



```
public class Employee
{
    private string name;
    private int id;

    public Employee(string s, int i)
    {
        name = s;
        id = i;
    }

    public string Name
    {
        get { return name; }
        set { name = value; }
    }

    public int ID
    {
        get { return id; }
        set { id = value; }
    }
}
```

IntelliTect

Essential C# 7.0 and Beyond

Mark Michaelis



IntelliTect.com

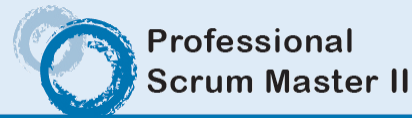
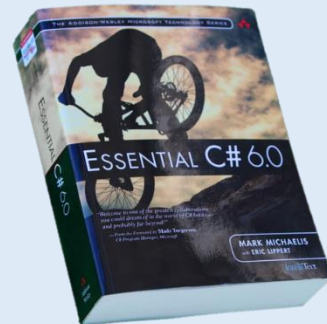
| info@intellitect.com

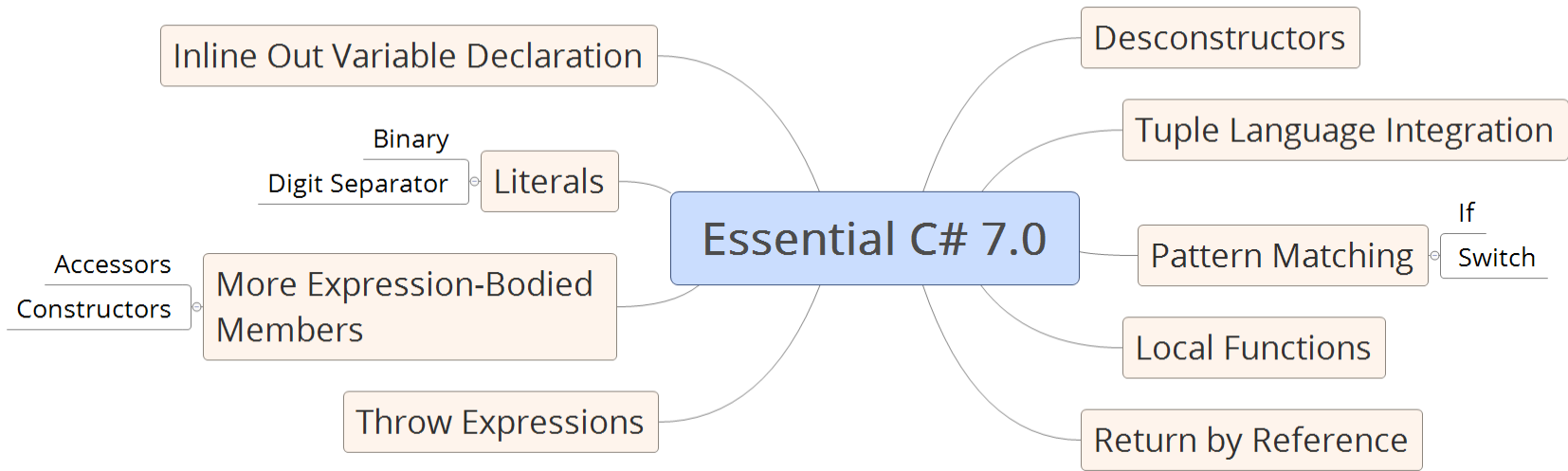
IntelliTect

About Us

Mark Michaelis

Chief Technical Architect,
Author, Trainer





Tuple: With Individually Declared Variables

Description	Example Code
Assigning a tuple to individually declared variables.	<pre>(string country, string capital, double gdp) = ("Malawi", "Lilongwe", 226.50); System.Console.WriteLine(\$@"The poorest country in the world in 2017 was { country}, {capital}: {gdp}");</pre>
Assigning a tuple to individually declared variables that are pre-declared.	<pre>string country; string capital; double gdp; (country, capital, gdp) = ("Malawi", "Lilongwe", 226.50); System.Console.WriteLine(\$@"The poorest country in the world in 2017 was { country}, {capital}: {gdp}");</pre>

Tuple: With Implicitly Type Variables

Description	Example Code
Assigning a tuple to individually declared and implicitly typed variables.	<pre>(var country, var capital, var gdp) = ("Malawi", "Lilongwe", 226.50); System.Console.WriteLine(\$@"The poorest country in the world in 2017 was { country}, {capital}: {gdp}");</pre>
Assigning a tuple to individually declared variables that are implicitly typed with a distributive syntax	<pre>var (country, capital, gdp) = ("Malawi", "Lilongwe", 226.50); System.Console.WriteLine(\$@"The poorest country in the world in 2017 was { country}, {capital}: {gdp}");</pre>

Tuple: With Named Items

Description	Example Code
Declaring a named item tuple and assigning it tuple values and then accessing the tuple items by name.	<pre>(string Name, string Capital, double Gdp) countryInfo = ("Malawi", "Lilongwe", 226.50); System.Console.WriteLine(\$"The poorest country in the world in 2017 was { countryInfo.Name}, {countryInfo.Capital}: { countryInfo.Gdp}");</pre>
Assigning a named item tuple to a single implicitly typed variable that is implicitly typed and then accessing the tuple items by name.	<pre>var countryInfo = (Name: "Malawi", Capital: "Lilongwe", Gdp: 226.50); System.Console.WriteLine(\$"The poorest country in the world in 2017 was { countryInfo.Name}, {countryInfo.Capital}: { countryInfo.Gdp}");</pre>

See Essential C# 7.0 by Mark Michaelis

Tuple: With Unnamed Items

Description	Example Code
Assigning an unnamed tuple to a single implicitly typed variable and then accessing the tuple elements by their Item-number property	<pre>var countryInfo = ("Malawi", "Lilongwe", 226.50); System.Console.WriteLine(\$"The poorest country in the world in 2017 was { countryInfo.Item1}, {countryInfo.Item2}: { countryInfo.Item3}");</pre>
Assigning a named item tuple to a single implicitly typed variable and then accessing the tuple items by their Item-number property.	<pre>var countryInfo = (Name: "Malawi", Capital: "Lilongwe", Gdp: 226.50); System.Console.WriteLine(\$"The poorest country in the world in 2017 was { countryInfo.Item1}, {countryInfo.Item2}: { countryInfo.Item3}");</pre>

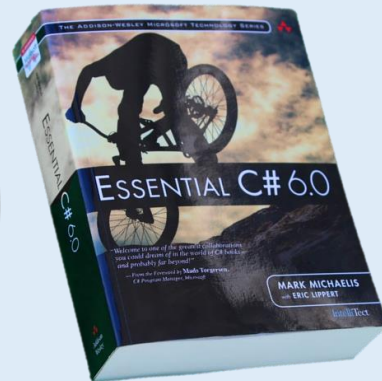
About Us

Mark Michaelis

Chief Technical Architect,
Author, Trainer

mark@IntelliTect.com

Twitter: @MarkMichaelis,
fb.com/MarkMichaelis



Professional
Scrum Master II

