**Main method in asp.net core**

**we will discuss**

* The significance of **Main() method in an ASP.NET Core application**
* What happens behind the scenes when a .NET core application is executed

In an ASP.NET Core project we have a file with name **Program.cs**. In this file we have a **public static void Main()** method 

public class Program

{

    public static void Main(string[] args)

    {

        CreateWebHostBuilder(args).Build().Run();

    }

    public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>

        WebHost.CreateDefaultBuilder(args)

            .UseStartup<Startup>();

}

If you have any experience with previous versions of .NET, a console application has a **Main()** method and it is the entry point for that console application.   
  
But here, we are creating an **asp.net core web application** and not a console application. So the obvious question that comes to our mind is why do we have a Main() method.  
  
Well, the important point to keep in mind is that, an **asp.net core application initially starts as a console application** and the **Main()** method in **Program.cs** file is the entry point.   
  
So, when the runtime executes our application **it looks for this Main() method**and this where the execution starts.  
  
This **Main() method configures asp.net core**and starts it and at that point it becomes an asp.net core web application.  
  
So, if you take a look at the **Main()** method, it calls CreateWebHostBuilder() method passing it the command line arguments.  
  
As you can see, CreateWebHostBuilder() method returns an object that implements IWebHostBuilder.  
  
On this object, **Build()** method is called which builds a web host that hosts our asp.net core web application.  
  
On the web host **Run()**method is called, which runs the web application and it begins listening for incoming HTTP requests.  
  
**CreateWebHostBuilder()** method calls CreateDefaultBuilder() static method of the WebHost class.  
  
CreateDefaultBuilder() method creates a web host with pre-configured defaults. CreateDefaultBuilder() method does several things to create a web host. We will discuss all that the CreateDefaultBuilder() method does in detail in our next video. For now, just understand that the CreateDefaultBuilder() method **sets up a web host with certain defaults**.  
  
As part of setting up a web host, Startup class is also configured using the UseStartup()extension method of IWebHostBuilder class. If you are new to the concept of extension methods, please check out the following video.   
  
[Extension Methods in C#](https://www.youtube.com/watch?v=VkrKNXscoto)   
  
By convention, the startup class in ASP.NET Core is named Startup. This class has 2 methods.

public class Startup

{

    public void ConfigureServices(IServiceCollection services)

    { }

    public void Configure(IApplicationBuilder app, IHostingEnvironment env)

    {

        if (env.IsDevelopment())

        {

            app.UseDeveloperExceptionPage();

        }

        app.Run(async (context) =>

        {

            await context.Response.WriteAsync("Hello World!");

        });

    }

}

**Startup class does the following 2 very important things**

* **ConfigureServices()**method configures services required by the application
* **Configure()** method sets up the application's request processing pipeline

It is very important we understand what these 2 methods does. We will be revisiting these 2 methods several times as we progress through this course and discuss them in detail.