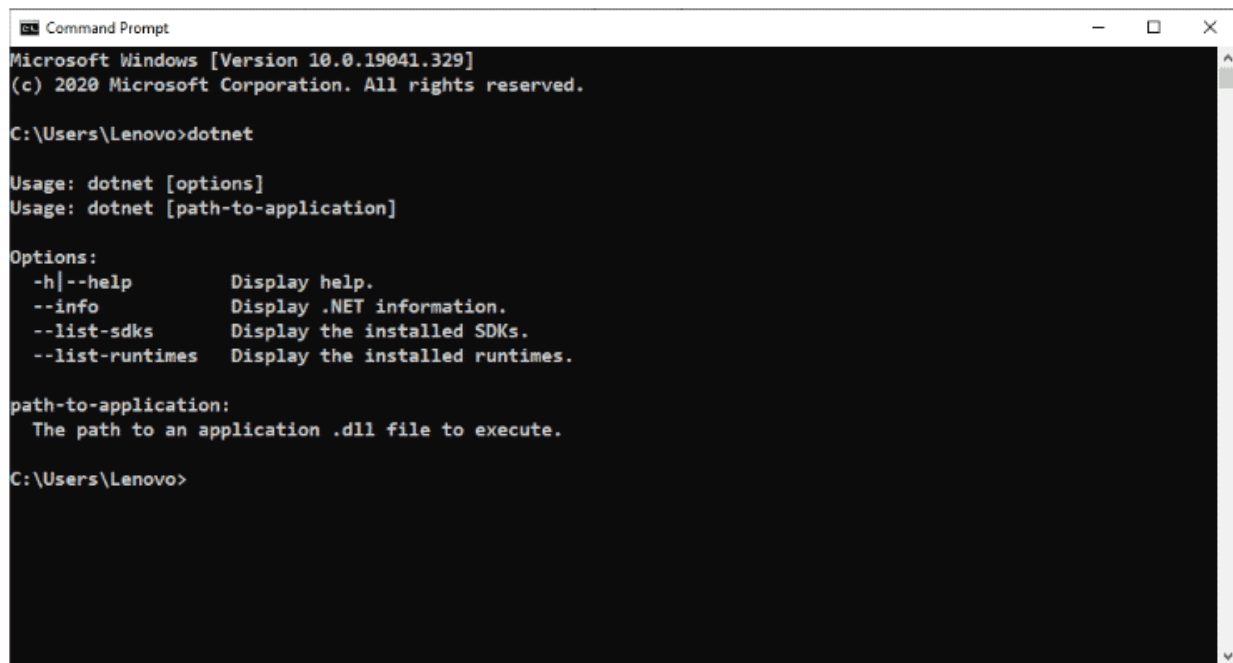


✓ .NET CLI Commands — Complete Cheat Sheet



```
Command Prompt
Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help          Display help.
  --info             Display .NET information.
  --list-sdks        Display the installed SDKs.
  --list-runtimes    Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\Lenovo>
```

◆ 1. Check .NET SDK & Runtime Versions

Command	Description
dotnet --version	Shows installed SDK version
dotnet --list-sdks	Lists all installed SDKs
dotnet --list-runtimes	Lists all installed runtimes

◆ 2. Create Projects

Command	Description
dotnet new console -n MyApp	Create Console App
dotnet new webapi -n MyApi	Create Web API

Command	Description
dotnet new mvc -n MyMvcApp	Create MVC App
dotnet new classlib -n MyLibrary	Create Class Library
dotnet new sln -n MySolution	Create Solution File

👉 To list all templates:

dotnet new --list

◆ 3. Work With Solutions

Command	Description
dotnet sln add MyApi/MyApi.csproj	Add project to solution
dotnet sln remove MyApi/MyApi.csproj	Remove project from solution

◆ 4. Build & Run Project

Command	Description
dotnet build	Builds project
dotnet run	Runs the project
dotnet run --project MyApi	Run a specific project

◆ 5. Restore Dependencies

Command	Description
dotnet restore	Restores NuGet packages

◆ 6. Add / Remove NuGet Packages

Command	Description
dotnet add package Newtonsoft.Json	Install package
dotnet remove package Newtonsoft.Json	Remove package
dotnet list package	List installed packages

◆ 7. Publish Application

Command	Description
dotnet publish -c Release	Publish in Release mode
dotnet publish -o ./publish	Publish to a folder
dotnet publish --self-contained true -r win-x64	Publish self-contained app

◆ 8. Run Tests

Command	Description
dotnet test	Run tests in solution/project

◆ 9. Create Global Tools

Command	Description
dotnet tool install -g dotnetsay	Install a global tool
dotnet tool list -g	List installed tools
dotnet tool uninstall -g dotnetsay	Remove tool

◆ 10. Useful Developer Commands

Command	Description
dotnet watch run	Auto-restart app on file changes
dotnet --info	Show environment info
dotnet nuget locals all --clear	Clear NuGet cache

◆ 11. EF Core CLI Commands

If EF Core tools not installed:

`dotnet tool install --global dotnet-ef`

Command	Description
dotnet ef migrations add Initial	Add migration
dotnet ef database update	Apply migration to DB
dotnet ef database drop	Drop DB
dotnet ef dbcontext list	List DbContexts

◆ 12. Docker Support for .NET

Command	Description
dotnet publish -c Release -o out	Build output for Docker
docker build -t myapi .	Build image
docker run -p 8080:80 myapi	Run container

◆ 13. Clean Solution

`dotnet clean`

✅ Create Projects Without Top-Level Statements

1 Console Application

```
dotnet new console -n MyApp --use-program-main
```

This generates:

```
public class Program
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

2 Web API Without Top-Level Statements

```
dotnet new webapi -n MyApi --use-program-main
```

This creates:

- Program.cs with Main method
- Startup.cs (older style)

Perfect if you want the traditional ASP.NET Core startup pattern.

3 MVC Project (already uses Program/Main by default)

But to be explicit:

```
dotnet new mvc -n MyMvc --use-program-main
```

4 Class Library (no top-level anyway)

```
dotnet new classlib -n MyLibrary
```

★ Bonus — Check All Options

To see all switches available for any template:

```
dotnet new webapi --help
```

```
dotnet new console --help
```