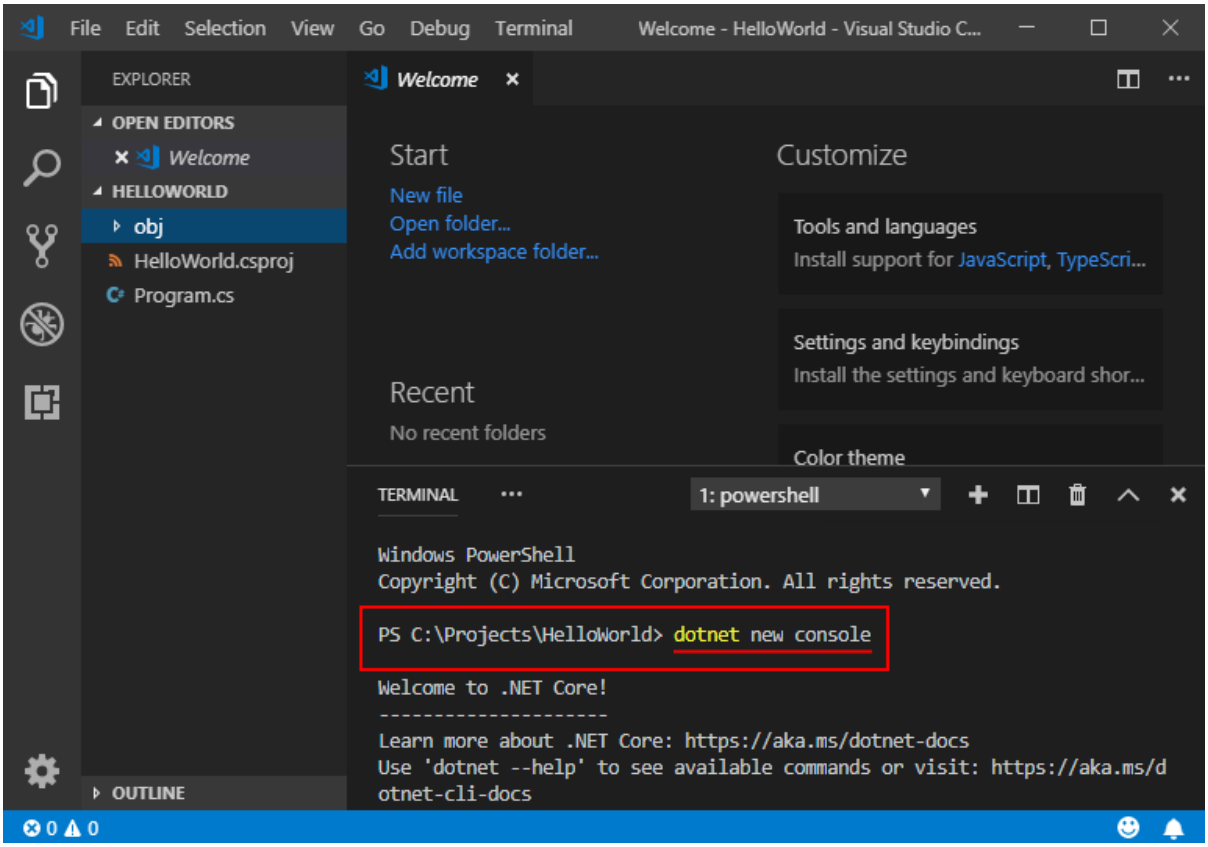
Se va a empezar con un programa "Hola mundo" sencillo basado en .NET Core:

1. Abrir un proyecto:
    - Abra Visual Studio Code.
    - Haga clic en el icono del explorador en el menú de la izquierda y después haga clic en **Abrir carpeta**.
    - Seleccione **Archivo** > **Abrir carpeta** en el menú principal para abrir la carpeta en la que quiere que esté el proyecto de C# y haga clic en **Seleccionar carpeta**. En el ejemplo se va a crear una carpeta para el proyecto denominada *HelloWorld*.



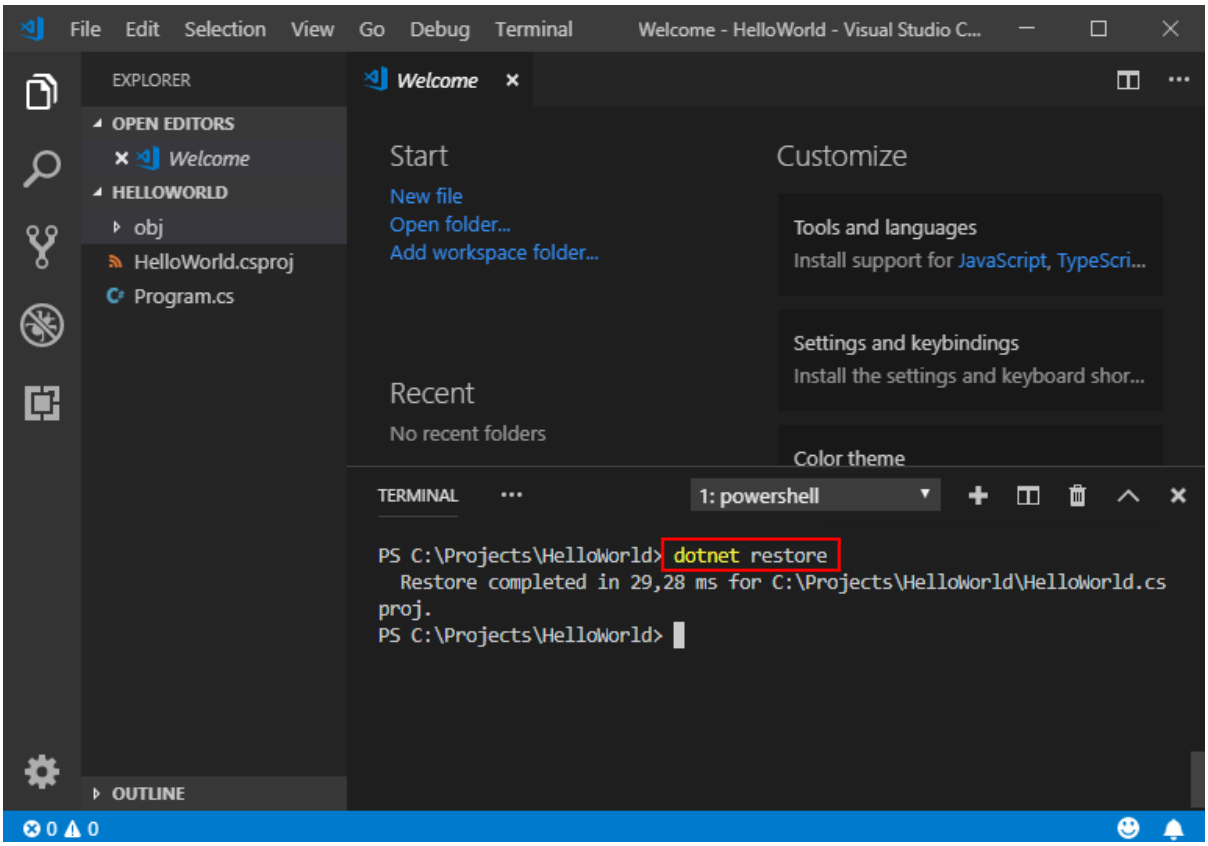
2. Inicializar un proyecto de C#:

   - Abra el terminal integrado de Visual Studio Code al seleccionar **Ver** > **Terminal integrado** en el menú principal.
   - En la ventana de terminal, escriba `dotnet new console`.
   - Este comando crea un archivo `Program.cs` en la carpeta con un programa "Hello World" sencillo ya escrito, junto con un archivo de proyecto de C# denominado `HelloWorld.csproj`.



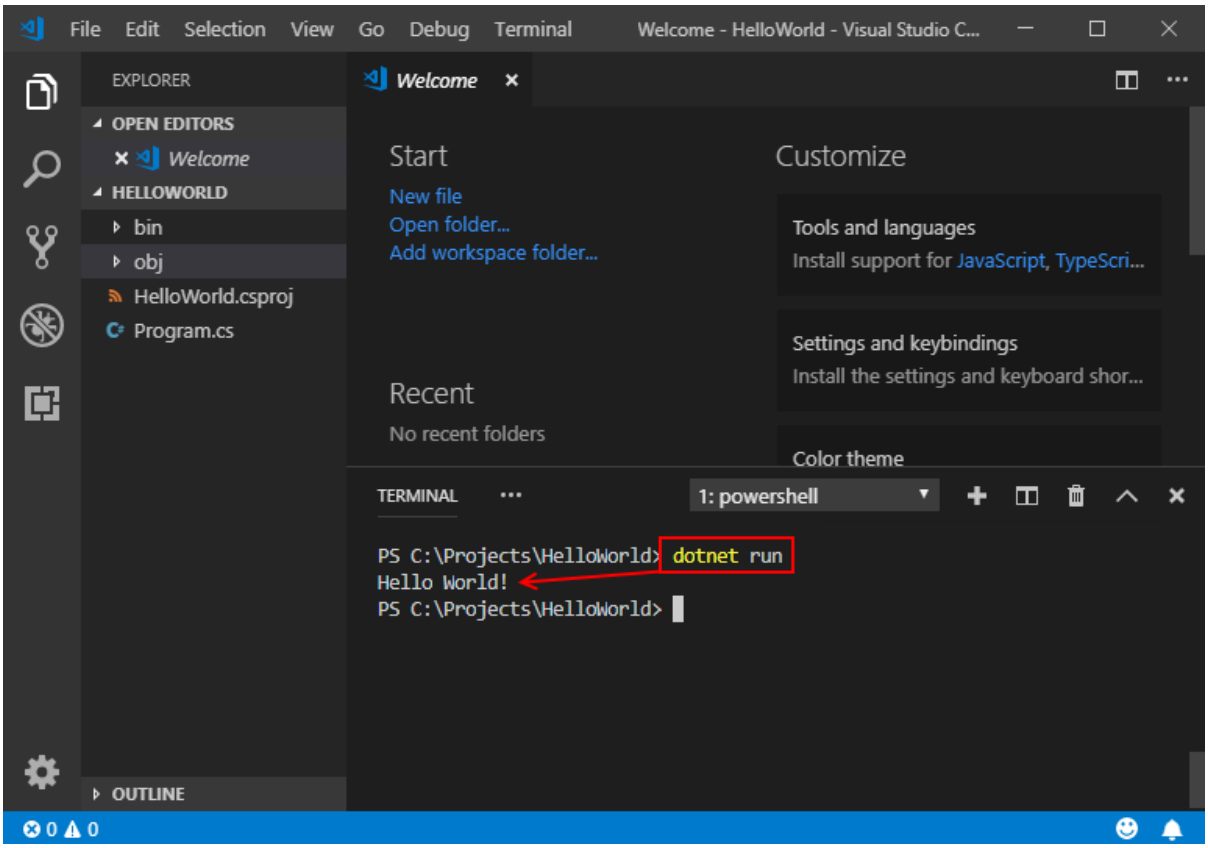

3. Resolver los recursos de compilación:

   - En **.NET Core 1.x**, escriba `dotnet restore`. Al ejecutar `dotnet restore`, se concede acceso a los paquetes de .NET Core necesarios para compilar el proyecto.




4. Ejecutar el programa "Hola mundo":

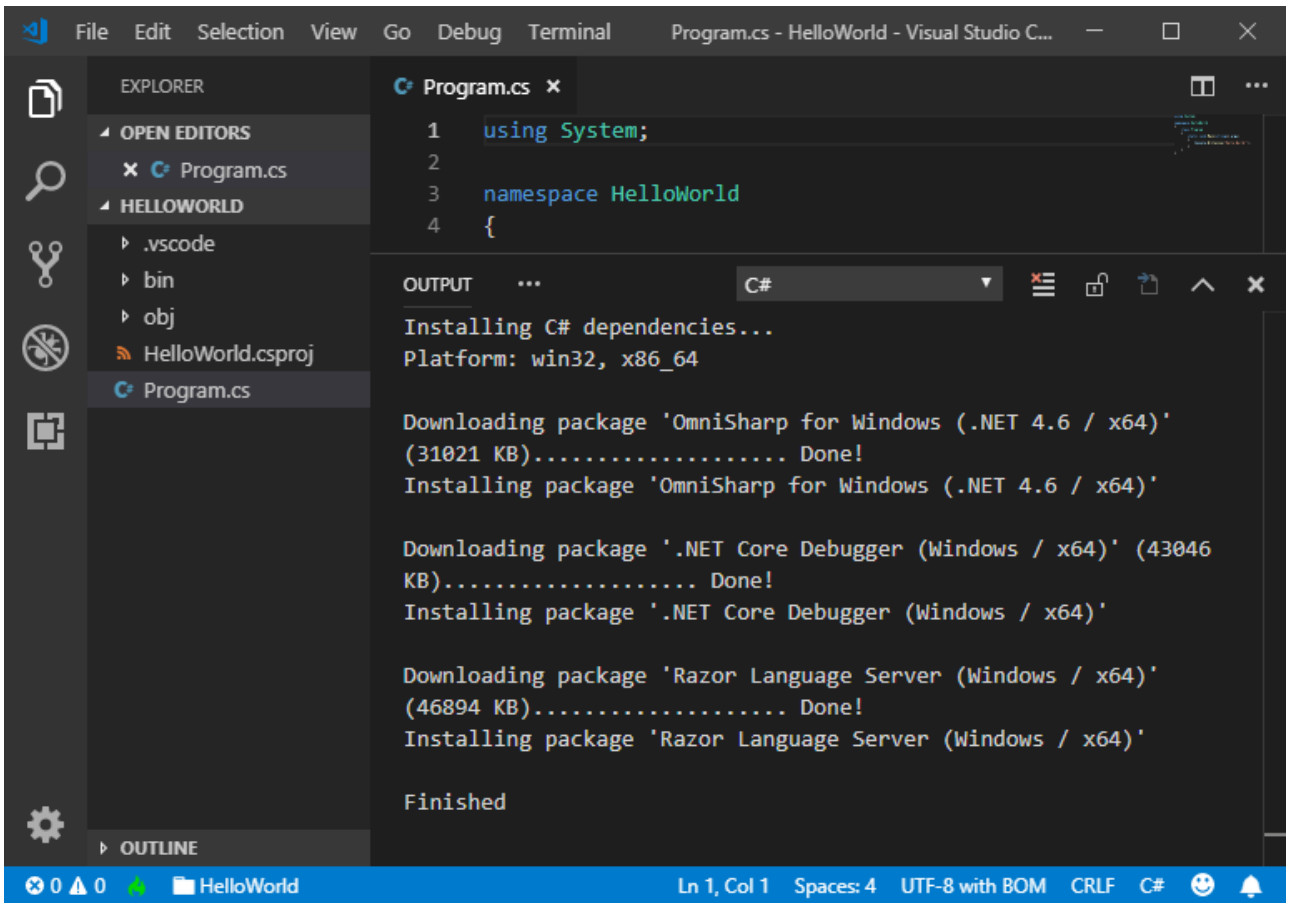
    - Escriba `dotnet run`.



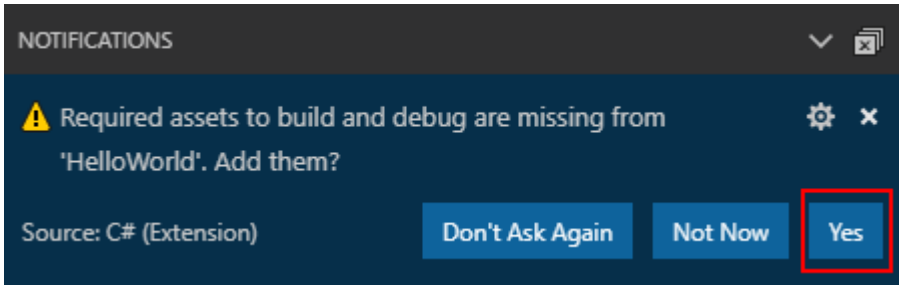

También puede ver un breve tutorial de vídeo para obtener ayuda del programa de instalación en Windows, macOS o Linux.

# Depuración

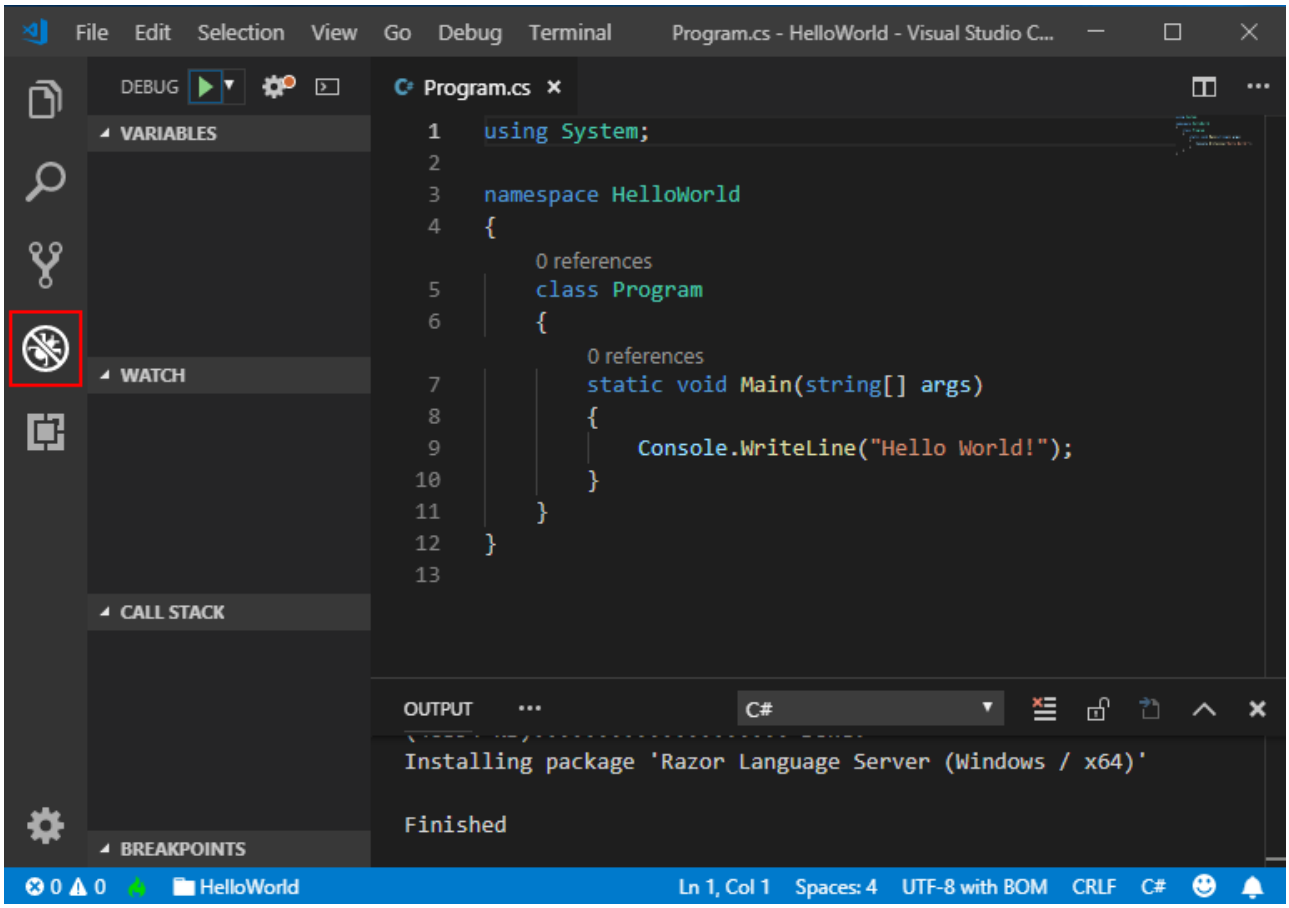
1. Haga clic en el archivo *Program.cs* para abrirlo. La primera vez que se abre un archivo de C# en Visual Studio Code, se carga OmniSharp en el editor.



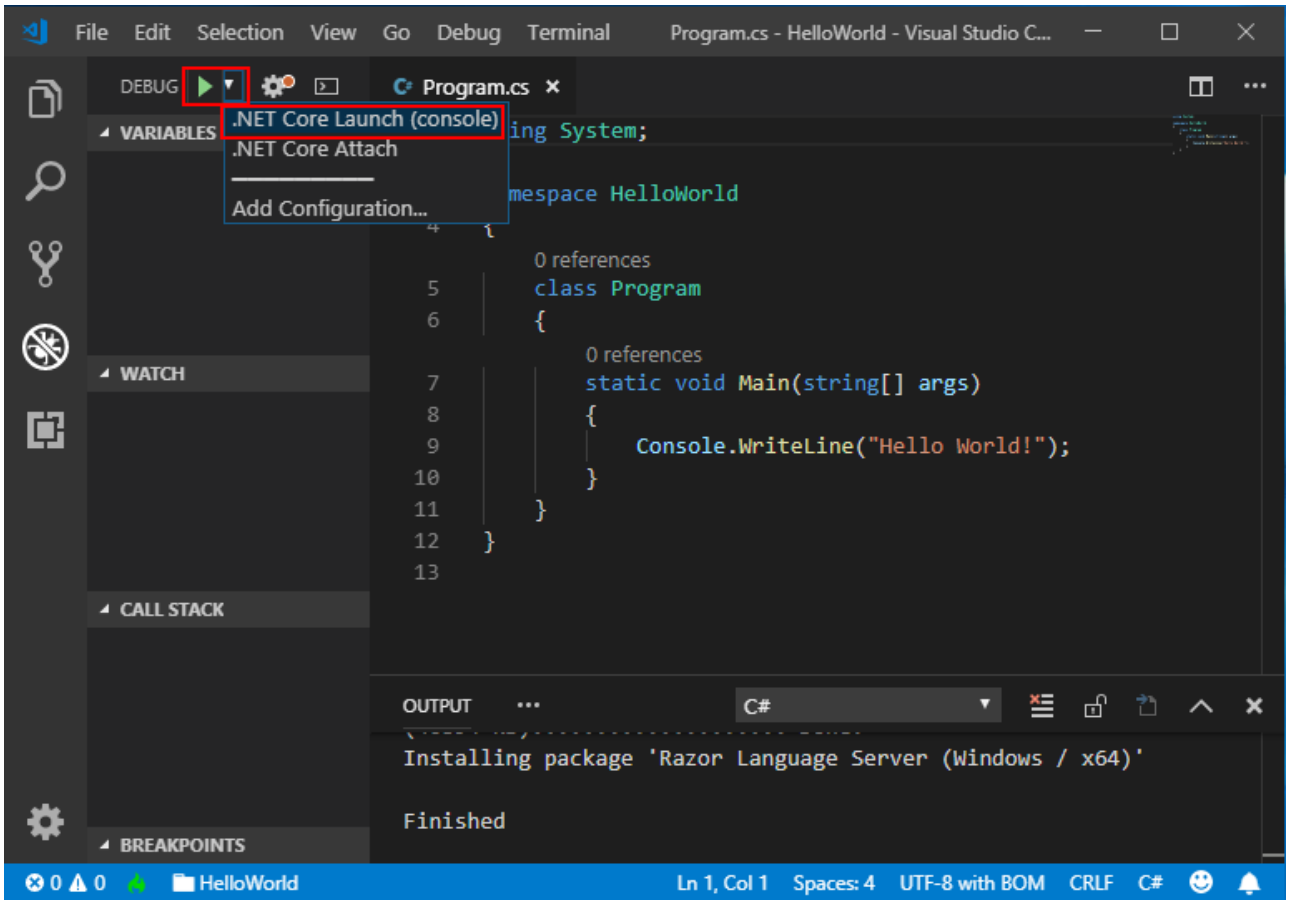

2. Visual Studio Code debe pedirle que agregue los recursos que faltan para compilar y depurar la aplicación. Seleccione **Sí**.



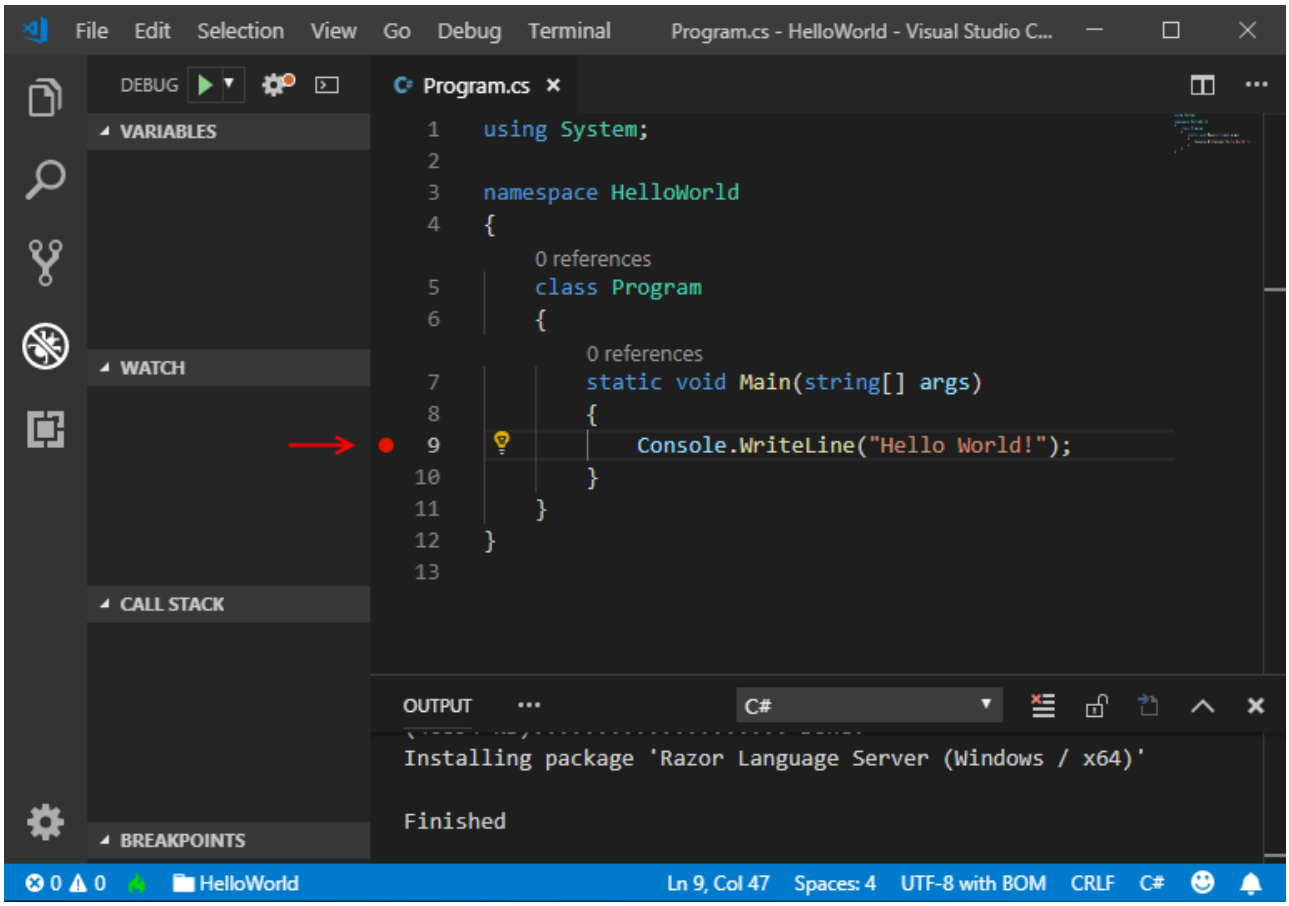

3. Para abrir la vista Depurar, haga clic en el icono de depuración en el menú de la izquierda.




4. Busque la flecha verde en la parte superior del panel. Asegúrese de que `.NET Core Launch (console)` está seleccionado en el menú desplegable que está junto a la flecha.



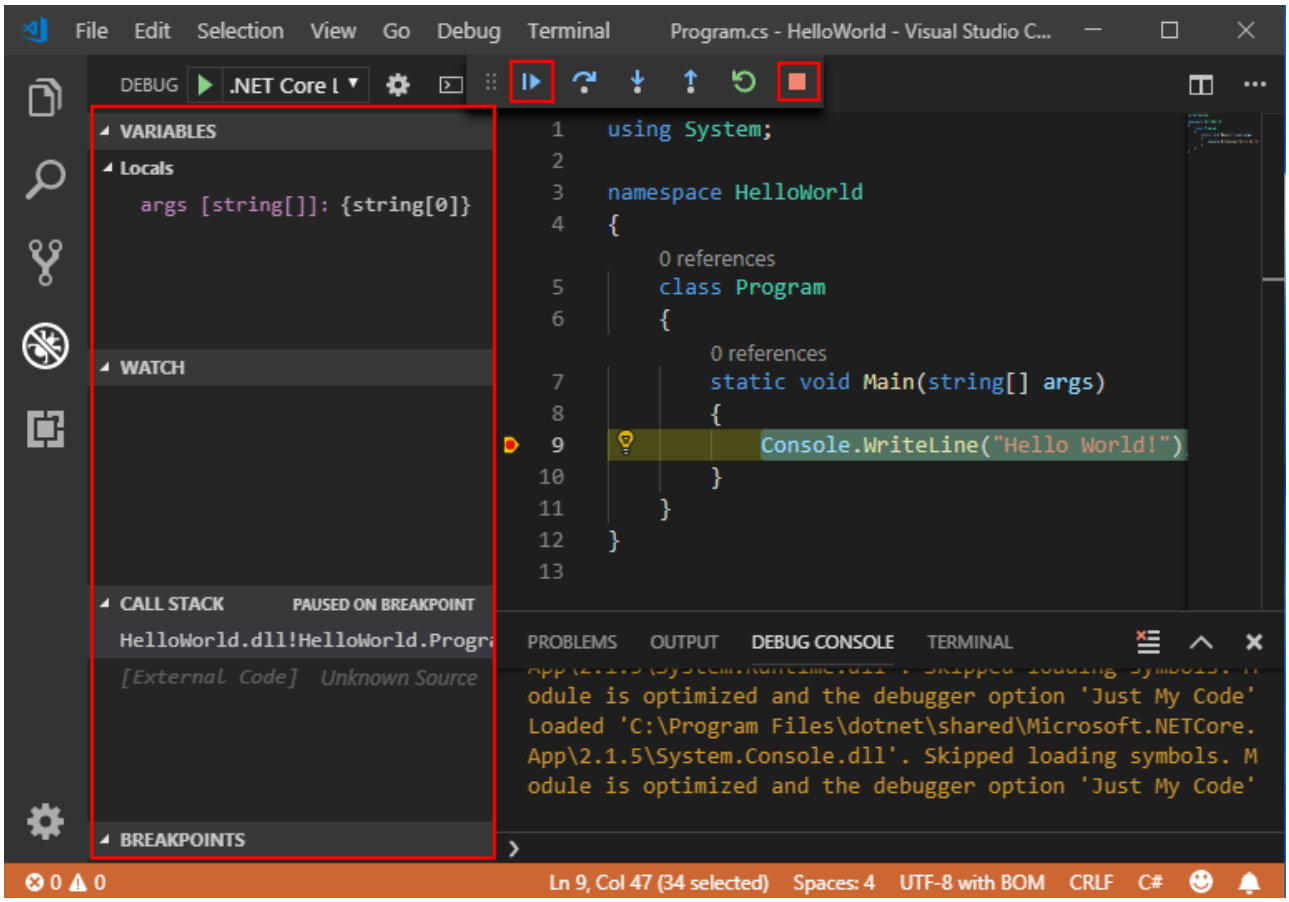
5. Agregue un punto de interrupción al proyecto; para ello, haga clic en el **margen del editor**, que es el espacio a la izquierda de los números de línea del editor, junto a la línea 9 o mueva el cursor del texto de la línea 9 en el editor y presione `F9`.




6. Para comenzar a depurar, seleccione `F5` o la flecha verde. El depurador detiene la ejecución del programa cuando alcanza el punto de interrupción establecido en el paso anterior.

    - Mientras depura, puede ver las variables locales en el panel superior izquierdo o usar la consola de depuración.

7. Seleccione la flecha azul de la parte superior para continuar la depuración o seleccione el cuadrado rojo de la parte superior para detenerla.




## Agregar una clase

1. Para agregar una nueva clase, haga clic con el botón derecho en el Explorador de Visual Studio Code y seleccione **Nuevo archivo**. Así se agrega un nuevo archivo a la carpeta abierta en Visual Studio Code.

2. Asigne un nombre al archivo `MyClass.cs`. Debe guardarlo con una extensión `.cs` al final para que se reconozca como archivo csharp.

3. Agregue el código siguiente para crear la primera clase. Asegúrese de incluir el espacio de nombres correcto para poder hacer referencia a él desde el archivo `Program.cs`.

```
using System;

namespace HelloWorld
{
    public class MyClass
    {
        public string ReturnMessage()
        {
            return "Happy coding!";
        }
    }
}
```

4. Llame a la nueva clase con el método principal de `Program.cs` agregando el código siguiente.

```
using System;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            MyClass c1 = new MyClass();
            Console.WriteLine($"Hello World! {c1.ReturnMessage()}");
        }
    }
}
```

5. Guarde los cambios y vuelva a ejecutar el programa. El nuevo mensaje debe aparecer con la cadena anexada.

```
> dotnet run
Hello World! Happy coding!
```