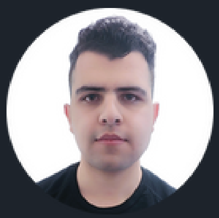
Primary Constructor

Primary Constructor syntax makes code less verbose and reduces lines of code.

```
public class Person(string name, int age)
{
  public void IntroduceYourself()
  {
    Console.WriteLine(
      $"Hello, my name is {name} and " +
      $"I am {age} years old.");
  }
}
```

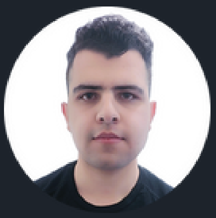


Elliot One

Fields are not read-only !

A simple inspection of the low-level C# code reveals that the generated fields are not read-only.

```
[NullableContext(1)]
[Nullable(0)]
public class Person
{
  [CompilerGenerated]
  [DebuggerBrowsable(DebuggerBrowsableState.Never)]
  private string <name>P;
  [CompilerGenerated]
  [DebuggerBrowsable(DebuggerBrowsableState.Never)]
  private int <age>P;

  public Person(string name, int age)
  {
    this.<name>P = name;
    this.<age>P = age;
    base..ctor();
  }
}
```



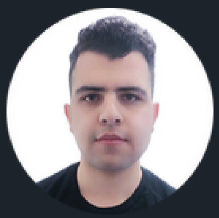
Elliot One

Usage in ASP .NET Core

Primary Constructors can reduce the lines of code in controllers.

```
public class UserController
{
  private readonly IUserService userService;

  // Constructor with dependency injection
  public UserController(IUserService userService)
  {
    this.userService = userService;
  }
}
```

```
public class UserController (IUserService userService)
{

}
```

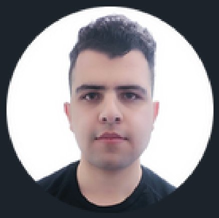


Elliot One

Inheritance and Chaining

Primary Constructor supports inheritance and constructor chaining. It can also be used simultaneously with the older version of constructors.

```csharp
// Base class
public class Vehicle(string model, int year)
{

}

// Derived class
public class Car(string model, int year, int numberOfDoors)
  : Vehicle (model, year)
{

}



public class Car(string model, int year, int numberOfDoors)
  : Vehicle(model, year)
{
  public Car(string model, int year) : this(model, year, 4)
  {

  }
}
```



Elliot One

Elliot One

Enjoyed Reading This?

Reshare and Spread Knowledge.