|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Expression** | **LINQ** | **Description** | **Result** |
| Primary | x.m | [ds.Value1] | Member access |  |
| x(...) | [ds.Result(Value1, Value2)] | Method and delegate invocation |  |
| x[...] | [ds.Values[1]] | Array and indexer access |  |
| new T(...) |  | Object and delegate creation |  |
| new T(...){...} |  | Object creation with initializer |  |
| new {...} |  | Anonymous object initializer |  |
| new T[...] |  | Array creation |  |
| Unary | -x | [-ds.Value1] | Negation |  |
| !x | [!ds.Value1] | Logical negation |  |
| ~x | [~ds.Value1] | Bitwise negation |  |
| (T)x | [(double)ds.Value1] | Explicitly convert x to type T |  |
| Binary | x \* y | [ds.Value1 \* ds.Value2] | Multiplication |  |
| x / y | [ds.Value1 / ds.Value2] | Division |  |
| x % y | [ds.Value1 % ds.Value2] | Remainder |  |
| x + y | [ds.Value1 + ds.Value2] | Addition, string concatenation, delegate combination |  |
| x – y | [ds.Value1 - ds.Value2] | Subtraction, delegate removal |  |
| x << y | [ds.Value1 << ds.Value2] | Shift left |  |
| x >> y | [ds.Value1 >> ds.Value2] | Shift right |  |
| x < y | [ds.Value1 < ds.Value2] | Less than |  |
| x > y | [ds.Value1 > ds.Value2] | Greater than |  |
| x <= y | [ds.Value1 <= ds.Value2] | Less than or equal |  |
| x >= y | [ds.Value1 >= ds.Value2] | Greater than or equal |  |
| x == y | [ds.Value1 == ds.Value2] | Equal |  |
| x != y | [ds.Value1 != ds.Value2] | Not equal |  |
| x & y | [ds.Value1 & ds.Value2] | Integer bitwise AND, boolean logical AND |  |
| x ^ y | [ds.Value1 ^ ds.Value2] | Integer bitwise XOR, boolean logical XOR |  |
| x | y | [ds.Value1 | ds.Value2] | Integer bitwise OR, boolean logical OR |  |
| x && y | [ds.Value1 && ds.Value2] | Evaluates y only if x is true |  |
| x || y | [ds.Value1 || ds.Value2] | Evaluates y only if x is false |  |
| X ?? y | [ds.Value1 ?? ds.Value2] | Evaluates to y if x is null, to x otherwise |  |
| Ternary | x ? y : z | [ds.Value2 = 1 ? ds.Value1 + ds.Value2 : ds.Value1] | Evaluates y if x is true, z if x is false |  |