II Cơ bản về lập trình C# lập trình trong trang ASP.NET

1. Kiểu dữ liệu.

C# đưa ra các kiểu dữ liệu dựng sẵn rất tiện ích, phù hợp với một ngôn ngữ lập trình hiện đại. Bảng sau đây sẽ miêu tả một số kiểu dữ liệu chính trong C#

Kiểu C#	Kiểu .Net	Số Byte	Mô tả
byte	Byte	1	số nguyên không dấu từ 0 đến 255
char	Char	2	Kiểu ký tự Unicode
bool	Boolean	1	Giá trị true/false
sbyte	Sbyte	1	Số nguyên có dấu, từ -128 đến 127
short	Int16	2	Số nguyên có dấu từ -32768 đến 32767
ushort	Int16	2	Số nguyên không dấu từ 0 đến 65.535
int	Int32	4	Số nguyên có dấu –2.147.483.647 đến 2.147.483.647
uint	Int32	4	Số nguyên không dâu 0 đến 4.294.967.295
float	Single	4	kiểu dấu chấm động, giá trị xấp xỉ từ 3,4E-38 đến 3,4E+38, với 7 chữ số có nghĩa.
Double	Double	8	Kiểu dấu chấm động có độ chính xác gấp đôi, giá trị xấp xỉ từ 1,7E-308 đến 1,7E+308, với 15,16 chữ số có nghĩa
Decimal	Decimal	8	Có độ chính xác đến 28 con số và giá trị thập phân, được dùng trong tính toán tài chính, kiểu này đòi hỏi phải có hậu tố m hoặc M kèm theo sau.

2. khai báo biến

Cú pháp: Kiểu Tên_biến;

Ví dụ:

```
string giatri_chuoi;
int giatri_nguyen;
```

chú ý biến có thể bao gồm các chữ cái, chữ số(không được đứng đầu) và ký tự _ (nối)

biến trong C# phân biệt chữ hoa và chữ thường.

3. Sử dụng các trình bày

a. trình bày if – if else

Khi bạn cần kiểm tra một điều kiện nào đó trước khi thực hiện công việc, hoặc kiểm tra điều kiện nếu đúng thì làm việc còn khác thì không làm bạn có thể dùng trình bày if – if else

cú pháp:

```
if(điều_kiên)
{
      //thực hiện công việc
}

if(điều_kiên)
{
      // thực hiện công việc 1
}
else
{
      //thực hiện công việc 2
}
```

Lưu ý bạn có thể dùng nhiều cặp if – else lồng nhau:

Ví dụ:

Vd1

```
      if (conn.State != ConnectionState.Open)
          conn.Open();
```

Vd2

```
      if (1 > 2)
          MessageBox.Show("1>2");
      else
          MessageBox.Show("2>1");
```

b, Sử dụng trình bày switch case

Khi công việc có nhiều lựa chọn và tuỳ vào từng trường hợp để bạn đưa ra công việc phù hợp với điều kiện đưa vào bạn có thể dùng trình bày switch case.

Ví dụ:

```
      string giatri = Request.QueryString["abc"];
      switch (giatri)
      {
          case "a":
              //thuc hien cong viec a
              break;
          case "b":
              //thuc hien cong viec b
```

```
                break;
            default:
                //thuc hien cong viec mac dinh
                break;
        }
```

c, Sử dụng trình bày for

Ví dụ

```
string giatri;
    for (int i = 0; i < 10; i++)
        giatri += i.ToString();
    MessageBox.Show(giatri);
```

Khi làm việc với mảng hay trong trường hợp thực hiện một công việc trong khoảng nào đó chúng ta có thể dùng trình bày for.

d, Sử dụng trình bày while

thực hiện công việc trong khi điều kiện đúng

Ví dụ

```
    int i = 0;

while (i < 5)

{

   Console.WriteLine(i.ToString());

   i++;

}
```

e, Sử dụng trình bày do while

ngược lại với while – do while làm việc cho đến khi điều kiện đúng thì thoát.

Ví dụ

```
int i = 0;
do
{
    MessageBox.Show(i.ToString());
    i++;
} while (i < 3);
```

f, Sử dụng trình bày break (*để thoát khỏi vòng lặp*)

Ví dụ

```
int i = 0;
do
{
    MessageBox.Show(i.ToString());
```

```
            i++;
            if (i == 1)
                break;
        } while (i < 3);
```

g, Sử dụng trình bày continue.

Ví dụ

```
        int j = 0;
        for ( int i = 0; i < 5; i++ )
        {
            j++;
            if ( j > 2 )
            {
                MessageBox.Show(j.ToString());
                continue;
            }
        }
```

h, Sử dụng trình bày return(*được sử dụng trong các hàm để trả về giá trị cụ thể cho hàm*)

Ví dụ

```
    public int sum(int a, int b)
    {
        return a + b;
    }
```

k, Sử dụng trình bày goto.

Ví dụ

```
        int i = 0;
        int j = 0;
        while (i < 5)
        {
            i++;
            j++;
            if (j == 2)
                goto jumpeddoutofloop;
        }
    jumpeddoutofloop:
        Console.WriteLine("I jumped out");
```

4. Trang asp.net

Trang asp.net có đuôi mở rộng là .aspx và kèm theo một lớp phục vụ ẩn đằng sau(Code behind).

Để viết code C# trong trang aspnet ta có thể khai báo và sử dụng trực tiếp trong trang asp.net, trong file code behind, hoặc từ một thành phần thư viện và ta gọi vào.

4.1 Viết code C# trong file .aspx:

về cơ bản bạn dùng các các thẻ sau

- <% %> bạn có thể khai báo biến hoặc viết các hàm, lớp trong thể này,
- <%= %> với thẻ này bạn dùng để gọi giá trị của biến hay của 1 hàm nào đó,
- <%# %> lấy giá trị dùng trang các đối tượng ràng buộc dữ liệu.

Đây là một ví dụ đơn giản

Trang basic.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"  CodeFile="Basic.aspx.cs"
Inherits="_Default" %>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">


<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
   <title>Basic</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <%
         string abc = "Hello World!";
      %>
      Biến abc của bạn vừa khai báo có giá trị <%=abc %>
   </div>
```

```
    </form>

</body>

</html>
```

4.2 Viết code trong trang code behind

Vì trang aspnet của chúng ta kế thừa từ trang aspx.cs lên trong trang .aspx chúng ta muốn gọi dữ liệu từ biến hay hàm trong file .aspx.cs chúng ta phải khai báo với bổ ngữ truy cập protected hoặc public.

Ví dụ sau:

Trang codebehind.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="codebehind.aspx.cs"
Inherits="codebehind" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <h3>Gán giá trị:</h3>
        <asp:Label ID="lblhello" runat="server" Text="Label"></asp:Label><br
/><br />
        <h3>Lấy giá trị từ code behind</h3>
        <%=_hello %>
    </div>
    </form>
</body>
</html>
```

Trang codebehind.aspx.cs

```
using System;

public partial class codebehind : System.Web.UI.Page
{
    protected string _hello;
    protected void Page_Load(object sender, EventArgs e)
    {
        _hello = "Hello World";
        lblhello.Text = _hello;
    }
}
```

Trong ví dụ trên có sử dụng một điều khiển asp.net là Label các bạn sẽ được học trong chương sau, ở chương này bạn hiểu nó là một điều khiển để hiển thị dữ liệu.

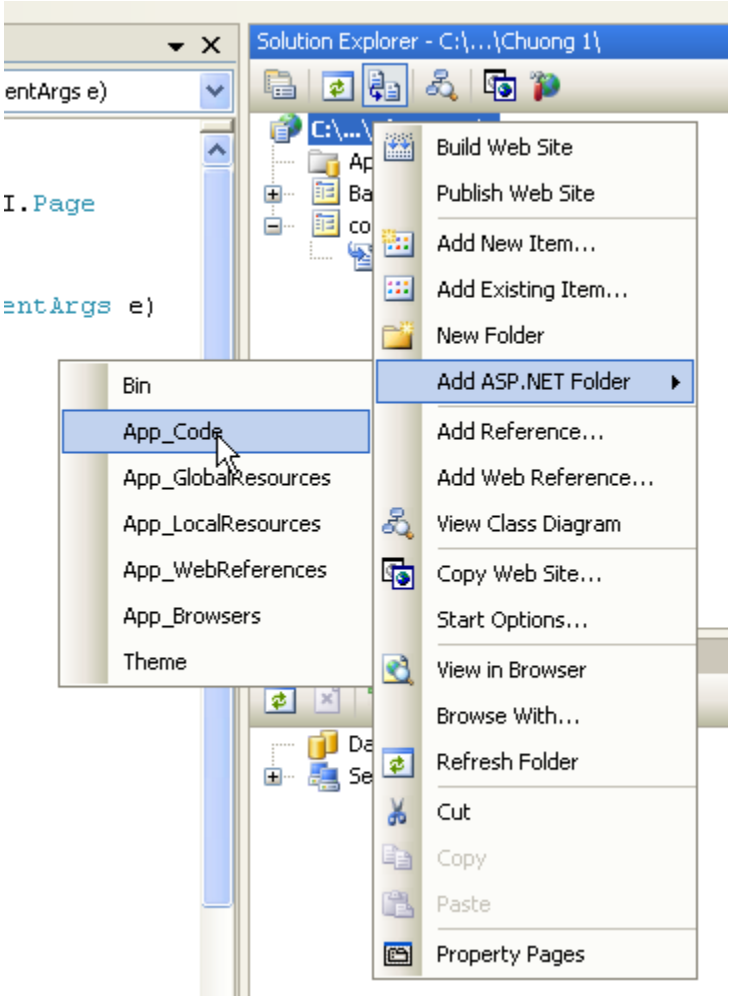
Bạn thấy trong phần code behind có khai báo một biến _hello kiểu string và bổ ngữ truy cập là protected trong sự kiện Page_Load(khi trang được tải lên) chúng ta gán _hello = "Hello World"; và sau đó gán giá trị cho Label bằng giá trị của _hello. Còn trong trang .aspx chúng ta có sử dùng thẻ <%= %> để lấy giá trị của _hello để in ra màn hình.

4.3 tạo một lớp thư viện

Để tạo một lớp thư viện phục vụ cho trang asp.net bạn có thể tạo một thành phần thư viện động DLL rồi nhập tham chiếu đến nó để sử dụng(chúng ta sẽ học nó trong phần asp.net nâng cao). Trong ứng dụng web ASP.NET Framework có một ASP.NET FOLDER là App_Code cho phép chúng ta viết các lớp thư viện ở đây và có thể sử dụng trong các trang của ứng dụng web.

để tạo thư mục App_code bạn làm theo các bước sau đây:

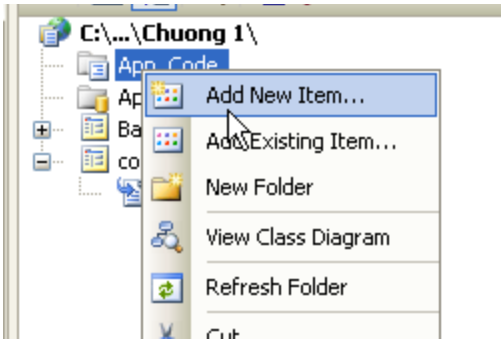
bước 1: nhấn chuột phải vào Solution và chọn theo đường dẫn của ảnh dưới đây.



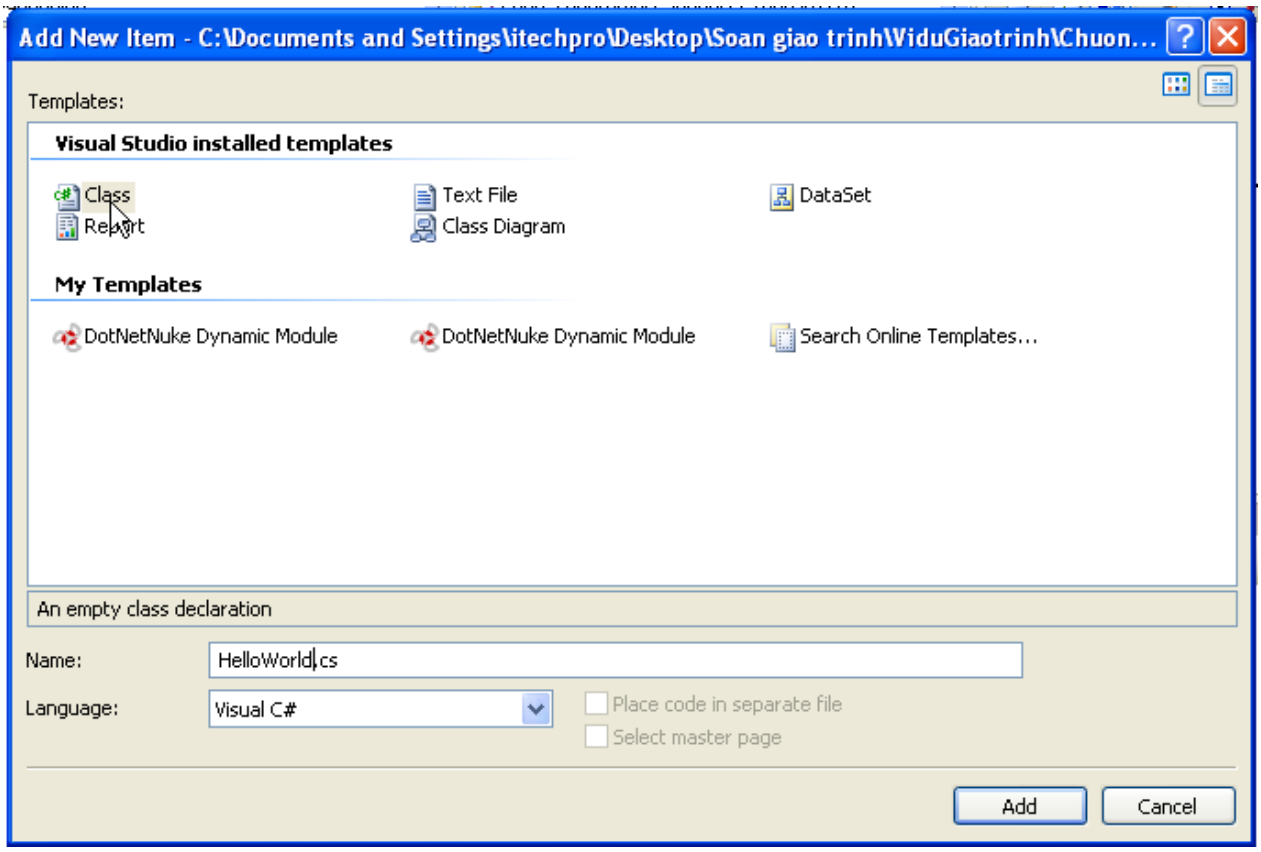
Trong ứng dụng web của chúng ta sẽ thêm vào một thư mục App_code

tại đây chúng ta có thể viết vào các lớp thư viện.

Để tạo một lớp thư viện trong thư mục này chúng ta nhấn chuột phải vào thư mục App_code rồi chọn Add New Item



Form Add New Item hiện ra



Bạn chọn Class và trong hộp TextBox Nam bạn nhập tên lớp muôn tạo và nhân nút Add.

a, Định nghĩa lớp:

Khai báo:

```
[Thuộc tính] [bổ sung truy cập] Class [Tên lớp] : [Lớp cơ sở]
{
  //các biến, phương thức hay thuộc tính của lớp
}
```

Ví dụ: Lớp HelloWorld.cs

```
class HelloWorld
  {
      public string SayMessage()
      {
          return "Hello World";
      }
  }
```

Trong ví dụ trên phương thức SayMessage sẻ về chuỗi "Hello World".

b, Sử dụng định nghĩa truy cập

Public: một lớp, một phương thức, hay thuôc tính khi sử dụng từ khoá này sẽ không bị hạn chế truy cập

Protected: Lớp, Phương thức, Thuộc tính chỉ được sử dụng ở lớp này hoặc lớp được dẫn xuất.

Internal: Một lớp, phương thức, thuộc tính Internal chỉ được truy cập trong một thành phần Assembly(file DLL).

Private: Một lớp Private, phương thức hoặc thuộc tính chỉ có thể truy cập tại chính lớp đó.

c, Hàm và thủ tục

Bạn có thể hiểu đơn giản hàm phải có giá trị trả về còn thủ tục như một đoạn mã chỉ thực hiện khi được chúng ta gọi. thủ tục còn được gọi là hàm không kiểu, hàm và thủ tục trong C# gọi chung là phương thức.

Ví dụ hàm:

```
public static int Sum(int _a, int _b)
 {
     return _a + _b;
 }
```

Trên là một hàm dùng để tính tổng của hai số, như bạn thấy trả về dữ liệu cho hàm chúng ta dùng từ khoá return, bổ sung truy cập public có ý nghĩa hàm được sử dụng trong toàn ứng dụng, từ khoá static đây là một phương thức tĩnh lên có thể sử dụng mà không cần phải khai báo khởi tạo đối tượng

Ví dụ về thủ tục

```
public static void HelloProcedure(string _bien)
{
    System.Web.HttpContext.Current.Response.Write(_bien);
}
```

Sử dụng lớp HellWorld trong trang aspx của chúng ta

Trang UseHelloworld.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="UseHelloworld.aspx.cs" Inherits="UseHelloworld" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Sử dụng Lớp Hello World trong thư mục App_Code</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Label ID="lblHello" runat="server" Text="Label"></asp:Label>
    </div>
    </form>
```

```
</body>
</html>
```

Trang UseHelloworld.aspx.cs

```
using System;
public partial class UseHelloworld : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        lblHello.Text = HelloWorld.sayMessage();
    }
}
```

Vì phương thức sayMessage trong lớp HelloWorld là một phương thức tĩnh lên ta không cần khởi tạo lớp để sử dụng.