
THIS DOCUMENT COVERS

Name	Code
Init-only setters	<pre>public class Person { public string First {get; init;} public string Last {get; init;} }</pre>
Positional Record	<pre>public record Person(string First, string Last);</pre>
Deconstruct Positional Record	<pre>var p = new Person("Kenny", "Wilson") var (f,s) = p;</pre>
Positional Records and with	<pre>var p2 = p with {First="John"};</pre>
Top-level statements	<pre>System.Console.WriteLine("Hello World!");</pre>
Type Pattern	<pre>var t = (o1, o2); if (t is (int, string)) {} // test if o1 is an int and o2 is a string switch (o1) { case int: break; // test if o1 is an int case System.String: break; // test if o1 is a string }</pre>
Negation Pattern	<pre>if (e is not null)</pre>
And/or pattern	<pre>bool IsLetter(char c) => c is >= 'a' and <= 'z' or >= 'A' and <= 'Z';</pre>
Omit type in new expression	<pre>private List<int> __ints = new();</pre>
Omit type in call	<pre>void Main() { UseDate(new()); } public void UseDate(DateTime d) {}</pre>
