C# 8 cheat sheet

Readonly members

```
public readonly struct Coords
{
    public Coords(double x, double y)
    {
        // ...
    }
}

var p1 = new Coords(1, 2);
var p2 = p1 with { X = 3 };
```

Default interface members

```
interface IA
{
    void M() { WriteLine("IA.M"); }
}

class C : IA { }

IA i = new C();
i.M();
```

Pattern matching switch expressions

```
public static Orientation ToOrientation(Direction direction) => direction switch
{
    Direction.Up => Orientation.North,
    Direction.Right => Orientation.East,
    Direction.Down => Orientation.South,
    Direction.Left => Orientation.West,
    _ => throw new ArgumentOutOfRangeException(nameof(direction), $"Not expected direction
value: {direction}"),
};
```

Pattern matching property pattern

```
static bool IsConferenceDay(DateTime date) => date is { Year: 2020, Month: 5, Day: 19 or 20
or 21 };
```

C# 8 cheat sheet

Using declarations

```
{
    using var f1 = new FileStream("...");
    using var f2 = new FileStream("..."), f3 = new FileStream("...");
    // ...
    // Dispose f3
    // Dispose f2
    // Dispose f1
}
```

Static local functions

```
int M()
{
    int y = 5;
    int x = 7;
    return Add(x, y);

    static int Add(int left, int right) => left + right;
}
```

Nullable reference types

```
#nullable enable
string notNull = "Hello";
string? nullable = default;
notNull = nullable!; // null forgiveness
#nullable disable
```

Asynchronous streams

```
await foreach (var item in GenerateSequenceAsync())
{
    Console.WriteLine(item);
}
```

Indices and ranges

```
int[] oneThroughTen = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
Write(oneThroughTen, ..3);    // 1, 2, 3
Write(oneThroughTen, 3..5);   // 4, 5
Write(oneThroughTen, ..^3);   // 1, 2, 3, 4, 5, 6, 7

static void Write(int[] values, Range range) =>
    Console.WriteLine($"{range}:\t{string.Join(", ", values[range])}");
```

Null-coalescing assignment

```
name ??= "default";
```