

C# 12.0

New

Features



Dipak Pakhale



Primary Constructors

Setting up new things in C# is now super easy with primary constructors. You can set up your objects right when you create them.



```
public class Person
{
    public string Name { get; }
    public int Age { get; }

    public Person(string name, int age) => (Name, Age) = (name, age);
}
```



Dipak Pakhale



Collection Expressions

Managing lists and arrays is much simpler now. Collection expressions help you initialize your collections neatly and save time.



```
var numbers = new List<int> { 1, 2, 3, 4, 5 };
```



Dipak Pakhale



Inline Arrays

Declaring arrays just got easier. With inline arrays, you can do it in a shorter, clearer way.



```
var daysOfWeek = new string[] { "Monday", "Tuesday", "Wednesday",
    "Thursday", "Friday", "Saturday", "Sunday" };
```



Dipak Pakhale



Optional Parameters in Lambda Expressions

You can now make your code more flexible with optional parameters in lambda expressions. You get to choose what you want to use.



```
Action<int, int> add = (x, y = 0) => Console.WriteLine(x + y);
```



Dipak Pakhale



ref Readonly Parameters

Keep your data safe with ref readonly parameters. It's like having a security guard for your data.



```
void Print(in int value) => Console.WriteLine(value);
```



Dipak Pakhale



Alias Any Type

Sometimes, code words can be confusing. With aliases, you can simplify them, making your code easier to understand.



```
using Year = System.Int32;
```



Dipak Pakhale



Experimental Attribute

Try out new things safely with the experimental attribute. It's like being an explorer in the world of coding.



```
public void NewFeature()  
{  
    // Your experimental code here  
}
```



Dipak Pakhale



Interceptors

Control how your code works with interceptors. You can shape your code to fit your vision.



```
public class LoggingInterceptor : IInterceptor
{
    public void Intercept(IInvocation invocation)
    {
        Console.WriteLine($"Calling method: {invocation.Method.Name}");
        invocation.Proceed();
    }
}
```



Dipak Pakhale





THANK YOU



Like



Comment



Share



Save



Dipak Pakhale

