Lesson 2 form property & lesson 3 showing messagebox & lesson 4 variables

Toolbox 里面有具体的项目，property 可以改颜色字体

在 property 上 有个 lightening button, 是可以改变它的 event，

从 messagebox 打印出东西来：

```
int a = 10;
MessageBox.Show(a.ToString());
```

Object 可以 fold anything, object myObj = true;

Lesson 5 change property:

button2.Text = "hover"; 用 button 的 design name

button1.Enabled = false; 会把 button disable 当点击后

button1.height = 60; 改变 button 的高度

textBox1.Text = “Adam” 通过 button 改变 textbox 的内容

textBox1.MaxLength = 2; 改变 textBox 的 string 的长度

lesson 6 If statement

lesson 7 more if statement

检查 checkbox 是否被 check

```
if (checkBox1.Checked == true) {
     MessageBox.Show("THe checkBox is checked");
 }
```

Lesson 9 switch statement

```
Switch(textBox1.text)
{
       Case “Adam”
        MessageBox.show(“Hello”);
        break;
       Case “bob”
        MessageBox.show(“yo”);
        break;
```

```
        Default
         MessageBox.show(“yoo”)’
         break;
}
```

Lesson 10 mathmatic operation

```
Int a = 5;
Int b = 3;
MessageBox.show((a-b).ToString());
A++; a--;
```

Lesson 11 array

```
String [] Names = {“Adam”,”Bob”,”Joe”};
String [] Names = new string [3];
```

Lesson 12 List

Do not need to speciliaze how many elements inside the list, 但是 array 中 需要说明多少个 element

```
List<object> Names = new List<object>();
Names.Add(1);
```

Lesson 13 for and foreach loop

```
 Names.Add("admas");
 Names.Add("yii");
 foreach (string s in Names)
 {
    MessageBox.Show(s);
 }
```

Lesson 14 while and do while loop

Lesson 15 Try, Catch and Finally

```
        try
            {
                string[] names = new string[2];
                string s = names[2];
            }
            catch {
                MessageBox.Show("There was a error");
            }
```

Finally: no matter we got error in try catch or not, we will

```
try
            {
                string[] names = new string[2];
                string s = names[2];
            }
            catch (Exception ex)
            {
                MessageBox.Show("There was a error " + ex.Message);
            }
            finally {
                MessageBox.Show("your code is done");
            }
```

Lesson 16, 17 Methods

Method is any blocks of codes that you want to use more than once

比如 function

```
        void Message(string message, string title) {
            MessageBox.Show(message,title);
        }
```

```
Messagebox.Show(Message(“adams”));
void Message(string name) {
      return name;

        }
```

Lesson 18 continue and break

Lesson 19 Namespace and classes
Namespace is used for organize class, struct and interface
Can create a namespace inside a namespace

```
Namespace myNamespace{
        Namespace space1{
        }
}
```

using Mynamespace; 加上 别的 cs file 里面当你想用这个 namespace

form 自带的 namespace , using System.Windows.Forms;

windows 是在 system 里面的 namespace, Forms 是在 system 里面的 namespace

```
System.Windows.Forms.MessageBox.show()
```

Lesson 20 Constructors

```
namespace Mynamespace
{
   class Class1
```

```
    {
        string Name;
        public Class1(string name){
            Name = name;
        }
    }
}
```

Constructor can has multiple constructors

Lesson 21 Access Modifier and static

```
        private string name() {
            return Name;
        }
```

如果不加 private 或者 public default 是 public

Static methods is to access without create the class

```
        public static void showMessage(string message) {
            System.Windows.Forms.MessageBox.Show("static method");
        }
```

Static is called modifier and public/private is called as access modifier

Lesson 22 Overloading Methods and Enumerations

```
        public static void showMessage(string message) {
            System.Windows.Forms.MessageBox.Show("static method");
        }
        public static void showMessage(int message) {
            System.Windows.Forms.MessageBox.Show(message.ToString());
        }
```

根据 parameter 的不同，overload function

```
        enum Names1
        {
            Adam = 1,
            Joe = 4,//similar to array, adam is 0, Joe = 0, 可以改变他们的值，next element
总比前一个大一
            Bob,
        }


Names1 myName = Names1.Adam; // My names hold adam which is default 0
```

可以改变 enum 的数值 成为 bytes

```
        enum Names1 : byte
        {
            Adam,
            Joe,
            Bob,
        }
```

```
Lesson 23 create your own properties
       public string Name
       {
           get; // let user to read
           set; // let user to change property
       }

namespace Mynamespace
{
   class Class1
   {

       public Class1(string name){
           Name = name;  // 把名字给到 property
       }
       public string Name  // 不用括号
       {
           get; // let user to read
           set; // let user to change property
       }
   }
}

Class1 mc = new Class1("adams");
MessageBox.Show(mc.Name); // 不用括号

public string Name
       {
           get; // let user to read
           private set; // let user to change property 只能在 class 内部修改，不能外部修改，
       }



Read only property
string mystring;
public Class1(string name){
      mystring = name;
}
public string Name
{
      get { return mystring; } // let user to read
           //private set; // let user to change property
 }


set {
          if (value == "") System.Windows.Forms.MessageBox.Show("You can't do that ");
 }
```

Lesson 24 Throwing Exceptions

```
if (name == "") throw new Exception("You can't set string equal to an empty string.");

       Exception exp = new Exception("You can't set string equal to an empty string.")
       public checkstring(string name)
       {
           if (name == "") throw new Exception(exp)        }
```

之后就可以在 main 加上打印 exception

```
Try{
       Myclass.checkstring(“”)
}
Exception (exception ex){
       MessageBox.show(ex.Message)
}
```

Lesson 25 Inheritance and overriding

```
   class Class1
    {
    }
   class MysecondClass : Class1 {
```

} 继承的可以所有 parent class function

假如用 base class 的内容(non private member(pubkic))

用 base.age

Class 也可以用 protected， protected int Age = 10;

Protected 的话，derived class 可以用 parent 的 protected 的，但是 outside class 不能用

Override function 用 new