# How to Run .NET Core CLI Commands on Windows

The .NET Core CLI (Command-Line Interface) is an essential tool for .NET developers, enabling them to create, build, run, and publish .NET applications directly from the command line. This is particularly useful for automating tasks, integrating with CI/CD pipelines, and managing projects without relying on a graphical interface. Given that many development environments run on Windows, it's important to understand how to effectively use the .NET Core CLI in this context.

**Examples:**

1. **Installing .NET Core SDK on Windows:** Before using the .NET Core CLI, you need to install the .NET Core SDK. You can download it from the official .NET Core download page.

2. **Creating a New .NET Core Project:** Open Command Prompt (CMD) or PowerShell and run the following command to create a new console application:

   ```
   dotnet new console -n MyConsoleApp
   ```

   This command creates a new directory named MyConsoleApp with the necessary files for a console application.

3. **Building the Project:** Navigate to the project directory and build the project using:

   ```
   cd MyConsoleApp
   dotnet build
   ```

   This compiles the application and generates the necessary binaries in the bin directory.

4. **Running the Application:** You can run the application directly from the command line:

   ```
   dotnet run
   ```

   This command compiles and runs the application in one step.

5. **Publishing the Application:** To create a self-contained executable that can be deployed to other systems, use the dotnet publish command:

   ```
   dotnet publish -c Release -r win-x64 --self-contained
   ```

This generates the executable in the publish directory under bin\Release\netcoreappX.X\win-x64.

6. **Adding NuGet Packages:** If your project requires additional libraries, you can add them using the dotnet add package command:

   ```
   dotnet add package Newtonsoft.Json
   ```

   This command adds the Newtonsoft.Json package to your project.

7. **Running Unit Tests:** If you have unit tests in your project, you can run them using:

   ```
   dotnet test
   ```

   This command discovers and runs all tests in the project, providing a summary of the results.