C# 7 cheat sheet

**Out variables**

int.TryParse("123", out int i);

**Tuples and deconstruction**

(double, int) t1 = (4.5, 3);

(double Sum, int Count) t2 = (4.5, 3);

(int min, int max) FindMinMax(int[] input)

{

// ...

}

**Pattern matching**

public void PrintStars(object o)

{

if (o is null) return; // constant pattern "null"

if (!(o is int i)) return; // type pattern "int i"

WriteLine(new string('\*', i));

}

**Local functions**

string GetText(string path, string filename)

{

var reader = File.OpenText($"{AppendPathSeparator(path)}{filename}");

var text = reader.ReadToEnd();

return text;

string AppendPathSeparator(string filepath) =>

filepath.EndsWith(@"\") ? filepath : filepath + @"\";

}

**Expanded expression bodied members**

public class Location

{

private string \_locationName;

public Location(string name) => \_locationName = name;

string Name => \_locationName;

}

**Discards**

var validInteger = int.TryParse("foo", out \_);

C# 7 cheat sheet

**Binary Literals and Digit Separators**

var oneMillion = 1\_000\_000;

var oneMillionBinary = 0b11110100001001000000;

**Throw expressions**

string first = args.Length >= 1 ?

args[0] :

throw new ArgumentException("Please supply at least one argument.");