



# برمجه صفحات الانترنت ASP



من منا لم يسمع بـ ASP.NET بعد من مطوري  
المواقع؟! ، فالجيل القادم سيكون للغات  
البرمجة التي تزيل العبء عن كاهل المطورين  
وتختصر الوقت عليهم ، ولذلك سأعرض هنا  
بعض الأسئلة التي يسألها أغلب من يريد البدء  
بتعلم هذه اللغة ، وستكون بقدر المستطاع  
بسيطة وسهلة الفهم

# قصة حياة : ASP.NET

- القصة بكل اختصار، من بدايتها كانت هنالك ASP بدون دوت نت و هي عبارة عن في بي سكربت
- وكانت مليئة بالمشاكل وبطيئة وبها عيوب كثيرة لست هنا بصدد ذكرها.
- وبعد صدور الدوت نت قامت مايكروسوفت بتطوير ال ASP وجعلها ضمن إطارها وموجهة لتطوير المواقع فالإصدار الأول كان في ٢٠٠٢.
- ثم في ال ٢٠٠٣ نزلت ASP.NET 1.1.
- في ال ٢٠٠٥ نزلت ASP.NET 2.0.
- في ال ٢٠٠٨ نزلت ASP.NET 3.0 و ASP.NET 3.5
- وسيصدر في عام ٢٠١٠ ASP.NET 4.0

# ما هي الـ ASP.NET

١. ASP.NET هي لغة برمجية ( مبدئيا ) من شركة مايكروسوفت موجهة لتطوير المواقع.
٢. " إختصار الـ Active Server Pages والتي تعني صفحات الخادم النشط " هو إطار لتطبيقات الويب تم تطويره وتسويقه من خلال شركة مايكروسوفت، من أجل إعطاء القدرة للمبرمجين على بناء مواقع ويب ديناميكية، تطبيقات ويب وخدمات ويب.
٣. وتقنيّة بلغات (Visual Studio.NET) تقوم على بناء التطبيقات في الإنترنت سواء مواقع ثابتة أو ديناميكية ويمكن تفصيلها كما يلي:

## المواقع الثابتة: static website

وهي عبارة عن مواقع عادية على الإنترنت تكون محتوية على صور ونصوص أو بشكل أبسط هي مواقع الإنترنت التي تم كتابتها على بـ (HTML) بحيث تكون كل صفحة مفصلة عن باقي الصفحات ولا يوجد أي قواعد للبيانات.

## مواقع الديناميكية : Dynamic website

هي عبارة عن مواقع تسمح بتغيير أو حذف أو إضافة أي صور أو معلومات من الصفحات بكل سهولة من خلال مسؤول المواقع بدون العودة الى مصمم المواقع، حيث يتم إظهار وإدخال البيانات والمعلومات المتواجدة بصفحاته بواسطة قواعد بيانات ((databases، والتي تسمح بالتحديث والإضافه بشكل مستمر للصفحات ولذلك تم تسميتها بـ "المواقع الديناميكية".

## أولاً: مفهوم العميل (Client) والخادم (Server) :

- هو الجهاز الذي يقوم بالاستجابة ومعالجة الطلب الذي تلقاه من جهاز العميل وتزويده بالموارد التي طلبها.

- عملية الاستجابة تُعرف بـ  
(Response)

- هو الجهاز الذي يقوم بطلب مصادر أو موارد (Resources)  
مثل: صفحة ويب أو صورة أو ملف صوتي.

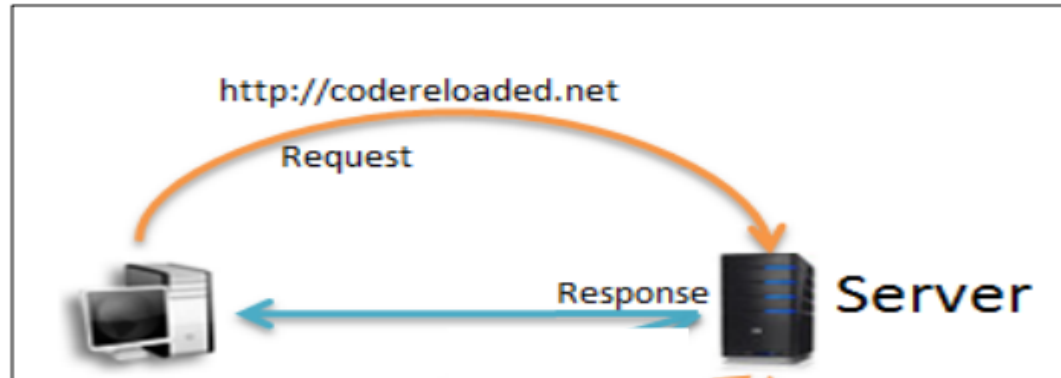
- عملية الطلب تُعرف بـ (Request)

- جهاز العميل بذاته لا يقوم بعملية Request.. إذن أي مكون يقوم بذلك؟

- فالعملية يقوم بها أحد البرامج وفي حالتنا المتصفح Browser

## كيف يجلب المتصفح الصفحات ؟

• في الحالة الافتراضية أن Request من جهاز و Response من جهاز آخر



• في حالة أن Request & Response قد حدثوا من نفس الجهاز فإن جهازك يعتبر Client & Server في آن واحد

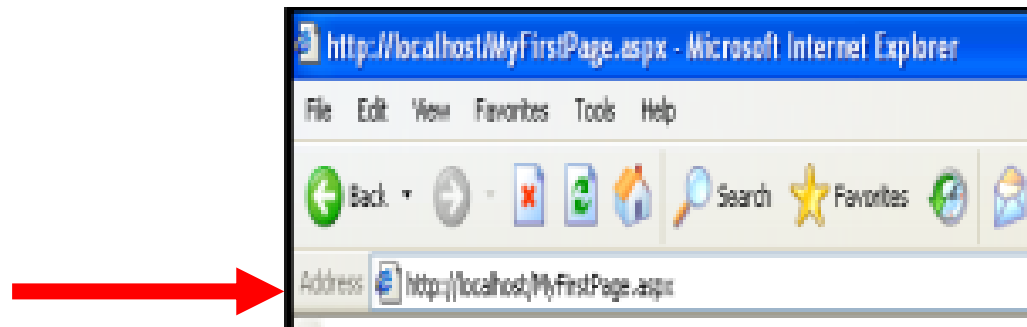
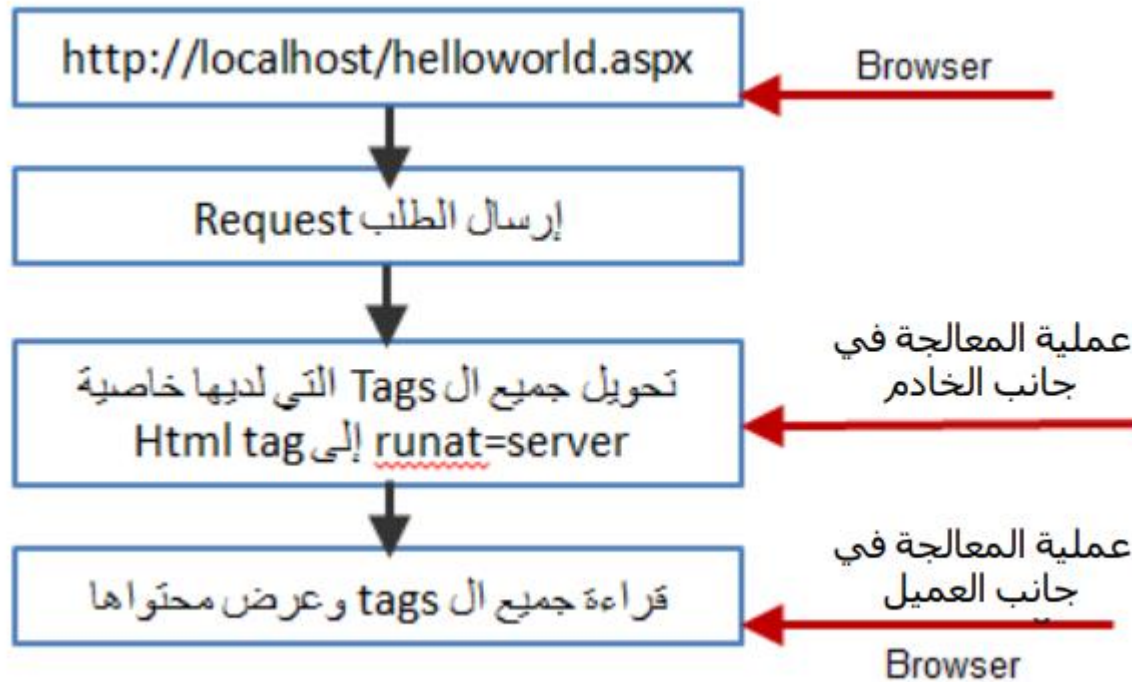




# الكود المصدري (source)

```
<html>
<head>
</head>
<body>
  <form runat="server">
    <!-- Insert content here -->
  </form>
</body>
</html>
```

# كيف يصبح عنوان الصفحة ..



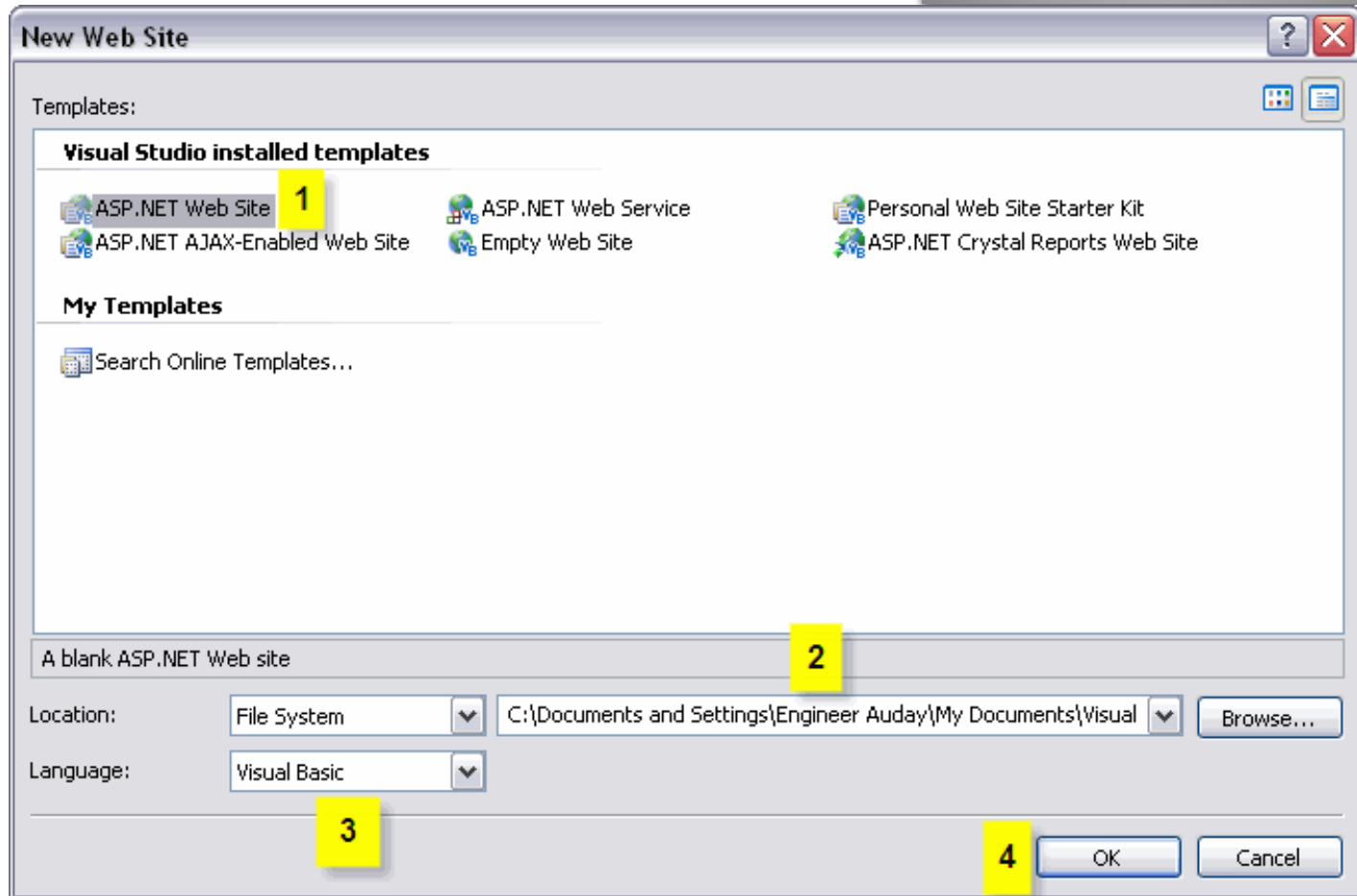
## !! الفرق بين ASP و ASP.net

١. الـ ( ASP ) يكون الكود والتصميم بها بالإضافة إلى Scripts مختلطاً مع بعضه البعض بنفس الصفحة.
٢. أمّا الـ ( ASP.net ) يكون الكود والتصميم بها منفصلان وذلك بواسطة الـ ( Code Behind )

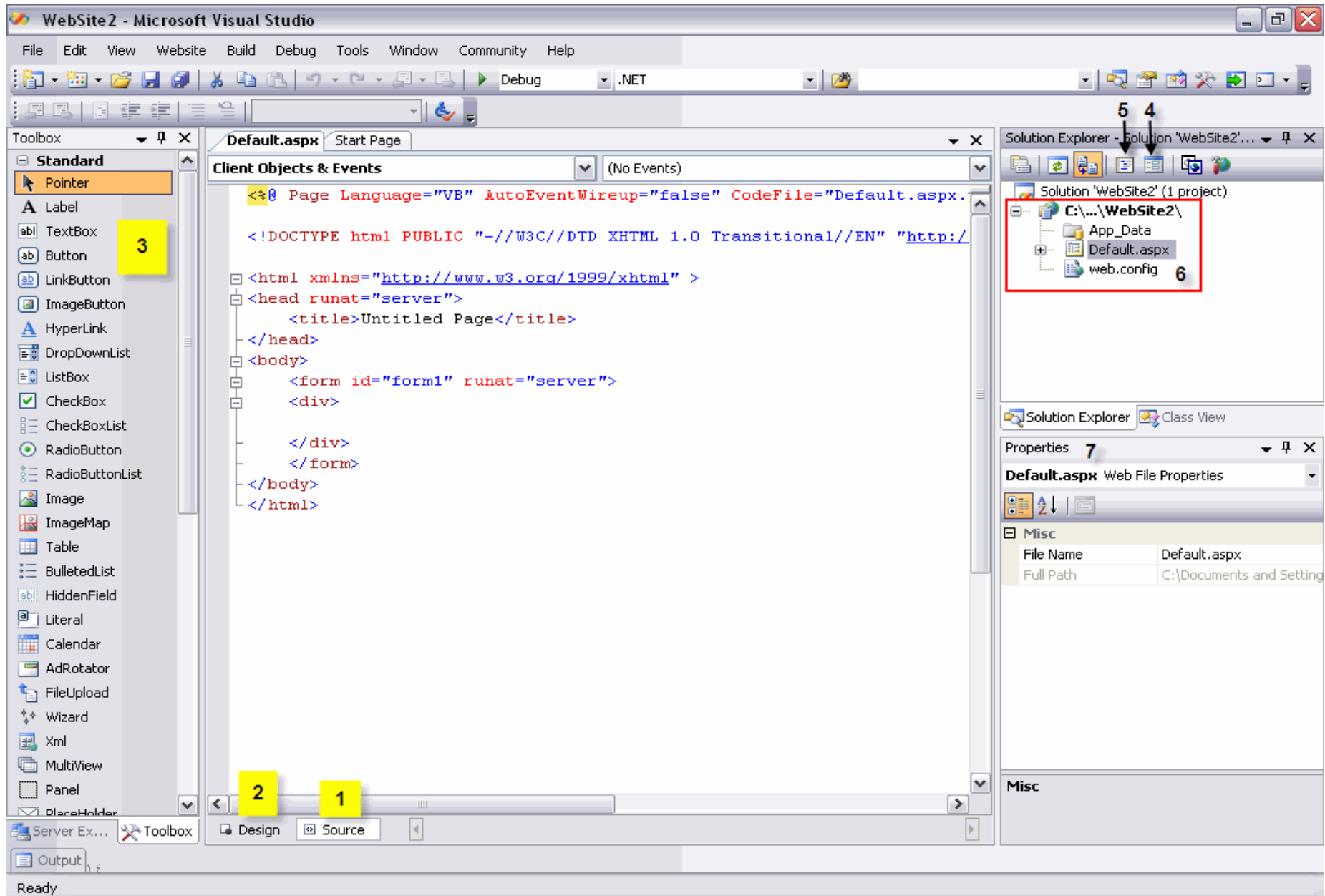
## ما الذي يميز ASP.NET عن غيرها!!!

١. تتميز بالسرعة بالمقارنة مع غيرها؛ وذلك لأنها تقوم بالترجمة وليس التفسير.
٢. الأمن حيث يوجد بها آليات تعمل على تخفيف الاختراق للمواقع التي تمت برمجتها من خلالها.
٣. مكتبة أكواد كبيرة وضخمة وهي مكتبة دوت نت. كثرة الأدوات الجاهزة.
٤. البيئة المتطورة المتكاملة وهي الفيچوال ستوديو.
٥. تعدد وتكامل اللغات، أي يمكننا البرمجة بالكثير من اللغات ومن أهمها:  
لغة (Visual Basic .Net) لغة (.Net) C# لغة ( .Net) #لغة

File → New → Web Site...



سيتم انشاء موقع جديد وتظهر لنا النافذة التالية..



## حيث تشير الارقام الى :

- ١ - نافذة الكود (HTML).
- ٣ - صندوق الادوات.
- ٦ - موقع الوب وملفاته.
- ٢ ، ٤ - نافذة التصميم.
- ٥ - نافذة الكود ( VB.Net ).
- ٧ - نافذة الخصائص.

داخل نافذة التصميم نستطيع اضافة الادوات (button, label, ...).



كود HTML الناتج عن اضافة الادوات:

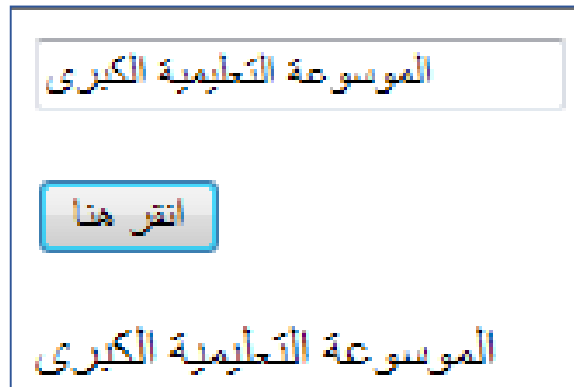
```
<asp: Label ID="Label1" runat="server" Text="Label"></asp: Label>  
<br />  
<asp: Button ID="Button1" runat="server" Text="Button" /></div>
```

id : يوضح الاسم البرمجي للأداة (المستخدم في الكود).  
runat : يوضح ان هذه الاداة هي اداة خادم وب (server)

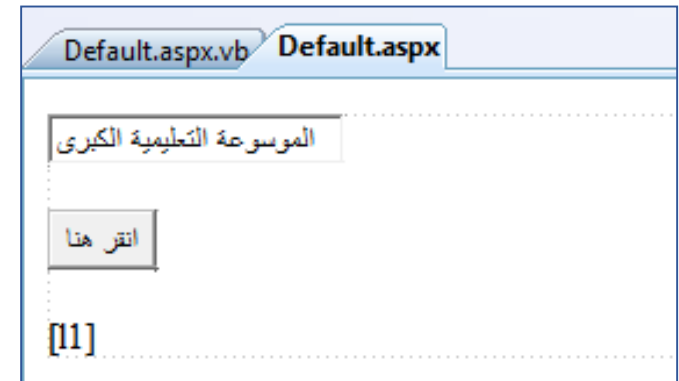
تغيير الخاصية text الى Hello ,this is my test من مربع الخصائص او من خلال الكود:

```
<asp:Label ID="Label1" runat="server" Text="Hello ,this is my test">
</asp:Label>
```

اضافة مربع نص ، زر ، عنوان.  
عند الضغط على الزر يطبع الرسالة الموجودة في مربع النص داخل العنوان.



l1.text=t1.text





```

<body>
  <form id="form1" runat="server">
    <div>

      <asp:TextBox ID="t1" runat="server">الموسوعة التعليمية الكبرى</asp:TextBox>
      <br />
      <br />
      <asp:Button ID="b1" runat="server" Text="انقر هنا" />
      <br />
      <br />
      <asp:Label ID="l1" runat="server"></asp:Label>

    </div>
  </form>
</body>

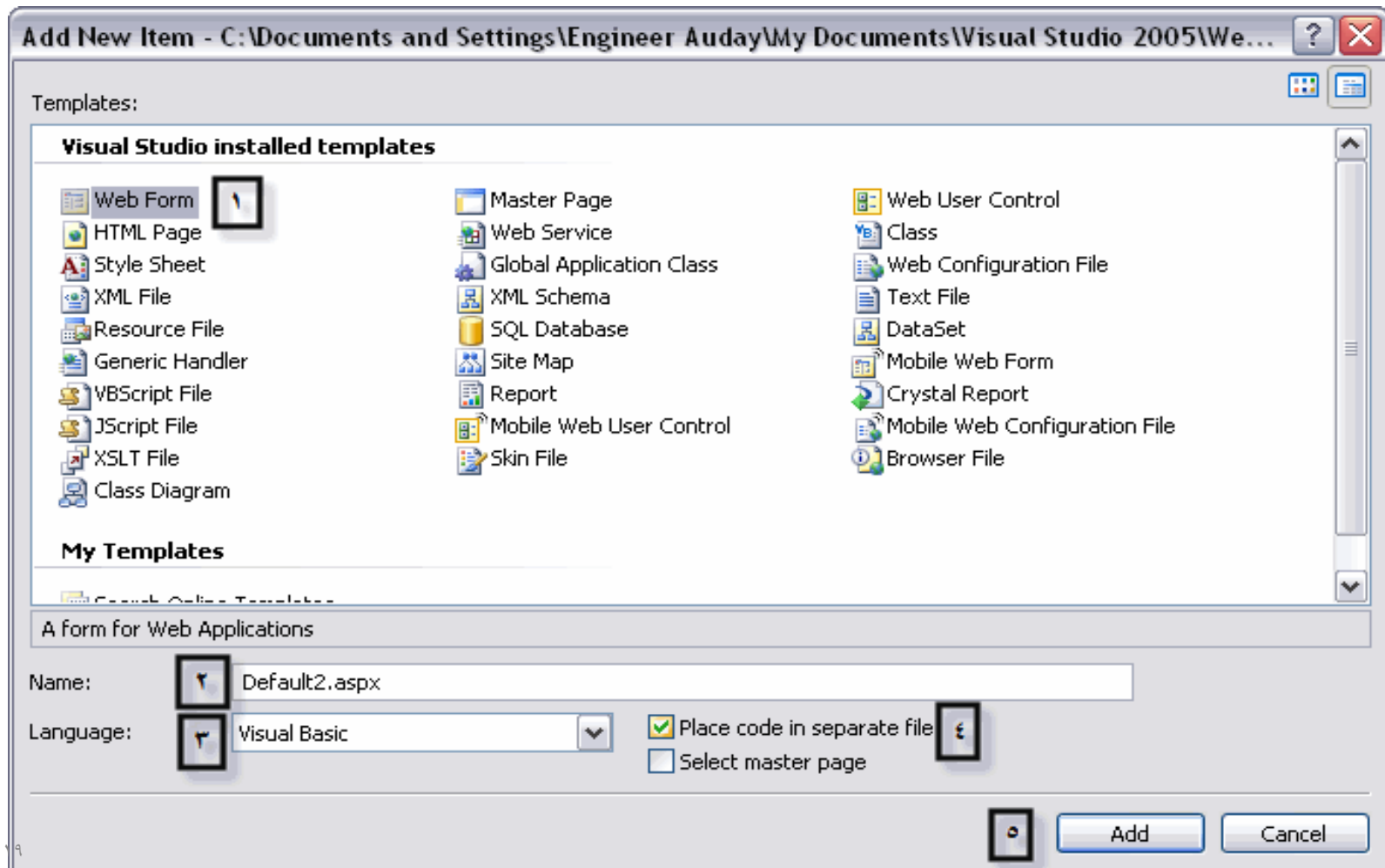
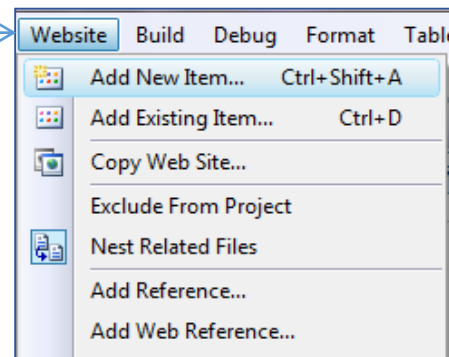
```

عند الضغط R-CLICK على صفحة التنفيذ سيتم تصدير الادوات على هيئة ادوات HTML.

File Edit Format

```
15 <div>
16
17     <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWAwKKwvXqDALM7+btDAK+7
+btDDGwUowYvPwdGmN5WSBGj5lBWBSj" />
18 </div>
19     <div>
20
21         <input name="t1" type="text" value="الموسوعة التعليمية الكبرى" id="t1" />
22         <br />
23         <br />
24         <input type="submit" name="b1" value="انقر هنا" id="b1" />
25         <br />
26         <br />
27         <span id="l1" style="display:inline-block;border-width:3px;border-style:Dotted;">الموسوعة التعليمية الكبرى</span>
28
29     </div>
30 </form>
31 </body>
32 </html>
```

إضافة صفحة جديدة للموقع



# استخدام الأدوات



ASP.NET

في هذا الدرس سوف نتعلم استخدام:

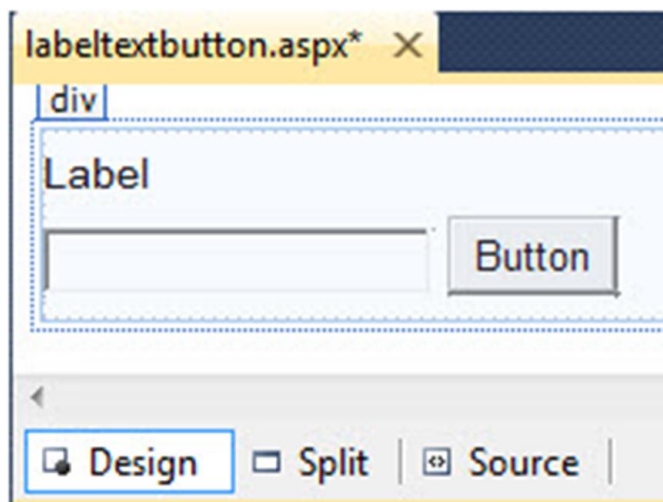
1- Label.

2- TextBox.

3- Button.

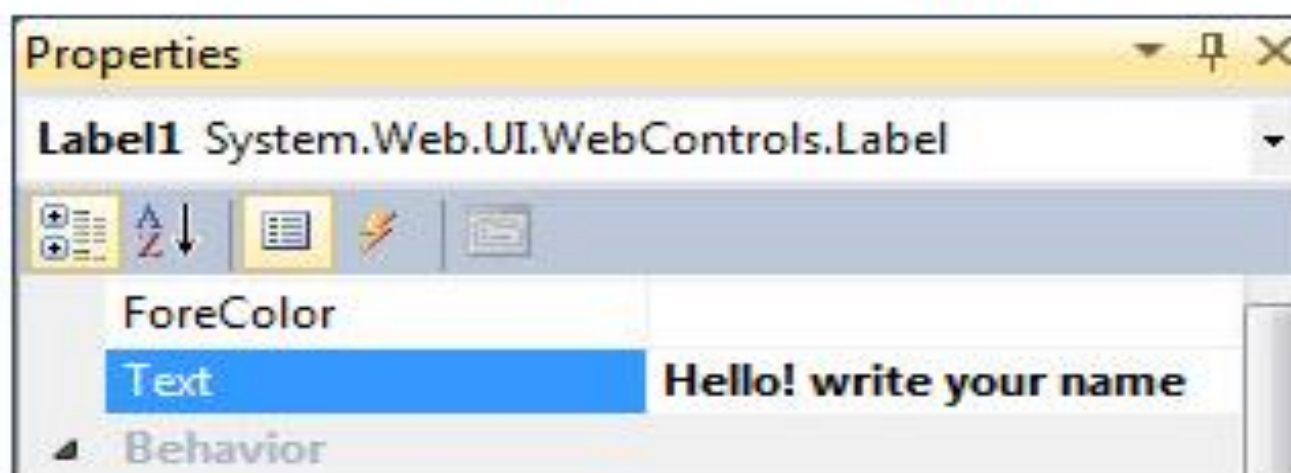
أولاً : إضافة الأدوات :

نقوم بإضافة الاداة Label ثم TextBox ثم Button كما نرى الشكل التالي



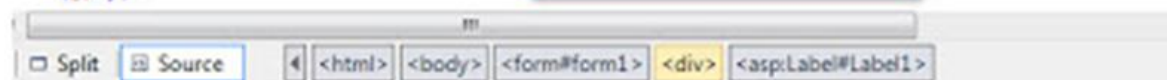
## ثانياً : نبدأ نتعرف على اهم خصائص الادوات المستخدمة :

ولنبدأ باداة Label واهم خصائصها هو ال Text ويمكن تغييرها من خلال نافذة الخصائص

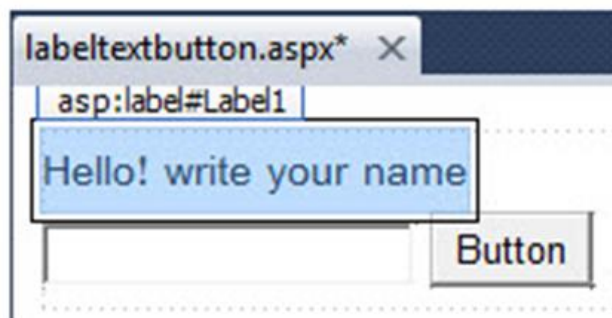


او من خلال نافذة ال Source

```
<asp:Label ID="Label1" runat="server" Text="Hello! write your name" </asp:Label>  
<br />
```



ليُغيّر النص المكتوب على ال Label بالشكل التالي



والآن الى الاداة TextBox و الاداة Button واهم الخصائص Text

Text	OK
Text	your name

ليصبح الشكل

Hello! write your name

your name	OK
-----------	----

وخاصية Enabled وهي المسئولة عن تفعيل الاداة ونلاحظ اذا تغيرت الى False

Enabled	False
---------	-------

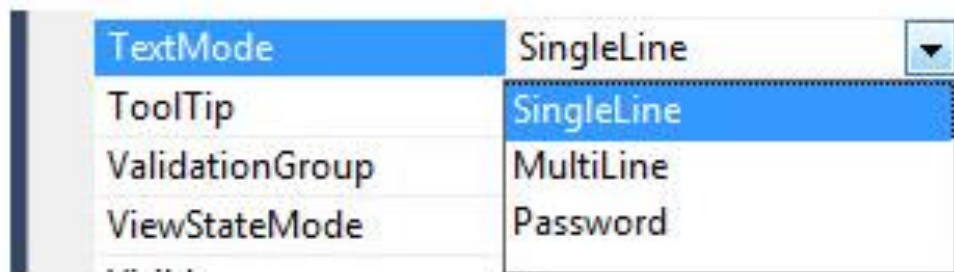
يصبح الشكل

Hello! write your name

your name	OK
-----------	----



وخاصية `TextMode` خاصة بال `TextBox` وهي المسئولة عن تغيير نمط مربع النص من سطر واحد او عدة سطور او باسورد



واذا تغير الى `MultiLine` يصبح الشكل

---

Hello! write your name

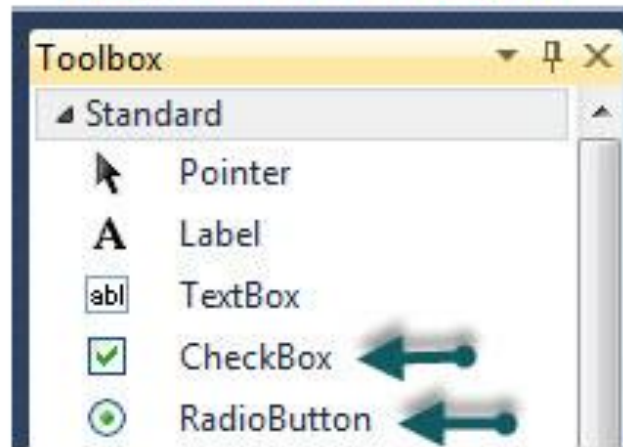
Button

وإذا تغير الى Password يصبح الشكل

Hello! write your name

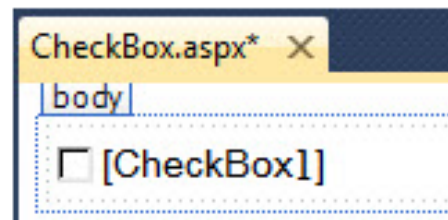
Button

### أدوات ASP.NET [3]



#### استخدام أداة CheckBox :

هي الأداة المربعة الشكل التي يوضع علامة داخلها عند اختيارها , تسمح هذه الاداة للمستخدم بالموافقة او عدم الموافقة على امرأ ما , تسمح لك مربعات الاختيار CheckBoxes بتحديد أكثر من أداة واحدة .



## خصائص أداة CheckBox :

**AutoPostBack**: إرسال خيار التحديد إلى السيرفر بشكل فوري عند أي تغيير بحالته تساوى **false** افتراضيا.

**Checked**: تصبح ب **false** أو **True** بحسب اختيار المستخدم.

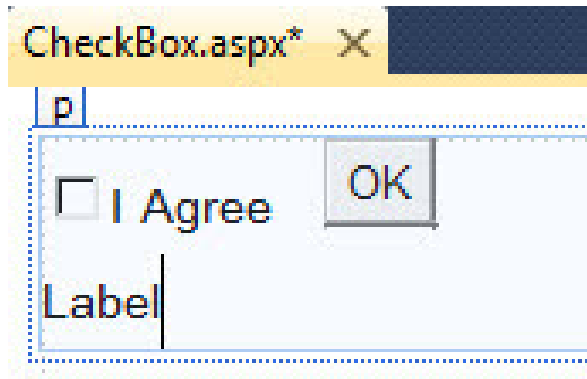
**Text**: تحديد النص الذي يظهر بجانب هذه الأداة.

**TextAlign**: موضع النص بالنسبة للأداة, على يمينها أم يسارها تأخذ إحدى القيمتين **Left, Right**.

## الأحداث في الأداة CheckBox:

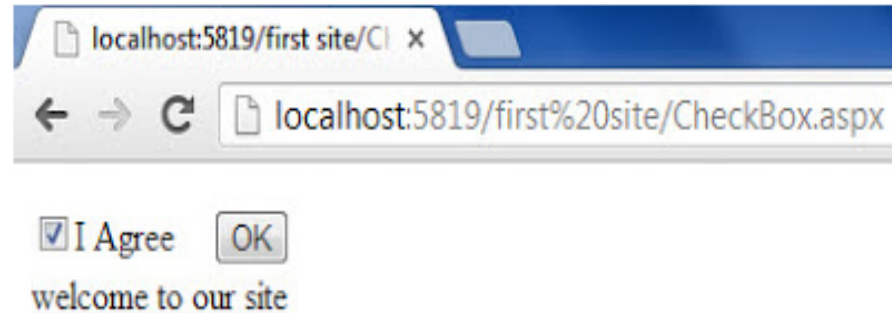
**CheckedChanged**: يقع عند تغيير الخاصية **Checked**

## مثال: باستخدام if الشرطيه ..



```
if (CheckBox1.Checked)
    Label2.Text = "welcome to our site";
else
    Label2.Text = "you must Agree";
```

ليصبح الشكل في صفحة ال asp.net

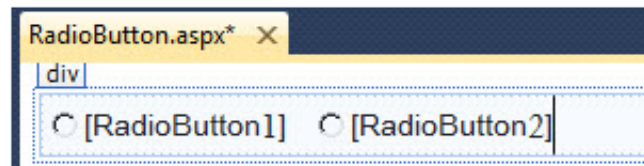


ملحوظة:

Asp.net تدعم الاداة CheckBoxList والتي تتيح إمكانية عمل قائمة عناصرها وايضا تسمح لك مربعات الاختيار CheckBoxList بتحديد أكثر من اختيار ..

## استخدام أداة RadioButton:

هي الأداة الدائرية التي يرسم في وسطها نقطة عند اختيارها , لا تظهر هذه الأداة لوحدها إنما تكون ضمن مجموعة حيث بإمكان المستخدم تحديد خيار واحد فقط من ضمن أدوات مجموعة ما .



## أهم الخصائص في الأداة RadioButton:

**GroupName :** لضم أداة RadioButton إلى مجموعة, من كل مجموعة لا يمكن الوجود فيها. تحديد سوى خيار واحد فقط من أدوات RadioButton.

**Checked:** تصبح ب `false` أو `True` بحسب اختيار المستخدم.

**Text:** تضبط و تعيد النص الظاهر بجوار زر راديو RadioButton.

**TextAlign:** تضبط وتعيد محاذاة النص الخاص بزر الراديو RadioButton.

## الأحداث في الأداة RadioButton:

**CheckedChanged:** يقع عند تغيير الخاصية `Checked`.

# مثال استخدام radiobutton



ونجعل كل ال RadioButton بنفس ال **GroupName**

GroupName	a
-----------	---

```
if (RadioButton1.Checked == true)
    Label1.Text = "your color : " + RadioButton1.Text;
else if (RadioButton2.Checked == true)
    Label1.Text = "your color : " + RadioButton2.Text;
else if (RadioButton3.Checked == true)
    Label1.Text = "your color : " + RadioButton3.Text;
else
    Label1.Text = "please! choose your color";
```

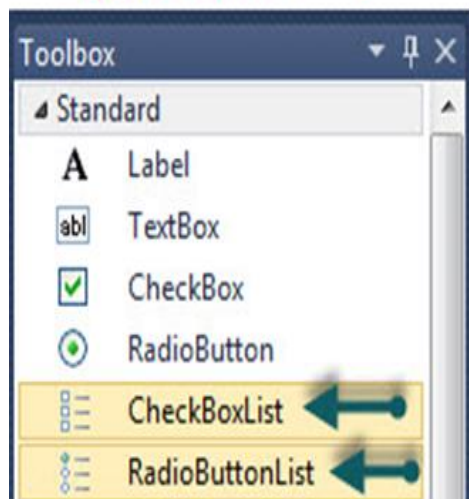
ليصبح الشكل في صفحة ال **asp.net**



choose your color :  
☒ Red  
☐ Blue  
☐ Black  
OK your color : Red

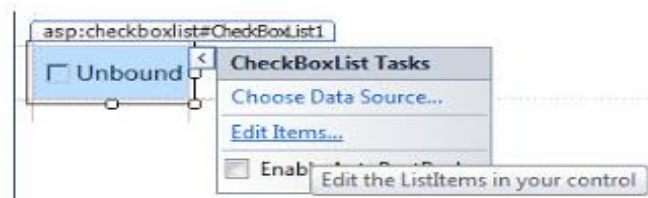


## أدوات ASP.NET [4]



### استخدم الأداة **CheckBoxList**:

هي قائمة من مربعات الاختيار **CheckBoxes** ويتم التعامل معها كمجموعة ويمكن تحديد أكثر من أداة واحدة (مربع) يلزم المرور على كل عناصر القائمة لمعرفة العناصر المحددة لأنها تسمح باختيار أكثر من عنصر.



### خصائص الأداة **CheckBoxList**:

**AutoPostBack**: إرسال خيار التحديد إلى السيرفر بشكل فوري عند أي تغيير بحالته تساوي **false** افتراضيا.

**SelectedItem**: تعيد العنصر المحدد.

**Items**: تعيد مجموعة العناصر الموجودة في **CheckBoxList**.

**SelectedIndex**: تحدد فهرس العنصر المحدد.

**DataSource**: تحدد مصدر البيانات (تستخدم عادة مع قواعد البيانات).

**DataMember**: تحدد مصدر بيانات (عادة جدول في قاعدة بيانات).

## الأحداث في الأداة `CheckBoxList`:

`SelectedIndexChanged`: يقع عند تغيير العنصر المختار في القائمة.

- وكل عنصر من ال `Items` داخل ال `CheckBoxList` له خصائص أهمها :

`Text`: تحديد النص الذي يظهر بجانب كل مربع.

`Value`: تحديد القيمة الموجودة في العنصر في القائمة `CheckBoxList`.

`Selected`: تصبح ب `false` أو `True` بحسب اختيار المستخدم.

`Enabled`: وهي المسئولة عن تفعيل الاداة.

`RepeatDirection`: و تحدد اتجاه عرض عناصر القائمة ، تأخذ هذه الخاصية إحدى قيمتين:

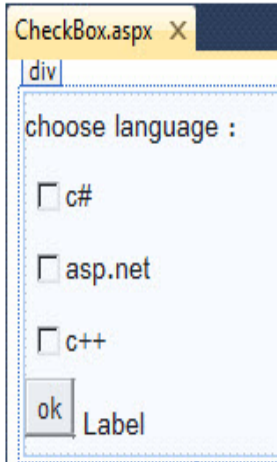
1. `RepeatDirection.Vertical` : تعني أن عناصر القائمة تظهر عمودياً ، و هي القيمة الافتراضية .
2. `RepeatDirection.Horizontal` : تعني أن عناصر القائمة تظهر أفقياً .

`RepeatLayout`: وتحدد طريقة عرض القائمة، تأخذ هذه الخاصية إحدى الخيارات التالية:

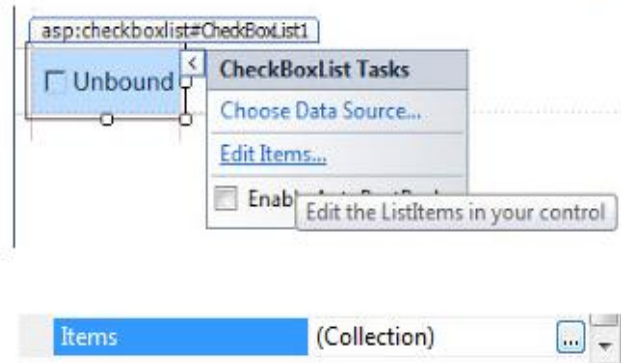
1. `RepeatLayout.Table` : تظهر القائمة على شكل جدول ، هي القيمة الافتراضية .
2. `RepeatLayout.Flow` : تظهر القائمة على بدون جدول .
3. `RepeatLayout.UnorderedList` : تظهر القائمة في شكل تعداد نقطي .
4. `RepeatLayout.OrderedList` : تظهر القائمة في شكل تعداد رقمي .

و من خلال النافذة المخصصة لإضافة عناصر جديدة نضيف مجموعة من العناصر ونعدل خاصية ال Text

**مثال : ننفذ الشكل التالي**

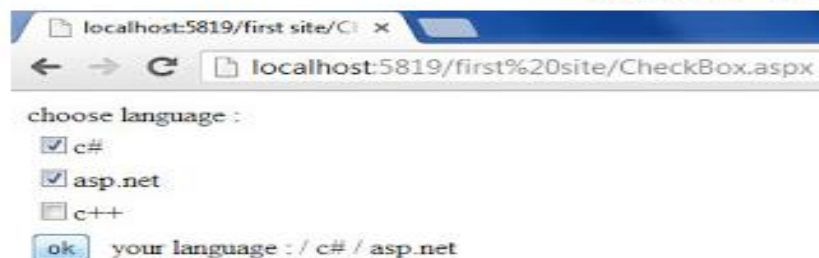


ويتم اضافة ال Items من خلال



او من قائمة الخصائص

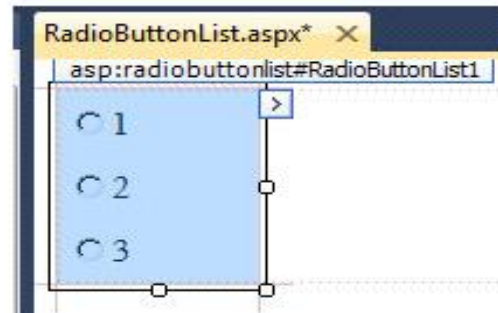
## ليصبح الشكل في صفحة .net asp



```
string a = "your language :";  
if (CheckBoxList1.Items[0].Selected)  
    a = a + " / " + CheckBoxList1.Items[0].Text;  
if (CheckBoxList1.Items[1].Selected)  
    a = a + " / " + CheckBoxList1.Items[1].Text;  
if (CheckBoxList1.Items[2].Selected)  
    a = a + " / " + CheckBoxList1.Items[2].Text;  
Label1.Text = a;
```

### استخدم الأداة RadioButtonList:

تستخدم الأداة RadioButtonList في إظهار قائمة من أزرار الراديو RadioButtons مع بعضها البعض و التعامل معها كمجموعة.



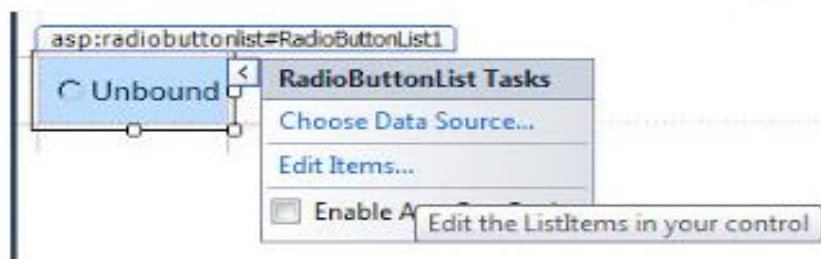
### خصائص الأداة RadioButtonList:

تملك نفس خصائص ال CheckBoxList. ولكن الاختلاف بينها وبين ال CheckBoxList في انه يمكن استخدام الخاصية selectedItem لمعرفة هل العنصر محدد أم لا ، و لمعرفة فهرس العنصر نستخدم الخاصية SelectedIndex ، كما انه لا يلزم المرور على كل عناصر القائمة لمعرفة العناصر المحددة كما كان في CheckBoxList لانه يلزم اختيار عنصر واحد فقط.

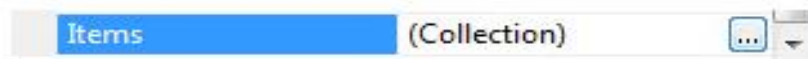
و من خلال النافذة المخصصة لإضافة عناصر جديدة نضيف مجموعة من العناصر ونعدل خاصية ال Text .



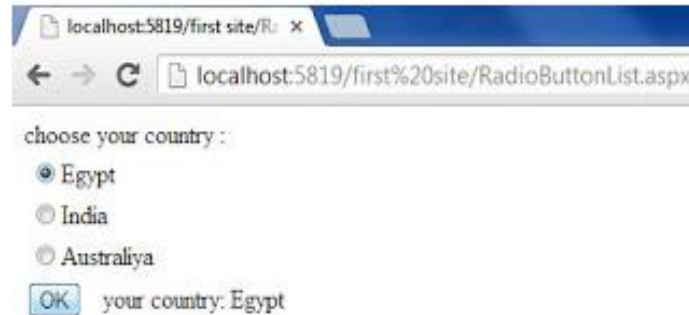
ويتم إضافة ال Items من خلال



او من قائمة الخصائص



ليصبح الشكل في صفحة .net asp



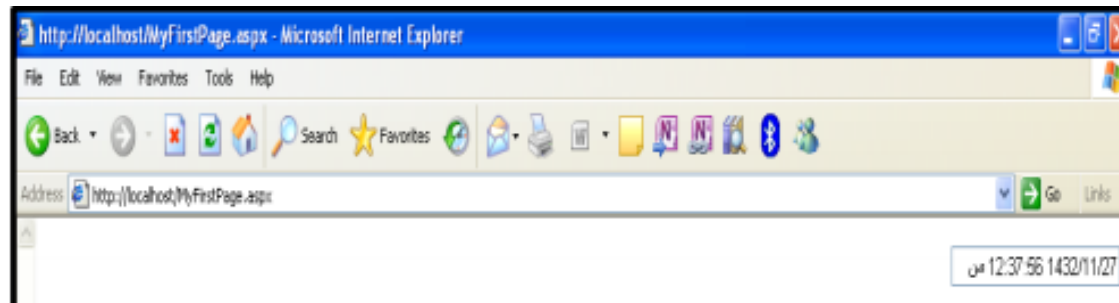
The screenshot shows a web browser window with the address bar displaying 'localhost:5819/first site/R...' and the full URL 'localhost:5819/first%20site/RadioButtonList.aspx'. The page content includes the text 'choose your country :', followed by three radio buttons labeled 'Egypt', 'India', and 'Australiya'. The 'Egypt' radio button is selected. Below the radio buttons is an 'OK' button and a text label 'your country: Egypt'.

```
Label1.Text = " your country: " + RadioButtonList1.Text;
```

## مثال / لعرض التاريخ والوقت على الصفحة

```
Partial Class MyFirstPage
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Text1.Value = DateTime.Now
    End Sub
End Class
```

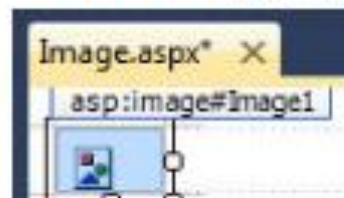






## استخدام الأداة Image:

تستخدم الأداة Image لعرض صور في صفحة ويب من نوع ASP.NET .



## خصائص هذه الأداة Image:

**AlternateText:** تضبط و تعيد النص الذي يظهر في العنصر Image في حال عدم ظهور الصورة ،و يظهر كتلميح في حال وجود الصورة خاصة مع المتصفحات الحديثة.

**ImageUrl:** تضبط و تعيد عنوان URL للصورة المراد ظهورها في الصفحة.

**Width:** تضبط و تعيد عرض الصورة.

**Height:** تضبط و تعيد ارتفاع الصورة.

**ImageAlign:** تضبط و تعيد محاذاة الصورة في الصفحة (تحديد موقع الصورة بالنسبة لأدوات HTML) و تأخذ عدة قيم

منها: Top, middle, Bottom, Left, Right.

مثال: نستخدم اداة Image فى صفحة ونغير خصائصها كالتالى, الخاصية ImageUrl نحدد عنوان الصورة عن طريق



وباقى الخصائص كما بالشكل

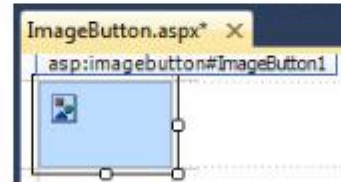
AlternateText	Minds Blog
ImageUrl	~/Images/2.png
Height	130px
ImageAlign	Right
Width	223px

وعند اختبار الصفحة يصبح الشكل كالتالى



### استخدام الأداة Image Button:

تستخدم أزرار الصور لعرض صور تستجيب لأحداث النقر ، وهذا يعطي هذه الأداة إمكانيات أكثر من الأداة Image التي تعرض صوراً فقط ولا تستجيب لأي أحداث.



### خصائص هذه الأداة ImageButton:

**AlternateText:** تضبط و تعيد النص الذي يظهر في العنصر ImageButton في حال عدم ظهور الصورة ، و يظهر كتلميح في حال وجود الصورة خاصة مع المتصفحات الحديثة.

**ImageUrl:** تضبط و تعيد عنوان URL للصورة المراد ظهورها في الصفحة.

**Width:** تضبط و تعيد عرض الصورة.

**Height:** تضبط و تعيد ارتفاع الصورة.

**ImageAlign:** تضبط و تعيد محاذاة الصورة في الصفحة (تحديد موقع الصورة بالنسبة لأدوات HTML) و تأخذ عدة قيم

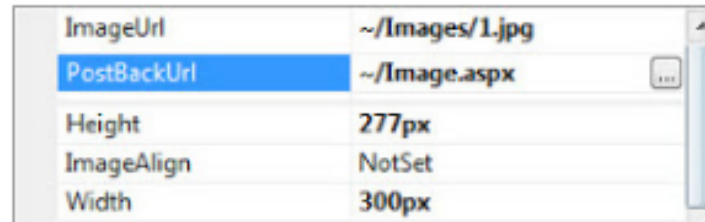
منها: Top, middle, Bottom, Left, Right.

**PostBackUrl:** للانتقال إلى صفحة معينة بعد إرسال البيانات إلى السيرفر.

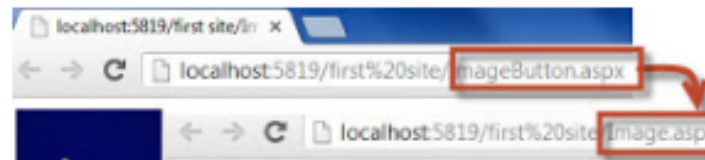
## الأحداث في الأداة ImageButton:

**Click:** يقع عند النقر على زر الصورة ImageButton.

**مثال:** نستخدم ImageButton ونغير الخصائص كالتالي وعند النقر عليها تنتقل الى الصفحة Image.aspx.



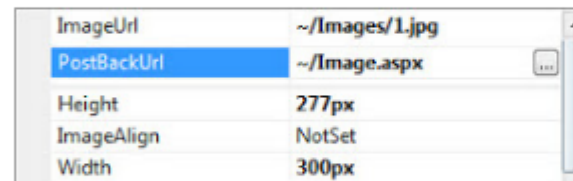
وعند اختبار الصفحة نجد ان عند النقر



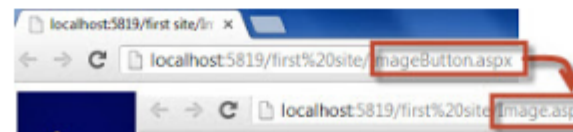
## الأحداث في الأداة ImageButton:

**Click:** يقع عند النقر على زر الصورة ImageButton.

**مثال:** نستخدم ImageButton ونغير الخصائص كالتالي وعند النقر عليها تنتقل الى الصفحة Image.aspx.



وعند اختبار الصفحة نجد ان عند النقر



### خصائص هذه الأداة ImageMap:

**AlternateText:** نص بديل يتم عرضه في حال عدم عرض الصورة المحددة.

**DescriptionUrl:** لتحديد رابط إلى صفحة تحتوي تفاصيل عن الصورة.

**HotSpots:** تكمننا من الحصول على جميع النقاط الساخنة الموجودة (HotSpots) ضمن أداة ImageMap.

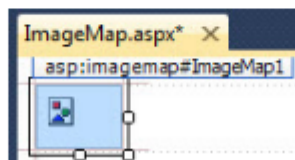
**ImageAlign:** تضبط و تعيد محاذاة الصورة في الصفحة (تحديد موقع الصورة بالنسبة لأدوات HTML) و تأخذ عدة قيم منها  
Top, middle, Bottom, Left, Right

**ImageUrl:** لتحديد مسار واسم الصورة التي سيتم عرضها.

**NavigateUrl:** تحدد عنوان URL المراد الانتقال إليه عند النقر.

### استخدام الأداة ImageMap:

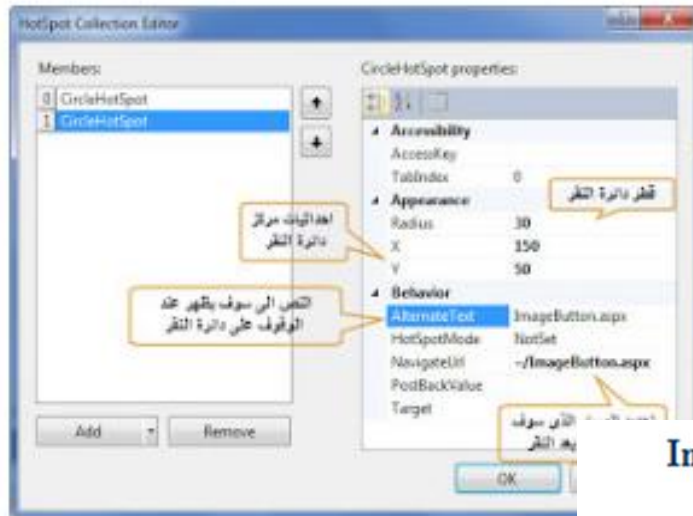
تستخدم هذه الأداة لعرض صورة ما بالإضافة عند نقر المستخدم على أجزاء مختلفة من الصورة فيمكننا الاستجابة بردود فعل مختلفة أيضا بحسب إحداثيات موقع النقر على الصورة , وعلى هذا الأساس فيمكننا استخدام هذه الأداة كوسيلة للتنقل بين الصفحات مثلا.



مثال : نستخدم اداة ImageMap ونغير الخصائص كالتالى

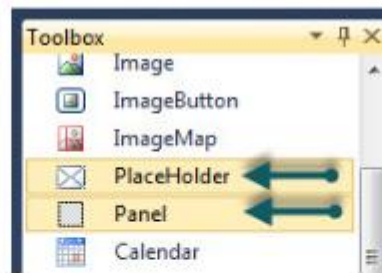
ImageUrl	~/Images/P1.png
HotSpots	(Collection) ...
Height	50px
ImageAlign	NotSet
Width	200px

ونحدد امان النقر من HotSpots ومسار الانتقال NavigateUrl



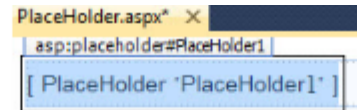
وعند اختبار الصفحة نجد عند النقر على اول HotSpot سوف ينتقل للصفحة Image.aspx





## استخدام الأداة Placeholder:

الأداة Placeholder عبارة عن مكان في الصفحة يستخدم لتخزين أدوات تحكم الويب (أدوات asp.net) التي تقوم بإضافتها إلى صفحة ويب في زمن التنفيذ Run Time (اثناء تشغيل الصفحات) لأنه لا يمكن إضافة الأدوات إلى صفحة.aspx ~ مباشرة في زمن التنفيذ.



\* لا تظهر هذه الاداة في الصفحة اثناء التنفيذ و لذلك لا تملك هذه الاداة خصائص التنسيق مثل تغيير لون الخلفية أو لون الخط أو غيرها من التنسيق .

\* لا تملك محددات المكان Placeholder مكان محدد في صفحة web وهذا يعني أنك تحدد موضع الأدوات التي تضيفها في الصفحة عن طريق الأدوات نفسها من خصائص التنسيق مثل الخاصية style أو cssClass أو ... .

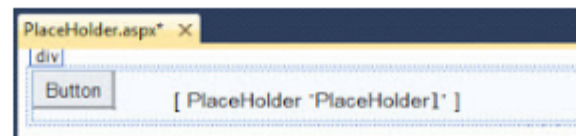
\* لا يمكن أن تضع أدوات أخرى بداخل الأداة Placeholder في صفحة التصميم , هي فقط تستخدم لإضافة الأدوات اثناء تشغيل الصفحات .  
\* تضاف الادوات إلى الأداة Placeholder في زمن التنفيذ عن طريق الإجراء...

(اسم الأداة المراد إضافتها) Placeholder1.Controls.Add:

مثال:

صفحة ASP.NET فيها زر Button و محدد مكان Placeholder عند النقر على الزر Button ينشئ Label ويضاف إلى الصفحة عن طريق الاداة Placeholder .

اولاً: نصمم الصفحة بالشكل

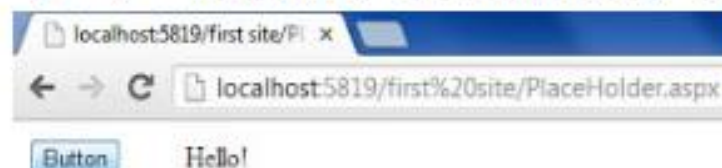




وثانيه تكتب الكود عند الحدث click

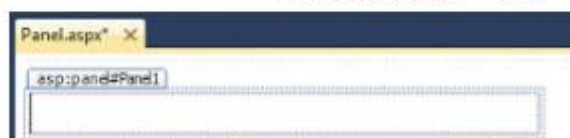
```
Label label1 = new Label();  
// هذا الأمر تعريف كائن جديد بالاسم label1 من نوع مربع تسمية label  
protected void Button1_Click(object sender, EventArgs e)  
{  
    label1.Text = "Hello!";  
    Placeholder1.Controls.Add(label1);  
}
```

وعند اختبار الصفحة نجد انه عند النقر على ال Button يظهر ال Label... بالشكل التالي



### استخدام الأداة Panel:

تستخدم الأداة Panel مكان لوضع ادوات اخرى، يمكن أن تضيف ادوات إليها في التصميم Design، كما يمكن إضافة ادوات إليها في زمن التنفيذ Run Time كما في الأداة Placeholder.



\* هذه الأداة مفيدة عندما أريد التعامل مع مجموعة من الأدوات مع بعضها البعض مثل إخفاءها أو إظهارها معاً أو تلوين جزء من الصفحة.

\* تضاف الادوات إلى الأداة panel في زمن التصميم عن طريق سحب الاداة ووضعها داخل الأداة Panel ، كما تضاف الادوات إلى الأداة Panel في زمن التنفيذ عن طريق الإجراء...

Panel1.controls.add (الأداة المراد إضافتها اسم);



### خصائص هذه الأداة panel:

**BackImageUrl:** تضبط و تعيد عنوان URL للصورة الخلفية للعنصر Panel.

**HorizontalAlign:** تضبط و تعيد المحاذاة الأفقية لمحتوى اللوحة Panel ، و تأخذ عدة قيم منها Left, center, right.

**Wrap:** تضبط و تعيد هل محتوى اللوحة Panel ملتفاً أم لا ، و تأخذ قيمتين:

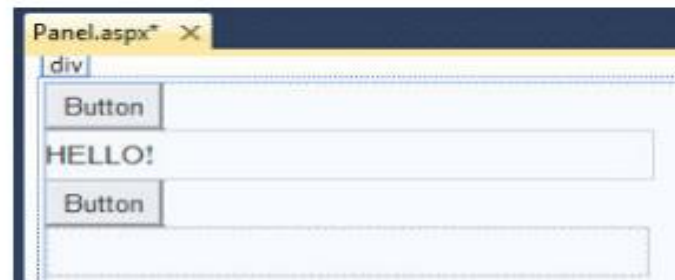
**True:** يصبح محتوى اللوحة Panel ملتفاً.

**False:** لا يصبح محتوى اللوحة Panel ملتفاً.

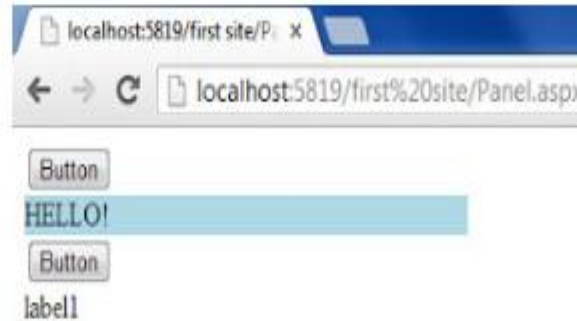
مثال:

صفحة ASP.NET فيها زرین Two Buttons و لوحتين Two Panels عند النقر على الزر الأول Button1 يجعل لون خلفية اللوحة الأولى اخضر ، و عند النقر على الزر الثاني Button2 ينشئ كائن من نوع مربع تسمية Label ويضاف إلى الصفحة عن طريق الاداة Panel.

اولاً: تصميم الصفحة بالشكل



وعند اختبار الصفحة نجد انه عند النقر على ال Button1 يتغير Panel1 اللون الى الازرق الفاتح وعند النقر على ال Button2 يظهر ال Label في ال Panel2 بالشكل التالي



وثانيه نكتب الكود عند الحدث click

```
protected void Button1_Click(object sender, EventArgs e)
{
    Panel1.BackColor = System.Drawing.Color.LightBlue;
}
Label label1 = new Label();
protected void Button2_Click(object sender, EventArgs e)
{
    label1.Text = "label1";
    Panel2.Controls.Add(label1);
}
```

# مميزات استخدام لوحة panel

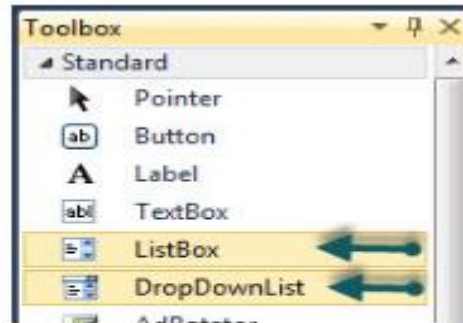


## اللوحة Panel :

- تستخدم لاحتواء الأدوات الأخرى ( تنظيم صفحة الويب).

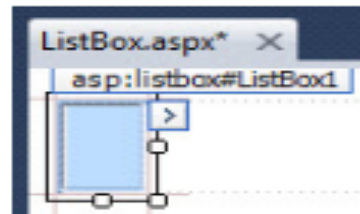
## المميزات:

- إظهار و إخفاء الأدوات في نفس الوقت
- تحجيم الادوات لتناسب مع عرض اللوحة نفسها وليس عرض المستعرض
- يمكنك اضافة اطار الى اداة اللوحة لظهارها بشكل جمالية مختلفة



### استخدام الأداة **ListBox**:

هي أداة تحكم تظهر على شكل قائمة مفتوحة في صفحة ASP.NET. تسمح بعمليات اختيار أحادية لعناصر القائمة **ListBox** ، أو بعمليات اختيار متعددة لعناصر القائمة **ListBox** .



### خصائص الأداة **ListBox**:

**SelectionMode**: تضبط و تعيد نمط الاختيار في مربع القائمة **ListBox** و تأخذ إحدى قيمتين:

- **Single** : تجعل القائمة تسمح باختيار أحادي .
- **Multiple** : تجعل القائمة تسمح باختيار متعدد .

## الأحداث في الأداة ListBox:

**SelectedIndexChanged:** يقع عند تغيير العنصر المختار في القائمة.

وكل عنصر من ال Items داخل ال ListBox له خصائص أهمها :

- **Text:** تحديد النص الذي يظهر بجانب كل مربع.
- **Value:** تحديد القيمة الموجودة في العنصر في القائمة ListBox.
- **Selected:** تصبح ب false أو True بحسب اختيار المستخدم.

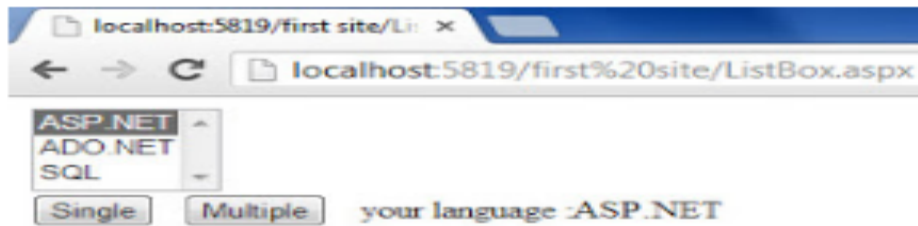
**ملحوظة:** يتم إضافة ال Items كما في ادوات الاختيار مثل **CheckBoxList**.

مثال: صفحة ASP.NET فيها أداة ListBox واداتين Button واداة Label وعند الضغط على الزر الاول يكتب العنصر المختار في مربع التسمية وعند الضغط على الزر الثاني يصبح ال ListBox ب Multiple وتكتب العناصر المختارة في مربع التسمية

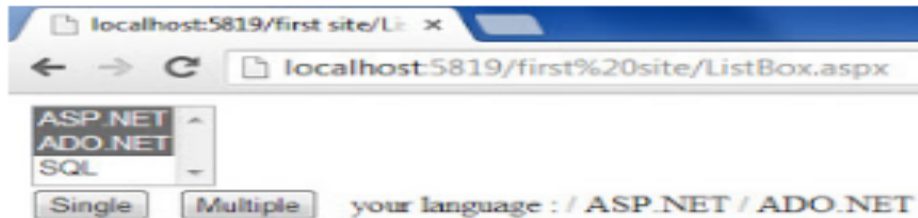
أولاً: تصميم الصفحة بالشكل التالي



وعند اختيار الصفحة..... الضغط على الزر الاول



وعند الضغط على الزر الثاني



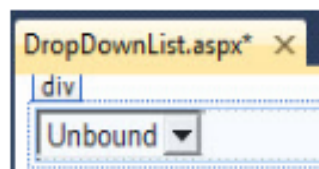
**ملحوظة:** يتم اختيار اكثر من عنصر في المتصفح بالضغط على زر **Ctrl** واختيار العناصر.

ثانيا: الكود كالتالى

```
string a = "your language :";  
protected void Button1_Click(object sender, EventArgs e)  
{  
    Label1.Text = a + ListBox1.Text;  
}  
protected void Button2_Click(object sender, EventArgs e)  
{  
    ListBox1.SelectionMode = ListSelectionMode.Multiple;  
    if (ListBox1.Items[0].Selected)  
        a = a + " / " + ListBox1.Items[0].Text;  
    if (ListBox1.Items[1].Selected)  
        a = a + " / " + ListBox1.Items[1].Text;  
    if (ListBox1.Items[2].Selected)  
        a = a + " / " + ListBox1.Items[2].Text;  
    Label1.Text = a;  
}
```

## استخدام الأداة DropDownList:

هي أداة تحكم تظهر على شكل قائمة منسدلة في صفحة ASP.NET. هي عبارة عن أداة تظهر على شكل مربع نص و زر متجاورين) عند النقر على الزر ينسدل قائمة من أسفل موقع مربع النص المجاور. تسمح أداة التحكم هذه بعمليات اختيار أحادية لعناصر ( بنود ) القائمة DropDownList فقط.



## خصائص الأداة DropDownList:

هي نفس خصائص الأداة ListBox ماعدا الخصائص التي تسمح بالاختيار المتعدد مثل **SelectionMode**.

## الأحداث في الأداة DropDownList:

**SelectedIndexChanged**: يقع عند تغيير العنصر المختار في القائمة.

وكل عنصر من ال Items داخل ال DropDownList له خصائص أهمها :

- **Text**: تحديد النص الذي يظهر .
- **Value**: تحديد القيمة الموجودة في العنصر في القائمة DropDownList.
- **Selected**: تأخذ القيمة بحسب اختيار المستخدم.



**مثال:** صفحة ASP.NET فيها أداة قائمة DropDownList و مربع نص TextBox ، عند تغيير العنصر المحدد في القائمة المنسدلة يطبع نص العنصر المحدد من القائمة المنسدلة DropDownList في مربع النص TextBox.

أولاً: نجعل الخاصية AutoPostBack تأخذ القيمة True ونصم الشكل كالتالي



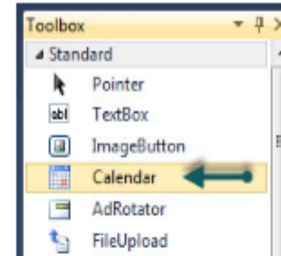
ثانياً: الكود كالتالي

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    TextBox1.Text = DropDownList1.SelectedItem.Text;
}
```

وعند اختبار الصفحة







في الدروس السابقة تحدثنا عن أدوات ASP.NET القياسية "Standard controls". وهنا سوف نتنقل إلى أدوات أكثر تشعباً والقوائم Menu.

### استخدام الأداة Calendar:

تستخدم لتمكين المستخدم من تحديد التواريخ بسهولة و التنقل إلى الشهر التالي أو الشهر السابق .

وخصائص وتعرف بالأدوات الغنية " Rich controls " ، من أمثلتها أداة التقويم Calendar وأداة عرض الشجرة TreeView.



### خصائص الأداة Calendar:

تضييق وتعديل المسافة الفاصلة بين جدار الخلية و محتواها .	CellPadding
تضييق وتعديل المسافة الفاصلة بين الخلايا في التكوين .	CellSpacing
تنسيق وتعديل النمط الخاص بأيام الأسبوع .	DayHeaderStyle
تنسيق وتعديل تنسيق اسم الأسبوع .	DayNameFormat
تنسيق وتعديل النمط الخاص بالأيام .	DayStyle
تنسيق وتعديل اليوم من الأسبوع الذي يظهر في أول عمود .	FirstDayOfWeek
تنسيق وتعديل النص المعنون لعنصر الانتقال إلى الشهر القادم .	NextMonthText
تنسيق وتعديل تنسيق عنصر الانتقال إلى الشهر القادم أو السابق .	NextPrevFormat
تنسيق وتعديل النمط الخاص بعنصر الانتقال إلى الشهر القادم أو السابق .	NextPrevStyle
تنسيق وتعديل النمط الخاص بالأيام المتبقية لغير الشهر المعروض .	OtherMonthDayStyle
تنسيق وتعديل نص عنصر الانتقال إلى الشهر السابق .	PrevMonthText
تنسيق وتعديل اليوم المحدد .	SelectedDate
تنسيق مجموعة أفراس من DateTime للأيام المحددة .	SelectedDates
تنسيق وتعديل النمط الخاص للأيام المحددة .	SelectedDayStyle
تنسيق وتعديل نمط الاختيار ، محددة فيما إذا كان بإمكانك اختيار يوم أو أسبوع أو شهر .	SelectionMode
تنسيق وتعديل النمط الخاص بعنصر اختيار الشهر .	SelectMonthText
تنسيق وتعديل النمط الخاص بعنصر اختيار الأسبوع أو الشهر .	SelectorStyle
تنسيق وتعديل النص الخاص بعناصر اختيار الأسبوع .	ShowWeekText
تنسيق وتعديل فيما إذا كان اليوم في رأس الأسبوع مرتباً .	ShowDayHeader
تنسيق وتعديل فيما إذا كانت أسطر الشبكة بين الأيام .	ShowGridLines
تنسيق وتعديل فيما إذا كان عناصر الانتقال إلى الشهر القادم أو السابق ظاهرة .	ShowNextPrevMonth
تنسيق وتعديل فيما إذا كان العنوان ظاهراً .	ShowTitle
تنسيق وتعديل تنسيق العنوان .	TitleFormat
تنسيق وتعديل النمط الخاص بالعنوان .	TitleStyle
تنسيق وتعديل تاريخ اليوم .	Today'sDate
تنسيق وتعديل تاريخاً للتأكد من أنه مرئي .	VisibleDate
تنسيق وتعديل النمط الخاص بأيام نهاية الأسبوع .	weekendDayStyle

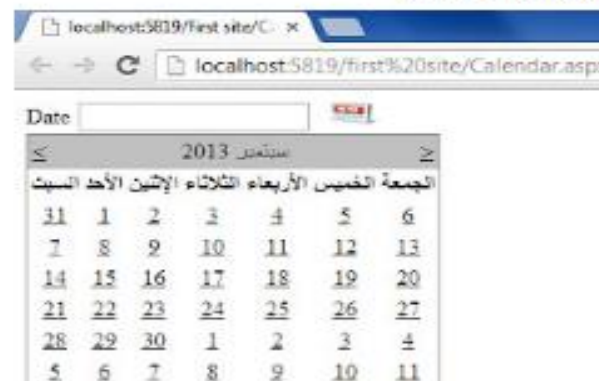
## الأحداث في الأداة Calendar:

**SelectionChanged:** يتم إطلاقه عند اختيار تاريخ جديد.

**VisibleMonthChanged:** يتم إطلاقه عند الضغط على رابط الشهر التالي أو السابق.

مثال:

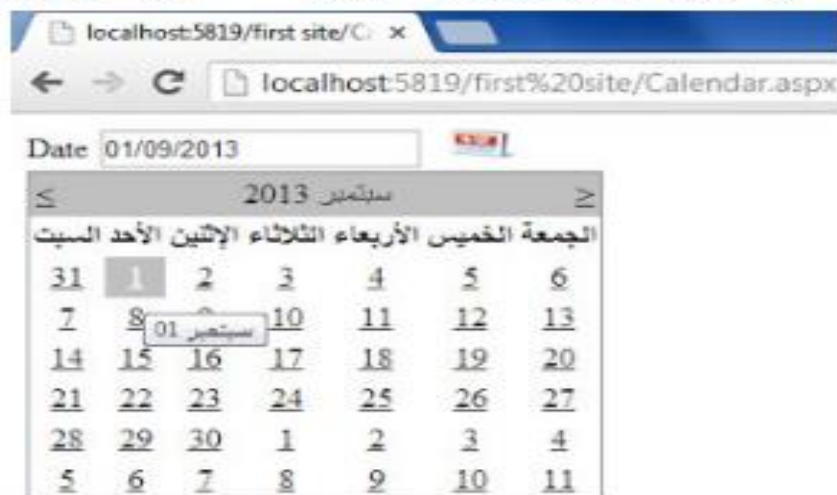
صفحة asp.net بها Calendar و TextBox و ImageButton بها صورة على شكل Calendar وعند الضغط عليها تظهر الـ Calendar وعند اختيار التاريخ يكتب داخل مربع النص .  
أولاً : تصمم الصفحة بالشكل التالي ..  
**ملحوظة:** تجعل خاصية الـ Visible في الـ Calendar بـ false.



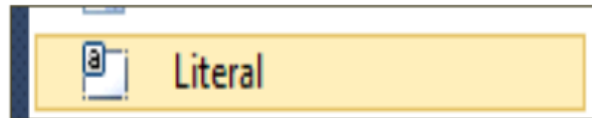
تالياً: كود الـ c#

```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    Calendar1.Visible = true;
}
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    TextBox1.Text = Calendar1.SelectedDate.ToShortDateString();
}
```

وعند اختيار الصفحة ... عند الضغط على صورة ال Calendar تظهر وعند اختيار التاريخ يظهر في مربع النص



# استخدام أداة literal



• أبسط أدوات الويب

• تستخدم في تعيين نص ثابت

• لا يمكن تغيير النص أثناء عرض الصفحة إلا من خلال الكود

مثال:

تعيين نص للأداة **Literal** عبارة عن الوقت والتاريخ الحالي ؟

خطوات الإنشاء:

- تصميم الواجهة
- كتابة الكود (تحديد الحدث)
- عرض الصفحة على المتصفح

---

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load  
    Literal1.Text = DateTime.Now  
End Sub
```

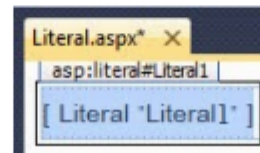
---

### الفرق بين الأدوات **Label** و **Literal**؟

أداة العنوان **Label** يمكن التحكم في خصائص النص الظاهر فيها وهو ما لا يمكن  
في الأداة **Literal**

## استخدام الأداة Literal:

تتشابه هذه الأداة مع الأداة Label في أنها تعرض نص في الصفحة ، ما يميز هذه الأداة عن الأداة Label أن هذه الأداة إذا أعطيت الخاصية Text نص فيه وسوم HTML (تنسيق الخط) فإنه يمكن جعل الوسوم هذه لا تنفذ و إظهارها و يمكن أيضا جعلها تنفذ.



## خصائص الأداة Literal:

**Mode:** و تأخذ أحد ثلاث قيم:

1. **Transform:** وهذا يعني أن الوسوم HTML في الخاصية Text تنفذ وتعرض النتائج بعد تنفيذها.
2. **Encode:** لا تنفذ وتظهر الوسوم كما كتبت مع النص.
3. **Pass Through:** إذا كان البيئة التي ضمنها تفهم وسوم HTML فإنها تنفذ، وإذا كانت البيئة لا تنفذ فإنها تظهر كما هي.

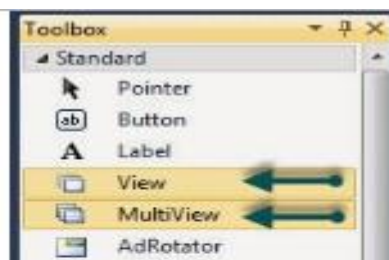
مثال:

صفحة ASP.NET بها ثلاث ادوات Literal وسوف تستخدم لكل اداة قسمة من قيم الخاصية Mode

\* تصمم الصفحة وتجعل كود صفحة ASP.NET كالتالى لتحديد قيم الخاصية Mode

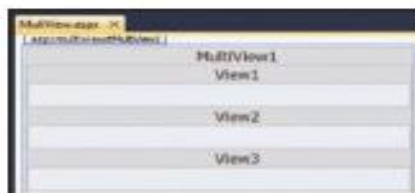
```
<asp:Literal ID="Literal1" runat="server"
    Text="<b> <i> Literal Tool </i> </b>" Mode="Transform"></asp:Literal>
<br />
<asp:Literal ID="Literal2" runat="server"
    Text="<b> <i> Literal Tool </i> </b>" Mode="Encode"></asp:Literal>
<br />
<asp:Literal ID="Literal3" runat="server"
    Text="<b> <i> Literal Tool </i> </b>" Mode="PassThrough"></asp:Literal>
```

وعند اختيار الصفحة تجد



استخدام الأداة MultiView :

تستخدم MultiView لتقسيم نموذج كبير إلى عدة نماذج فرعية أصغر , تضم بداخلها أدوات من النوع View.



## خصائص الأداة MultiView :

**ActiveViewIndex :** لتحديد رقم المنظور View الذي سيتم عرضه والترقيم يبدأ من الصفر.  
**Views:** للحصول على مجموعة المناظير التي ضمن الأداة MultiView.

## الأحداث في الأداة MultiView :

**ActiveViewChanged:** يحدث عند اختيار واحد من المناظير التي بداخلها.

## استخدام الأداة View :

تستخدم لاحتواء الادوات داخل اداء اخرى تحتويها كال MultiView .  
خصائص الأداة View :

أداة التحكم View لا تدعم أي خصائص, وذلك لأنها لا تستخدم إلا داخل أداة أخرى تحتويها.  
**الأحداث في الأداة View :**

**Activate:** تحدث عندما نختار هذا المنظور من مجموعة مناظير الأداة MultiView ليتم عرضها.  
**Deactivate:** تحدث عندما نختار منظور آخر, أي عندما يلغى عرض هذا المنظور.  
**ملحوظة:**

يمكن إضافة Button داخل كل منظور يبحث ينقلنا إلى منظور آخر عند الضغط حيث ان الأداة MultiView لها الاوامر التالية:

**NextView:** لعرض المنظور التالي للمنظور المعروض حاليا ضمن MultiView

**PrevView:** لعرض المنظور السابق للمنظور المعروض حاليا ضمن MultiView

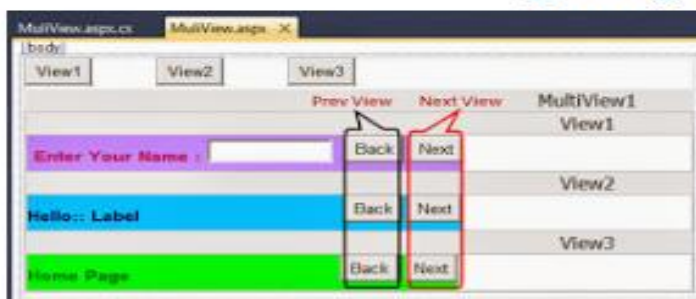
حيث نستخدم الخاصية **CommandName** التي في ال Button لنعطيهها قيمة **NextView** او **PrevView**.



مثال:

صفحة ASP.NET بها أداة MultiView وتحتوي على ثلاثة أدوات View ويوجد ثلاث أدوات Button تستخدم لظهور ال View وفي كل منظور يوجد اثنين Button واحدة NextView والآخرى PrevView.

**أولاً:** نصمم الصفحة بالشكل التالي



**ثالثاً:** الكود بالشكل التالي

```
protected void Button1_Click(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 0;
}
protected void Button2_Click(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 1;
}
protected void Button3_Click(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 2;
}
protected void Button6_Click(object sender, EventArgs e)
{
    Label1.Text = TextBox1.Text;
}
```

**ثانياً:** نحدد خاصية CommandName

CommandName	NextView
CommandName	PrevView