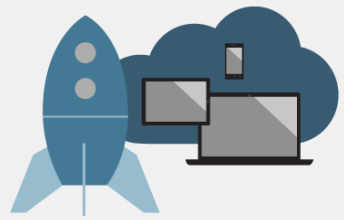
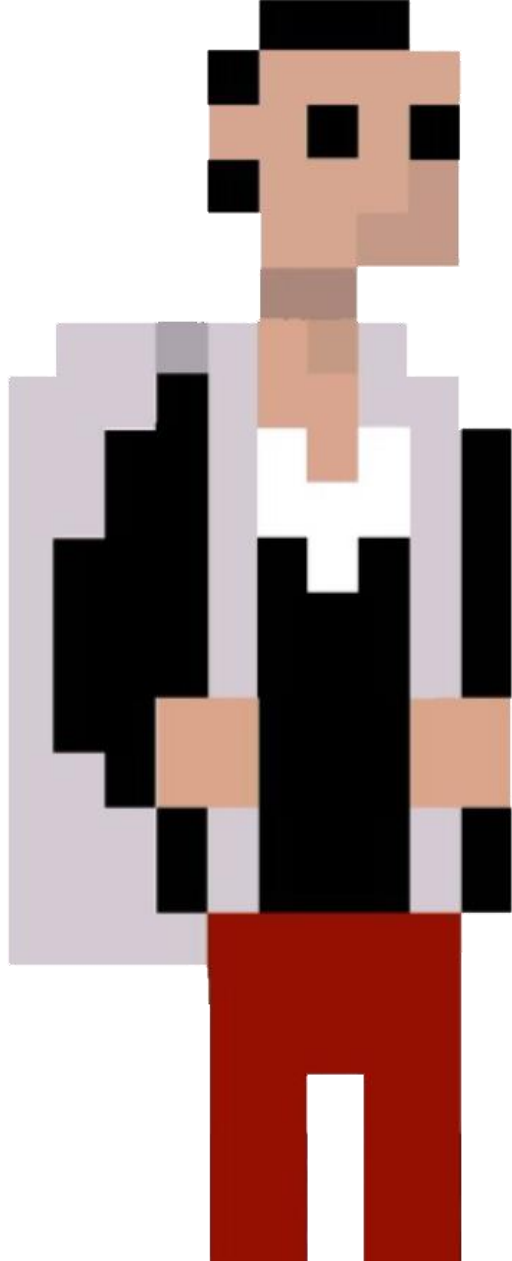


Formation



Développer des applications Windows Form avec C#





AHMED OUMEZZINE

.Net software engineer



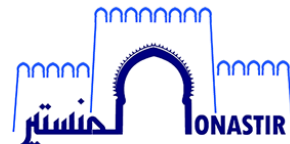
ahmedoumezzine@outlook.fr



facebook.com/Aoumezzine



twitter.com/AOumezzine





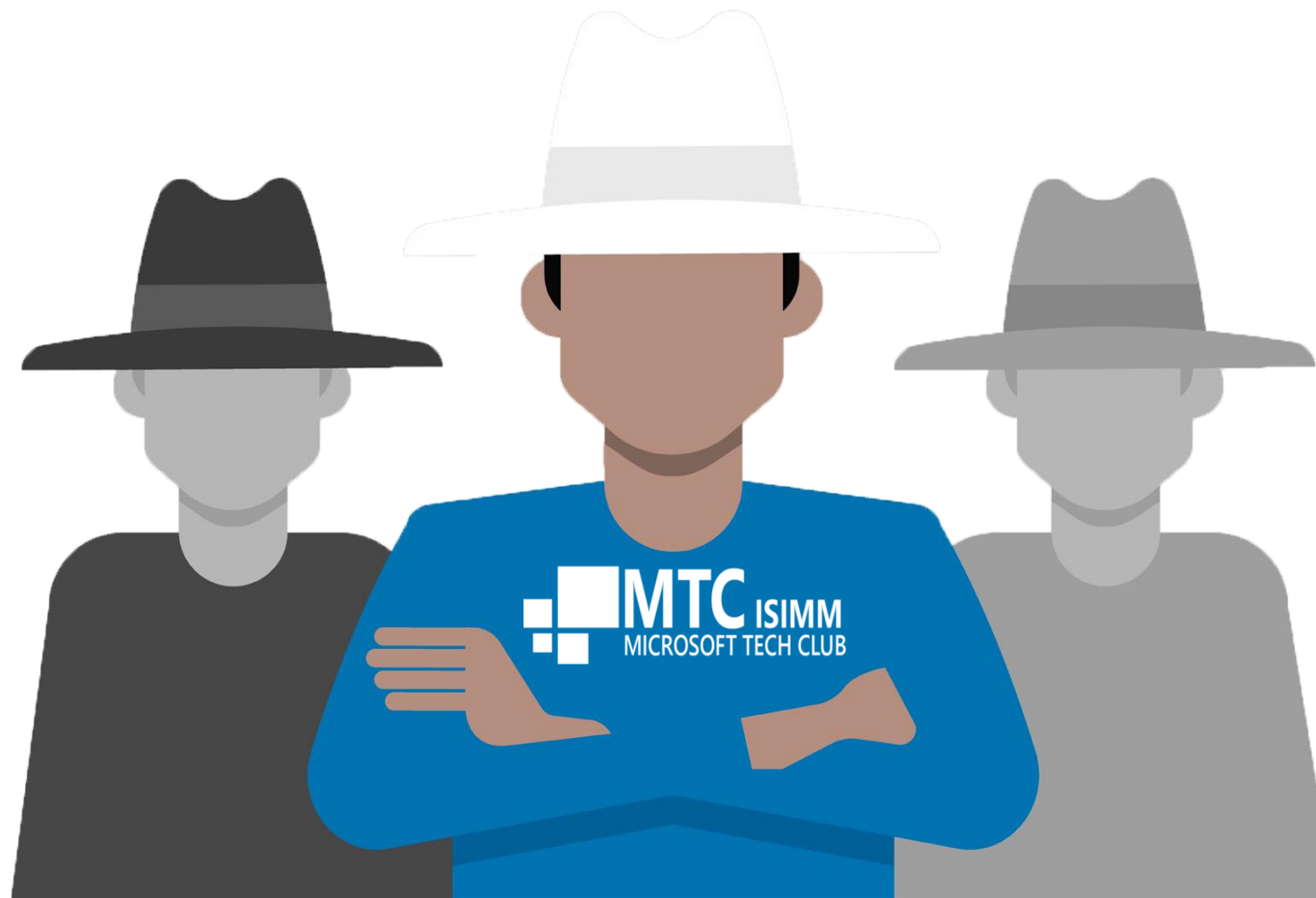
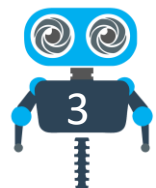
Windows Forms Application



**Development
(Orienté Object C#)**



**Accès à
une base de données**



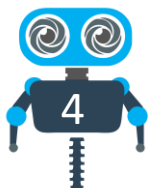
Le C# prononcé 'Ci Sharp' est un langage de programmation orienté objet, créé par la société Microsoft, et notamment par un de ses employés, **Anders Hejlsberg**.

Il a été créé afin que la plate-forme Microsoft .NET soit dotée d'un langage permettant d'utiliser toutes ses capacités.

Il est très proche du Java dont il reprend la syntaxe générale, ainsi qu'au niveau des concepts



Anders Hejlsberg



Windows Forms

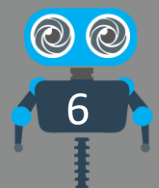
Les Windows Forms permettent de créer des interfaces graphiques sous Windows assez simplement.

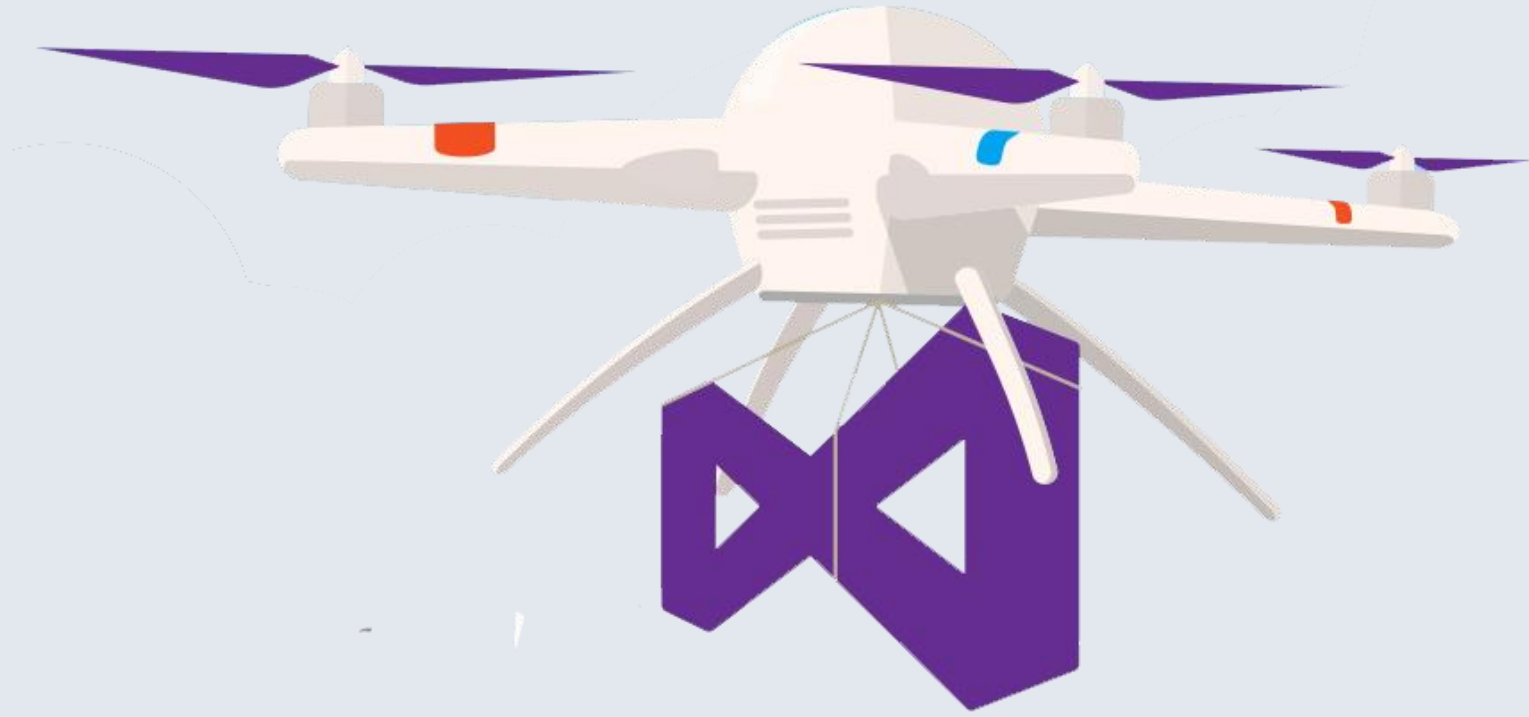




Microsoft Visual Studio s'agit d'un ensemble complet d'outils de développement permettant de générer des applications. Ils permettent de développer des applications dans divers langages s'appuyant sur .NET tels que C# , F# , Visual C++ et Visual Basic.

Il est possible de développer des types d'applications très variés : applications graphiques, en console, des bibliothèques de classes, des services Windows ou encore des sites web.





Visual Studio Community

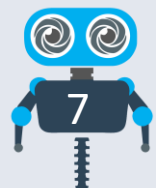
Il peut être utilisée que pour des projets open source, des recherches académiques et des scénarios d'environnement d'apprentissage en classe.

Visual Studio Professional

Des outils de développement professionnels et des services pour des développeurs individuels ou de petites équipes.

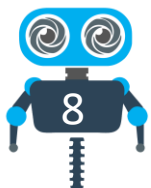
Visual Studio Enterprise

Une solution intégrée de bout en bout pour les équipes de toute taille avec des exigences de qualité et d'échelle exigeantes



Supported Features	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise
⊕ Supported Usage Scenarios	● ● ● ○	● ● ● ●	● ● ● ●
Development Platform Support ²	● ● ● ●	● ● ● ●	● ● ● ●
⊕ Integrated Development Environment	● ● ● ○	● ● ● ○	● ● ● ●
⊕ Advanced Debugging and Diagnostics	● ● ○ ○	● ● ○ ○	● ● ● ●
⊕ Testing Tools	● ○ ○ ○	● ○ ○ ○	● ● ● ●
⊕ Cross-platform Development	● ● ○ ○	● ● ○ ○	● ● ● ●
⊕ Collaboration Tools and Features	● ● ● ●	● ● ● ●	● ● ● ●

Pour en savoir plus : <https://www.visualstudio.com/vs/compare/>



What would you like to build?



Windows

Develop apps and games to reach every device running Windows.



Mobile Apps

Create native or hybrid apps targeting Android, iOS, and Windows.



Azure Apps

Build, manage, and deploy cloud apps that scale to Azure.



Web Apps

Develop modern web apps with powerful open tools and an extensible ecosystem.



Office

Utilize comprehensive tools for all types of Office development.



Games

Create cross-platform games with leading engines and frameworks.



Extensions

Write your own extensions for Visual Studio.

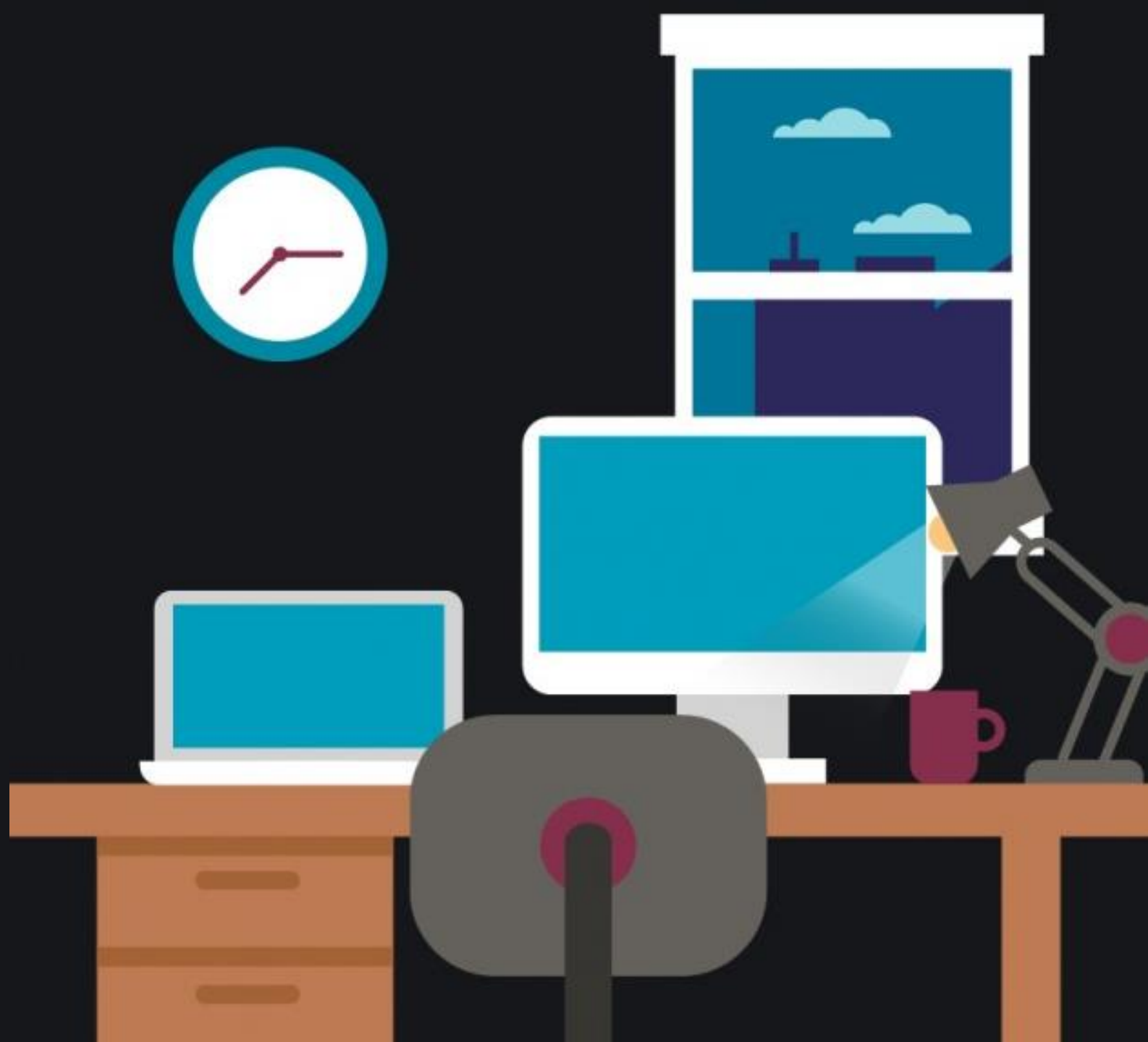


Database

Develop and deploy SQL Server and Azure SQL databases with ease.

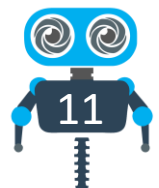
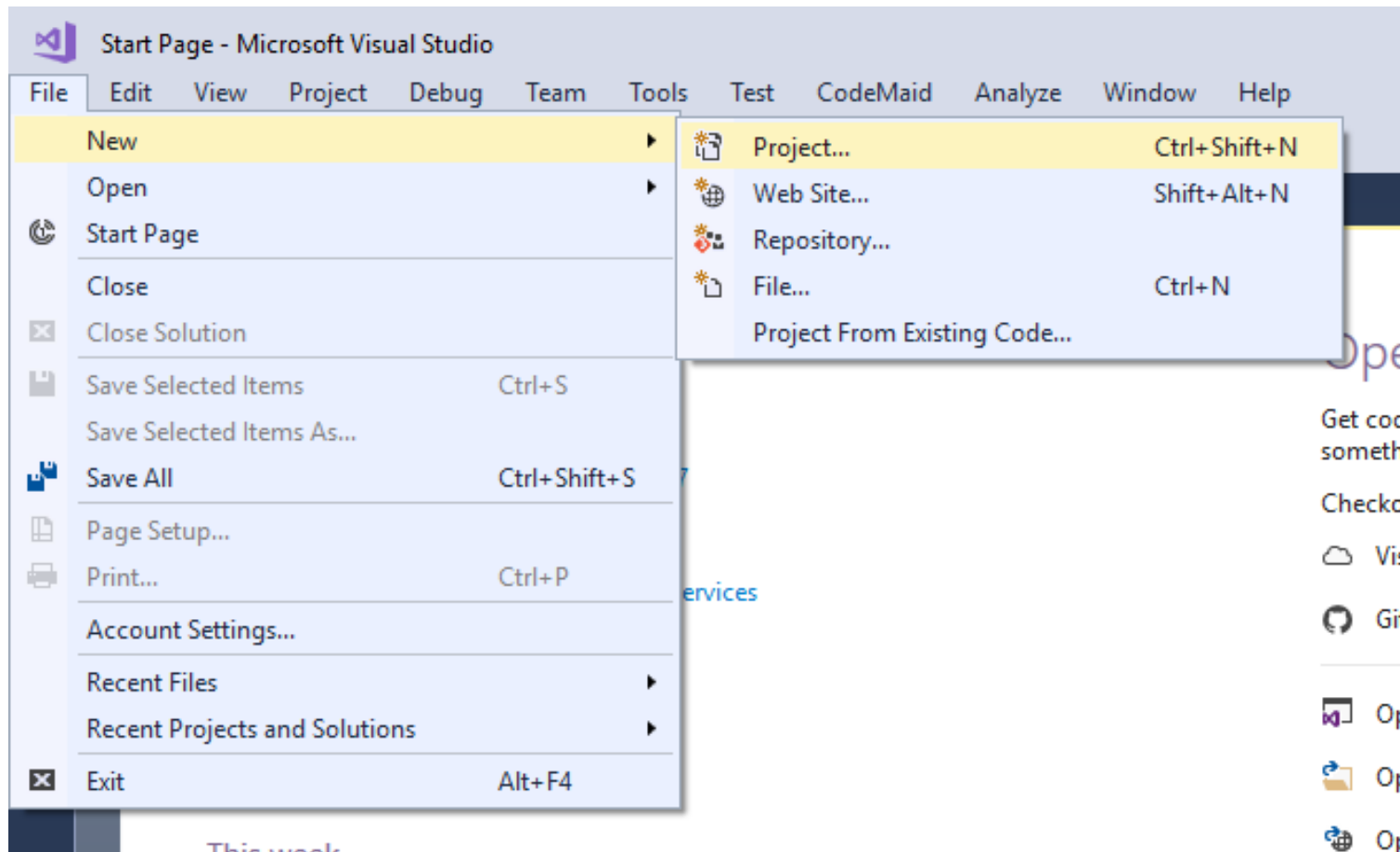


Installation Visual Studio

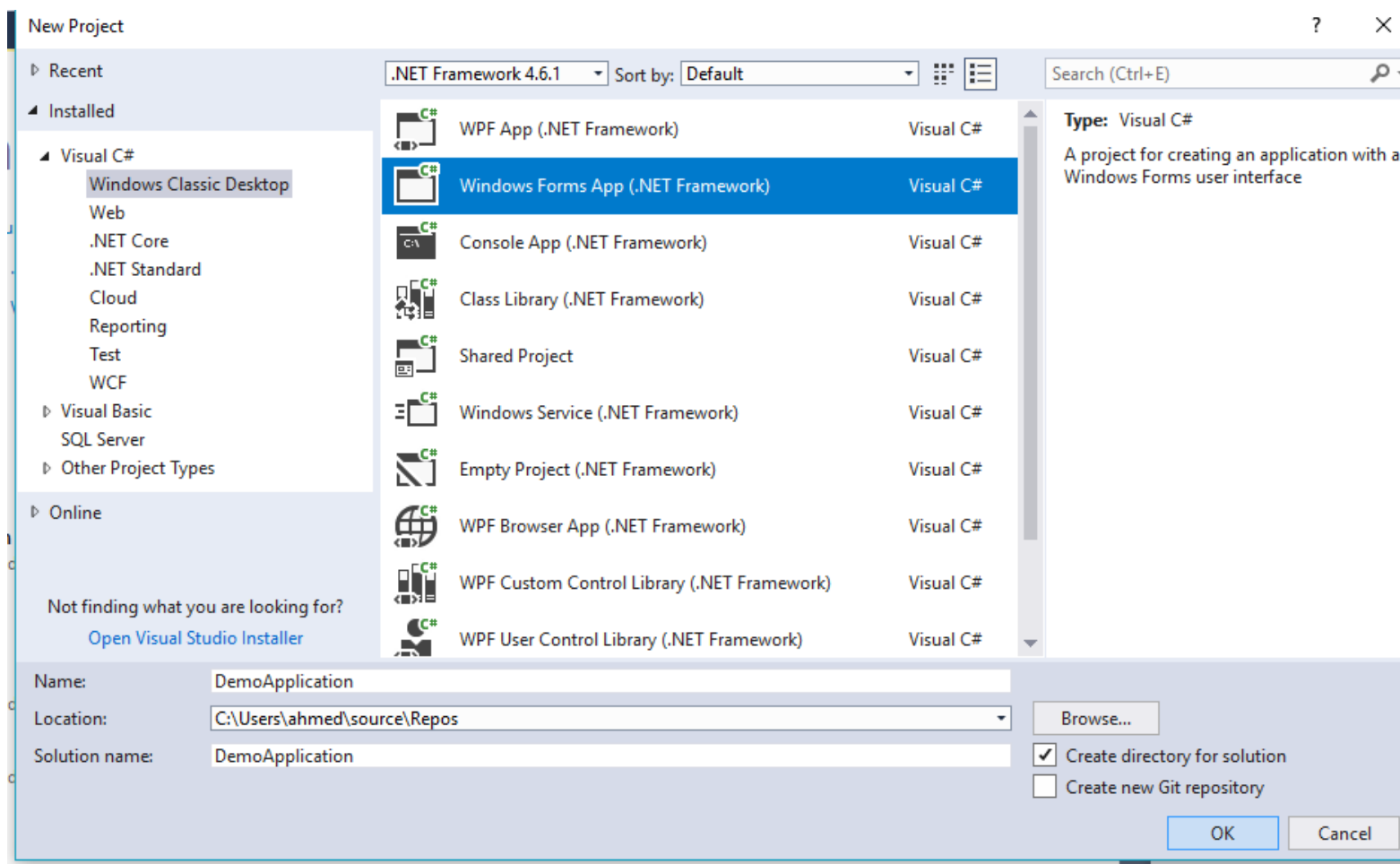


Étape 1)

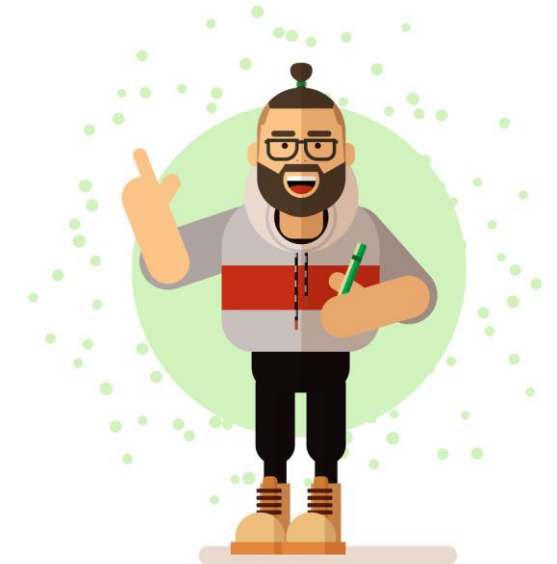
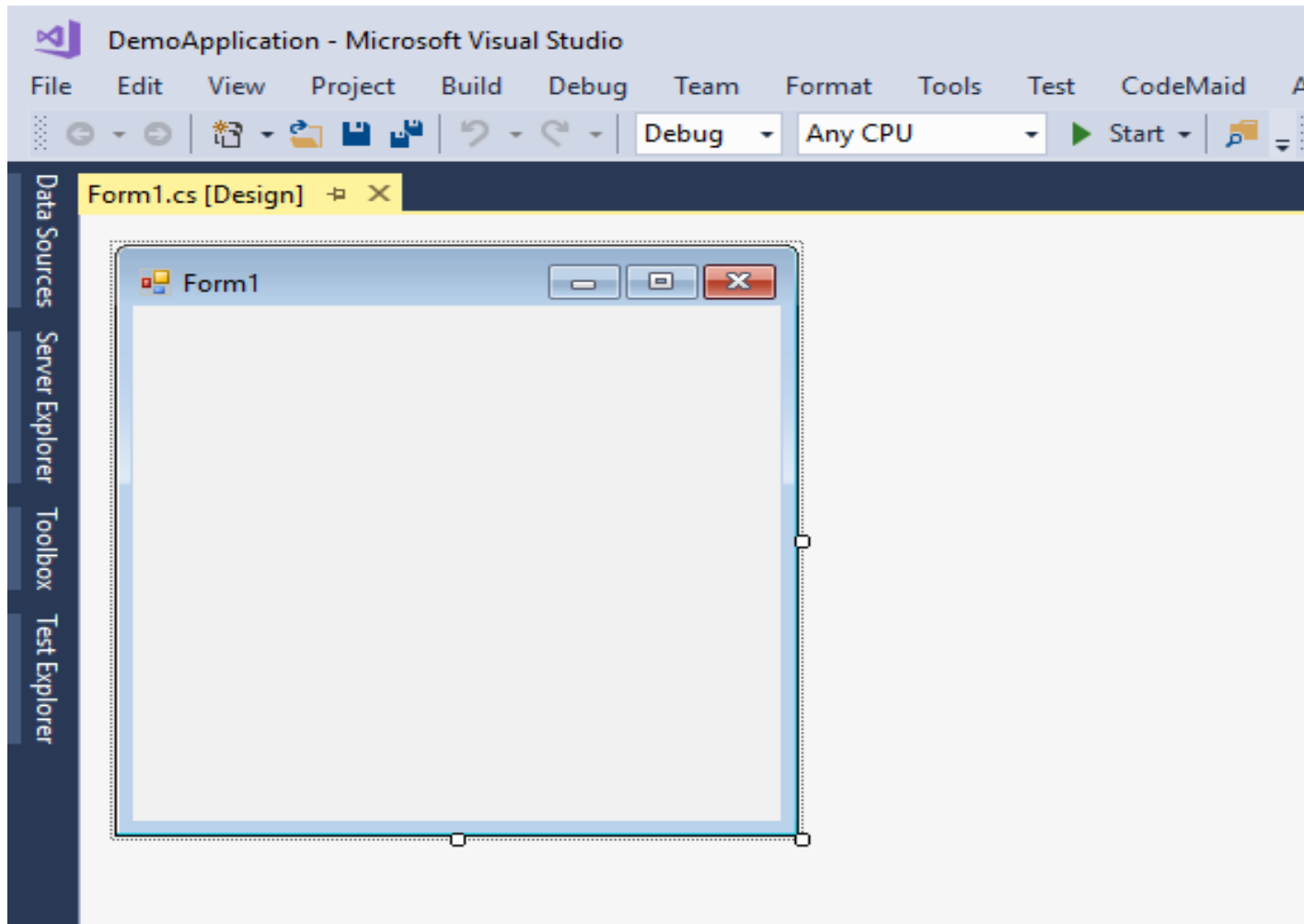
La première étape consiste à créer un nouveau projet dans Visual Studio.
Après le lancement de Visual Studio, vous devez choisir l'option de menu Nouveau -> Projet.



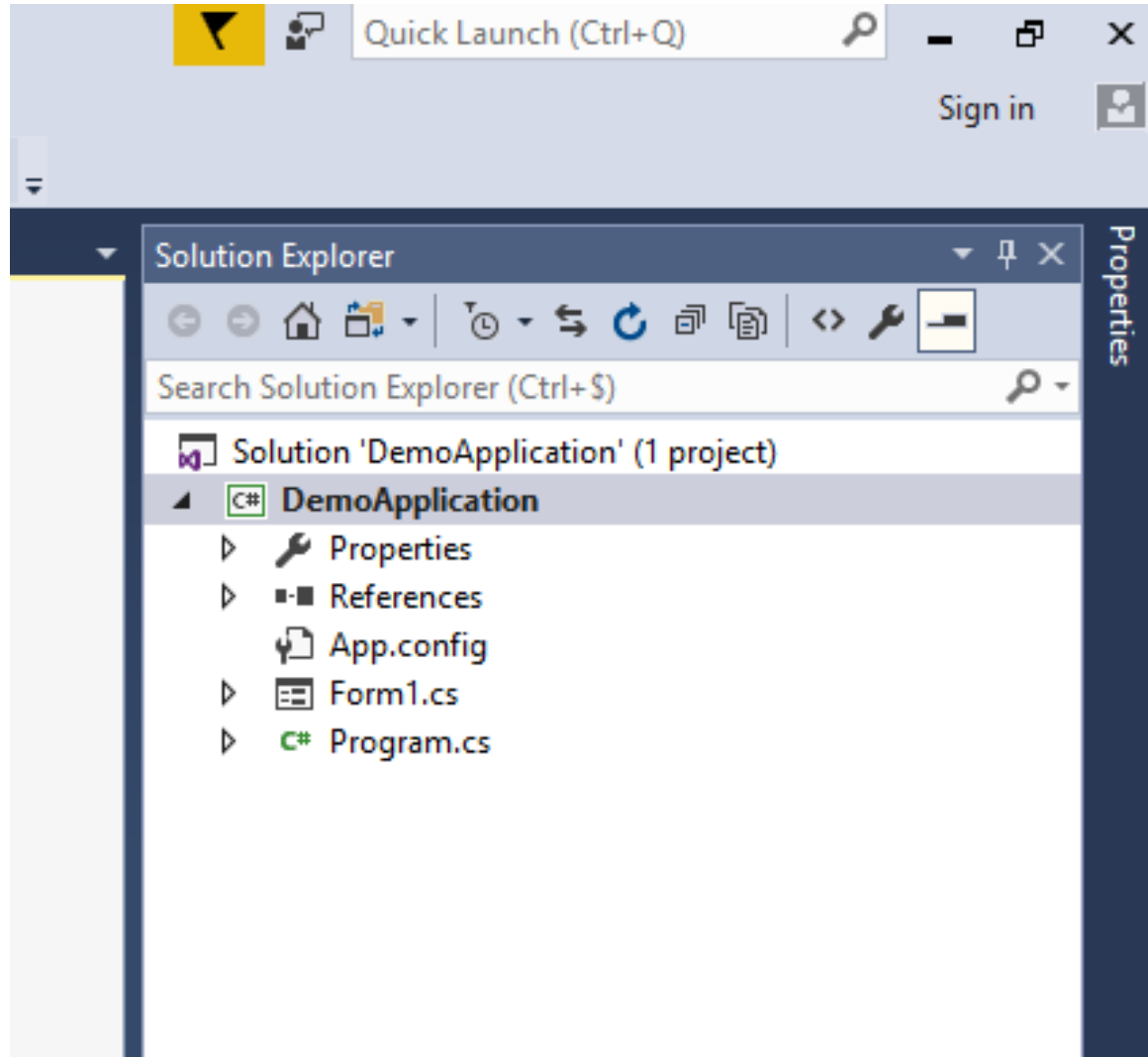
L'étape suivante consiste à choisir le type de projet en tant qu'application Windows Forms. Ici, nous devons également mentionner le nom et l'emplacement de notre projet.



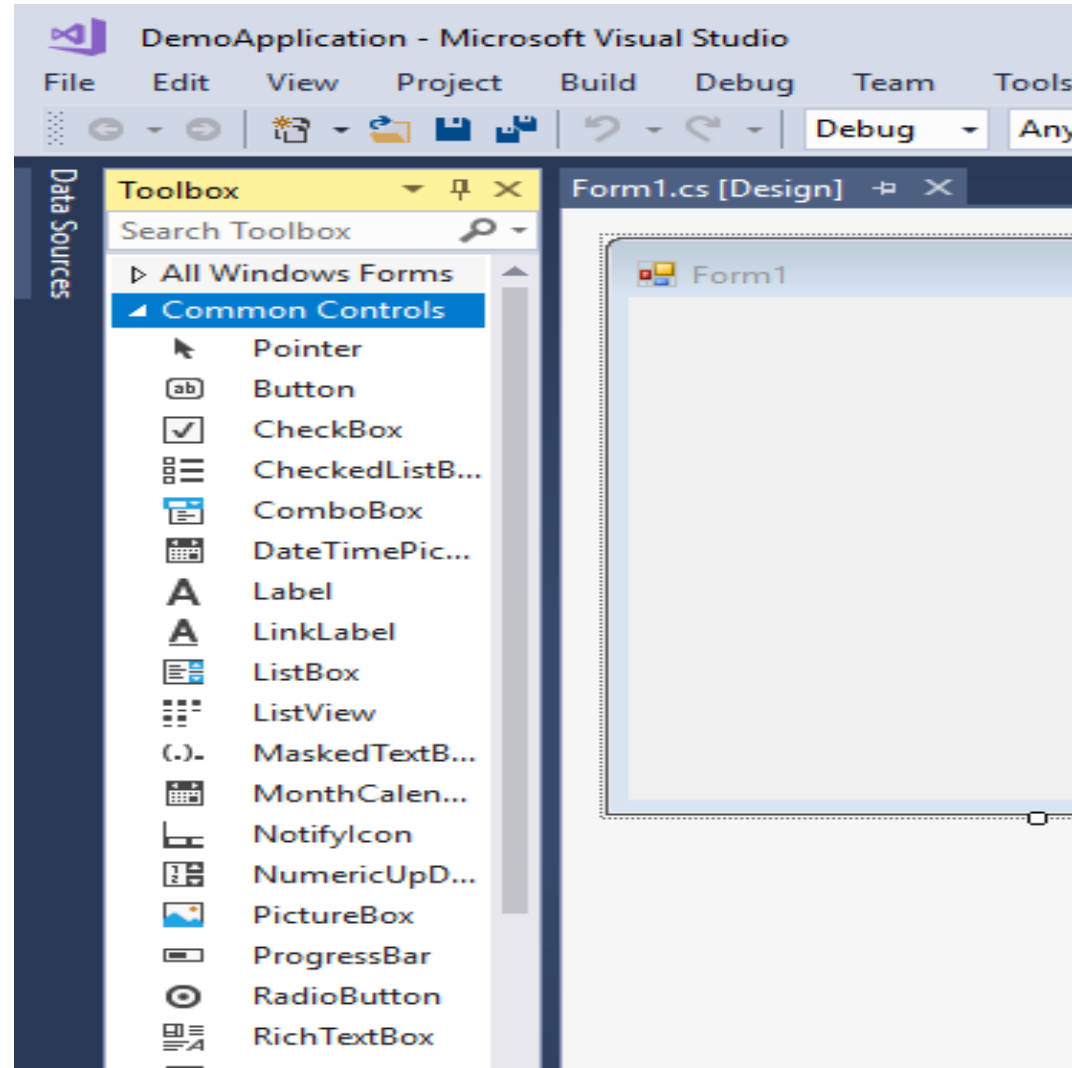
Si les étapes ci-dessus sont suivies, vous obtiendrez la sortie ci-dessous dans Visual Studio.



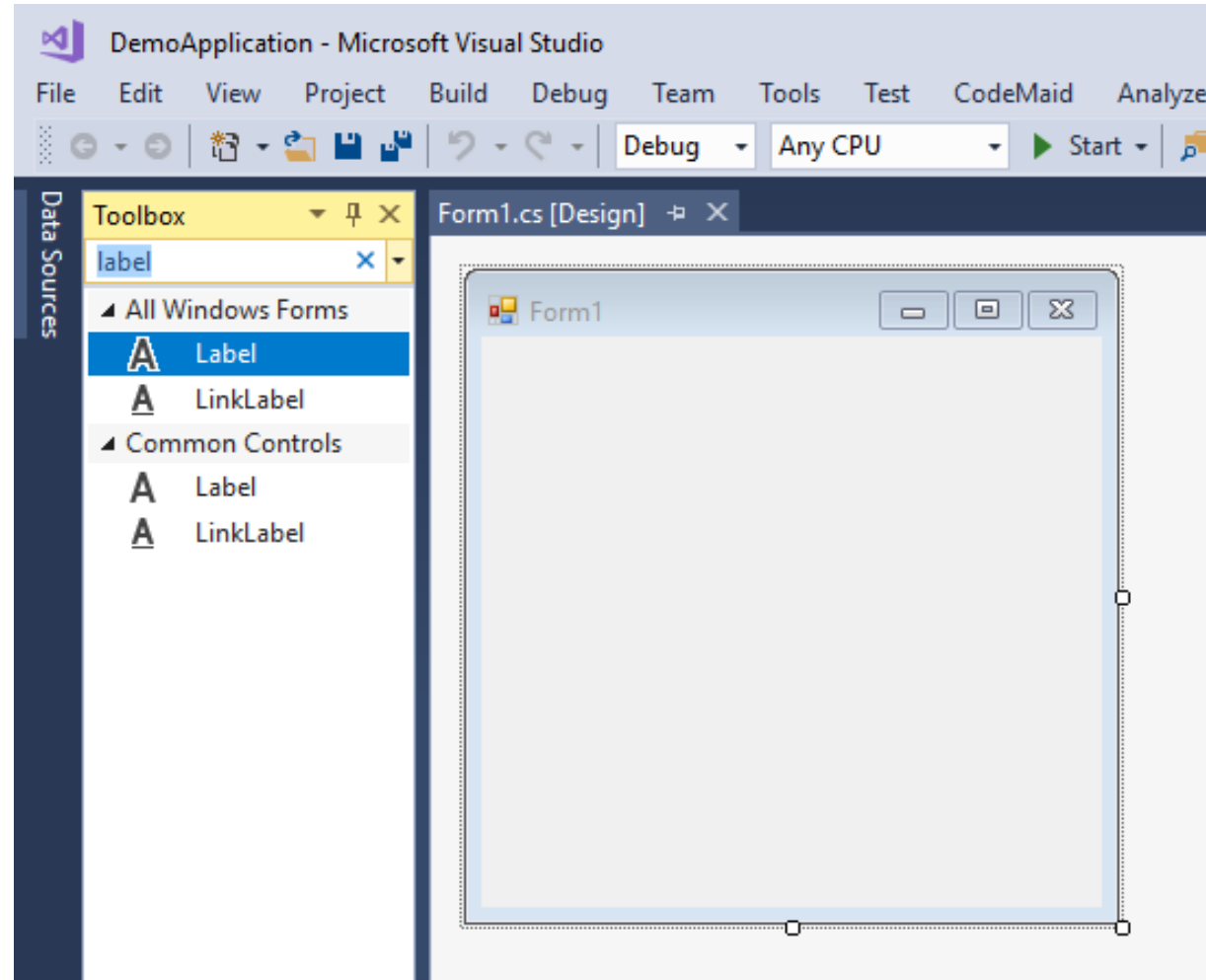
Dans l'**explorateur de solutions**, vous pourrez également voir la solution DemoApplication. Cette solution contiendra les deux fichiers de projet ci-dessous



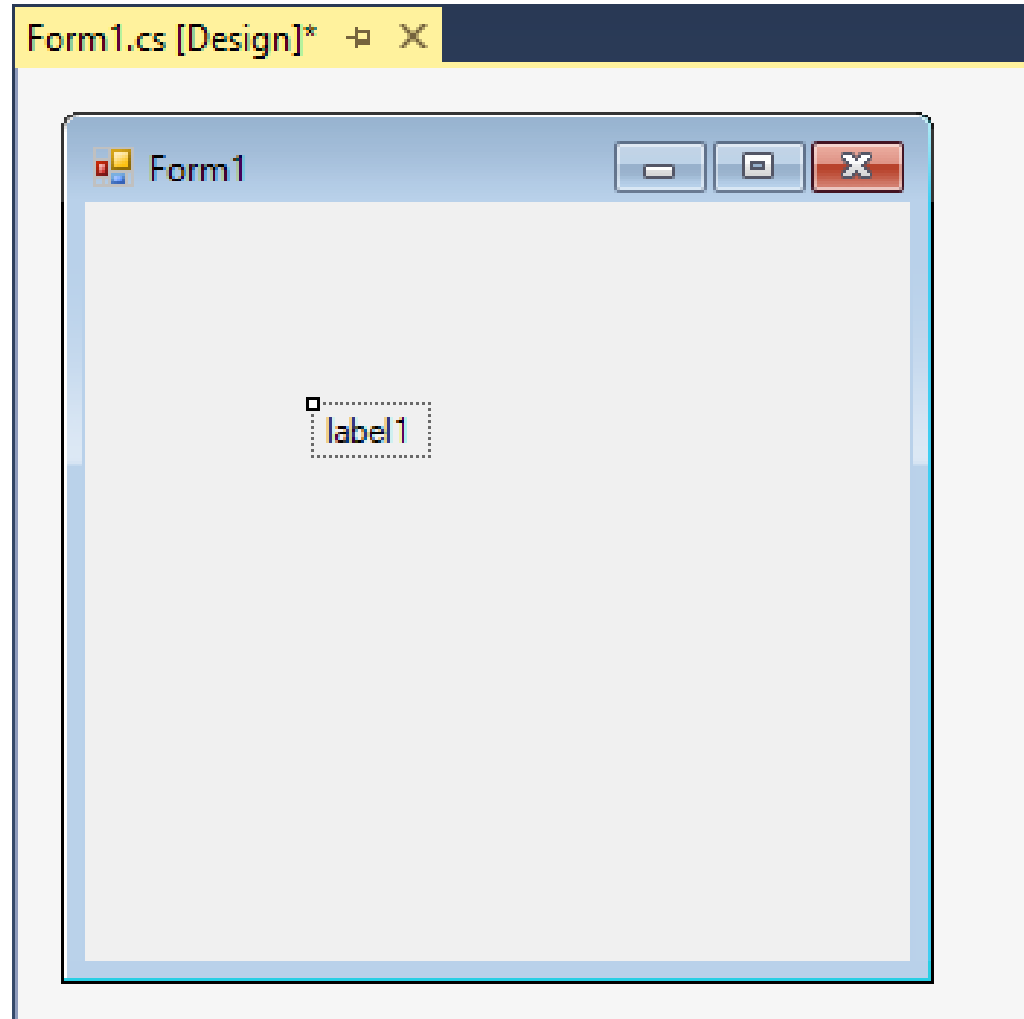
Étape 2) Sur le côté gauche de Visual Studio, vous verrez également un ToolBox. La boîte à outils contient tous les contrôles qui peuvent être ajoutés à un Windows Forms.



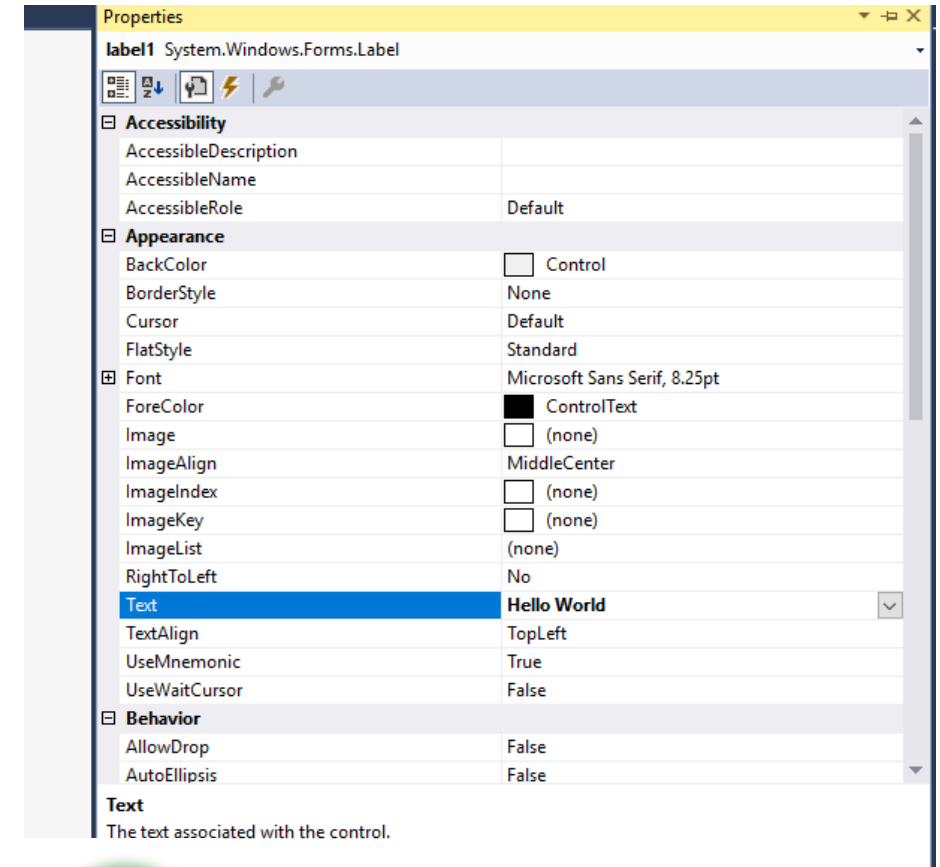
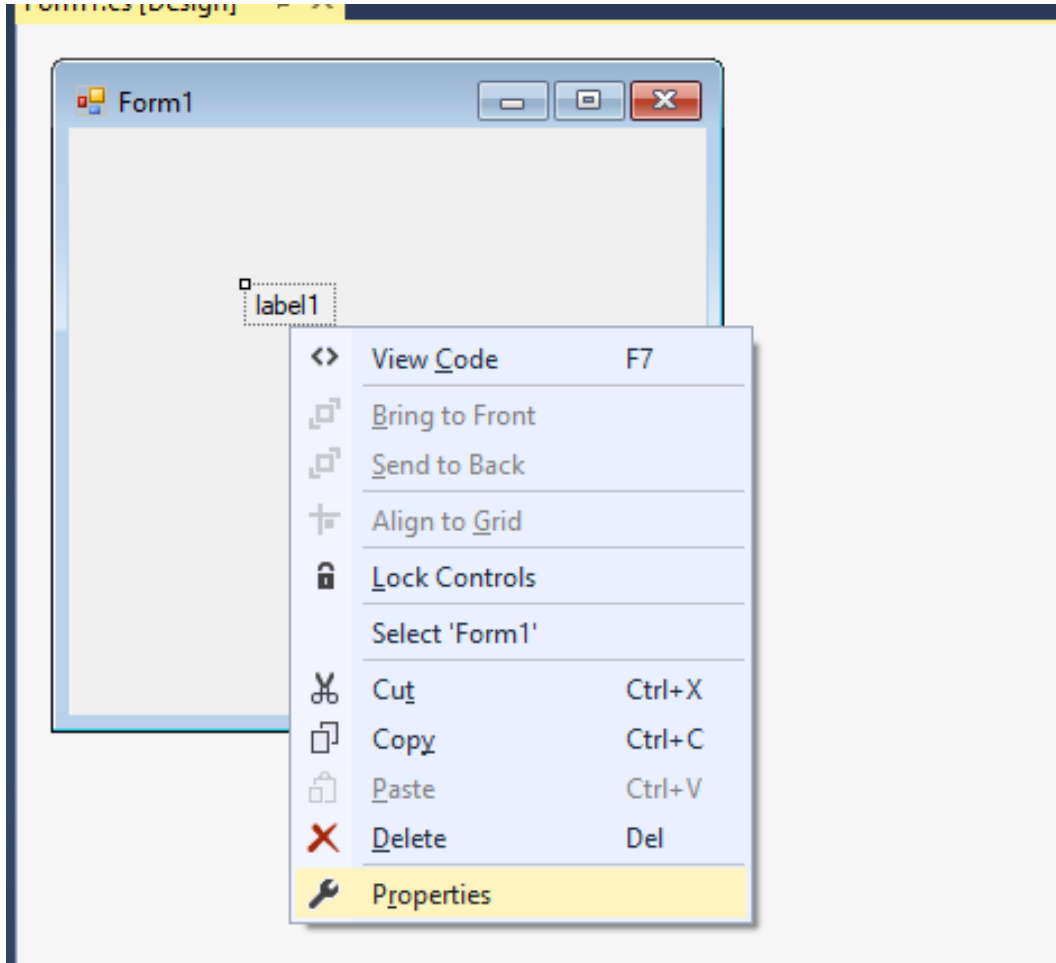
Étape 3) Dans cette étape, nous allons maintenant ajouter une Label au Forms qui affichera "Hello World". Dans la boîte à outils, vous devrez choisir le contrôle Label et le faire simplement glisser sur le Forms.



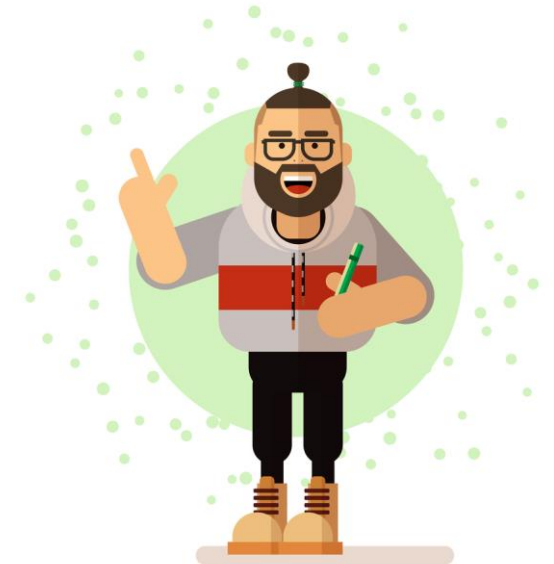
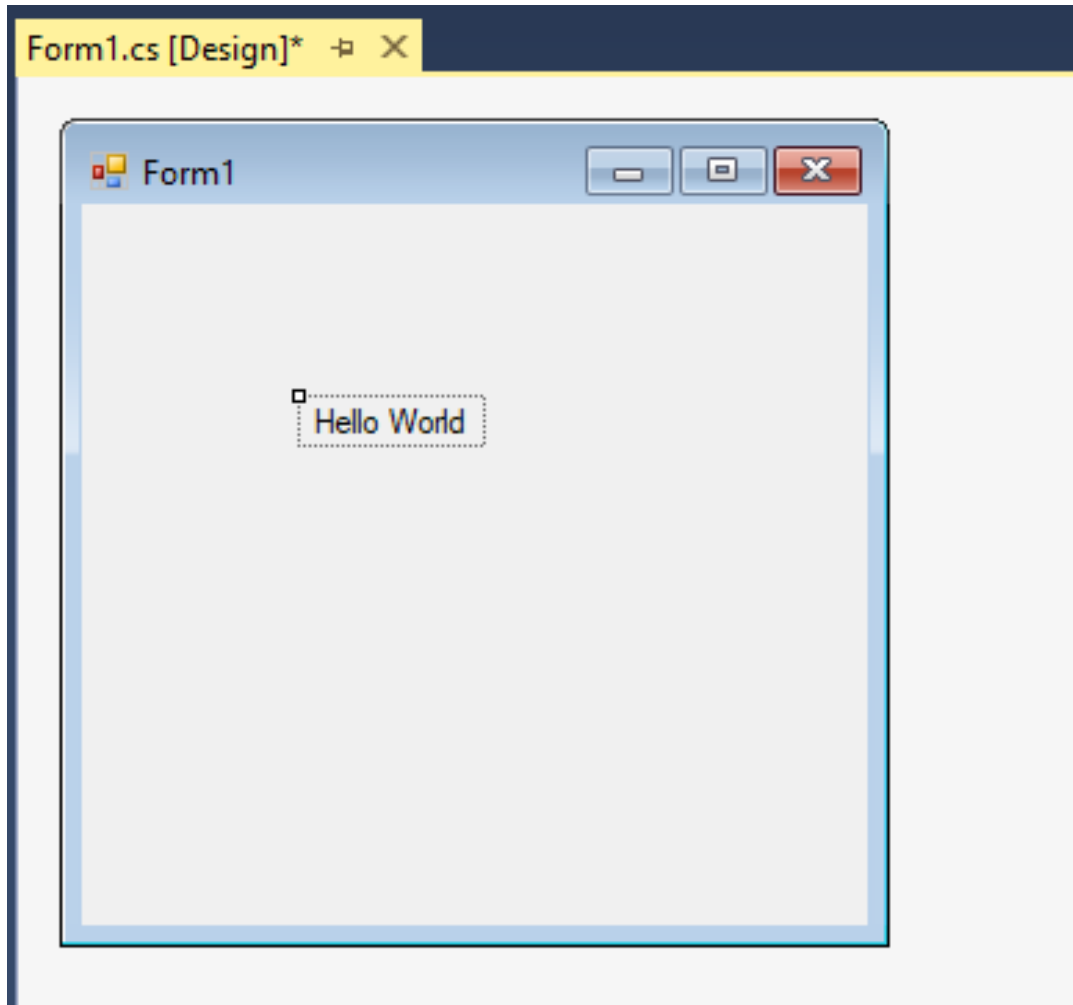
Une fois que vous avez déplacé Label dans le Forms, vous pouvez voir Label incorporée dans le Forms, comme indiqué ci-dessous.

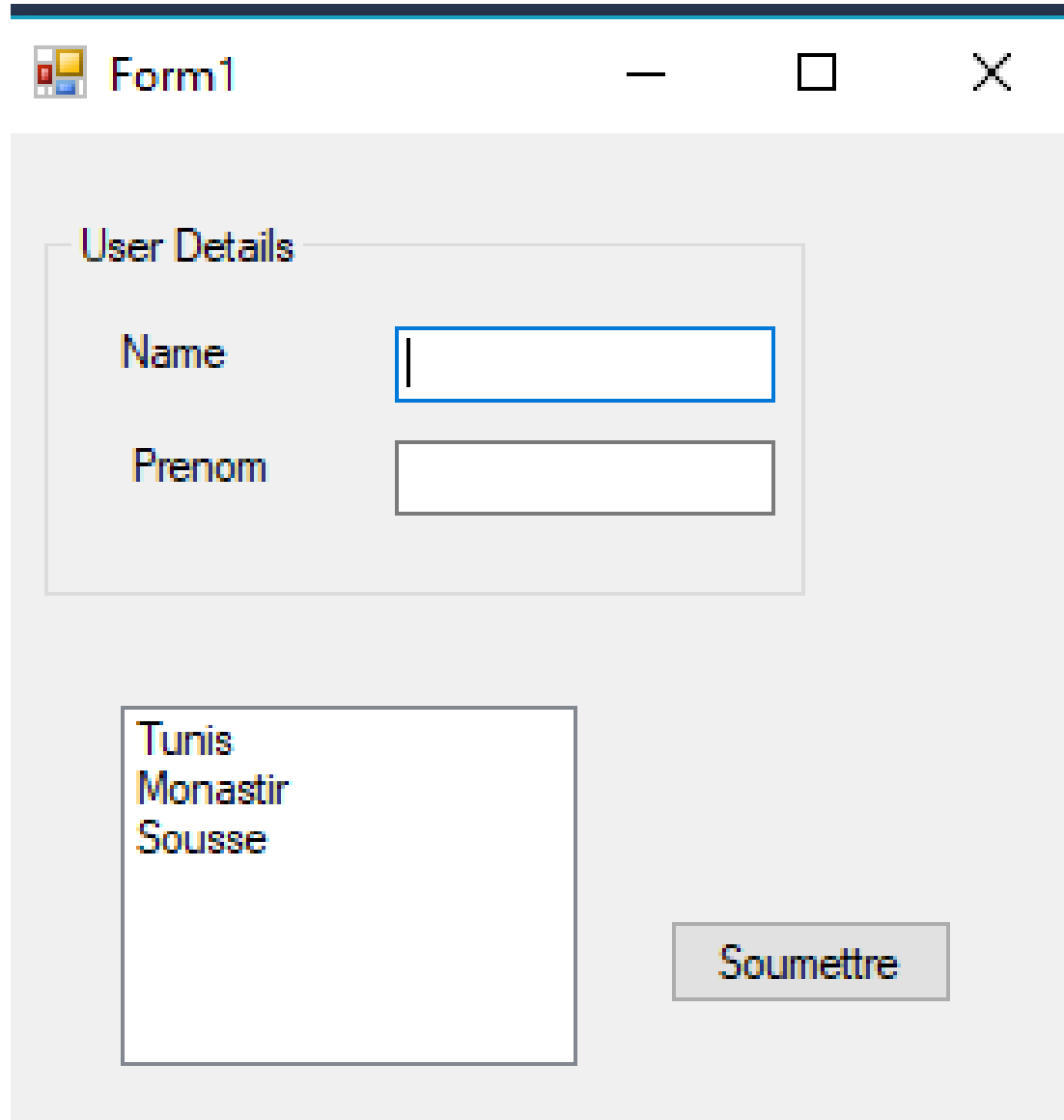


Étape 4) L'étape suivante consiste à accéder aux propriétés du contrôle et à modifier le texte en «Hello World».



Si vous suivez toutes les étapes ci-dessus et exécutez votre programme dans Visual Studio, vous obtiendrez la sortie suivante.





Form1

User Details

Name

Prenom

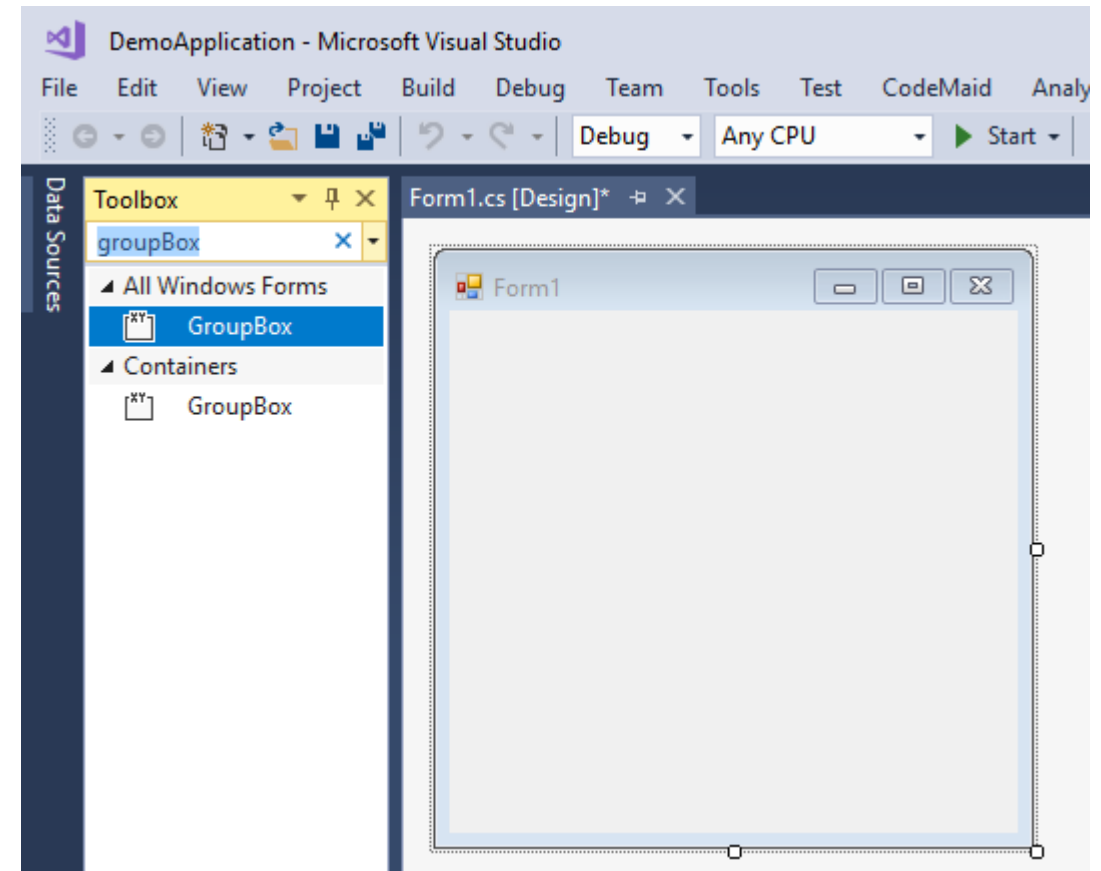
Tunis
Monastir
Sousse

Soumettre

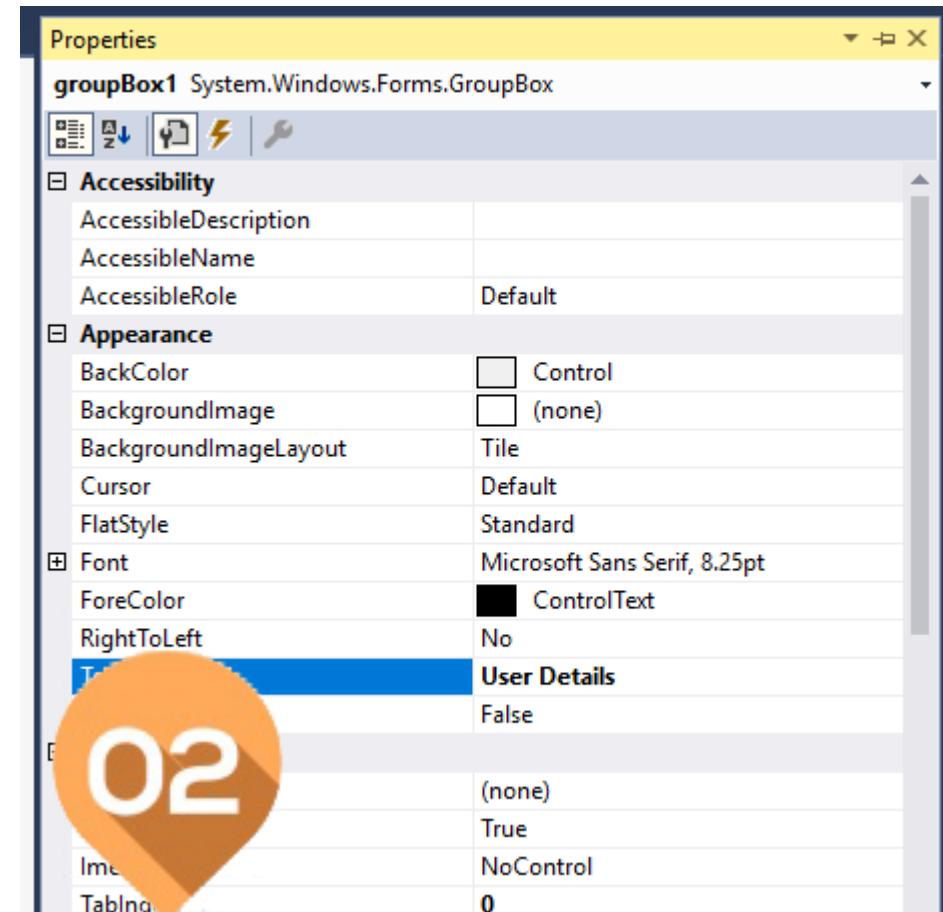
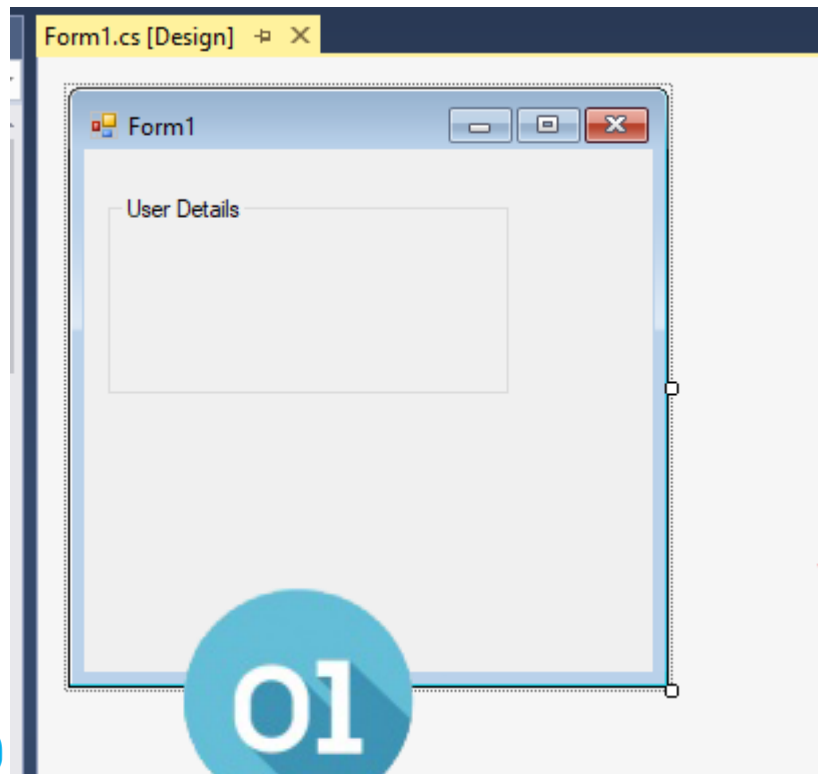
GroupBox

Une zone de groupe est utilisée pour regrouper les contrôles dans une section. Prenons un exemple, si vous aviez une collection de contrôles pour entrer des détails tels que le nom et l'adresse d'une personne.

La première étape consiste à faire glisser le contrôle Groupbox sur le Windows Form à partir de la boîte à outils comme indiqué ci-dessous

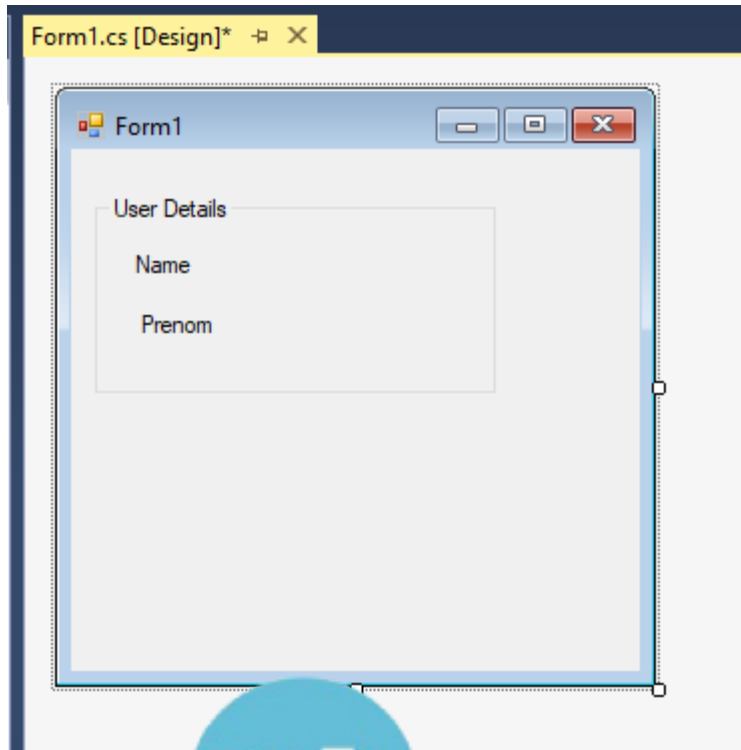


Étape 2) Une fois la zone de groupe ajoutée, accédez à la fenêtre des propriétés en cliquant sur le contrôle de la zone de liste. Dans la fenêtre des propriétés, accédez à la propriété Text et remplacez-la par "User Details".

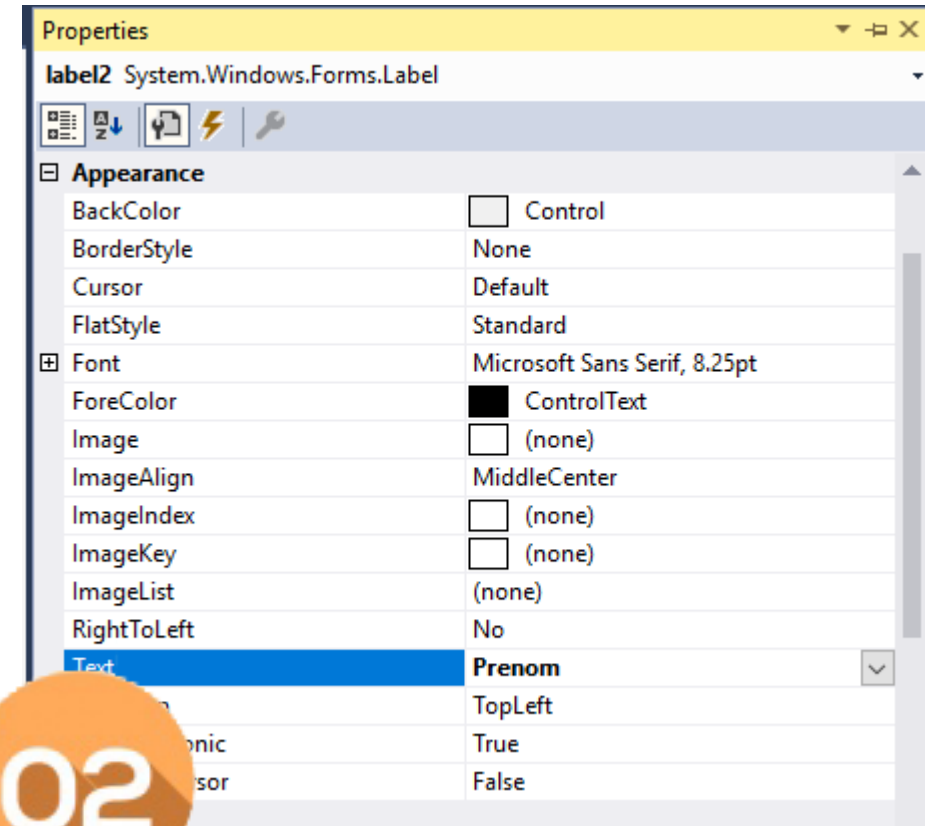


Label

Le contrôle Label est utilisé pour afficher un texte ou un message à l'utilisateur sur le Forms.



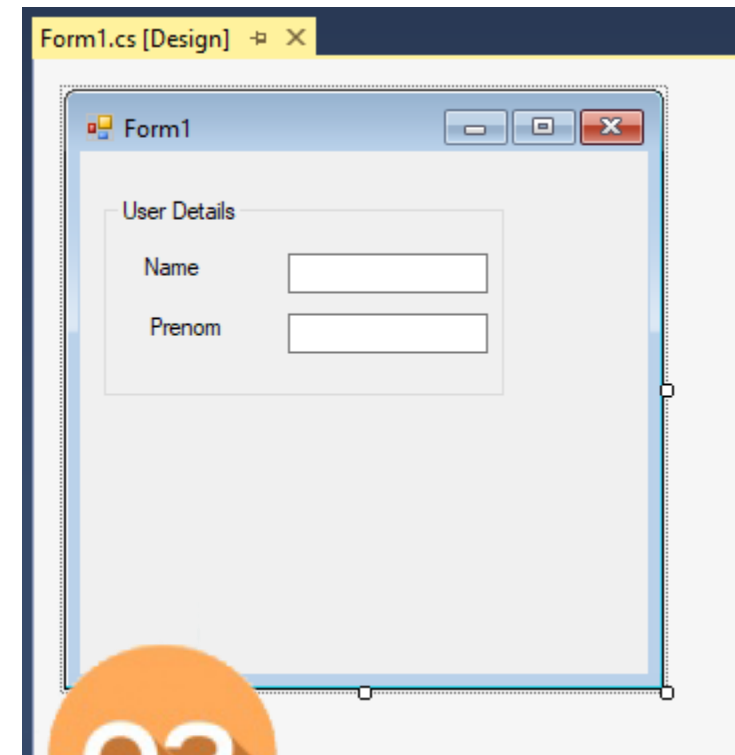
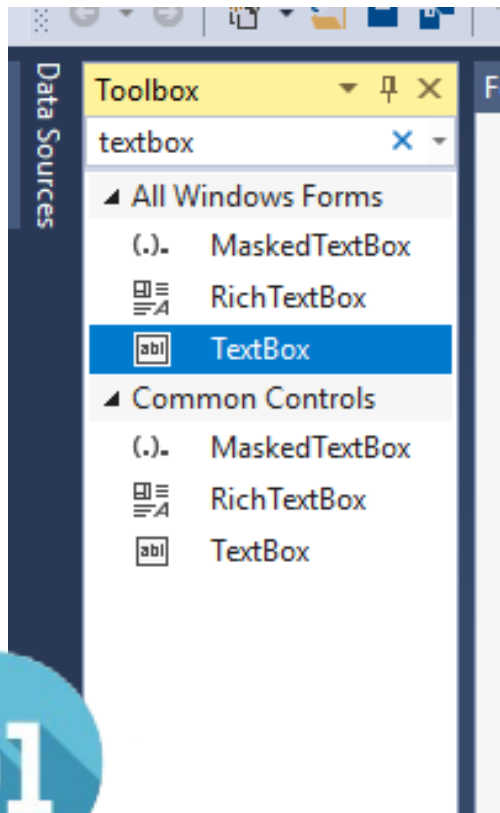
01



02

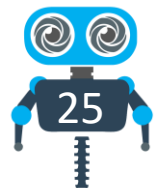
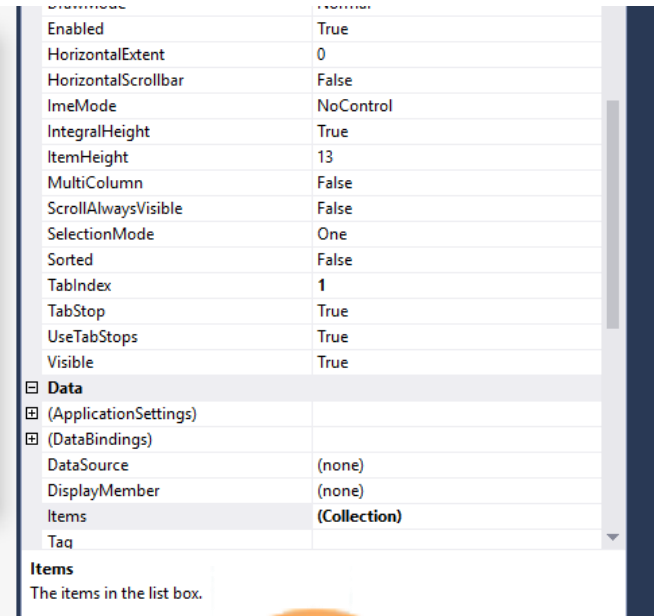
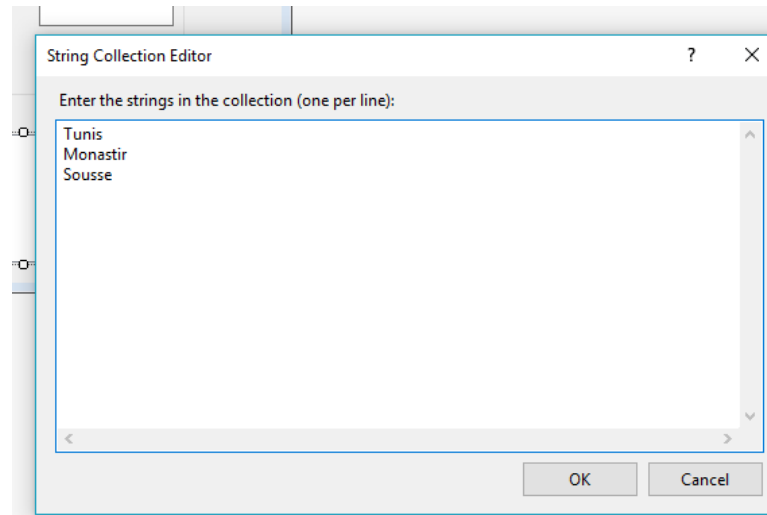
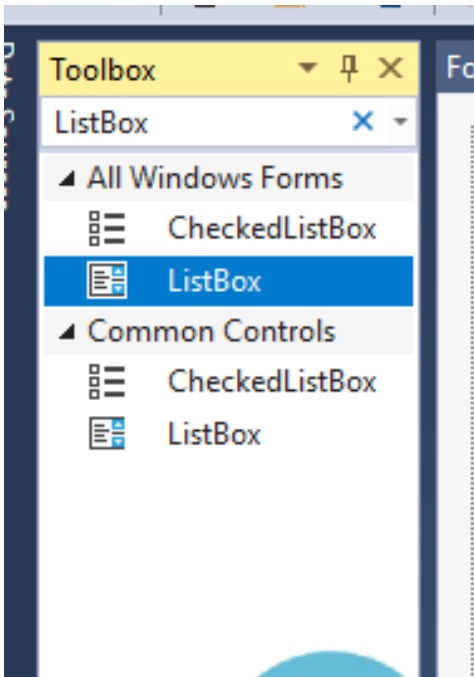
Zone de texte (TextBox)

Une zone de texte est utilisée pour permettre à un utilisateur d'entrer du texte dans l'application de formulaires.

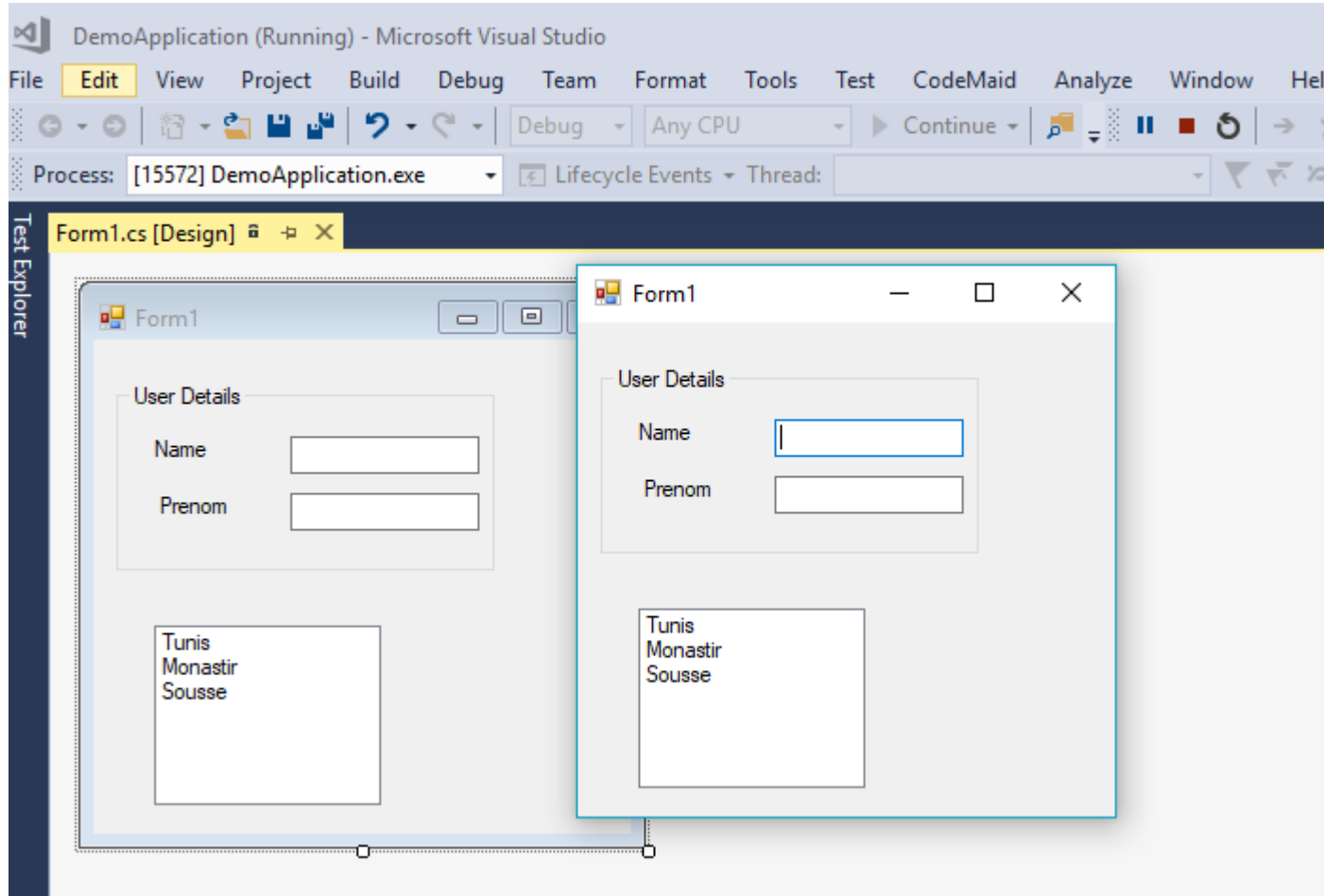


ListBox

listBox est utilisée pour présenter une liste d'éléments sur le Windows Forms.

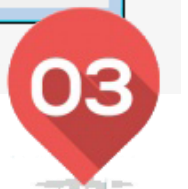
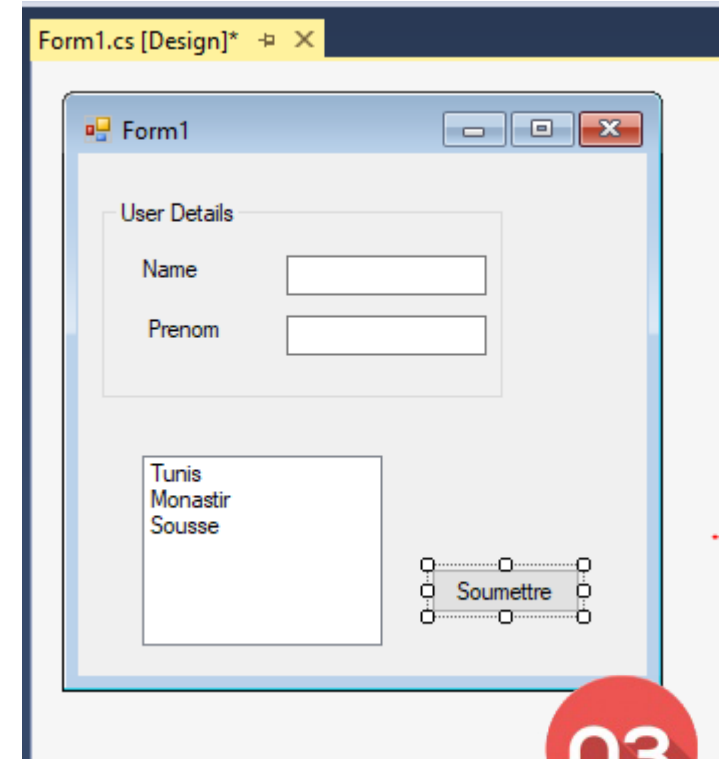
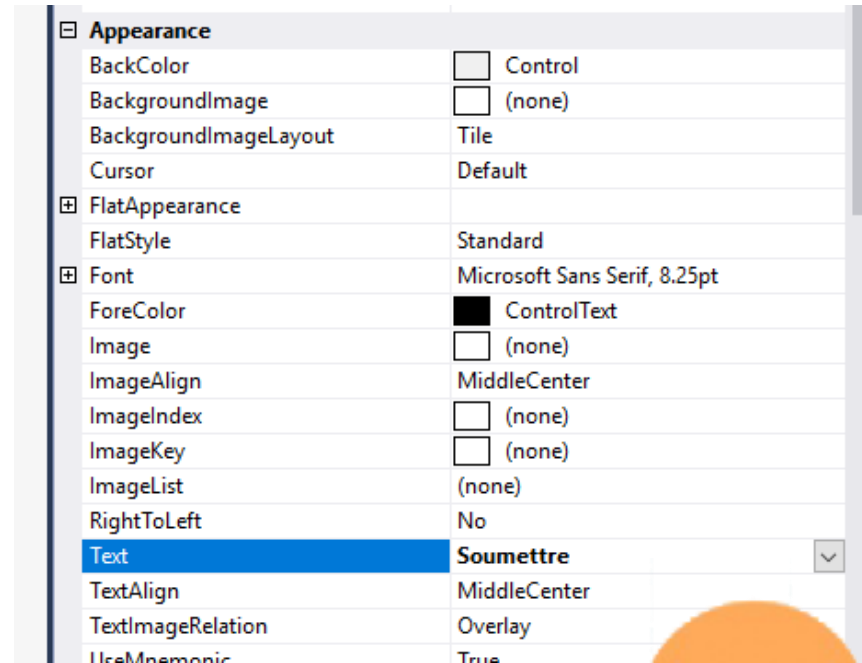
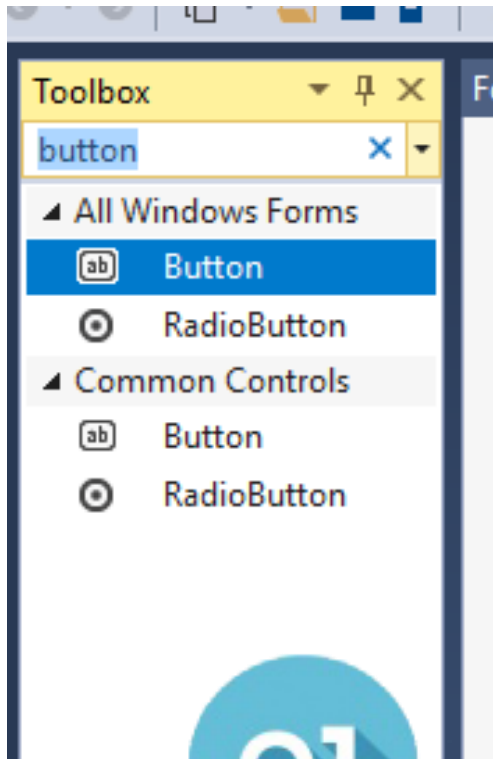


Celui que vous faites les changements ci-dessus, vous verrez la sortie suivante
Sortie:

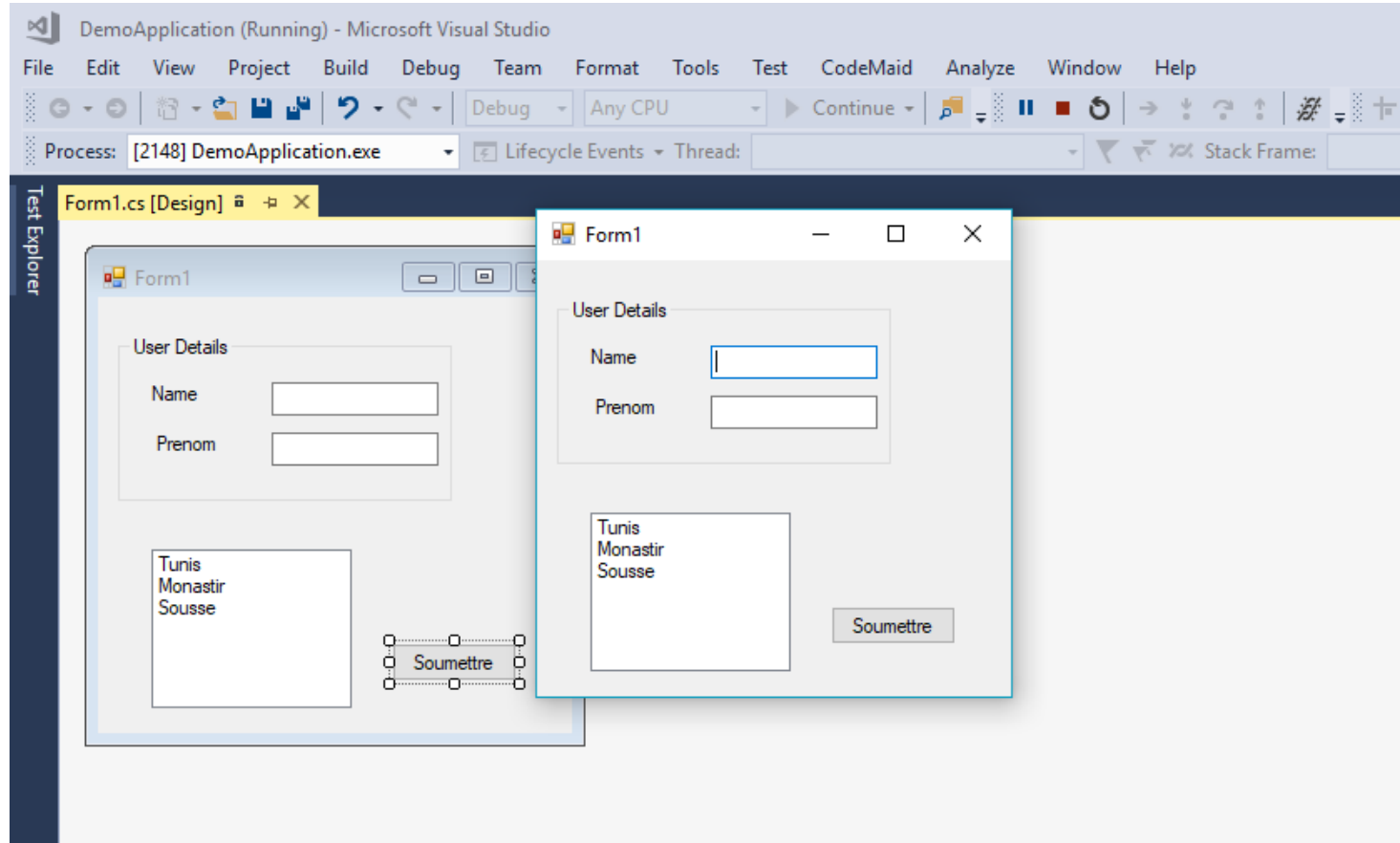


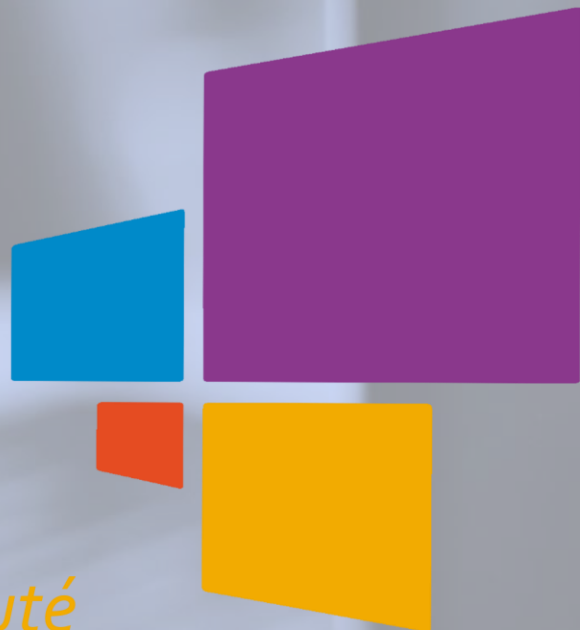
Bouton

Un bouton permet à l'utilisateur de cliquer sur un bouton pour démarrer le traitement du formulaire.



Celui que vous faites les changements ci-dessus, vous verrez la sortie suivante
Sortie:





Communauté
DOTNET Bettouensi

```
346 .widget-area-sidebar: newWindow {
347     font-size: 13px;
348 }
349 }
350
351 /* =Menu
352 -----
353
354 #access {
355     display: inline-block;
356     height: 69px;
357     float: right;
358     margin: 11px 28px 0px 0px;
359     max-width: 800px;
360 }
361
362 #access ul {
363     font-size: 13px;
364     list-style: none;
365     margin: 0 0 0 -0.8125em;
366     padding-left: 0;
367     z-index: 99999;
368     text-align: right;
369 }
370
371 #access li {
372     display: inline-block;
373     text-align: left;
```

Tout d'abord, une classe est un groupe de méthodes et de variables apparentées.

```
public class MyClass
{
    public MyClass()
    {
    }

    public void MyMethod(int parameter1, string parameter2)
    {
        Console.WriteLine("First Parameter {0}, second parameter {1}", parameter1, parameter2);
    }

    private int myPropertyVar;

    public int MyProperty
    {
        get { return myPropertyVar; }
        set { myPropertyVar = value; }
    }
}
```

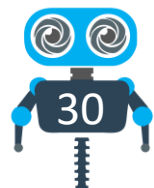
Access Modifier

Class name

Constructor

Method\Function

Property

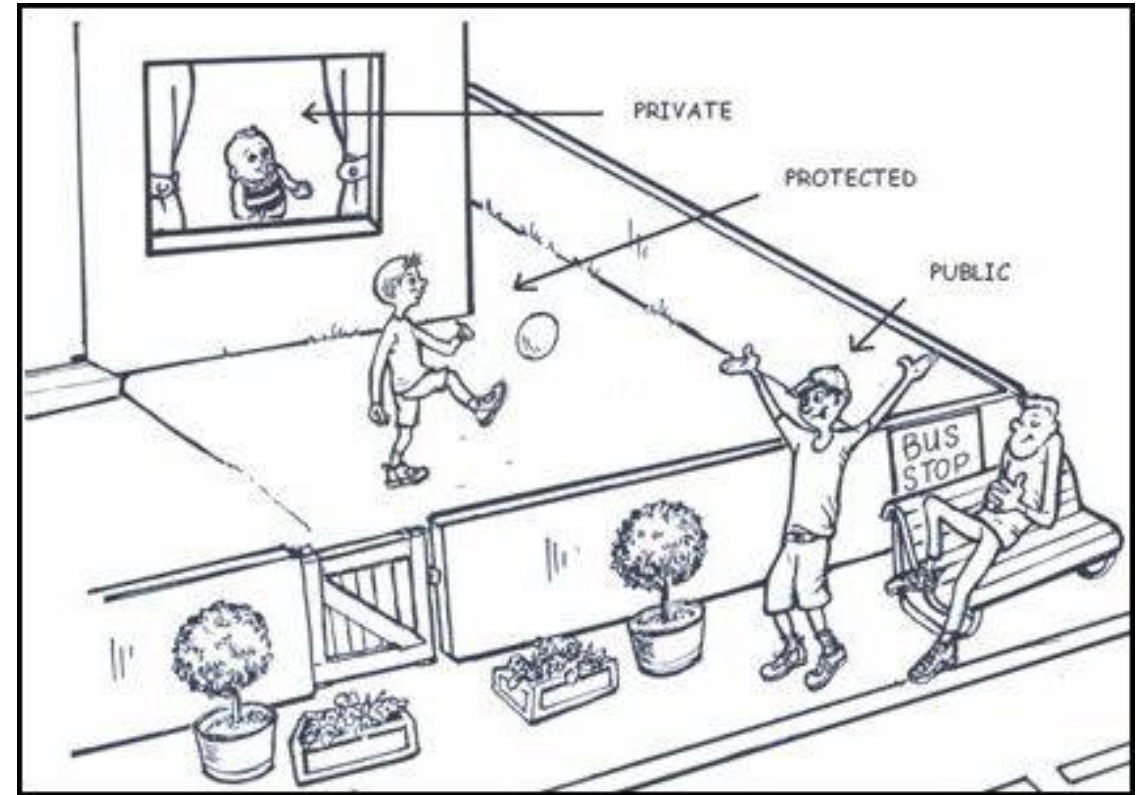


La visibilité d'une classe, d'une méthode, d'une variable ou d'une propriété nous indique comment accéder à cet élément.

public - le membre peut être atteint de n'importe où.

protected - les membres ne peuvent être atteints que depuis la même classe ou depuis une classe

privé - ne peut être atteint que par des membres de la même classe.



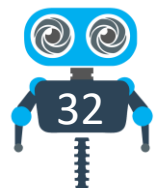
Server: mysql wampserver » Database: workshop » Table: Person

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Triggers](#)

Table name: Add column(s)

Structure

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	
PersonID	INT	11	None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
Name	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
Prenom	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
Ville	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>



```
using DotNetTN.Connector.SQL.Mapping;
```

```
namespace DemoApplication.Entity
```

```
{
```

```
    [Table("Person")]
```

```
    0 references
```

```
    public class Person
```

```
    {
```

```
        [Column(IsPrimaryKey = true, IsIdentity = true, ColumnName = "PersonID")]
```

```
        0 references
```

```
        public int Id { get; set; }
```

```
        [Column(ColumnName = "Name")]
```

```
        0 references
```

```
        public string Name { get; set; }
```

```
        [Column(ColumnName = "Prenom")]
```

```
        0 references
```

```
        public string Prenom { get; set; }
```

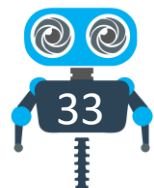
```
        [Column(ColumnName = "Ville")]
```

```
        0 references
```

```
        public string Ville { get; set; }
```

```
    }
```

```
}
```



```
using DotNetTN.Connector.SQL;
using DotNetTN.Connector.SQL.Entities;

namespace DemoApplication.Base
{
    0 references
    public class DotNetTNBase
    {
        public static string ConnectionString = "server=localhost;Database=aotn;Uid=root;Pwd=";

        0 references
        public static SqlClient GetInstance()
        {
            SqlClient db = new SqlClient(
                new Config()
                {
                    ConnectionString = ConnectionString,
                    DbType = DbType.MySql,
                    IsAutoCloseConnection = true
                }
            );

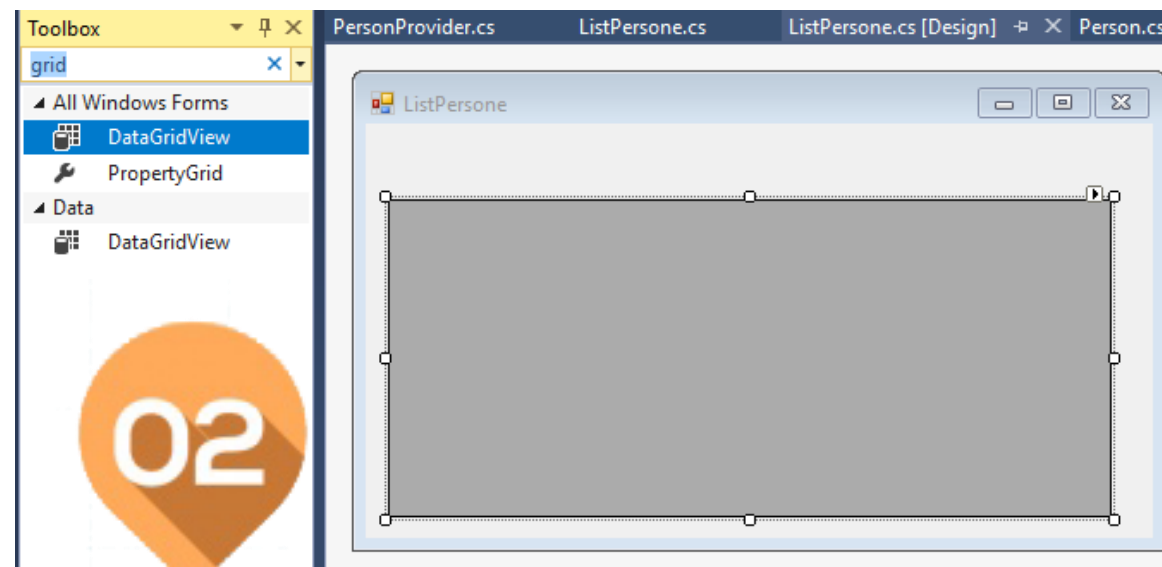
            return db;
        }
    }
}
```

```
using DemoApplication.Base;
using DemoApplication.Entity;
using System;

namespace DemoApplication.Provider
{
    0 references
    public class PersonProvider : DotNetTNBase
    {
        private DotNetTN.Connector.DotNetTNConnector DotNetTNConnector = new DotNetTN.Connector.DotNetTNConnector(GetInstance());

        0 references
        public void Add(Person insertObj)
        {
            try
            {
                DotNetTNConnector.Insert(insertObj);
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }
    }
}
```

```
0 references
public List<Person> GetLists()
{
    try
    {
        return DotNetTNConnector.GetList<Person>();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

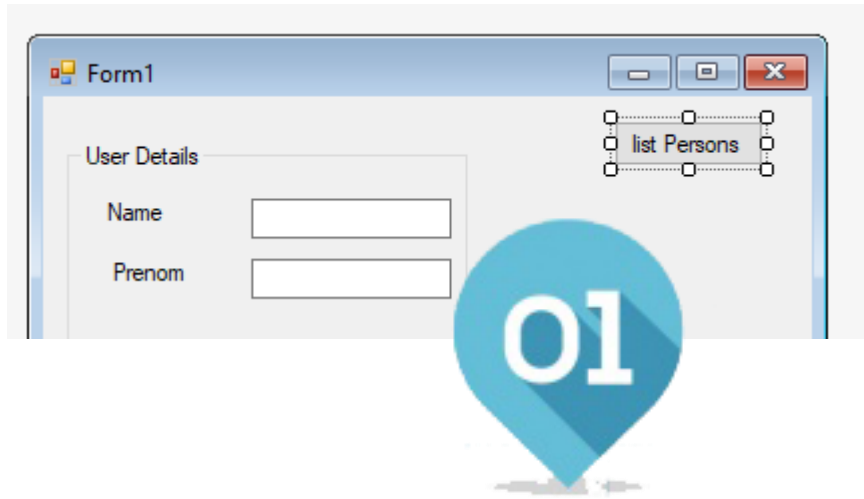


```
using System;
using System.Windows.Forms;

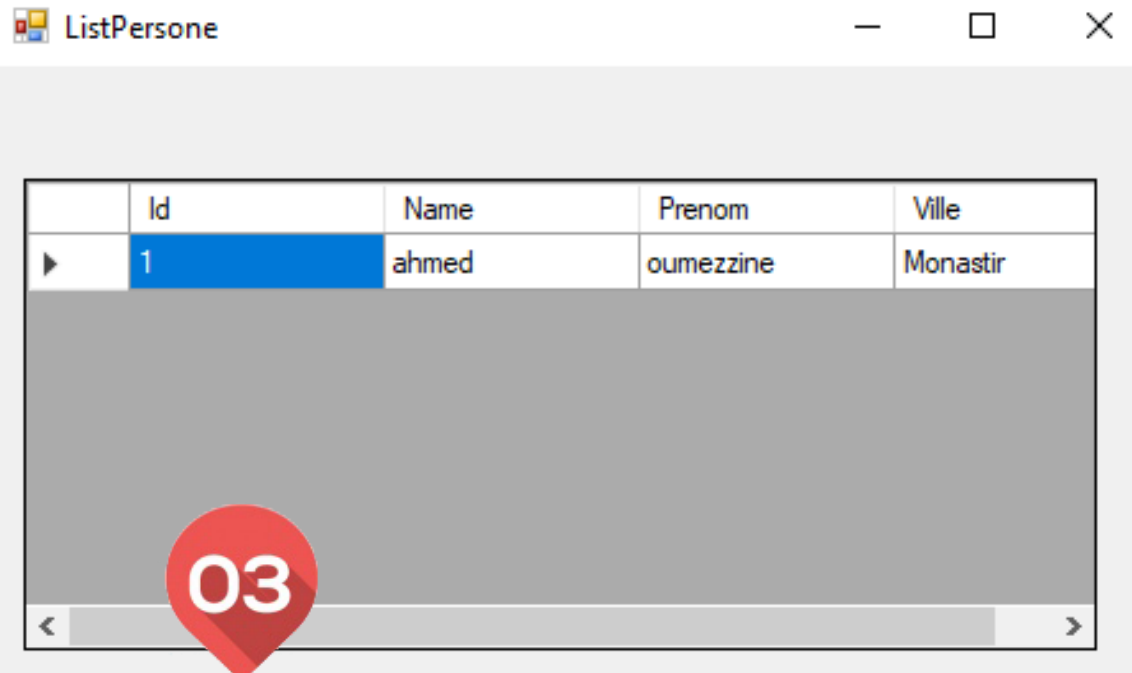
namespace DemoApplication
{
    2 references
    public partial class ListPersone : Form
    {
        0 references
        public ListPersone()
        {
            InitializeComponent();
        }

        1 reference
        private void ListPersone_Load(object sender, EventArgs e)
        {
            DemoApplication.Provider.PersonProvider provider = new Provider.PersonProvider();
            dataGridView1.DataSource = provider.GetLists();
        }
    }
}
```





```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    ListPersone list = new ListPersone();
    list.Show();
}
```

	Id	Name	Prenom	Ville
▶	1	ahmed	oumezzine	Monastir

Microsoft Virtual Academy

www.microsoftvirtualacademy.com

MVA

Microsoft Virtual Academy



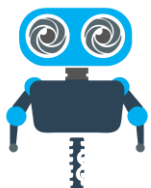
Channel 9

Channel 9

<https://channel9.msdn.com>

Notion De base C#

<http://csharp.net-tutorials.com/basics/introduction/>





THANK YOU

.....

Microsoft[®]

Your potential. Our passion.[™]

