Cheat Sheets

Cheating

This document covers

## Operators / Compiler Fluff

|  |  |
| --- | --- |
| Name | Code |
| Null Conditional (6.0) | var r = s?.toString(); |
| Null-coalescing (8.0) | s ??= “something”; |

## Patterns

|  |  |
| --- | --- |
| Name | Code |
| Pattern Variables (7.0) | if (o is string s )  WriteLine(s); |
| Tuple Pattern (8.0) | Switch based on multiple values |
| Positional Patterns |  |

## Indices and Ranges

|  |  |  |
| --- | --- | --- |
| Name | Code | Result |
| Last element | WriteLine(new [] {1,2,3}[^1]); | 3 |
| Second last | WriteLine(new [] {1,2,3}[^2]); | 2 |
| First two elements | WriteLine(new [] {1,2,3}[..2]); | [1,2] |
| Slice from index 1 to index 2 (second index is exclusive) | WriteLine(new [] {1,2,3}[1..3]); | [2,3] |
| Last two elements | WriteLine(new [] {1,2,3}[^2..]); | [2,3] |

### 7.0

|  |  |
| --- | --- |
| Name | Code |
| Throw Expression | void ThrowException() =>  throw new NotImplementedException(); |
| Tuples  (System.ValueTuple) | var tuple = (name: "Kenny", age:45);  var (nm, ag) = tuple; |
| Deconstructor | void Main()  {  var (nm, street) = new Address();  }  class Address  {  public void Deconstruct(out int nm, out string st )  {  nm = 45;  st = "North Road";  }  } |

#### Expression Bodied Constructor

public class Person

{

public Person() => Console.WriteLine("Person");

}

#### Expression Bodied read/write properties

private int \_age;

public int Age

{

get => \_age;

set => \_age = value;

}