

HABIB UL REHMAN

01

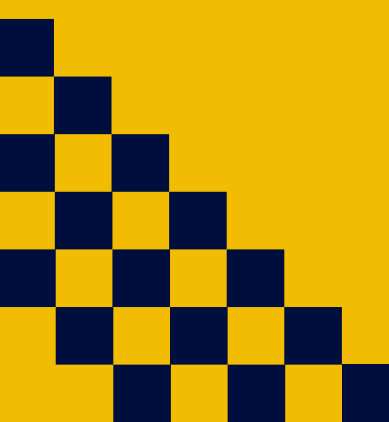
C#




DELEGATES

@HABIBDEVELOPER





Simple delegates provide a straightforward way to encapsulate and call methods with a specific signature.




```
1 delegate void SimpleDelegate(string message);  
2  
3 SimpleDelegate myDelegate = (msg) =>  
    Console.WriteLine($"Simple: {msg}");  
4 myDelegate("Hello, Delegates!");
```



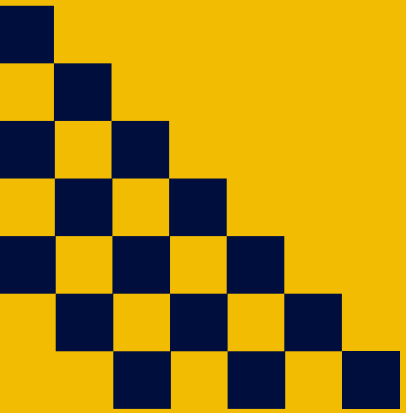
Func delegates take simplicity further, enabling you to define functions with input parameters and return types concisely.

```
1 Func<int, int, int> add = (a, b) => a + b;  
2 int result = add(5, 3);  
3  
4 Console.WriteLine($"Result: {result}")
```

Action delegates are perfect when your method doesn't return a value. They simplify the syntax and make your code cleaner.



```
1 Action<string> logMessage = (msg) =>  
    Console.WriteLine($"Action: {msg}");  
2  
3 logMessage("Logging a message.");
```



Predicates are specialized delegates that focus on conditions. Explore how they bring sharp conditions to your code.



```
1 Predicate<int> isEven = (num) => num % 2 == 0;  
2 bool result = isEven(6);  
3  
4 Console.WriteLine($"Is 6 even? {result}");
```

**REPOST THIS
POST IF YOU
FIND IT USEFUL**

