



برمجه صفحات الانترنت ASP



من هنا لم يسمع بـ ASP.NET بعد من مطوري المواقع؟!! ، فالجيل القادم سيكون للغات البرمجة التي تزيل العبء عن كاهل المطورين وتخصر 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 هناك ASP بدون دوت نت و هي عبارة عن في بي سكريبت وكانت مليئة بالمشاكل وبطبيعة وبها عيوب كثيرة لست هنا بصدده ذكرها.
- وبعد صدور الدوت نت قامت مايكروسوفت بتطوير ال ASP وجعلها ضمن إطارها ووجهة لتطوير الواقع 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. " اختصار ال 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 الإنترنّت سواء مواقع ثابتة أو ديناميكيّة ويمكن تفصيلها كما يلي:

المواقع الثابتة: static website:

وهي عبارة عن مواقع عاديّة على الإنترنّت تكون محتويّة على صور ونصوص أو بشكل أبسط هي مواقع الإنترنّت التي تم كتابتها على بـ (HTML) حيث تكون كلّ صفحة مسؤولة عن باقى الصفحات ولا يوجد أيّ قواعد للبيانات.

مواقع الديناميكية : Dynamic website

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

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.

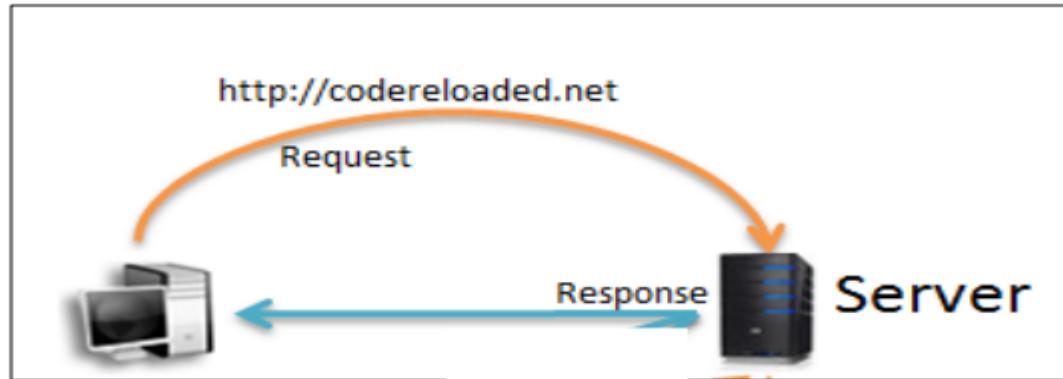
- عملية الطلب تُعرف بـ (Request)

- 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



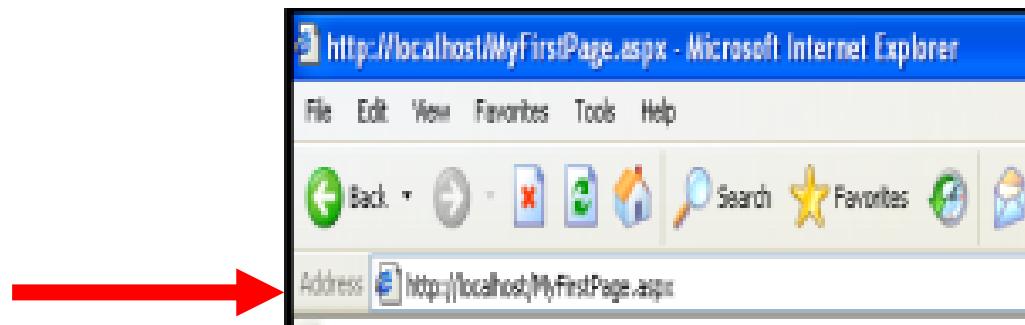
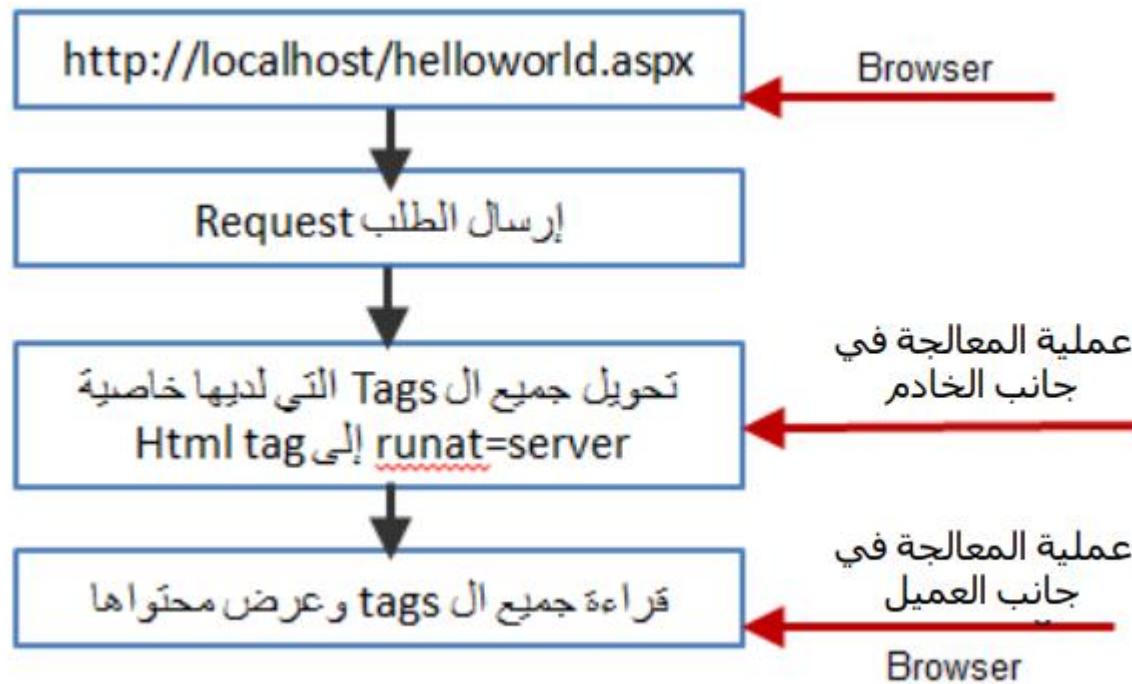
• في حالة أن Request & Response قد حدثوا من نفس الجهاز فإن جهازك يعتبر Client & Server في آن واحد



الكود المصدرى (source)

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

كيف يصبح عنوان الصفحة ..



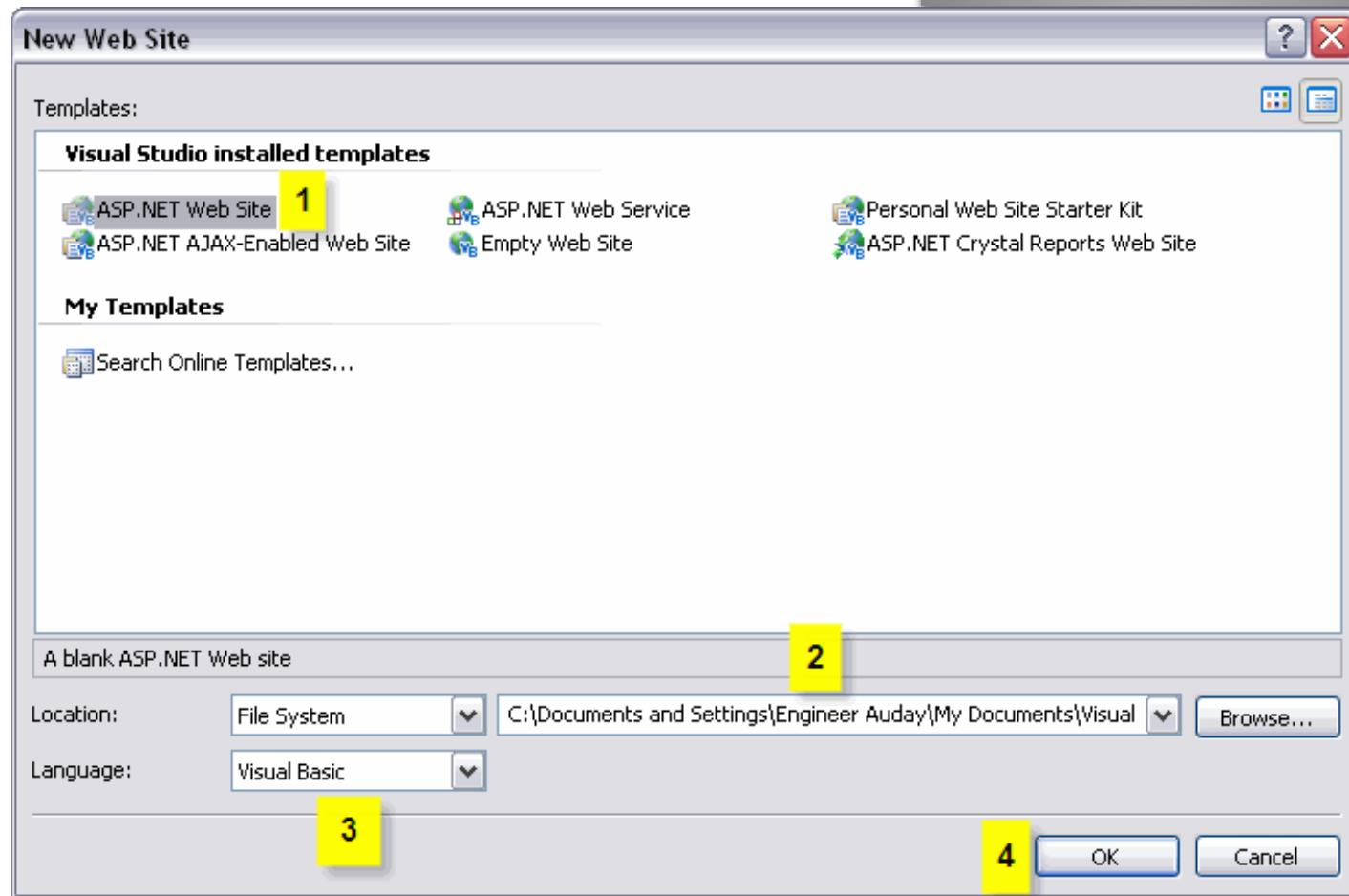
!! الفرق بين ASP و ASP.net !!

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

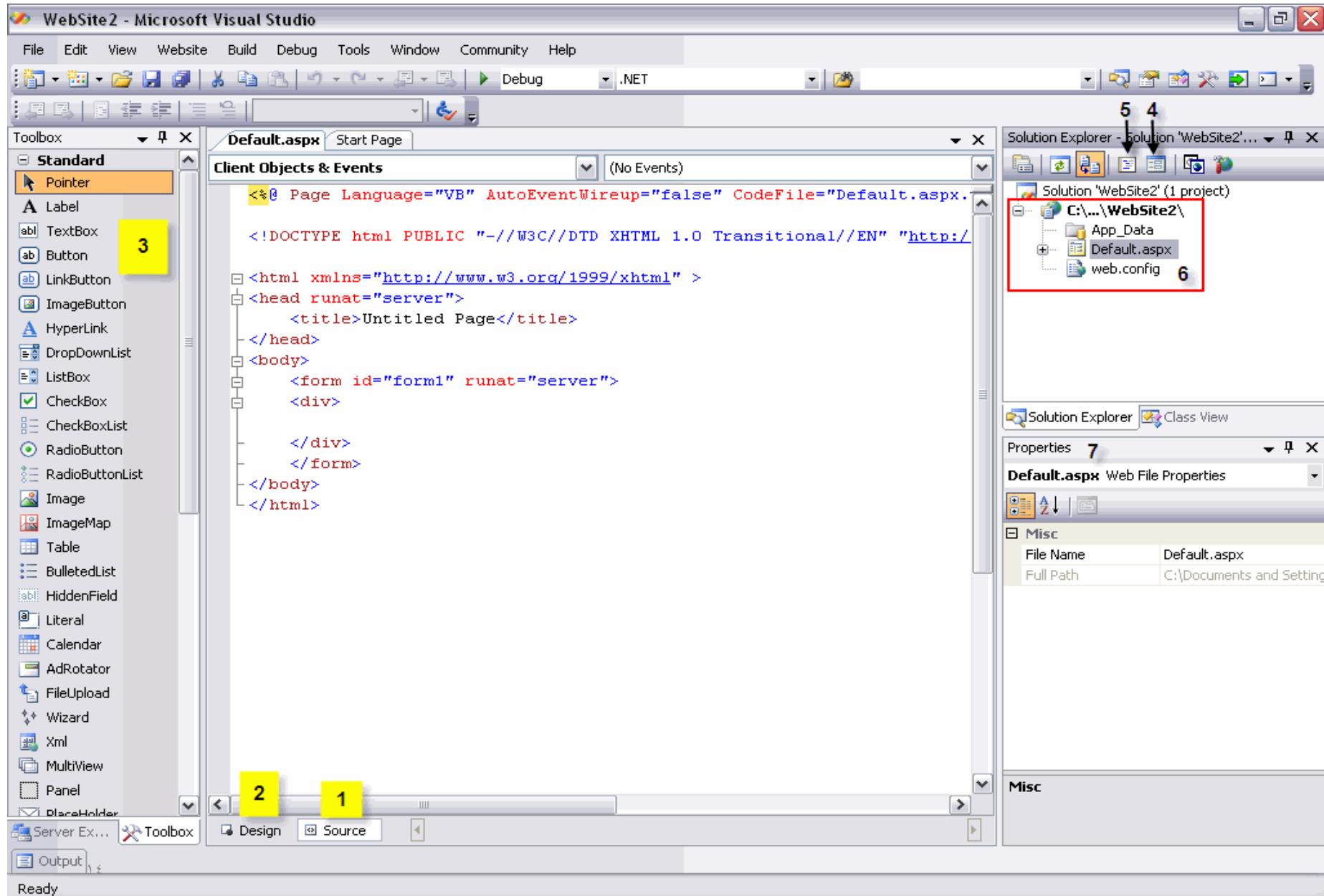
ما الذي يميز ASP.NET عن غيرها!!

١. تتميز بالسرعة بالمقارنة مع غيرها؛ وذلك لأنّها 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 الفيجوال ستوديو.
٥. تعدد وتكامل اللغات، أي يمكننا البرمجة بالكثير من اللغات ومن أهمها:
لغة Visual Basic .Net). لغة C#.Net).

File → New → Web Sit...



سيتم انشاء موقع جديد و تظهر لنا النافذة التالية ..



حيث تشير الارقام الى :

- ٢ ، ٤ - نافذة التصميم.
- ٥ - نافذة الكود (VB.Net).
- ٧ - نافذة الخصائص.

- ١ - نافذة الكود (HTML).
- ٣ - صندوق الادوات.
- ٦ - موقع الويب وملفاته.

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



كود HTML الناتج عن اضافة الادوات:

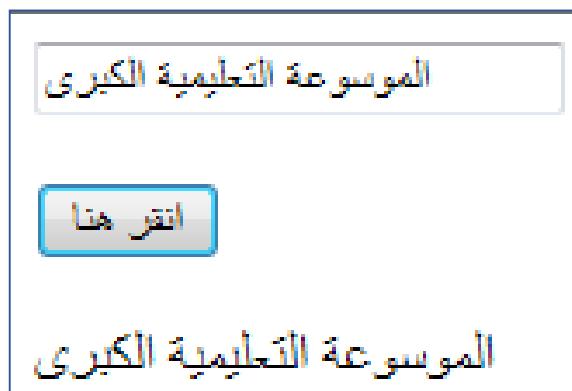
```
<asp: Label ID="Label1" runat="server" Text="Label"></asp: Label>
<br />
<asp: Button ID="Button1" runat="server" Text="Button" /></div>
```

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

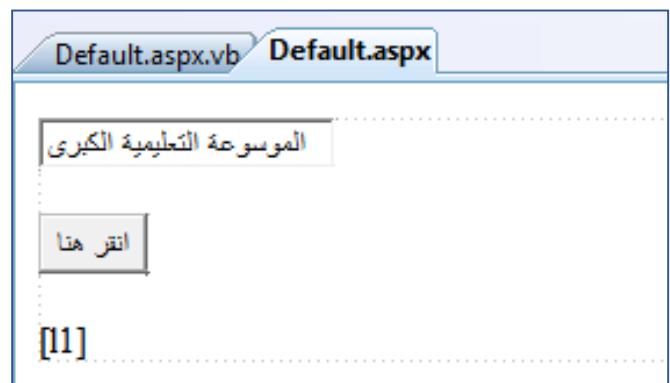
تغيير الخاصية text الى Hello ,this is my test من مربع الخصائص او من خلال الكود:

```
<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.



`t1.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>

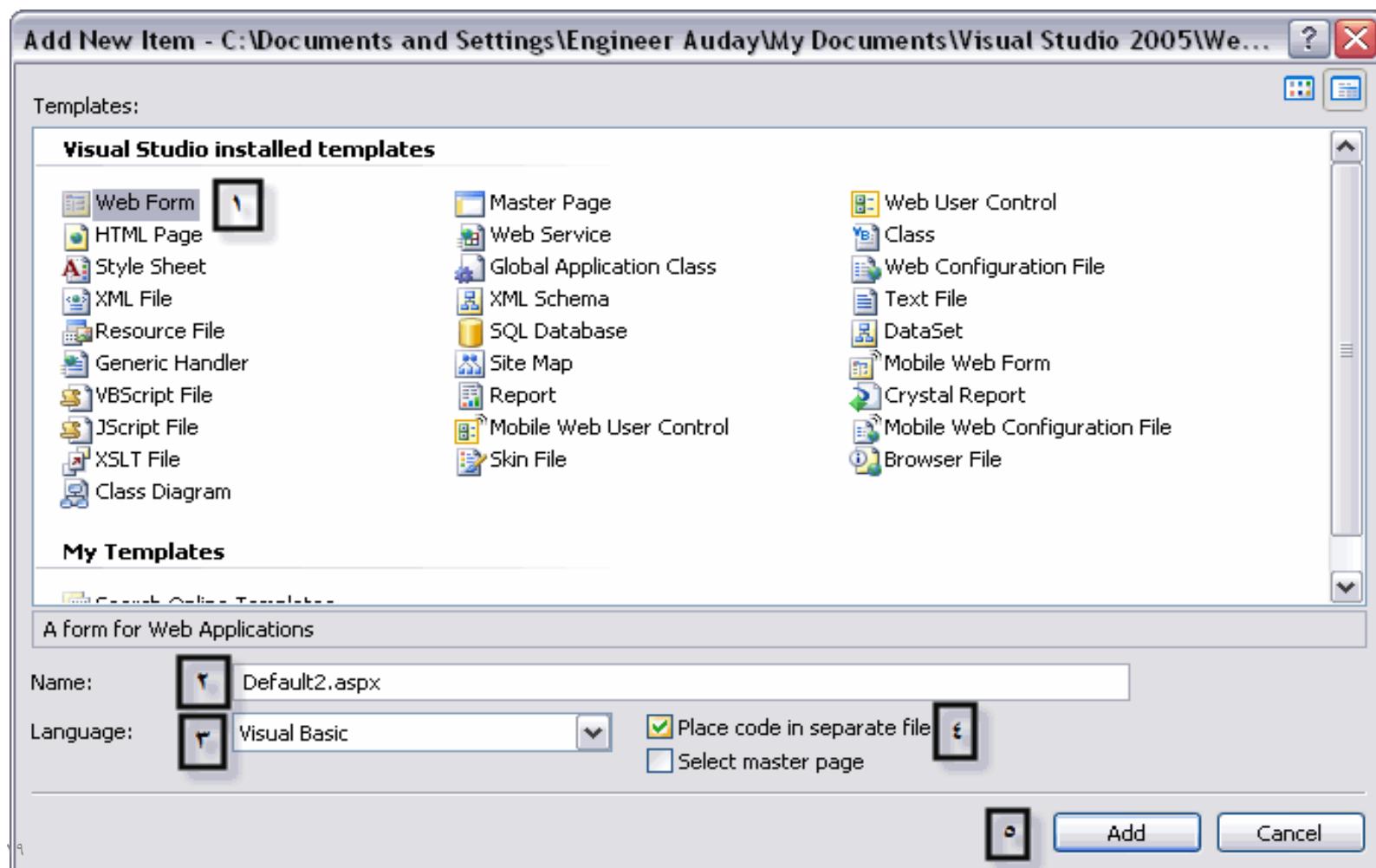
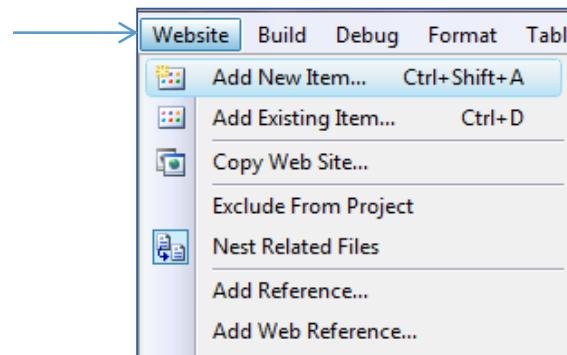
    </div>
  </form>
</body>
```

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

File Edit Format

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

اضافة صفحة جديدة للموقع



استخدام الآدوات



ASP.NET

في هذا الدرس سوف نتعلم استخدام:

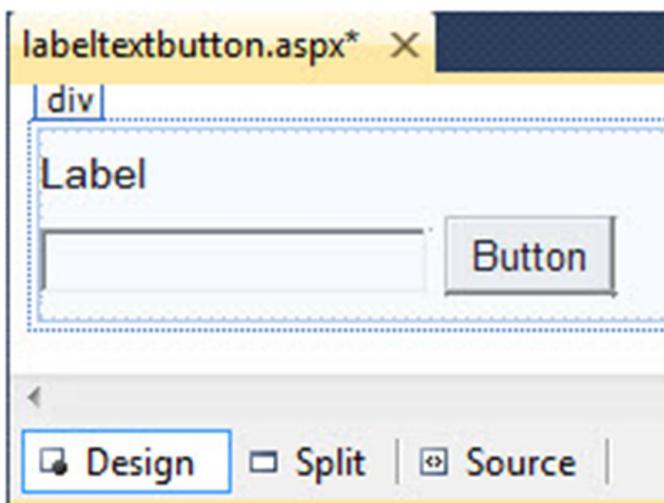
.Label -1

.TextBox -2

.Button -3

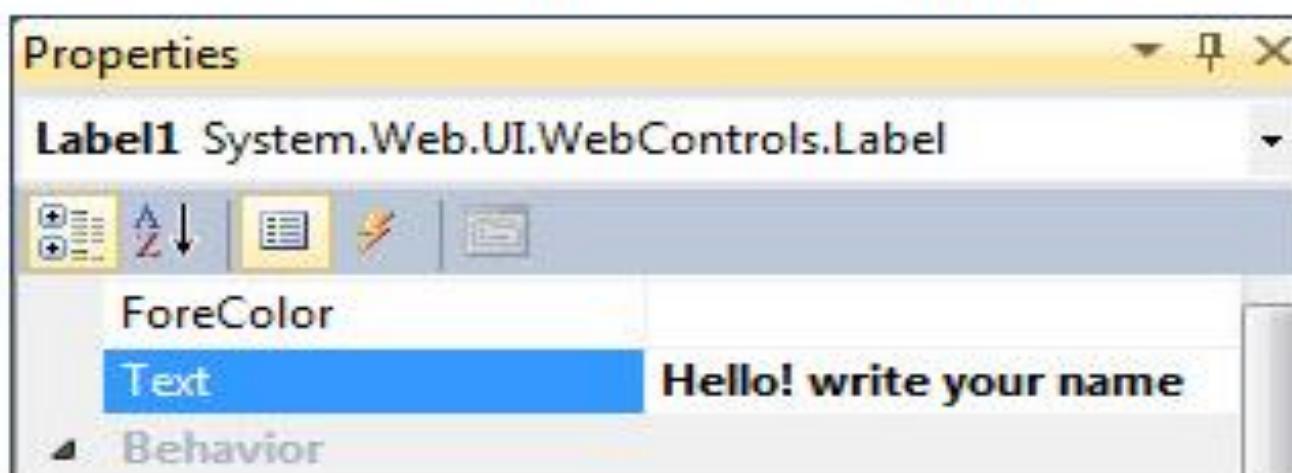
أولاً : إضافة الأدوات :

نقوم بإضافة الأدوات **Button** **TextBox** **Label** كمان التشكيل التالي



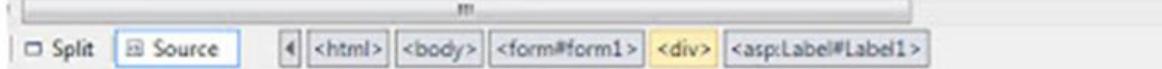
ثانياً : نبدأ نتعرف على اهم خصائص الادوات المستخدمة :

ولنبدأ باداة **Label** و اهم خصائصها هو ال **Text** ويمكن تغييرها من خلال نافذة الخصائص

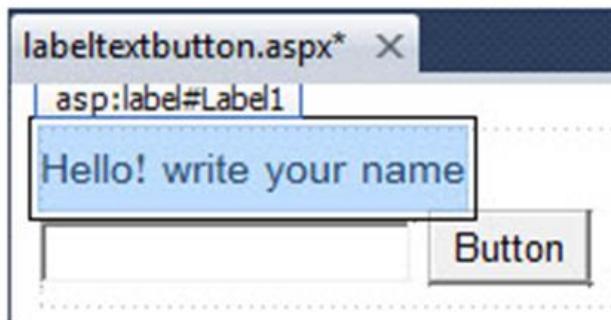


او من خلال نافذة ال Source

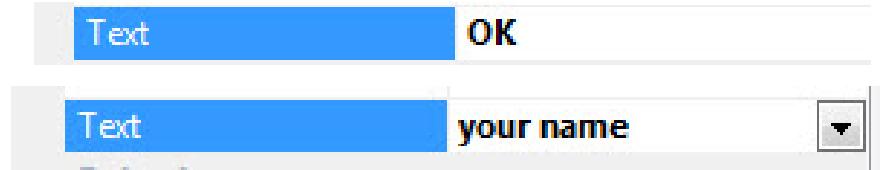
```
<asp:Label ID="Label1" runat="server" Text="Hello! write your name" </asp:Label>  
<br />
```



لتغير النص المكتوب على ال Label بالشكل التالي



والآن الى الاداء **Button** و **TextBox** و اهم **الخصائص**

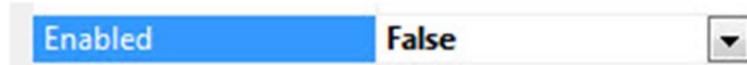


يصبح التك

Hello! write your name

A simple application window is shown. It contains a text input field with the placeholder text 'your name' and an 'OK' button to its right. The window title is 'Hello! write your name'.

وخاصية **Enabled** وهي المسئولة عن تفعيل الاداء ونلاحظ اذا تغيرت الى False



يصبح التك

Hello! write your name

A simple application window is shown. It contains a text input field with the placeholder text 'your name' and an 'OK' button to its right. The window title is 'Hello! write your name'.

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



ولذا نغير الى MultiLine يصبح التشكيل

Hello! write your name

your name



Button

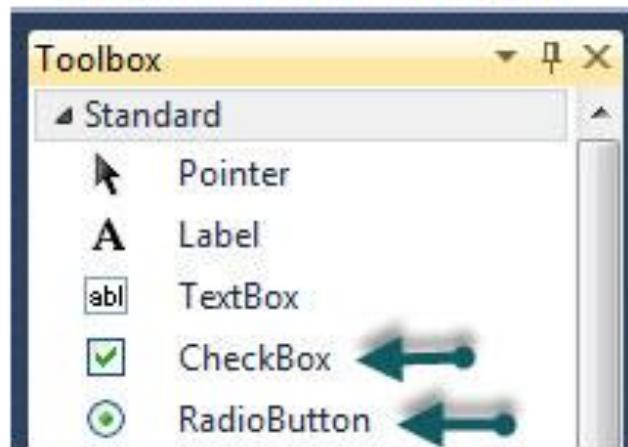
وإذا تغير إلى Password يصبح الشكل

Hello! write your name

.....

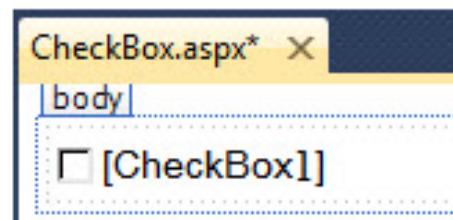
Button

[3] ASP.NET أدوات



: CheckBox

هي الأداة المربعة الشكل التي يوضع علامة داخلها عند اختيارها ، تسمح هذه الأداة 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 تغير بحالته تساوى `false` افتراضيا.

Checked: تصبح بـ `True` أو `false` بحسب اختيار المستخدم.

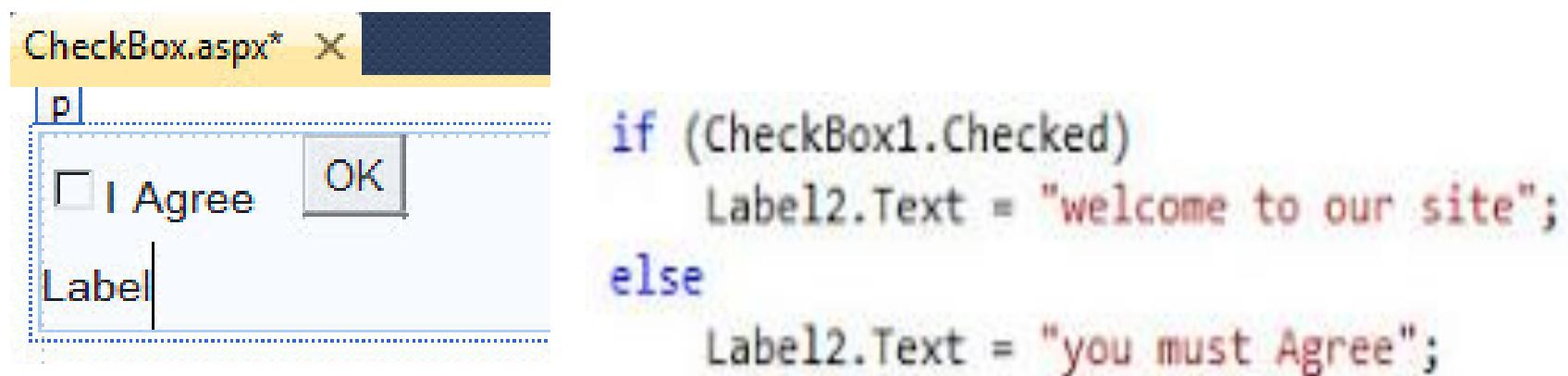
Text: تحديد النص الذي يظهر بجانب هذه الأداة.

TextAlign: موضع النص بالنسبة للأداة، على يمينها أم يسارها تأخذ احدى القيمتين `Left`, `Right`.

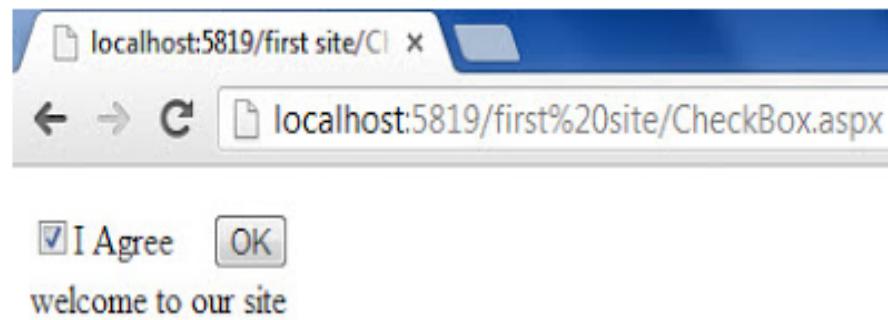
الأحداث في الأداة CheckBox

CheckedChanged: يقع عند تغيير الخاصية `Checked`

مثال : باستخدام if الشرطية ..



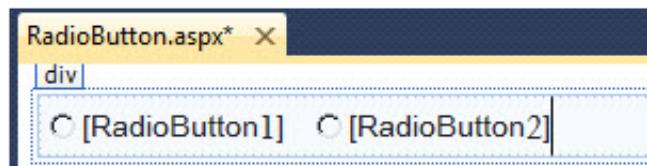
ليصبح الشكل في صفحة ال asp.net



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

استخدام أداة :RadioButton

هي الأداة الدائرية التي يرسم في وسطها نقطة عند اختيارها ، لاظهر 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 الموجودة فيها. تحديد سوى خيار واحد فقط من أدوات RadionButton .

Checked : تصبح بـ True أو false بحسب اختيار المستخدم.

Text : تضبط و تعيّد النص الظاهر بجوار زر راديو RadioButton .

TextAlign : تضبط و تعيّد محاذاة النص الخاص بزر الراديو RadioButton .

الأحداث في الأداة :RadioButton

CheckedChanged : يقع عند تغيير الخاصية Checked .

مثال استخدام radiobutton

RadioButton.aspx

choose your color :

Red
 Blue
 Black

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

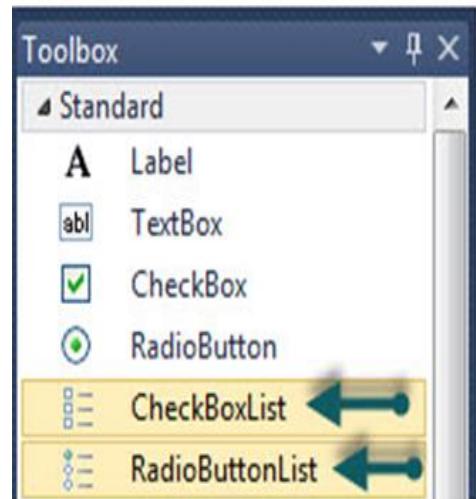


choose your color :

Red
 Blue
 Black

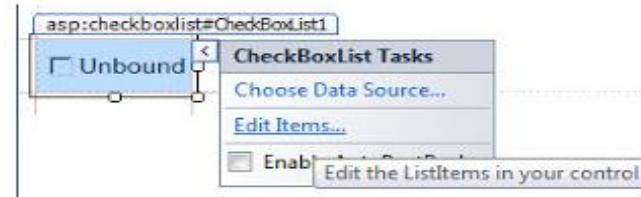
your color : Red

[4] ASP.NET أدوات



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 تغير بحالتة تساوى `false` افتراضيا.

SelectedItem: تعيين العنصر المحدد.

Items: تعيين مجموعة العناصر الموجودة في `.CheckBoxList`.

SelectedIndex: تحدد فهرس العنصر المحدد.

DataSource: تحدد مصدر البيانات (تستخدم عادةً مع قواعد البيانات).

Datamember: تحدد مصدر بيانات (عادةً جدول في قاعدة بيانات).

الأحداث في الأداة : CheckBoxList

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

- وكل عنصر من ال **Items** داخل ال **CheckBoxList** له خصائص اهمها :

Text : تحديد النص الذي يظهر بجانب كل مربع.

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

Selected : تصبح بـ **True** أو **false** بحسب اختيار المستخدم.

Enabled : وهي المسئولة عن تفعيل الاداء.

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

: تعني أن عناصر القائمة تظهر عمودياً ، و هي القيمة الافتراضية . **RepeatDirection.Vertical** .1

: تعني أن عناصر القائمة تظهر أفقياً . **RepeatDirection.Horizontal** .2

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

: تظهر القائمة على شكل جدول ، هي القيمة الافتراضية . **RepeatLayout.Table** .1

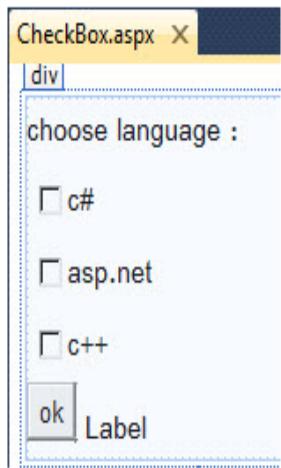
: تظهر القائمة على بدون جدول . **RepeatLayout.Flow** .2

: تظهر القائمة في شكل تعداد نقطي . **RepeatLayout.UnorderedList** .3

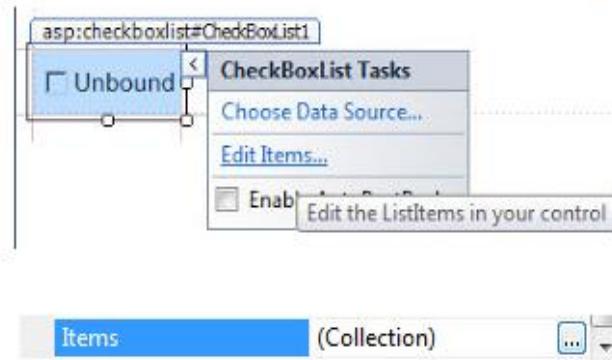
: تظهر القائمة في شكل تعداد رقمي . **RepeatLayout.OrderedList** .4

و من خلال النافذة المخصصة لاضافة عناصر جديدة نضيف مجموعة من العناصر و نعدل خاصية ال Text

مثال : نتفق الشكل التالي

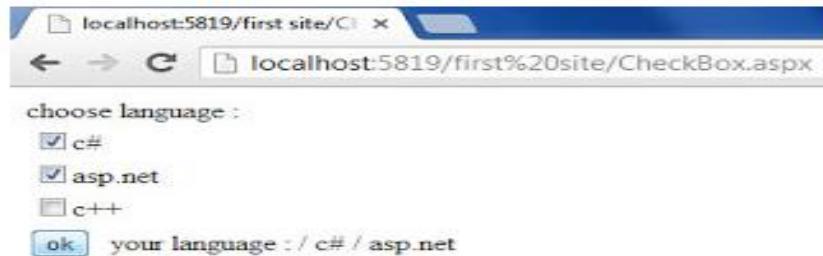


ويتم اضافة ال Items من خلال



او من قائمة الخصائص

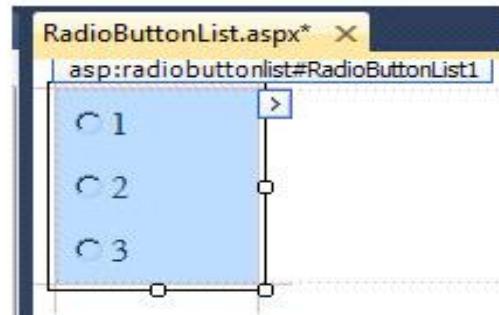
ليصبح الشكل في صفحة .net



```
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

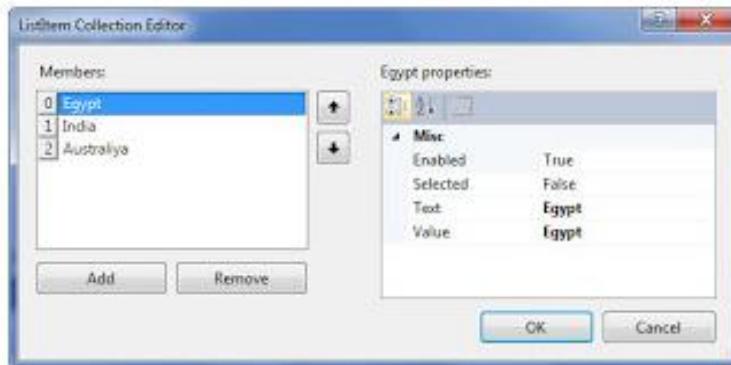
تستخدم الأداة **RadioButtonList** في إظهار قائمة من أزرار الراديو **RadioButtons** مع بعضها البعض و التعامل معها كمجموعة.



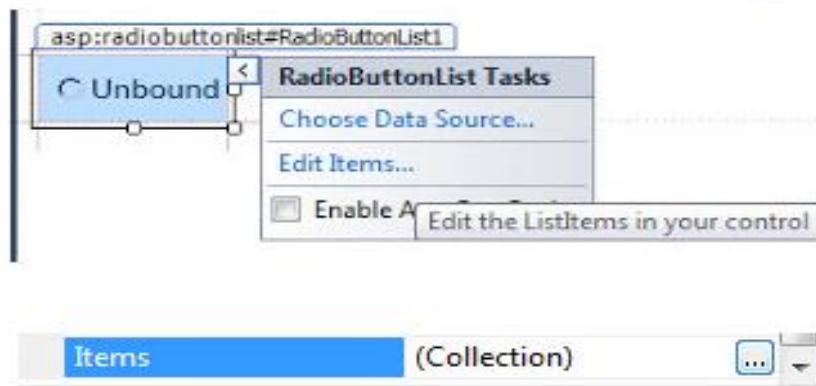
خصائص الأداة :RadioButtonList

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 لاضافة عناصر جديدة نضيف مجموعة من العناصر ونعدل خاصية ال Text .

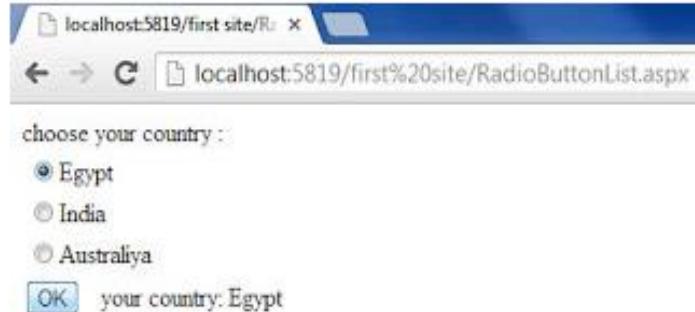


ويتم اضافة ال Items من خلال



او من قائمة الخصائص

ليصبح الشكل في صفحة asp.net

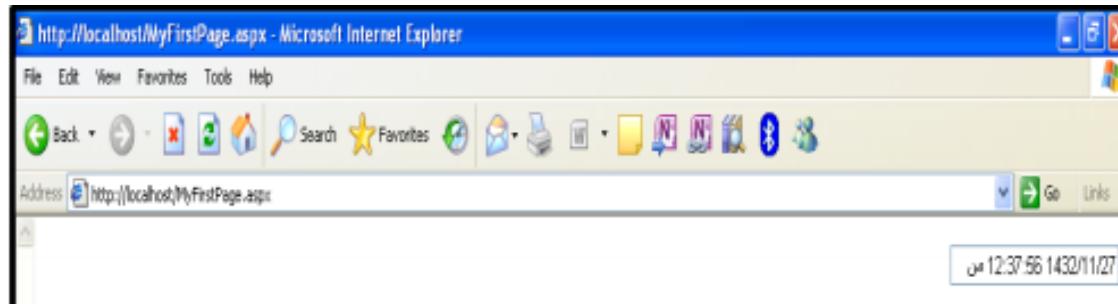


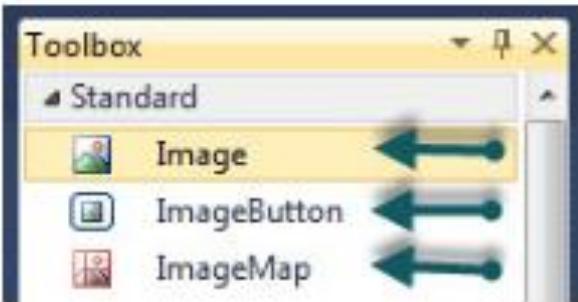
```
Label1.Text = " your country: " + RadioButtonList1.Text;
```

مثال /عرض التاريخ والوقت على الصفحة

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

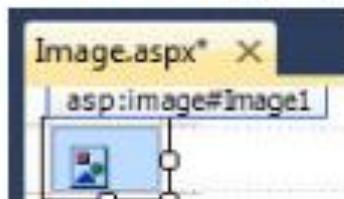
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Text1.Value = DateAndTime.Now
    End Sub
End Class
```





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 ، ويظهر كتلمينج في حال وجود الصورة خاصة مع المتصفحات الحديثة.

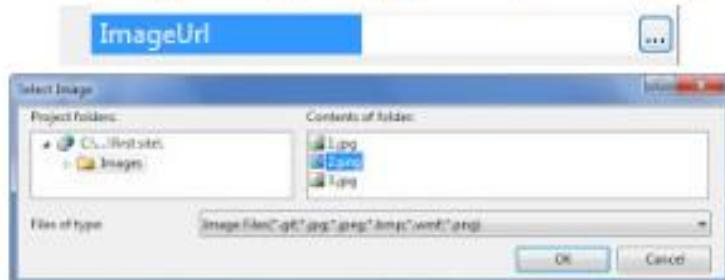
ImageUrl: تضبط و تعيد عنوان URL للصورة المراد ظهورها في الصفحة.

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 في صفحة ونغير خصائصها كالتالى ، الخاصية ImageUrl نحدد عنوان الصورة عن طريق



وباقي 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 وتعيد النص الذي يظهر في الغذر **ImageButton** في حال عدم ظهور الصورة ، و يظهر كتلميح في حال وجود الصورة خاصة مع المتصفحات الحديثة.

ImageUrl: تضبط وتعيد عنوان URL للصورة المراد ظهورها في الصفحة.

Width: تضبط وتعيد عرض الصورة.

Height: تضبط وتعيد ارتفاع الصورة.

ImageAlign: تضبط وتعيد 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.

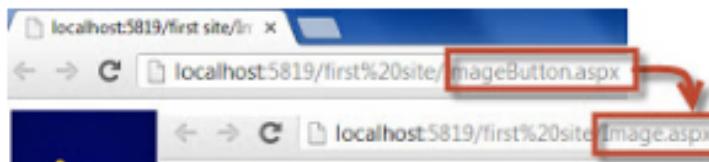
الأحداث في الأداة :ImageButton

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

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

ImageUrl	~/Images/1.jpg
PostBackUrl	~/Image.aspx
Height	277px
ImageAlign	NotSet
Width	300px

وعند اختبار الصفحة نجد ان عند النقر



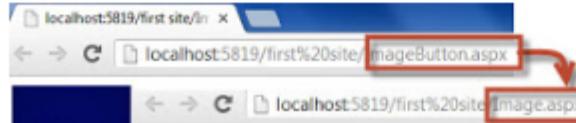
الأحداث في الأداة :ImageButton

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

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

ImageUrl	~/Images/1.jpg
PostBackUrl	~/Image.aspx
Height	277px
ImageAlign	NotSet
Width	300px

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**

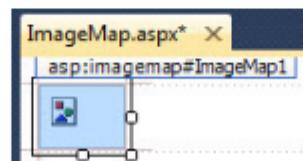
: **ImageAlign** تضبط وتعيد محاذاة الصورة في الصفحة (تحديد موقع الصورة بالنسبة لأدوات HTML) وتأخذ عدة قيم منها .Top, middle, Bottom, Left, Right

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

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

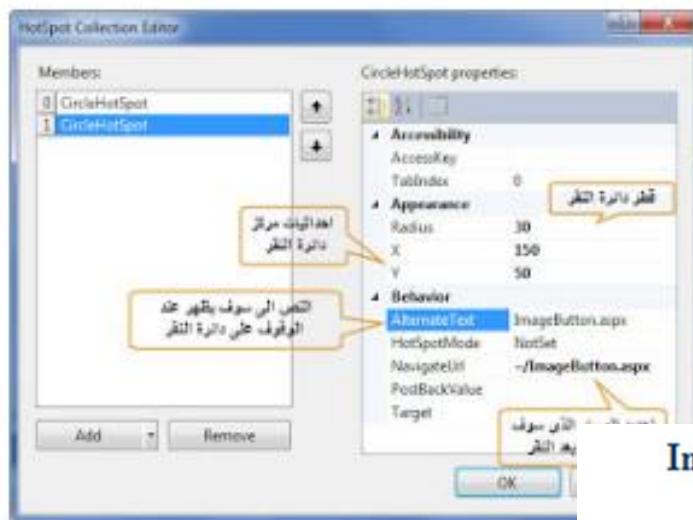
استخدام الأداة :ImageMap

تستخدم هذه الأداة لعرض صورة ما بالإضافة عند نقر المستخدم على أجزاء مختلفة من الصورة فيإمكاننا الاستجابة بردود فعل مختلفة أيضاً بحسب إحداثيات موقع النقر على الصورة ، وعلى هذا الأساس فيإمكاننا استخدام هذه الأداة كوسيلة للتنقل بين الصفحات مثلـ.



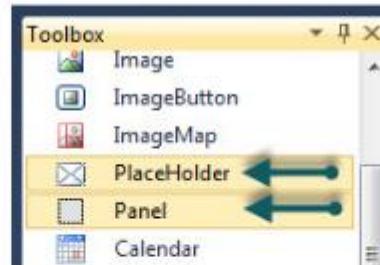
ImageUrl	~/Images/P1.png
HotSpots	(Collection)
Height	50px
ImageAlign	NotSet
Width	200px

ونحدد امان النقر من HotSpots ومسار الانتقال



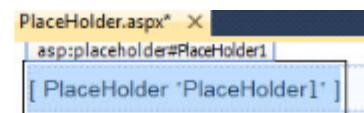
وعند اختبار الصفحة نجد عند النقر على اول HotSpot سوف ينتقل للصفحة





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 (الثناء تشغيل الصفحات) لأنه لا يمكن إضافة الأدوات إلى صفحة .aspx. ~ مباشرة في زمن التنفيذ.

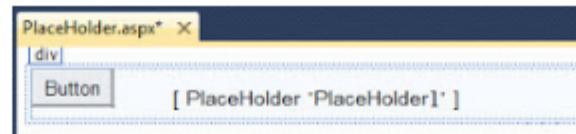


- * لا تظهر هذه الأداة في الصفحة أثناء التنفيذ و لذلك لا تملك هذه الأداة خصائص التصيير مثل تغيير لون الخلفية أو لون الخط أو غيرها من التصييرات.
- * لا تملك محددات المكان PlaceHolder مكان محدد في صفحة web و هذا يعني أنك تحدد موضع الأدوات التي تصييرها في الصفحة عن طريق الأدوات نفسها من خصائص التصييرات مثل الخاصية style أو cssClass أو
- * لا يمكن أن تضع أدوات أخرى يداخل الأداة PlaceHolder في صفحة التصميم ، هي فقط تستخدم لإضافة الأدوات أثناء تشغيل الصفحات.
- * تضاف الأدوات إلى الأداة PlaceHolder في زمن التنفيذ عن طريق الإجراء ...

(اسم الأداة المراد إضافتها):

مثال:

صفحة ASP.NET فيها زر Button و محدد مكان PlaceHolder عند النقر على الزر Button ينشئ Label ويضاف إلى PlaceHolder .
الصفحة عن طريق الأداة PlaceHolder او لا: نصمم الصفحة بالشكل



وثائقيه نكتب الكود عند الحدث click

```
Label label1 = new Label();
// هنا الأمرتعريف كائن جديد بالاسم label1 من نوع مربع تسمية
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 أدوات أخرى، يمكن أن تضيف أدوات إليها في التصميم **Design**، كما يمكن إضافة أدوات 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 الأدوات إلى الأداة **panel** في زمن التصميم عن طريق سحب الأداة ووضعها داخل الأداة **Panel** ، كما تضاف الأدوات إلى الأداة **Panel** في زمن التنفيذ عن طريق الإجراء...

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

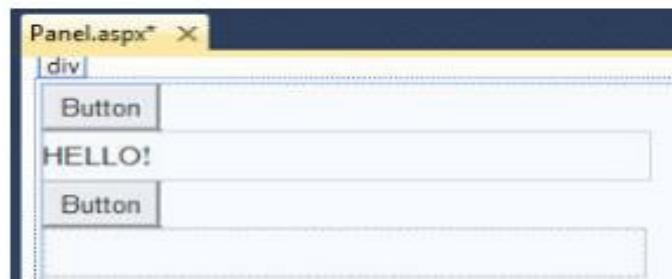
خصائص هذه الأداة :panel

- .Panel: تضبط و تعيّد عنوان URL للصورة الخلفية للعنصر Panel.
- .Left, center, right: تضبط و تعيّد المعاذلة الأفقية لمحتوى اللوحة Panel ، و تأخذ عدة قيم منها HorizontalAlign.
- :Wrap تضبط و تعيّد هل محتوى اللوحة Panel ملتفاً أم لا ، و تأخذ قيمتين:
 - True: يصبح محتوى اللوحة Panel ملتفاً.
 - False: لا يصبح محتوى اللوحة Panel ملتفاً.

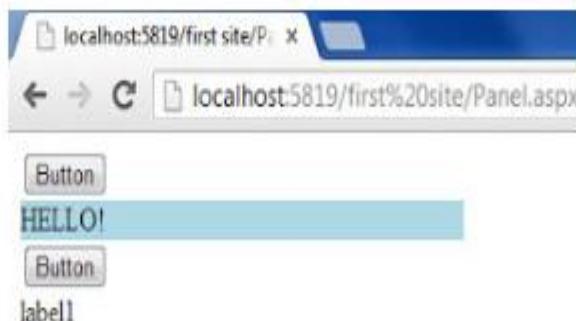
مثال:

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

(ولا: تصمم الصفحة بالشكل



و عند اختبار الصفحة نجد انه عند النقر على ال Button1 يتغير اللون الى الازرق الفاتح و عند النقر على ال Panel1 يظهر ال Label2 في ال Panel2... بالشكل التالي



وثانية 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

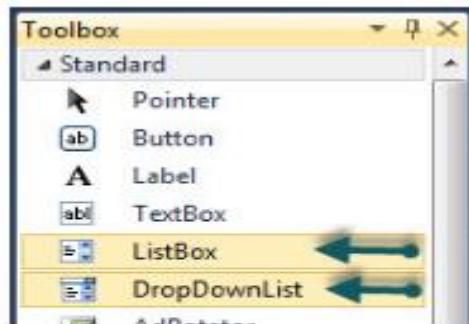


اللوحة : **Panel**

- تستخدم لاحتواء الأدوات الأخرى (تنظيم صفحة الويب).

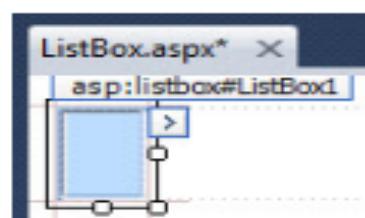
المميزات:

- إظهار و إخفاء الأدوات في نفس الوقت
- تحجيم الأدوات لتتناسب مع عرض اللوحة نفسها وليس عرض المستعرض
- يمكنك إضافة إطار إلى لوحة لإظهارها باشكال جمالية مختلفة



استخدام الأداة :ListBox

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



خصائص الأداة :ListBox

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

- **Single** : تجعل القائمة تسمح باختيار أحدى .
- **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 اهمها :

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

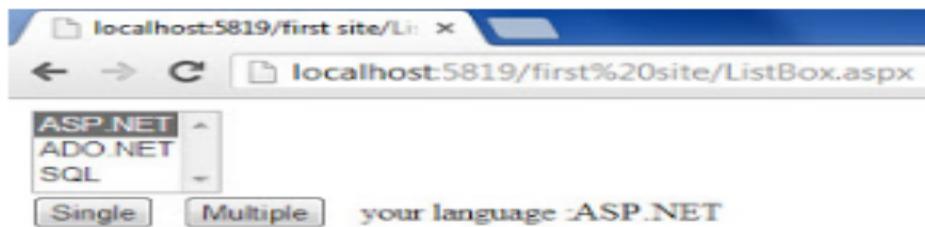
ملاحظة: يتم اضافة ال **Items** كما في ادوات الاختيار مثل **.CheckBoxList**

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

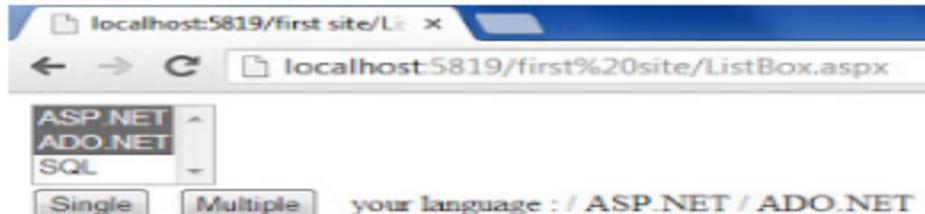
أولاً: تصميم الصفحة بالشكل التالي



وعند اختبار الصفحة..... الضغط على الزر الاول



وعند الضغط على الزر الثاني



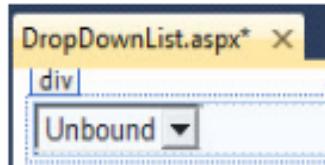
ملاحظة: يتم اختيار اكثرب من عنصر فى المتصفح بالضغط على زر **Ctrl** واختيار العناصر.

ثانياً: 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 متلاصق) عند النقر على الزر ينسدل قائمة من أسفل موقع مربع النص المجاور. تسمح أداة التحكم هذه بعمليات اختيار أحدية 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 اهمها :

- `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 في مربع النص TextBox .

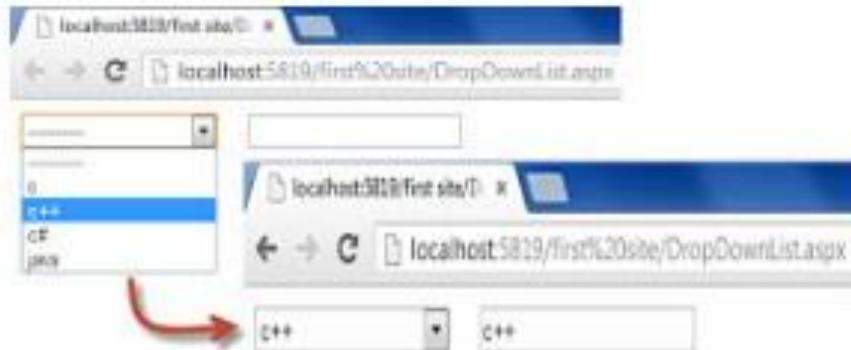
أولاً: نجعل الخاصية AutoPostBack تأخذ القيمة True وتصمم الشكل كالتالي



ثانياً: الكود كالتالى

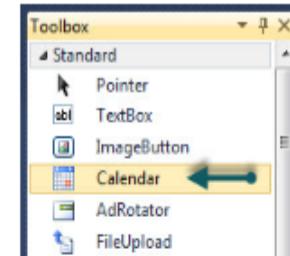
```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    TextBox1.Text = DropDownList1.SelectedItem.Text;
}
```

وعند اختبار الصفحة



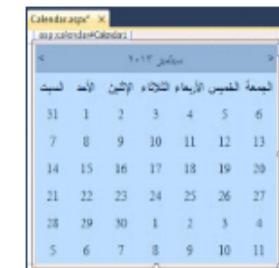
Menu

استخدام الأداة :Calendar

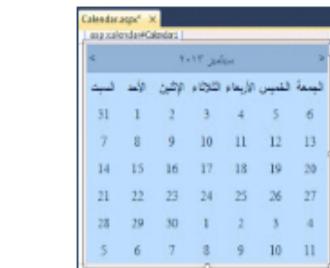


في الدرس السابق تحدثنا عن أدوات ASP.NET القياسية "Standard controls" ..وهذا سوف ننتقل إلى أدوات أكثر تشبيعاً والقوانين

خصائص الأداة :Calendar	
CellPadding	تضييق وتعيد المسافة الفاصلة بين حطار الخلية و محتواها .
Cellspacing	تضييق و تعيد المسافة الفاصلة بين الخلايا في التقويم .
DayHeaderStyle	تضييق و تعيد النمط الخاص باليوم الأسبوع .
DayNameFormat	تضييق و تعيد تنسيق اسم الأسبوع.
DayStyle	تعيد النمط الخاص بالأيام .
FirstDayOfWeek	تضييق و تعيد اليوم من الأسبوع الذي يظهر في أول عبود .
NextMonthText	تضييق و تعيد النص المعنون لمنصر الانتقال إلى الشهر القادم .
NextPrevFormat	تضييق و تعيد تنسيق هناري الانتقال إلى الشهر القادم أو السابق .
NextPrevStyle	تضييق و تعيد النمط الخاص بعنصري الانتقال إلى الشهر القادم أو السابق.
OtherMonthDayStyle	تضييق و تعيد النمط الخاص بال أيام المنوبة لغير الشهر المعروض .
PrevMonthText	تضييق و تعيد نص عنصر الانتقال إلى الشهر السابق .
SelectedDate	تضييق و تعيد اليوم المحدد .
SelectedDates	تعيد مجموعة أخرى من DateTime للأيام المختلفة .
SelectedDayStyle	تعيد النمط الخاص لثبات المختارة .
SelectionMode	تضييق و تعيد نمط الاختيار ، مختلفة فيما إذا كان يمكنك اختيار يوم أو أسبوع أو شهر .
SelectMonthText	تضييق و تعيد النمط الخاص بعنصري اختيار الشهر .
SelectorStyle	تعيد النمط الخاص بعنصري اختيار الأسبوع أو الشهر .
ShowWeekText	تضييق و تعيد النص الخاص بعنصري اختيار الأسبوع .
ShowDayHeader	تضييق و تعيد فيما إذا كان اليوم في رأس الأسبوع مرتباً .
ShowGridLines	تضييق و تعيد فيما إذا كانت خطوط الشبكة بين الأيام .
ShowNextPrevMonth	تضييق و تعيد فيما إذا كان منصر الانتقال إلى الشهر القادم أو السابق ظاهرة .
ShowTitle	تضييق و تعيد فيما إذا كان العنوان ظاهراً .
TitleFormat	تضييق و تعيد تنسيق العنوان .
TitleStyle	تعيد النمط الخاص بالعنوان .
TodaysDate	تضييق و تعيد تاريخ اليوم .
VisibleDate	تضييق و تعيد تاريخاً تنتهي منه من آلة مرتدي .
weekendDayStyle	تعيد النمط الخاص باليوم نهاية الأسبوع .



وخصائص وتعرف بالادوات القيمة " Rich controls " ، من امثلتها اداة التقويم Calendar واداة عرض الشجرة TreeView



الاحداث في الاداء :Calendar

يتم اطلاقه عند اختيار تاريخ جديد . SelectionChanged

يتم اطلاقه عند الضغط على رابط الشهر التالي أو السابق . VisibleMonthChanged

مثال :

صفحة asp.net بها Calendar و TextBox و ImageButton وبها صورة على شكل Calendar وعند الضغط عليها تظهر الـ Calendar وعند اختيار التاريخ يكتب داخل مربع النص .

أولاً : نصمم الصفحة بالشكل التالي ..

ملحوظة: تجعل خاصية الـ Visible في الـ Calendar بـ false



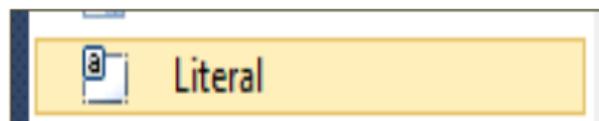
ناتجاً: كود الـ c#

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

و عند اختيار الصفحة ... عند الضغط على صورة الـ Calendar تظهر و عند اختيار التاريخ يظهر في مربع النص



استخدام أداة Literal



أبسط أدوات الويب

- تستخدم في تعين 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

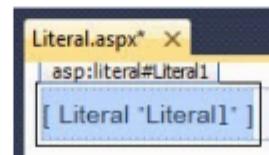
```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    Literal1.Text = DateAndTime.Now
End Sub
```

الفرق بين الأداتين **Literal** و **label**

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

استخدام الأداة :Literal

تشابه 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 أيضاً جعلها تنفذ.



خصائص الأداة :Literal

و تأخذ أحد ثلاثة قيم : **Mode**

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

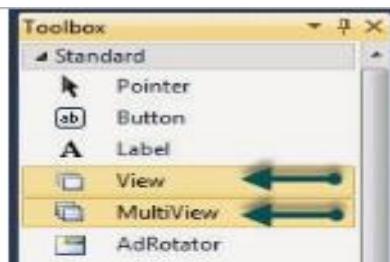
مثال:

صفحة ASP.NET بها تلات أدوات Literal وسوف تستخدم لكل إداء قسمة من قيم الخاصية Mode

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

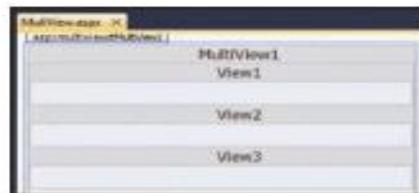
```
<asp:Literal ID="Literal1" runat="server"
    Text=<b><i> Literal Tool </i></b>" Mode="Transform"></asp:Literal>
<br />
<asp:Literal ID="Literal2" runat="server"
    Text=<b><i> Literal Tool </i></b>" Mode="Encode"></asp:Literal>
<br />
<asp:Literal ID="Literal3" runat="server"
    Text=<b><i> Literal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

وعند اختبار الصفحة تجد



استخدام الأداة MultiView

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



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

ActiveViewChanged : يحدث عند اختيار واحد من المناظير التي بداخلها.

استخدام الأداة : View

تستخدم لاحتواء الأدوات داخل أداء آخر تحتويها كالتالي .
MultiView :
View : خصائص الأداة
أداة التحكم View لا تدعم أي خصائص، وذلك لأنها لا تستخدم إلا داخل أداء آخر تحتويها.
View في الأحداث :

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 :

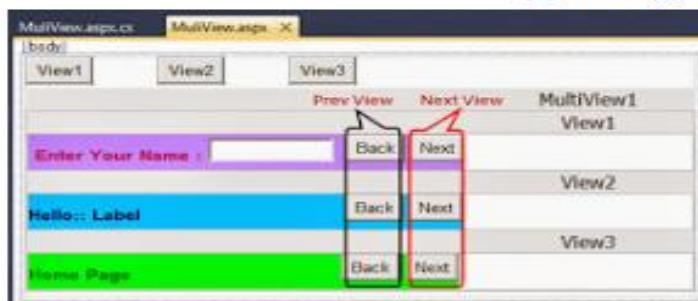
يمكن إضافة Button داخل كل منظور يبحث بعده إلى منظور آخر عند الضغط حيث أن الأداة MultiView لها الأوامر التالية:

NextView : لعرض المنظور التالي للمنظور المعروض حاليا ضمن MultiView
PrevView : لعرض المنظور السابق للمنظور المعروض حاليا ضمن MultiView حيث تستخدم الخاصية CommandName التي في ال Button لتعطيها قيمة NextView أو PrevView .

مثال:

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

أولاً: نصمم الصفحة بالشكل التالي



ثانياً: 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