# ASP.Net Web Forms Course

## الدرس الثانى

## مقدمة إلى ASP.Net (2)

**المحتويات :-**

- **حين تبدأ فى إنشاء تطبيقاً جديدا .**
- **المجلدات الخاصة داخل ASP.Net .**
- **كيفية إضافة أكواد إلى الصفحة .**
  - **Server-Side Code**
  - **Client-Side Code**
- **نافذة الآدوات Toolbox .**
- **كيف يمكن تثبيت IIS على جهازك ؟**
- **إنشاء أول موقع لك عبر IIS ؟**

**الهدف من هذا الدرس :-**

- التفرقة بين Web Site , Web Application .
- التعرف على المجلدات الخاصة بـ ASP.Net والتى لا يمكنك إستخدامها إلا للغرض الذى أعدت له .
- التعرف على كيفية إضافة أكواد إلى الصفحة سواءً كانت Server أو Client .
- التعرف على بعضاً من الآدوات ومعرفة خصائصهم وأحداثهم .
- التفاعل مع IIS وكيفية تثبيته , وإنشاء مواقع عبر IIS .

## حين تبدأ فى إنشاء تطبيقا جديدا

بعد تشغيل Visual Web Developer ,ستحتاج أن تبدأ فى إنشاء موقعك ,فحين تقوم بالذهاب إلى قائمة File ومن ثم تريد إنشاء مشروع ,فهل ستختار Web Site أم Web Application ؟ ___ وما الفرق بينهم ؟ ومتى أختار أحدهم دون الآخر ؟ .

يتوقف إختيار أحدهم دون الآخر على ما بهم من مزايا , فتجد أن أحدهم يقدم مزايا ليست فى الآخر , ولكن المزايا الغير موجودة فى الآخر يمكن الوصول إليها بطرق غير مباشرة .ولكن هناك ما لم يمكن الوصول إليه بطريقة مباشرة أو غير مباشرة .

فعلى سبيل المثال :-

- يقدم لك Web Application وجود namespace لملفات التطبيق إفتراضياً , بينما لا يقدم Web Site هذا , ولكن يمكن إضافته يدوياً فى Web Site .
- لا يسمح Web Application أن يحتوى المشروع الواحد على ملفات لغات vb و C# معاً , ولكن Web Site يسمح بوجود Class بـ C# وآخر vb . ولكن يمكن إضافة اللغة المختلفة فى dll ومن ثم إضافتها , فيمكن إنشاء مكتبة أكواد (dll) بلغة vb وإضافتها إلى مشروع بلغة C# .
- فى Web Application يتم إخراج أكواد المشروع على هيئة ملف dll واحد إفتراضياً, ولكن فى Web Site لا يتم هذا ,ولكن يمكن عمل Pre-Compilation , وإن كان Pre-Compilation يتيح التعديل فى صفحات aspx أيضاً ,على خلاف ما يوجد فى Web Application وذلك بعد نشر الكود. ولكن لا يكون فى ملف dll واحد كما هو الحال فى Web Application . كذلك فى Web Application يمكن التحكم فى إضافة معلومات عن ملف dll المعروف بإسم Assembly وذلك من خلال الملف AssemblyInfo.cs الموجود ضمن مجلد Properties ضمن ملفات مشروعك ,ولا يمكن ذلك فى Web Site .
- فى Web Application يتم نشر أكواد الموقع على هيئة dll ولا يمكن التعديل عليها .ولكن فى Web Site يمكن نشر الموقع على هيئة Source Code , أى كما هى على بيئة التطوير (ملفات cs. ) مما يجعل التعديل عليها سهلاً .
- فى Web Application لا يمكن فتح الموقع من خلال Visual Studio وهو فى بيئة الإستضافة , أى بعد نشره , ولكن فى Web Site يمكن أن تقوم بفتح الموقع من خلال FTP والتعديل كيفما شئت .
- لا يمكنك إنشاء Web Application عبر IIS ولكن يمكنك نشره على IIS بعد إنشائه , بينما فى Web Site يمكن إنشاء موقعك عبر IIS ,مما لا يحتاج بعد ذلك إلى نشره على IIS , فهو فى الأصل موجود على IIS .
- فى Web Application يمكن الإشارة إلى الـ Classes المصاحبة للصفحات والموجودة بداخل Code-behind داخل أى Class مستقلة , بينما لايمكن ذلك فى Web Site .

# المجلدات الخاصة داخل ASP.Net

تحتوى ASP.Net على ثمانية مجلدات خاصة بها , فلا تستطيع إستخدام أحدهما إلا للغرض الذى وجد من أجله .فعلى سبيل المثال , تفاعلنا قبل ذلك مع المجلد App_Data وتعرفنا عليه من خلال إضافة قاعدة بيانات بداخله ,ورأينا كيف تعرفت عليه ASP.Net حينما كتبنا فى نص الإتصال فى ملف Web.Config مايلى :-

```
<connectionStrings>
     <add name="TestConnStr" connectionString="Data
     Source=.\SQLEXPRESS;AttachDbFileName=|DataDirectory|\TestDB.mdf;Integrated Security =true ;
     User Instance = true "/>
</connectionStrings>
```

سنفهم هذا الكود فيما بعد ولكن لاحظ وجود DataDirectory يتبعها إسم قاعدة البيانات ,ومع ذلك حين نقوم بتشغيل التطبيق , نرى أنه قام بالفعل بالإتصال بقاعدة البيانات المحددة , والتى توجد بالفعل داخل مجلد App_data , وذلك لإن App_Data يمثل Data Directory بالنسبة إلى ASP.Net , أى أنه يمثل مجلد البيانات للتطبيق .

مثال آخر , وهو مجلد App_Themes , وهو مجلد يختص بتنسيقات الموقع , حيث يمكن أن يحتوى هذا المجلد على ملفات CSS وSkin وكذلك الصور وكلها خاصة بمظهر الموقع الخارجى بما يمثله من تنسيقات , فيمكننا بإستخدام الخاصية Theme الموجودة ضمن Page Directive مثال التالى :-

```
<%@ Page Language="C#" AutoEventWireup="true" Theme="Default"  %>
```

كما ترى , قمنا بتحديد إسم Theme وهو Default , وهو مجلد تم وضعه داخل App_Themes وبه كافة التنسيقات المراده . وسنعرف كيف يتم هذا فى وقته , ولكن ما أريد أن أذكره هنا ,هو ,كيف تعرفت الصفحة على مجلد Default الموجود داخل App_Themes؟ نحن هنا قد إستخدمنا الخاصية Theme والتى بدورها تتفاعل تلقائياً مع مجلد App_Themes الذى نحن بصدد الحديث عنه , وهذا ما أعنيه بكلمة "مجلدات خاصة" , أى أن ASP.Net تتعرف عليها تلقائياً وتستخدمها فى آداء كثير من المهما .

وإليك جدول يوضح المجلدات الخاصة التى نحن بصدد الحديث عنها :-

| إسم المجلد | إستخدامه |
|---|---|
| Bin | للإحتفاظ بمكتبات الأكواد والمعروفة بإسم Assemblies (ملفات DLL ). |
| App_Data | للإحتفاظ بقواعد البيانات . |
| App_Themes | للإحتفاظ بتنسيقات الموقع مثال CSS, Skin, Images |
| App_Code | للإحتفاظ بالأكواد العامة لكامل التطبيق مثال إنشاء Class تراها كل الصفحات .وذلك فى حالة العمل مع WebSite وليس WebApplication.وسنعرف الفرق بينهم فيما بعد إن شاء الله |
| App_LocalResources | للإحتفاظ بملفات تعدد اللغات للصفحة الواحدة . |
| App_GlobalResources | للإحتفاظ بملفات تعدد اللغات لكامل صفحات الموقع , وهذة المجلد والمجلد السابق يتم إستخدامها فى وجود صفحة أو أكثر تعمل بأكثر من لغة , فيتم إنشاء ملف Resource.resx لكل لغة |

| | |
|---|---|
| | وسيأتى إن شاء الله تفصيل هذة النقطة فى درس قادم . |
| App_Browser | للإحتفاظ بملفات بإمتداد browser. وهى لإعدادات المتصفحات |
| App_WebReference | يستخدم عند الإشارة إلى Web Service,وهذة العملية بمثابة إنشاء قناة إتصال بين التطبيق وبين الخدمة (WebService). |

## إضافة أكواد إلى الصفحة

إذا تحدثنا عن الأكواد فإننا بصدد الحديث عن أمرين , الآول , أكواد تعمل على جهاز المستخدم وتسمى Client-Side Code ,والثانى, أكواد تعمل على جهاز الخادم وتسمى Server-Side Code . فالآولى , يتم كتابتها بإستخدام لغة JavaScript ,والثانية لديك خيارين إما C# أو Vb.net .

### Server-Side Code

**الطريقة الآولى :-**

فقد تحدثنا منذ قليل عن ملف Code-behind, وهو ملف ظاهره أنه منفصل ولكن فى الحقيقة لا ,ولتعرف لماذا , فعليك معرفة ما يدور فى عملية compilation ,وبالفعل أصبح لدينا علماً بإنه يتم إستخدامه لكتابة أكواد Server-Side Code الخاص بالصفحة .

وتستطيع أن تحدد هذا الملف لصفحة aspx من خلال الخاصية Codebehind كما ترى :-

```
<%@  Page Language="C#" AutoEventWireup="true" CodeBehind="Home.aspx.cs" Inherits="Class1_Part2.Home" %>
```

**الطريقة الثانية:-**

وهى إستخدام Script Tag داخل صفحة aspx بحيث يكون متداخلاً مع أكواد Html على مثل هذا الشكل ولاحظ أنه يحمل الخاصية runat وهذا شرطاً أساسياً فى عمله:-

```
<%@ Page Language="C#" AutoEventWireup="true" %>

<script runat="server">
    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Write("Hello World!");
    }
</script>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Button ID="Button1" runat="server" Text="Button" OnClick="Button1_Click" />
    </div>
    </form>
</body>
</html>
```

كما ترى أمامك فى الكود , لاحظ غياب الخاصية Codebehind والخاصية Inherits , وذلك لعدم وجود حاجة لهم هنا , كذلك لاحظ الجزء المظلل بالأعلى , فإنه ما نتحدث عنه ,إنه Script Tag الذى يحمل الخاصية runat,حيث تجد بداخله أحد أحداث الآداة Button وهو الحدث Click ,وكذلك يمكن وضع جميع أكوادك بداخله ولكن إذا كان الكود الذى ستكتبه بداخله كبير وبه مهام كثيرة , فالأفضل أن تفصل بوضعه داخل Codebehind .

### الطريقة الثالثة :-

وهى أن تستخدم Server Tags كالتى رأيتها فى الدرس السابق , ولكن لا تكثر من إستخدام هذا الطريقة , ولذلك لصعوبة صيانة الكود فيما بعد, وإليك مثال على هذة الطريقة :-

```
<div>
    <% for (int i = 6; i > 0; i--)
       {
    %>
    <h<%=i %>> Hello World! </h<%=i %>>
    <% } %>
</div>
```

ليظهر أمامك الشكل التالى :-

Hello World !

Hello World !

Hello World !

Hello World !

Hello World !

Hello World !

## Client-Side Code

يمكن كتابة أكواد JavaScript بأكثر من طريقة , بل إنى أذكر يوماً إختبرت نفسى فى معرفة أكبر عدد من الطرق التى يمكن بها إضافة أكواد JavaScript إلى صفحة aspx , فحينها رحت أجرب وأجرب حتى وصلت إلى 10 طرق ,ولكن سأسرد لك الطرق الأساسية :-

### الطريقة الأولى :-

وهى الأفضل لا شك ,وهى أن تستخدم ملفاً منفصلاً لتضع به أكواد JavaScript , ومن ثم تشير إليه داخل صفحة aspx. وسيكون هذا الملف بإمتداد js إختصار لـ JavaScript. وبالطبع يمكن أن تقوم بإنشاء هذا الملف من خلال Add New Item .

فيما يلى صورة لملف js بعد إنشائه ,وبداخله دالة SayHello والتى تقوم بإخراج رسالة ترحيب للمستخدم.

TestJavaScript.js

```
function SayHello() {
    alert('Hello World!');
}
```

بعد ذلك تستطيع أن تشير إلى هذا الملف بداخل صفحة aspx كما يلى :-

```
<script src="TestJavaScript.js" type="text/javascript"></script>
```

ينتج هذا السطر من الكود عند سحب ملف js وإدراجه بداخل الصفحة , ولكن أين ستدرجه ؟ يمكنك أن تضعه بداخل العنصر Head , ولكن من الآ فضل أن تضعه أسفل الصفحة وسنعرف لماذا حين نصل إلى الدرس الخاص بـ Scripts .

**الطريقة الثانية :-**

أن نستخدم Script Tag ولكن هذة المرة بشكلاً مختلفاً ,حيث أنه لن يحتوى على runat, وفيما يلى شكلاً لإستخدام Script Tag :-

```
<script type="text/javascript">
    function ValidateUser(obj) {
        //do something with object
        if (obj) {
            return true;
        }
        return false;
    }
</script>
```

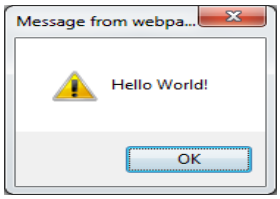
لاحظ الخاصية type والتى من خلالها تم تحديد أن اللغة المستخدم هى javascript.وتستطيع أن تضعه أيضاً داخل العنصر Head أو كما قلنا الآفضل أن تضعه أسفل الصفحة .

**الطريقة الثالثة :-**

ليست كثيرة الإستخدام ولكن قد تحتاج إليها يوماً ما , وهى أن تضع كود JavaScript بجانب الحدث الذى سيتسبب فى تنفيذ هذا الكود على مثل الشكل التالى :-

```
<input type="button" value="Say Hello" onclick="javascript:alert('Hello World!')" />
```

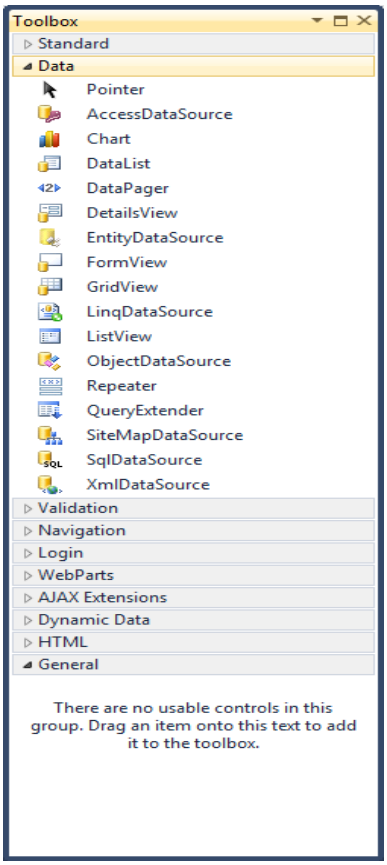
ليكون الناتج تلك الرسالة :-




كذلك يمكنك التحكم فى هذة الآكواد من أكواد C#/vb.net ,أى من خلال Server-Side Code

فيما بعد إن شاء الله سنتعرف كيف نتفاعل مع أكواد JavaScript من خلال أكواد Server-Side , وذلك سيكون فى الدرس الخاص بـ Scripts إن شاء الله .

# نافذة الأدوات Toolbox




تنقسم الأدوات إلى مجموعات كما ترى فى الصورة :-

| الأدوات | وظيفتها |
|---|---|
| **Standard** | هى الأدوات الأساسية لتصميم صفحة مثل Buttonأو Text Box. وغيرهم . |
| **Data** | هى أدوات التعامل مع قواعد البيانات من عرض وتحرير . |
| **Validation** | هى أدوات التحقق من صحة المدخلات . |
| **Navigation** | هى أدوات التنقل فى الموقع مثل القائمة وغيرها. |
| **Login** | هى أدوات التسجيل فى موقعك كإنشاء عضوية وتسجيل دخول وما شابه . |
| **Web Parts** | هى أدوات تمكنك من صنع صفحة موقع مقسمة لإجزاء تتشكل حسب ما يريد كل مستخدم على حده , مثال ذلك my.yahoo.com |
| **Ajax Extensions** | أداوت تعمل بمساعدة من Ajax لأداء بعض المهام . |
| **Dynamic Data** | أداوت تعمل بمساعد أدواتData مثال GridView و Details View وغيرهم تسمح بتفاعل جيد بين هذة الآدوات وبين Database Model مثال Entity Framework و LINQ to SQL بما فى ذلك دعم للتفاعل مع Data |

Anontations وسيأتى تفصيل ذلك فى مكانة . ولكن لتعرف الآن أنها تساعد فى تيسير العمل مع عمليات التفاعل مع قواعد البيانات .

**HTML** هى أدوات Html مثال input , Div , Table .

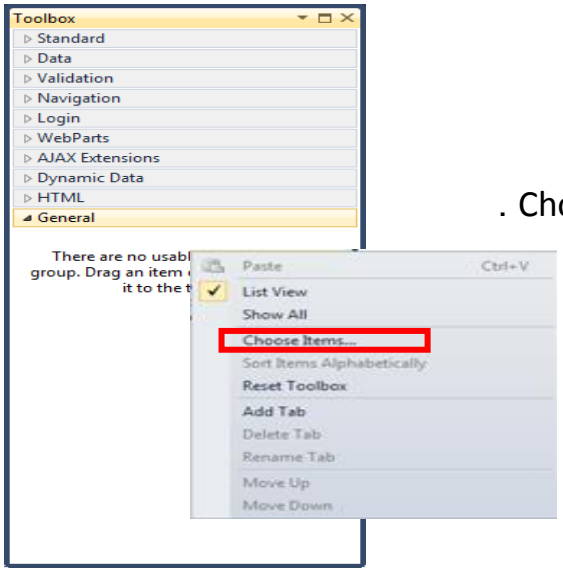
**General** يمكن أن تضع بهذا المكان بعضاً من الأدوات التى تريد .

إن شاء الله سيأتى تفصيلاً لكل مجموعة أدوات على حده .

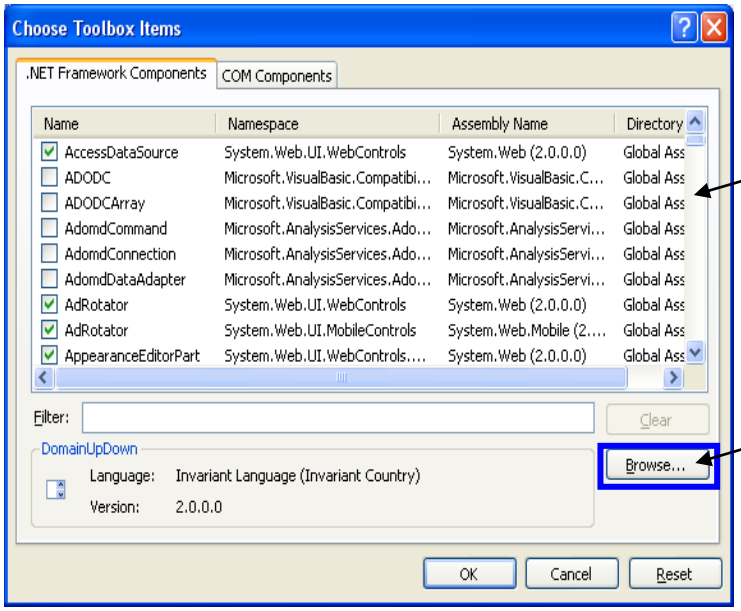
## كيف تضيف أدوات أخرى ؟

إليك الخطوات لفعل هذا :-

1 -قف على General .
2 -إضغط بيمين الماوس (R.C)عليها .
3 -تظهر لك قائمة إختر منها Choose Item .




4 - تظهر لك هذة النافذة :-




**إختر من هنا الأداة التى تريد ثم ضع علامة أمامها ثم إضغط Ok**

**وبعدها ستجد الأداة التى حددتها موجود أسفل General**

**أويمكن لك أن تأتى بأداة من ملف خارجى من صنعك . وسنستخدم هذا الأمر عند إضافة أداوات AjaxControlToolKits**

**والآن لنتعرف على ما تحتويه نافذة الآدوات :-**

# Label

يستخدم لعرض بيانات للمستخدم يكون عند إدراجة او كتابته فى الكود هكذا :-

```
<asp:Label ID="Label1" runat="server" Text="Label"/>
```

تحليل هذا الكود

**<asp:Label**

بداية تعريف أمر Label ويكتب على الشكل الذى ترى حيث asp تسمى Tag Prefix أى علامة بادئة , وLabel تسمى Tag Name .

**ID="Label1"**

إعطاء هذا الـ Label إسم برمجياً يمكن أن نتعامل معه من خلال كود C# أو Vb.net

**runat="server"**

وذلك لإخبار البرنامج أن هذة الأداة Server Control أى تعمل على الـ Server وعليه تنفذ أكوادها على الـ Server .

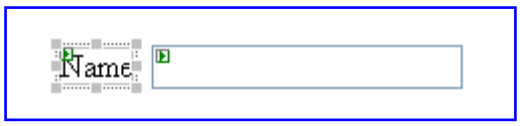
**Text="Label"**

وذلك لتعريف نص يعرض فى الأداة يراه المستخدم .

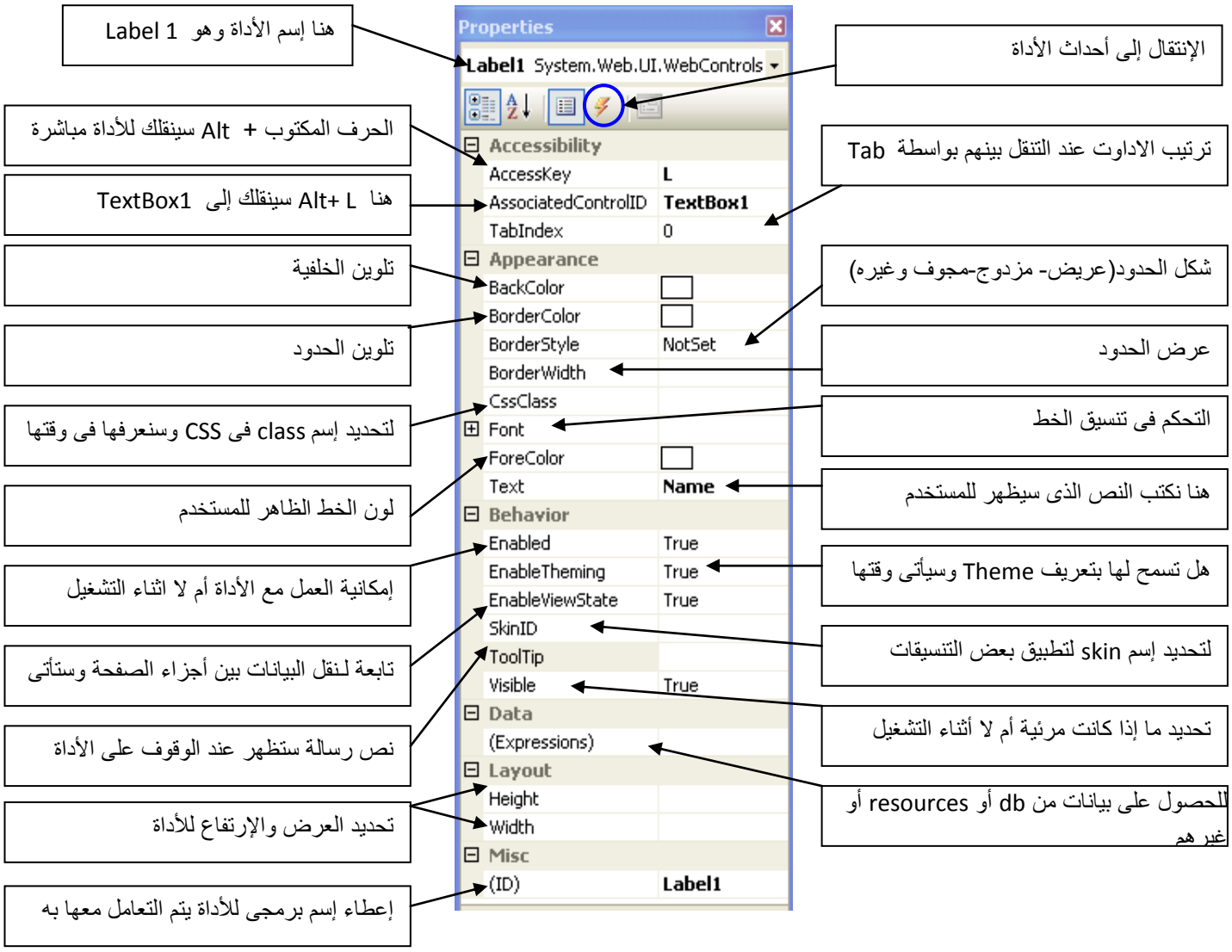
**/>**

و ذلك لغلق الأمر . فلابد أن لكل أمر يفتح أن يغلق . بهذا الشكل الذى رأيته .

## خصائص الـ Label

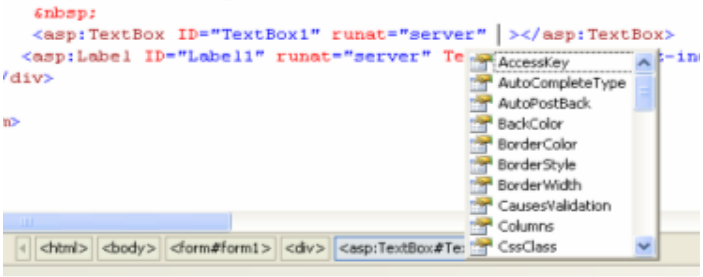



- فى هذة الصورة تجد أداتين هما Label , Text Box .
- عند الوقوف على Label فى وضع التصميم نذهب لنافذة الخصائص , كذلك يمكن أن تقف على أى أداة وتضغط Alt + Enter لتذهب لنافذة الخصائص أيضاً .
- نرى هذة النافذة التى أمامك بها الخصائص التى يمكن أن نقوم بتعديل كل خصائص الـ Label .




لماذا تم شرح خصائص الآداة Label ؟ ____ فى الحقيقة ليست الآداة Label هى المقصودة بعينها , ولكن لإن الخصائص للأداة الواحدة تتكرر فى بقية الآدوات , فلذلك تم إختيار آداه لتقديم شرحاً لخصائصها , وبالتالى فهى خصائص متكررة مع جميع الآدوات ولا حاجة لشرحها مع كل آداة .

كذلك يمكن لك ضبط هذة الخصائص من خلال كود التصميم هكذا :-




كما يمكن أن تغير الخصائص فى ملف كود C# هكذا :-

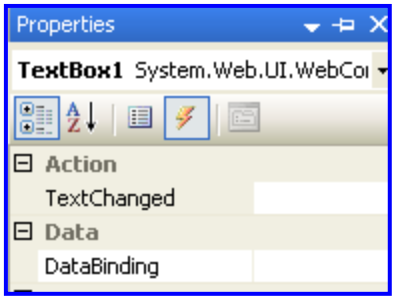
```
protected void Page_Load(object sender, EventArgs e)
   {
    Label1.Text = "First Name";
    Label1.BackColor = System.Drawing.Color.Gray ;
   }
```

## Text Box

وتستخدم لإخراج وإدخال بيانات من وإلى المستخدم . وتشترك كما أشرنا مع باقى الأدوات فى الكثير من الخصائص , أما الجديد فى الخصائص الخاصة بـ Text Box هى كالتالى :-

| الخاصية | وظيفتها |
|---|---|
| AutoCompleteType | التكملة التلقائية كالتى تراها فى Google . تكتب كلمة فيأتى لك بالإقتراحات . فإنك عادة ما تجد خانة تحتفظ بالبريد الإلكترونى فقط (كمثال). وذلك لأنك تحدد قيمة هذة الخاصية لهذا الـ Text Box بـ Email . كما يوجد غيرها الكثير المتاح ضبطه, ولاحظ أن الآمر متعلقاً بالمتصفح فى حفظ مثل هذة البيانات .<br><br>ولكن منها مايأتى من قاعدة بيانات أو ملف Xml أو حتى من الذاكرة ولكن فى هذة الحالة يجب أن نستعين بأحد أدوات Ajax أوJQuery UI . |
| AutoPostBack | تمكننا هذة الخاصية من إعادة تحميل الصفحة بأمر جديد لتغيير لبيانات ,أى القيام بإرسال HttpPost إلى الخادم للتحديث . |
| CousesValidation | تقرر هذة الخاصية ما إذا كانت الأداة ستقوم بإجراء التحقق من صحة البيانات المدخلة أم لا . ( أوامر التحقق من صحة البيانات لم تأتى معنا بعد .سنراها فى القريب العاجل إن شاء الله ) |
| MaxLength | تحدد عدد الأحرف المتاح كتابتها داخل هذة الأداة وتحدد بالأرقام |
| ReadOnly | تجعل الأداة للقراءة فقط . أى لا تقبل التعديل أو الكتابة بداخلها . |
| Rows | تحدد عدد الصفوف التى ستعرض فى الأداة وليست التى ستكتب .وتعمل عندما مع الخاصية TextMode |
| TextMode | تحديد نوع الـText Box هل هو متعدد الأسطر أم أنه سيحتوى على سطر واحد أم أنه على شكل Password . وهما كا لأتى ( Single Line – Multiline – Password) |
| ValidationGroup | وتساعد هذة الخاصية على ضم الأداة إلى مجموعة ادوات أخرى يتم التحقق من صحة بياناتهم مع بعضهم البعض . وذلك يتم من خلال إعطاء إسم لمجموعة تتبعها هذة الأدوات . |
| Warp | تعمل على إلتفاف النص داخل الأداة . وتقوم بإخفاء شريط التمرير الأفقى فى الأداة وهذا معنى إلتفاف . |

### أهم الأحداث الخاصة بها :-




### TextChanged

وتحدد ماذا سيحدث عند تغير النص داخل الأداة . ومن اجل أن تصل إلى هذا الحدث قف فوقه وإضغط مرتين ينقلك تلقائياً إلى ملف الكود هكذا

```
protected void TextBox1_TextChanged(object sender, EventArgs e)
 {
  // ضع هنا الكود الذى تريد تطبيقه عند تغير النص
 }
```

ترى أيضاً أن هناك أحداث آخرى مثال , Data Binding ,Init, Load , PreRender Unload, Disposed,فسنراها إن شاء الله فى الحديث عن دورة حياة الصفحة والمعروفة بإسم Page Life Cycle , وذلك لتكرارها مع جميع الآدوات وإرتباطها بدورة حياة الصفحة والله المستعان .

## Button

جديد الخصائص هنا كالتالى :-

### OnClientClick

تستخدم هذة الخاصية عند الحاجة إلى أن ينفذ هذا الـ Button والذى هو Server Control كود javaSceript والذى Client Code .

مثال ذلك :-

```
<asp:Button ID="Button1" runat="server" Text="Show"
            OnClientClick ="javascript:alert('اهلا بك معنا ');" />
```

وعند الشغيل والضغط على الـ Button تخرج لك هذة الرسالة

### PostBackUrl

وهى لنقل المستخدم لمسارجديد (URL ). مثال ذلك نقل المستخدم لصفحة اخرى

```
<asp:Button ID="Button1" runat="server" Text="Button"
                PostBackUrl ="~/default2.aspx" />
```

فعند الضغط على هذا الـ Button يتم نقل المستخدم إلى الصفحة default2.aspx كما فى المثال .

## LinkButton

يشبه كثيرا الـ Button العادى فى خواصة وأوامره ولكنه على شكل Link. ويتم كتابة أمر الإنتقال فى خاصية PostBackUrl هكذا :-

```
<asp:LinkButton ID="LinkButton1" runat="server"
    PostBackUrl="http://www.asp.net/learn" > المرجع للـ ASP.Net
</asp:LinkButton>
```

يظهر فى الصفحة على هذا الشكل وعند الضغط ينقلك للموقع المشار إليه فى PostBackUrl

المرجع للـ ASP.Net

## ImageButton

تشبه أيضاً Button ولكنها على شكل صورة وبها أيضاً خصائص الـ Button .

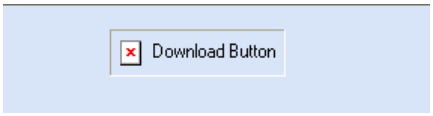
```
<asp:ImageButton ID="ImageButton1" runat="server"
     ImageUrl="~/icontexto-webdev-arrow-down-048x048.png"
     AlternateText ="Download Button
     PostBackUrl="~/download.aspx?id=10"  />
```

**الخصائص :-**

| الخاصية | وظيفتها |
|---|---|
| **Imageurl** | تحدد الصورة التى ستكون على هيئتها الأداة . |
| **AlternateText** | النص داخل الصورة إذا لم تظهر الصورة وهو ضرورى جداً تفادياً للمشاكل. |
| **PostBackUrl** | رابط هذة الصورة .أى المسار التى تؤدى إليه وفى المثال تؤدى إلى صفحة Download . |

فعند التشغيل نجد الصور على هذا الشكل

وإذا كان هناك مشكلة فى عرض الصورة فستكون على الشكل التالى




لاحظ النص المكتوب داخل الأداة هو النص المكتوب فى خاصية AlternateText .

## Hyperlink

نستخدمها لعمل رابط فى الصفحة مثال ذلك

```
<asp:HyperLink ID="HyperLink1" runat="server"
                               NavigateUrl="http://www.google.com"
                               Target="_blank" > Google
</asp:HyperLink>
```

**الخصائص :-**

| الخاصية | وظيفتها |
|---|---|
| **NavigateUrl** | لكتابة عنوان الرابط . (الرابط هو المسار أو العنوان الذى سننتقل إليه ) |
| **Target** | نحدد طريقة فتح الصفحة . هنا ستفتح الرابط فى صفحة جديدة . |

## Image

لوضع صورة فى الصفحة وتكتب على هذا الشكل

```
<asp:Image ID="Image1" runat="server"
                       ImageUrl ="~/Images/photo.gif"
                       AlternateText ="Photo Text" />
```

## DropDownList

هى قائمة تحتوى على عناصر تتيح للمستخدم ان يختار منها ما يريد . ثم ينتج بعد إختياره حدث معين . و يمكن ملئ هذة القائمة بعدة طرق منها , والهدف من سرد هذة الطرق هو توسيع مدارك ليس أكثر :-

### 1 - يدوياً

```
<asp:DropDownList ID="DropDownList1" runat="server" >
   <asp:ListItem >المنتج إختار</asp:ListItem>
   <asp:ListItem>Hard Disk</asp:ListItem>
   <asp:ListItem>Scanner </asp:ListItem>
   <asp:ListItem>Camera </asp:ListItem>
</asp:DropDownList>
```

### 2 - بإستخدام الخاصية Items

```
<asp:DropDownList ID="DropDownList2" runat="server">
</asp:DropDownList>
```

وبعد ذلك ملئها بالكود , على سبيل المثال : تكرار رقم فيحفظ بداخلها

```
for (int i = 0; i < 5; i++)
 {
    DropDownList2.Items.Add(i.ToString ());
 }
```

فقد تحتاج يوماً أن تقوم بالدوران داخل بيانات قادمة من قاعدة بيانات أو ملف xml لعرضها داخل هذة الآداة .

### 3 - بإستخدام مصدر بيانات مع الخاصية DataSource

```
<asp:DropDownList ID="DropDownList3" runat="server">
</asp:DropDownList>
```

وملئها كالتالى

```
string[] drop = new string[] { "ASP.Net", "Csharp", "JavaScript" };
DropDownList3.DataSource = drop;
DropDownList3.DataBind();
```

فى الكود السابق إسندنا قيمة مصدر البيانات إلى Array , ووظيفة DataBind( ) هنا هو إعطاء الأمر ببداية ربط الآداة بمصدر البيانات سواءً كان من قاعدة بيانات أو غيرها .

### 4 - بإستخدام ملف XML والخاصية DataSource

أولاً :- ننشأ ملف XML ونسميه XMLFile.xml ليكون هكذا

```xml
<?xml version="1.0" encoding="utf-8" ?>
<Links>
  <Link>
    <PageName>ASP.Net</PageName>
    <URL>~/asp.aspx</URL>
  </Link>
  <Link>
    <PageName>C#</PageName >
    <URL>~/CSharp.aspx</URL>
  </Link>
  <Link>
    <PageName>JavaScript</PageName >
    <URL>~/javascript.aspx</URL>
  </Link>
  <Link>
    <PageName>Data Base</PageName >
    <URL>~/Database.aspx</URL>
  </Link>
</Links>
```

بعد أن أنشأنا الملف السابق ندرج أداة DropDownList فى الصفحة , ثم ملئها بالكود هكذا :-

```
DataSet ds = new DataSet();
ds.ReadXml(Server.MapPath("XMLFile.xml"));
DropDownList3 .DataSource = ds;
DropDownList3.DataTextField = "PageName";
DropDownList3.DataBind();
```

تحليل الكود السابق

1 -أنشأنا Object من الـ DataSet .

2 -قمنا بإستخدام الدالة ReadXML لقراءة ملف XML ووضعه بداخل DataSet .

3 -أسندنا قيمة مصدر البيانات إلى ds وهى نسخة مأخوذة من DataSet تعمل بعملها .

4 -تخصيص قيمة العنصر PageName الموجود فى ملف XMl وجعله هو نص القائمة .

5 -إعطاء أمر بربط الآداة بمصدر البيانات من خلال () DataBind ليكون الناتج هكذا .

**ما هى فائدة الـ DataSet ؟**

تعمل كمصدر لتخزين البيانات فى الذاكرة , وظيفتها تجميع البيانات فى جداول يتكون الواحد منهم من صفوف وأعمدة .

## تمرين بسيط

ما رأيك أن نستخدم ملف XML السابق فى عمل قائمة نتصفح بها الموقع , المعنى أننا حينما نختار العنوان من القائمة يذهب بنا إلى الصفحة المختارة , الهدف من هذا التمرين هو التفاعل أكثر مع DropDownList وليس إنشاء القائمة لذاتها .

يحتوى الملف السابق على عنصرين هما PageName و URL ونريد أن نعرض فى القائمة إسم الصفحة فقط (PageName) وعند الضغط ينقلنا إلى مسار (URL)

**خطوات العمل :-**

1 -قم بإنشاء أربع صفحات بالأسماء الموجودة وأكتب فى كلاً منها شئ حتى تعرفها حين تفتح أمامك.

2 -قم بتفعيل الخاصية AutoPostBack الخاصة بتنشيط الصفحة فى كل إختيار .

**إليك كود التصميم**

```
<asp:DropDownList ID="DropDownList5" runat="server"
                                  AutoPostBack ="true" >
</asp:DropDownList>
```

**كود C#**

```
protected void Page_Load(object sender, EventArgs e)
 {
    if (!IsPostBack)
    {
        DataSet ds = new DataSet();
        ds.ReadXml(Server.MapPath("XMLFile.xml"));
        DropDownList3 .DataSource =  ds;
        DropDownList3.DataTextField = "PageName";
        DropDownList3.DataValueField = "URL";
        DropDownList3.DataBind();
        DropDownList3.Items.Insert(0, "Choose");
    }
```

**تحليل الكود السابق :-**

1 -كما ترى فى الكود , فإنه تم وضعه بداخل شرطاً وهو IsPostBack !, ومعنى علامة التعجب هنا (Not , أى " ليست " ) والهدف من هذا الشرط , أن يتم تحميل وملئ الآداة DropDownList فقط فى المرة الآولى التى يتم فيها تحميل الصفحة وليس مع كل مرة يحدث فيها Postback مما يترتب عليه تنفيذ حدث Page_Load .

2 -المشكلة الثانية أننا نريد ان نضع كلمة Choose قبل عناصر القائمة . فكان الحل إدراج عنصر إسمه Choose من خلال هذا الأمر

```
DropDownList3.Items.Insert(0, "Choose");
```

ومعناه إدراج العنصر فى المكان الأول للقائمة وهو 0 . ولاحظ وضعه فى نهاية الكود حتى يتم تنفيذة فى النهاية لتجنب الأعطال .

3 -ثم قم بالضغط مرتين على الأداة لنكتب كود التنقل بين الصفحات هكذا

```
protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
  {
      Response.Redirect(DropDownList3.SelectedValue);
  }
```

هنا نخبره أن ينتقل إلى قيمة العنصر المختار وهى Value المأخوذة من قيمة URL الموجود فى ملف الـ XML .

## ListBox

تستخدم هذة الأداة لعرض البيانات أيضاً ولكن تختلف عن DropDownList فى أنها تعرض أكثر من عنصر فى نفس الوقت للمستخدم كما يمكن إختيار أكثر من عنصر فى المرة الواحدة.

```
<asp:ListBox ID="ListBox1" runat="server" SelectionMode="Multiple" >
   <asp:ListItem >Mohamad</asp:ListItem>
    <asp:ListItem > Ali </asp:ListItem>
    <asp:ListItem> Omr  </asp:ListItem>
    <asp:ListItem >Abd Allah </asp:ListItem>
</asp:ListBox>
```

يتم التفاعل معها بنفس طريقة التفاعل مع آداة DropDownList , من إستخدام خصائص DataSource و DataBind وDataTextField وDataValueField , وغيرهم .

## Checkbox Tool

أداة تتيح للمستخدم ان يختار أمر معين بوضع علامة أمامه . هكذا

```
<asp:CheckBox ID="CheckBox1" runat="server" AutoPostBack="true" Text="I Agree " />
```

**أهم خصائصها :-**

| الخاصية | وظيفتها |
|---|---|
| **AutoPostBack** | عند تفعيلها , فإننا بذلك نسمح للآداة أن تقوم بعمل Postback أى عودة للخادم وذلك عند قيام المستخدم بالتفاعل مع الآداة بالتحديد أو إلغاء التحديد . |
| **Text Align** | محاذاة النص لليمين أو لليسار |
| **Checked** | نستطيع من خلالها , أن نكشف إذا ما قام المستخدم بالتحديد أم لا . |

**الأحداث :-**

**Checked Changed :-**

يتم تنفيذ هذا الحدث عند تفاعل المستخدم مع آداة Checkbox, سواء بالتحديد أو بإلغاء التحديد.

## CheckBoxList

هى مجموعة من آداوات Checkbox تم تجميعهم فى شكل قائمة لتقوم بالإختيار منها .

```
<asp:CheckBoxList ID="CheckBoxList2" runat="server" >
  <asp:ListItem>One </asp:ListItem>
  <asp:ListItem>Two</asp:ListItem>
  <asp:ListItem>Three</asp:ListItem>
</asp:CheckBoxList>
```

**الخصائص**

| الخاصية | وظيفتها |
|---|---|
| AutoPostBack | تنشيط الصفحة بالإختيار الجديد |
| RepeatColumns | لعرض عناصر القائمة فى أكثر من عمود |
| ReapeatDirection | لعرض القائمة أفقى او رأسى حسب الإختيار |

**الأحداث**

**SelectedIndexChanged**

تستخدم لتنفيذ كود عند التنقل بين عناصر القائمة وتستخدم بعد تنشيط إختيار AutoPostBack إلى True .

## RadioButton

يستطيع المستخدم أن يقوم بتحديد أمر ما من خلالها .

```
<asp:RadioButton ID="RadioButton1" runat="server" Text="Yes"  />
<asp:RadioButton ID="RadioButton2" runat="server"  Text="No" />
```

**الخصائص :-**

| الخاصية | وظيفتها |
|---|---|
| AutoPostBack | تنشيط الصفحة بالإختيار الجديد |
| GroupName | تجميع أكثر من عنصر فى مجموعة واحدة ليتيح للمستخدم أن يختار إختيار واحد فقط من هذة المجموعة . |
| Checked | للكشف هل تم الإختيار أم لا , والناتج يكون True , False . |

**الاحداث**

**CheckedChange**

نستخدمه عندتنفيذ كود عند الإختيار, للتفاعل مع الخادم طبقاً لإختيار المستخدم سواءً كانوا بالتحديد أو إلغاء التحديد.

## RadioButtonList

أداة تحدد مجموعة إختيارات للمستخدم يجب أن يختار واحد فقط منهم .تشبه CheckBoxList ولكن هنا لابد من إختيار واحداً على الأكثر ,ويعد الإختلاف فى أن هذة الأداة تتيح إختيار عنصر واحد من مجموعة عناصر .

```
<asp:RadioButtonList ID="RadioButtonList1" runat="server"  >
  <asp:ListItem>الصف الأول</asp:ListItem>
  <asp:ListItem>الصف الثانى</asp:ListItem>
  <asp:ListItem>الصف الثالث</asp:ListItem>
  <asp:ListItem>الصف الرابع</asp:ListItem>
</asp:RadioButtonList>
```

**الأحداث**

**SelectedIndexChanged**

وذلك لتنفيذ بعض المهام عند تحديد المستخدم أحد عناصر القائمة .

**كود C#**

```
protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs e)
 {
   Response.Write ("هو إختيارك  : " + RadioButtonList1.SelectedItem.Text  );
 }
```

## BulltedList

```
<asp:BulletedList ID="BulletedList1" runat="server" BulletImageUrl="~/Icon1.png"
                  BulletStyle="CustomImage"  DisplayMode="HyperLink" >
     <asp:ListItem Value ="http://www.msdn.com" > Microsoft </asp:ListItem>
     <asp:ListItem Value ="http://www.asp.net" > ASP.Net </asp:ListItem>
     <asp:ListItem Value ="http://www.vb4arab.com" >VB4arab </asp:ListItem>
</asp:BulletedList>
```

هى أداة لعرض عناصر قائمة على أشكال مختلفة, منها مثلاً شكل عادى هكذا قائمة عرض فقط

---

- Microsoft
- ASP.Net
- VB4arab

كما يمكن تغير العلامة التى امام كل عنصر وجعلها على هيئة صورة مثلاً :-

ومنها ما يكون على شكل أرقام مسلسلة

1. Microsoft
2. ASP.Net
3. VB4arab

كما يمكن بداية الترقيم من رقم معين, وكذلك بدلاً من الترقيم يمكن أن تكون حروف وعلامات اخرى .

أهم الخواص الجديدة التى تنفذ ما سبق :-

| الخاصية | وظيفتها |
|---|---|
| BulletImageUrl | تحديد مسار الصورة التى ستكون أمام العنصر . |
| BulletStyle | تحديد شكل العلامة إما أرقام او حروف او علامات. |
| FirstBulletNumber | تحديد رقم أول عنصر . |
| DisplayMode | تحديد أن تكون العناصر عرض فقط او على شكل رابط . |

**لاحظ أن :-**

إذا قمت بتحديد صورة فى BulletImageUrl ولم تحدد BulletStyle بـ CustomImage لن تعرض الصورة. وإذا قمت بإستخدام FirstBulletNumber ولم تحدد BulletStyle بـ Numbers لن يحدث شئ .

بعد أن تعرفت على بعضاً من الآدوات الأساسية وإن شاء الله سنتعرف على البقية فى الدروس القادمة , ننتقل الإن إلى تمرين عملى , سنرى فيه كيف نقوم بإنشاء تطبيق عبر IIS .

## تمرين عملى

## إنشاء Web Site عبر IIS

(هذا التمرين تم تسجيله فيديو أيضاً )

فى هذا التمرين سنقوم بإنشاء تطبيق عبر IIS , ولنرى معا الخطوات :-

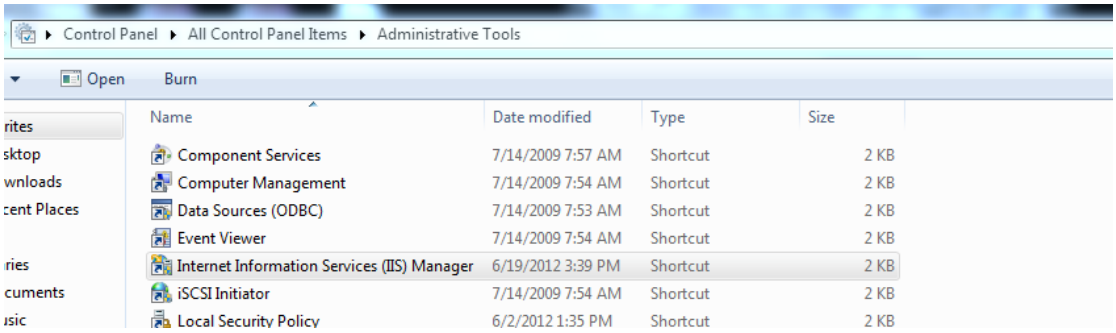
**أولاً :- التأكد من وجود IIS**

هل IIS مثبت على جهازك ؟ إن لم يكن كذلك , فإتبع الخطوات التالية وستعرف إن شاء الله إن كان مثبتاً لديك أم لا .

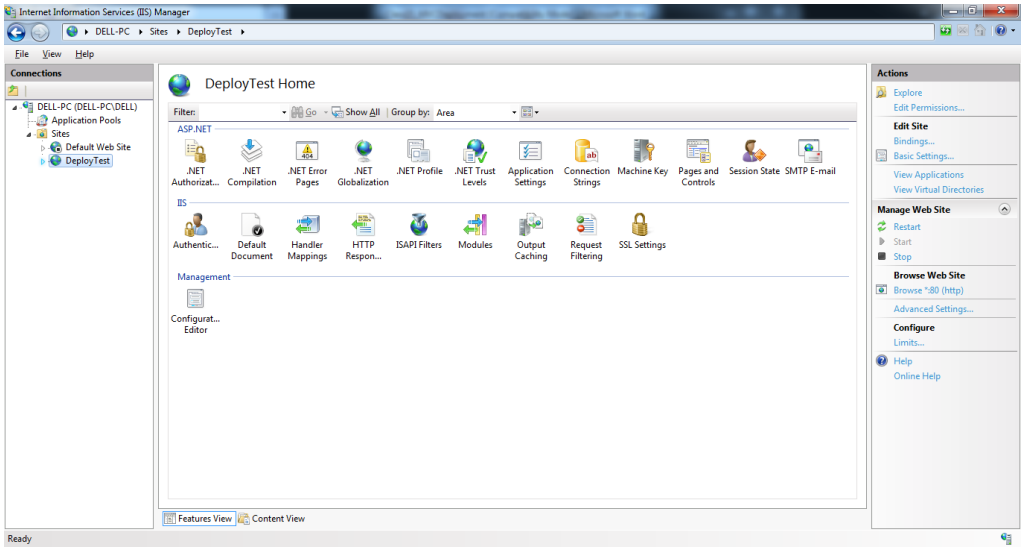
## كيف تقوم بتثبيت IIS على جهازك

سنمر بخطوات والإفضل لنا أن نراها فى ترتيب مرقم كما سيلى إن شاء الله , ولاحظ أن العمل يتم فى Windows 7 :-

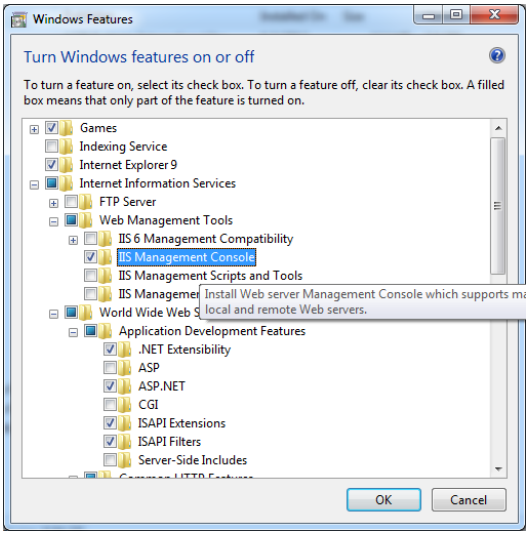
1 - التأكد من تثبيت IIS أولاً ,حيث يمكن أن تجده ضمن Administrative Tool بداخل Control Panel :-




بالطبع إن وجد Internet Information Services (IIS) Manager , فلديك بالفعل IIS , ويمكن فتحه لترى النافذة التالية :-




ولكن إن لم يكن لديك IIS , ماذا تفعل ؟ _____ يمكن تفعيل IIS من خلال لوحة التحكم Control Panel , فمن خلال الذهاب إلى Program and Features , ومن ثم إختيار Turns Windows Features On or Off لتظهر أمامك النافذة التالية :-



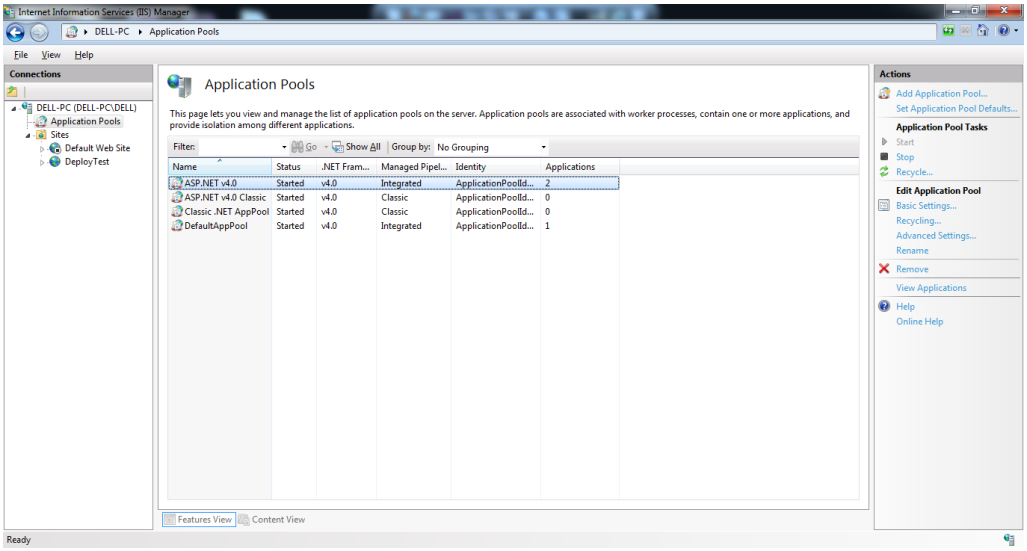

بعد تحديد ما تراه أمامك , والضغط على Ok , سيكون لديك IIS , قد تم تثبيته على جهازك .

بعد ذلك يمكنك فتح نافذة IIS Manager للتفاعل معه ,كما ترى فى الخطوة الآولى .

2 -حيث أننا نعمل على إصدارات 2010 فتأكد من أن لديك ASP.Net 4 قد تم تثبيته بداخل Application Pools كما بالشكل التالى , ولكن ما هو Application Pool ؟

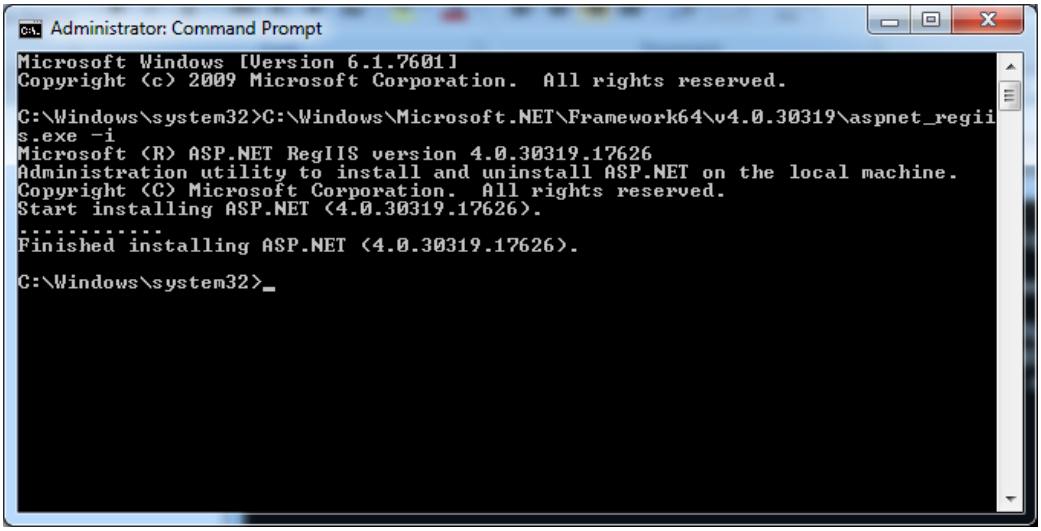
## Application Pool

عبارة عن حاجز يفصل بين التطبيقات وبعضها, وذلك لتوفير عزلة للتطبيق الواحد عن بقية التطبيقات حتى لا يتأثر بما يحدث فى التطبيقات الآخرى, فقد يحدث خطأ فى تطبيق ما, فهل سيؤثر فى بقية التطبيقات وهم على نفس الخادم ؟, فهنا تظهر الحاجة إلى Application Pool والعزلة التى نتحدث عنها .



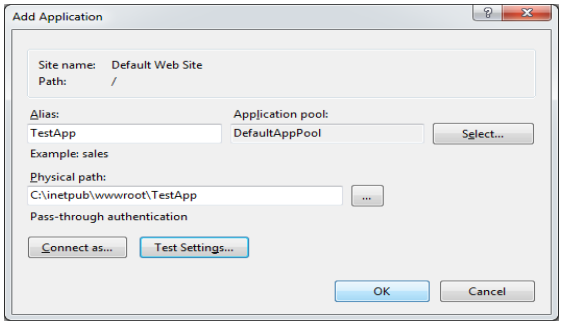

إن لم يكن لديك ASP.Net 4.0 , فيمكنك تثبيت ذلك من خلال الآداة aspnet_regiis.exe الموجودة ضمن ملفات تثبيت Microsoft .Net بداخل مجلد c:\Windows .

من خلال Command Prompt وتشغيل أدادة aspnet_regiis.exe لنرى التالى :-




## ثانياً :- إنشاء تطبيق عبر نافذة IIS .

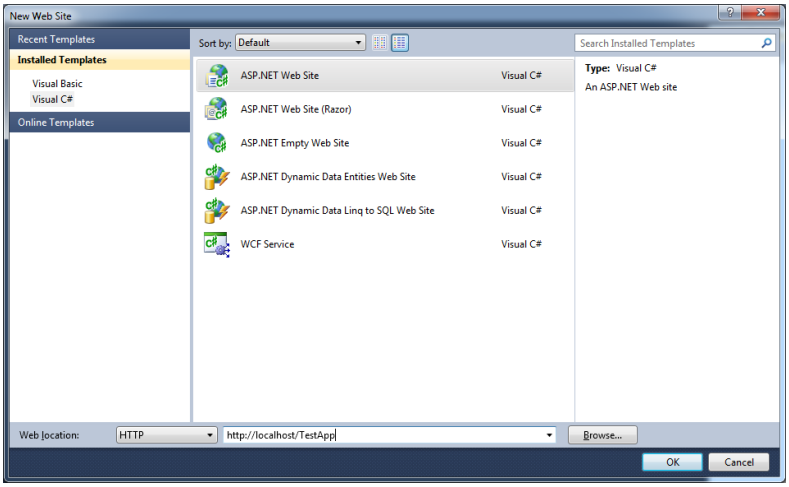
- بعد فتح نافذة IIS Manager , إذهب إلى الجانب الأيسر وقم بتحديد Default Web Site والوقوف عليه ثم R.C وقم بإختيار Add Application , لتظهر لك النافذة التالية :-



- كما ترى فإننا قمنا بتحديد اسماً مستعاراً للتطبيق ويمكن أن يكون الاسم الذى يناسبك وأيضا قمنا بتحديد مسار المجلد الذى نريد حفظ ملفات الموقع الذى نريد القيام بإنشائه بداخله .

**ثالثاً :- الإتصال بالتطبيق من خلال Visual Web Developer**

- أولاً لابد من تشغيل VWD بكامل الصلاحيات حتى يتسنى له التفاعل مع IIS, ولهذا يجب تشغيله كـ Administrator , وذلك من خلال الوقوف على إسم البرنامج فى قائمة Start ومن ثم R.C ونختار Run as Administrator .
- يمكننا الذهاب إلى قائمة file ومن ثم إختيار New WebSite
- لتظهر لك النافذة التالية , فقم بتحديد الإختيارات التى ترى :-

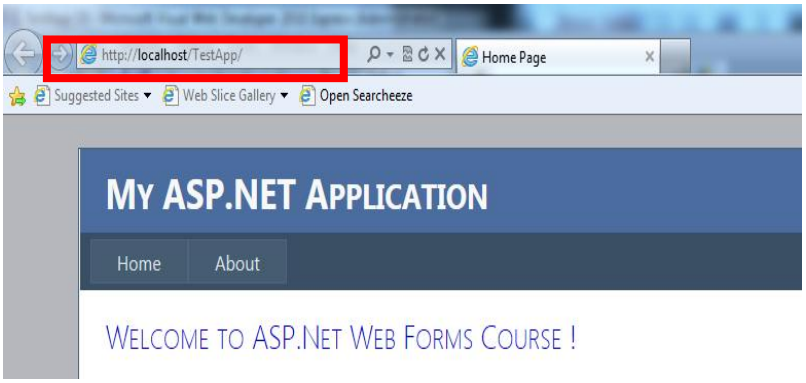



كما ترى قمنا بتحديد Http ومن ثم الإشارة إلى IIS من خلال Localhost , ثم نقوم بتحديد إسم التطبيق المراد العمل معه .

- ثم بعد ذلك نقوم بالضغط على OK , لتظهر لك نافذة بملفات المشروع الذى قمنا بإنشائه , ليتسنى لنا التفاعل معه والبدء فى بنائه .

قم بالتغيير فى محتويات الصفحات على سبيل التجربة .

- الآن يمكنك أن تتوجه إلى تشغيل المتصفح الخاص بك , وتكتب العنوان به localhost/testapp , لتستطيع به تصفح الموقع الذى قمت بإنشائه :-




نهاية الدرس والسلام عليكم