

## الفصل الثامن ضبط الخصائص Setting Properties

بعد تصميم واجهة التطبيق وتوقيع الأدوات المناسبة عليها ، يجب ضبط خصائص الأدوات التي قمت بوضعها على النموذج ، وفي هذا الفصل نتناول ضبط خصائص الأدوات والنماذج.  
بانتهاء هذا الفصل ستتعرف على:

- ◆ المقصود بالخصائص.
- ◆ ضبط وتغيير الخصائص أثناء التصميم باستخدام مربع الخصائص.
- ◆ ضبط وتغيير الخصائص أثناء التنفيذ باستخدام نافذة الكود.
- ◆ الإشارة إلى النموذج وعناصر التحكم داخل الكود.
- ◆ خصائص النماذج.
- ◆ الخصائص المشتركة.
- ◆ خصائص الوظائف والأحداث.

يتم التحكم في مظهر وسلوك النماذج وعناصر التحكم من خلال ثلاثة أشياء رئيسية وهي:

- الخصائص **Properties**
- الوظائف **Methods**
- الأحداث **Events**

سنركز في هذا الفصل بصفة أساسية على شرح خصائص الأدوات والنماذج وكيفية التحكم فيها وضبطها سواء باستخدام مربع الخصائص أو في مرحلة التنفيذ . ثم نلقي نظرة خاطفة على الوظائف **Methods** والأحداث **Events** لتوضيح العلاقة بينهما وبين الخصائص على أن نخصص لكل منهما فصلا مستقلا يشرح المقصود بهما مع إعطاء الأمثلة المناسبة .

### المقصود بالخصائص

في مرحلة التصميم والإعداد للبرامج يحدد **Visual Basic** مجموعة من الخصائص **Properties** ترتبط بكل نموذج أو أداة في التطبيق. أى أن لكل نموذج أو أداة في التطبيق مجموعة خصائص تختص بها. فمثلا يخصص لكل نموذج حجم ومكان على الشاشة، وكذلك يخصص لها مجموعة ألوان لون الكتابة (**ForeColor**) ولون الخلفية (**BackColor**) وتوجد هذه الخصائص داخل مربع خصائص الواجهة.

كما أن لكل أداة من الأدوات التي تظهر على النموذج مجموعة محددة من الخصائص، تحدد هذه الخصائص سلوك الأداة ومظهرها. فمثلا من الخصائص التي تحدد سلوك أداة مربع النص خاصية **MultiLine** أى الأسطر المتعددة ومعناها هل يقبل مربع النص أكثر من سطر أم لا، وأيضا من الخصائص التي تحدد شكل الأداة ومظهرها خاصية اسم خط الكتابة **FontName** وحجمه **FontSize** وخاصية اللون **ForeColor**. وتغيير هذه الخصائص يمكنك تغيير الوظيفة التي ستقوم بها الأداة أو الشكل الذي ستظهر به على الواجهة .

يمكنك النظر إلى خصائص الكائن (نافذة النموذج أو عنصر التحكم) على أنها وصف يميز

هذا الكائن، مثل أوصاف الشخص التي تميزه شكلا أو سلوكا مثل: طويل ، قصير ، أسمر، عصبي. فكل هذه يمكن اعتبارها خصائص للشخص بمصطلحات **Visual Basic**.

## ضبط أو تغيير الخصائص

يتم ضبط أو تغيير خصائص النموذج أو الأدوات بإحدى الطريقتين التاليتين:

- ضبط الخصائص أثناء تصميم التطبيق باستخدام مربع الخصائص.
- ضبط الخصائص أثناء تشغيل التطبيق باستخدام نافذة الكود.

واستخدام مربع الخصائص هي الطريقة الأسرع والأدق في مرحلة التصميم حيث تتم عملية التغيير من خلال عدة اختيارات تتم بواسطة النقر بالفأرة، ولذلك ننصح بإتباعها خصوصا في هذا المستوى من الدراسة. بينما تعتمد طريقة استخدام نافذة الكود على كتابة التعليمات بلغة **Visual Basic** وهي طريقة تعتمد على مهارة المبرمج في كتابة هذه التعليمات وتنطوي على مخاطر الوقوع في الخطأ أثناء كتابته لها. سنوضح فيما يلي كيفية ضبط الخصائص أثناء تصميم التطبيق وأثناء تشغيله.

### ضبط الخصائص أثناء تصميم التطبيق

يتم ضبط الخصائص أثناء مرحلة التصميم بعد توقيع الأدوات على الواجهات، باستخدام مربع الخصائص **Properties Window**. لاستدعاء مربع الخصائص الخاص بإحدى الأدوات أو النماذج الموجودة على الواجهة - إذا لم يكن ظاهرا- اتبع إحدى الطرق الآتية:

- اختر الأداة التي تريد ضبط خصائصها (يمكنك اختيار أكثر من أداة) ثم اضغط مفتاح **F4** أو انقر زر الخصائص  من شريط الأدوات أو من نافذة مستكشف الحل.
- اختر الأداة ثم افتح قائمة **View** واختر أمر **Properties Window** من القائمة المنسدلة.
- انقر بزر الفأرة الأيمن النموذج المطلوب أو الأداة ثم اختر أمر **Properties** من

القائمة الموضوعية.

- في جميع الأحوال، يظهر مربع الخصائص كما في شكل ٨-١ التالي ويتكون هذا المربع من:
- شريط العنوان ويحتوى من اليسار على الكلمة **Properties** والتي تعنى الخصائص ومن اليمين على زر الإغلاق  المستخدم في إغلاق مربع الخصائص وزر الإخفاء التلقائي  المستخدم في إخفاء المربع وإظهاره عند الإشارة إليه فقط وزر موضع المربع  الذى يظهر بنقره عدد من الخيارات التى يمكنك من خلالها التحكم فى المربع وإخفائه إن أحببت.
  - اسم الكائن أو النموذج المختار ويظهر تحت شريط العنوان مباشرة.
  - شريط العرض الذى يمكنك من خلاله عرض الخصائص فى ترتيب أبجدى من خلال الزر  أو عرضها طبقاً لمجموعات متجانسة من خلال الزر .
  - قائمة الخصائص: يعرض هذا الجزء من مربع الخصائص كل الخصائص المتعلقة بالأداة أو النافذة المختارة.
  - تعليق أو شرح للأداة المختارة ويظهر فى أسفل مربع الخصائص وتظهر الخاصية المختارة تحت الشريط المضاء.



شكل ٨-١ مربع الخصائص بين خصائص مربع نص.

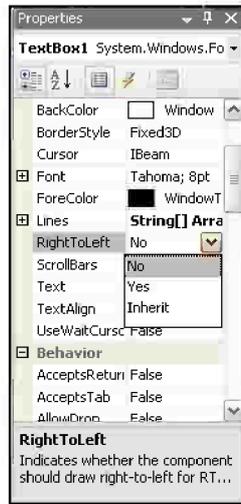
### تغيير قيم الخصائص

يختار **Visual Basic 2008** دائماً قيمة تلقائية لكل الخصائص. بعض الخصائص يتم تغييرها بتغيير الكتابة الموجودة لأنها ليس لها قيم أخرى أو ليس لها قيم محددة مثل خاصية **Text** أو **Name** لأن اسم الأداة أو التعليق تحدده أنت. بعض الخصائص لها قيمتين أو أكثر. وعادة تختار قيمة واحدة للخاصية وهي القيمة التي تناسبك. في الخصائص التي لها أكثر من قيمة، يظهر على يمين الخاصية المختارة سهم  أو زر . هذا الزر أو السهم يتغير شكله تبعاً للخاصية التي ترغب في تغييرها، وتفصيل ذلك على النحو التالي:

- في حالة الخصائص التي لها قيم محددة من **Visual Basic** يظهر السهم . فمثلاً لتغيير اتجاه الكتابة، اختر الخاصية **RightToLeft** يظهر هذا السهم  على يمين القيمة المختارة للخاصية ومعناه إمكانية اختيار قيمة واحدة من مجموعة من القيم. انقر على السهم بالفأرة تظهر لك مجموعة القيم التي يمكن أن تختار منها وتناسب هذه الخاصية كما في الشكل ٨-٢.
- في حالة الخصائص التي ليس لها قيم محددة سلفاً وإنما تحددها أنت، يظهر الزر 

. فمثلاً لتغيير خط الكتابة انقر على الخاصية **Font** الخاصة بالخط يظهر لك الرمز  بجانب خانة القيمة مباشرة ومعناه إمكانية اختيار أو تحديد قيم أخرى للخاصية، وبمجرد النقر على الرمز  الموجود على يمين الخاصية سيظهر لك مربع الحوار الخاص بمجموعة الخطوط التي يمكن التعامل معها لتتنقى منه الخط المناسب وحجمه ونمطه وهو نفس مربع **Font** الذي يظهر في برنامج **Microsoft Word** الشهير. بعد تحديد مواصفات الخط المطلوبة، قم بإغلاق مربع **Font** للعودة إلى مربع الخصائص مرة أخرى.

- في اختيارات أخرى للخصائص مثل خاصية النص **Text** في مربع الخصائص نلاحظ عدم ظهور السهم أو الزر على يمين الخاصية المختارة وهذا يدل على أنه لا يوجد اختيارات يمكن تحديدها من خلال هذا الزر وعلى المستخدم أن يستخدم لوحة المفاتيح في إدخال عنوان جديد أو قيم جديدة بصفة عامة .

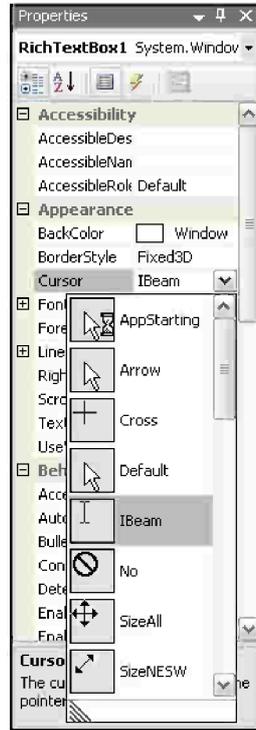


شكل ٨-٢ اختيار قيمة للخاصية من بين مجموعة من القيم.

### استخدام النقر المزدوج لتغيير الخصائص

هناك بعض الخصائص التي تعمل بطريقة مفصلية، بمعنى أنه يخصص لها قيمة من قيمتين.

فمثلاً الخاصية **Enabled** تأخذ دائماً واحدة من قيمتين. إما القيمة **True** بمعنى صح أو القيمة **False** بمعنى خطأ. مثل هذه الخصائص يكفي أن توجه إليها مؤشر الفأرة ثم تنقر عليها نقرًا مزدوجاً لتغير القيمة الحالية إلى القيمة الأخرى. يمكنك أيضاً استخدام النقر المزدوج مع الخصائص لتغير قيمة بعض الخصائص التي يخصها أكثر من قيمتين، أي الخصائص المحددة بمجموعة من القيم فمثلاً الخاصية **Cursor** تتغير قيمتها تبعاً لترتيب قائمة هذه الخاصية. فمثلاً إذا كان مؤشر الفأرة من نوع **Default** وأردنا أن نغيره إلى **I-Beam** فإننا نلاحظ أن ترتيب القيمة **I-Beam** في خاصية **Cursor** يأتي بعد **Default** مباشرة كما في الشكل ٨-٣. فإذا أردت تغيير القيمة الحالية للخاصية إلى التالية لها في الترتيب، انقر على الخاصية **Cursor** نقرًا مزدوجاً. وهكذا يمكنك بنقر الخاصية نقرًا مزدوجاً عدة مرات الحصول على القيمة المناسبة من مجموعة القيم المتاحة.



شكل ٨-٣ أشكال مؤشر الفأرة.

### تغيير خاصية في أكثر من أداة

لتغيير خاصية في أكثر من أداة مرة واحدة اتبع الآتي:

١. اختر الأدوات التي ترغب في تغيير أو ضبط خاصيتها.
٢. اضغط مفتاح **F4** أو انقر زر  من شريط الأدوات لإظهار مربع الخصائص حيث تظهر بداخله الخصائص المشتركة بين الأدوات التي اخترتها في الخطوة الأولى فقط .
٣. أجر التغييرات التي تراها. فمثلا لتغيير خط الكتابة في كل الأدوات المختارة، قم بتغيير خاصية **Font** من مربع الخصائص. ستتأثر كل الأدوات التي اخترتها بهذا التغيير وسيتغير الخط بالنسبة لها جميعا .

### ضبط الخصائص أثناء تنفيذ التطبيق

أحيانا يكون من الضروري تغيير خصائص إحدى الأدوات أثناء تشغيل التطبيق وليس أثناء تصميمه. منها مثلا إذا قررت تغيير لون الخلفية أثناء تشغيل التطبيق. كما أن بعض الأدوات لا يمكن ضبط خصائصها أثناء التصميم.

لتغيير الخصائص من نافذة الكود اتبع الصيغة العامة التالية :

**ControlName.PropertyName = NewValue**

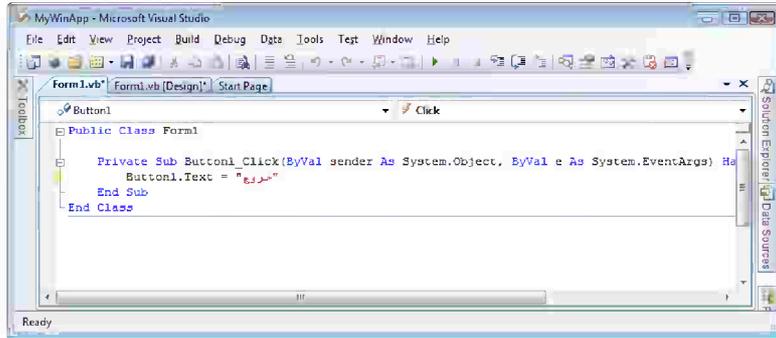
حيث تعنى **ContolName** اسم الأداة و **PropertyName** هى اسم الخاصية و **NewValue** هى القيمة الجديدة التى سيتم ضبط الخاصية بها.

وفيما يلى نوضح مثالا لاستخدام هذه الصيغة لتغيير عنوان أداة زر الأمر  . تابع معنا الخطوات الآتية:

١. افتح نموذجاً جديداً وضع عليه زر الأمر **Button1** .
٢. انقر نقرا مزدوجاً أداة زر الأمر الموجودة على النموذج، تظهر نافذة الكود.
٣. اكتب الأمر التالى:

**Button1.Text = "خروج"**

ومن ذلك نستنتج أن **Button1** هو اسم الأداة، و**Text** هو اسم الخاصية التي ترغب في تغيير قيمتها و"خروج" هو القيمة الجديدة التي سيتم ضبط الخاصية بها. وهذا الأمر معناه خصص القيمة "خروج" لتظهر على زر الأمر الخاص بالأداة المختارة وهي **Button1** (انظر شكل ٨-٤).



شكل ٨-٤ تغيير الخصائص أثناء التشغيل

## الإشارة إلى النماذج ومخازن التحكم داخل الكود

من الخصائص المشتركة بين كل كائنات **Visual Basic** خاصية الاسم **Name** وهي تعطي تمييز فريد للكائن ومن ثم تستطيع أن تعزو إليه النداءات والدوال وسط الكود. كل نموذج في **Visual Basic** له اسم مميز وكل كائنات النموذج لابد أن يكون لها أسماء مميزة إلا أنها قد تتماثل مع أسماء كائنات في نموذج آخر حيث لن يسبب ذلك مشاكل كما سنرى.

يعطي **Visual Basic** كما رأينا في الفصل السابق اسما افتراضيا للعنصر عند إضافته للنافذة (مثل **TextBox1** أو **Label1** وهكذا) إلا أنه من المفضل جدا أن تغير اسم العنصر عندما تضيفه لأن هذا الاسم سترجع إليه عند مخاطبة أو التحكم في هذا العنصر من خلال الكود فلا بد أن يكون ذو معنى ( ويجب أيضا ألا يكون طويلا حتى لا يرهقك في كتابته).

من العادات الجيدة والتي يستخدمها كثير من المبرمجين استخدام البادئة المكونة من ثلاثة

حروف والتي تعبر عن نوع العنصر عند تسميته، فمثلا الاسم frmMain معناه أن الاسم لنموذج (لأن البادئة frm تعني Form أو نموذج) وفيما يلي البادئات الشهيرة للعناصر المختلفة التي يستخدمها Visual Basic .

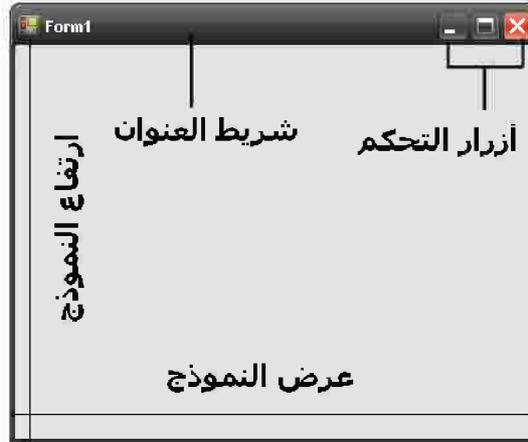
جدول ٨-١ البادئات المميزة لنوع الكائن

البادئة	نوع الكائن
Chk	CheckBox مربع اختيار
Btn	Button زر أمر
Cbo	ComboBox مربع تحرير وسرد
Frm	Form نموذج نافذة
Pnl	Panel إطار
Hsb	HScrollBar شريط تمرير أفقي
Pic	PictureBox مربع صورة
Lbl	Label عنوان
Lin	Line خط
Lst	ListBox مربع سرد
Mnu	Menu قائمة

البادئة	نوع الكائن
Rad	RadioButton زر اختيار
Txt	TextBox مربع نص
Tmr	Timer مؤقت
Vsb	VScrollBar شريط تمرير رأسي

### خصائص النماذج

عند النظر إلى نافذة نموذج سترى مساحة مستطيلة على الشاشة مثل شكل ٨-٥، ولكن وراء ظهورها هكذا تكمن الكثير من الخصائص، فعلى سبيل المثال موقع الركن الأيسر العلوي من النافذة يتحدد بخاصتي X، Y بينما تتحكم الخاصتان Width، Height في حجم النافذة. وتعني خاصية Text بإظهار التعليق الذي يظهر في شريط عنوان النافذة. ويمكن أيضا باستخدام الخصائص تحديد أي أزرار التحكم ستظهر أعلى يمين النافذة.



شكل ٨-٥ العناصر الرئيسية لنموذج النافذة تمثل خصائصها

### نظرة أخرى على خصائص النموذج

لقد شرحنا بعض خصائص الكائنات (نموذج أو أدوات). فيما يلي سنهتم بالخصائص التي لا توجد إلا في النموذج أو ما نسميه بالخصائص الخاصة أو الرئيسية **Key Properties**. الجدول التالي يبين هذه الخصائص بالنسبة لنموذج النافذة وهل يمكن تغيير هذه الخاصية أثناء التشغيل **Run-Time** أم لا.

جدول ٨-٢ الخصائص الرئيسية للتحكم في النافذة

اسم الخاصية	وظيفتها	يمكن تغييرها أثناء التشغيل
<b>FormBorderStyle</b> نمط الإطار	تحدد نمط إطار النافذة	لا
<b>ControlBox</b> مربع التحكم	تحدد ظهور قائمتي التحريك والإغلاق أعلى يسار النافذة	لا
<b>Font</b> الخط	تحدد الخط الذي يكتب به على سطح النافذة	نعم
<b>Icon</b> الرمز	تحدد الرمز المستخدم عند تصغير النافذة	نعم
<b>Maximize Box</b> زر التكبير	تحدد ظهور زر التكبير	لا
<b>IsMdiContainer</b> تحتوى على أكثر من نافذة	تحدد هل النافذة تحتوى على أكثر من نافذة وليدة أم لا	لا
<b>Minimize Box</b> زر تصغير	تحدد ظهور زر التصغير	لا
<b>StartPosition</b> الوضع الابتدائي	تحدد الوضع الذي تظهر عليه النافذة حين بدء ظهورها	نعم

اسم الخاصية	وظيفتها	يمكن تغييرها أثناء التشغيل
WindowState حالة النافذة	تحدد حالة النافذة (ملء الشاشة، مصغرة، عادية)	نعم

فالخاصية **FormBorderStyle** لها سبع قيم مختلفة تتحكم في نوع الإطار وهل النافذة متغيرة الحجم عند سحب أطرافها بالفأرة أم لا وما هي الأزرار التي تظهر على النافذة حال التشغيل وتحدد أيضا سمك شريط العنوان (انظر شكل ٦-٨). الجدول الآتي يعطي تفاصيل القيم السبع.



شكل ٦-٨ تغيير الخاصية **FormBorderStyle** يمكنها إعطاء النافذة عدة مظاهر مختلفة

جدول ٨-٣ قيم الخاصية FormBorderStyle

التأثير	القيمة
لا يظهر إطار للنافذة ولا شريط عنوان ولا أزرار تحكم	<b>None</b> بلا إطار
إطار غير سميك . يظهر شريط عنوان وأزرار تحكم إلا أن النافذة غير قابلة لتغيير حجمها	<b>FixedSingle</b> إطار ثابت الحجم
إطار غير سميك. يظهر شريط عنوان وأزرار تحكم إلا أن النافذة قابلة لتغيير حجمها	<b>Fixed3D</b> إطار ثابت ثلاثي الأبعاد
إطار اعتيادي به شريط عنوان وأزرار تحكم والنافذة قابلة لتغيير حجمها وهذه القيمة هي الافتراضية للخاصية	<b>Sizable</b> إطار متغير الحجم
إطار ثابت الحجم ، شريط العنوان وأزرار التحكم تظهر إلا أن زر التكبير وزر التصغير لا يظهران	<b>FixedDialog</b> مربع حوار ثابت الحجم
إطار ثابت الحجم وهناك شريط عنوان وزر إغلاق فقط وسمك شريط العنوان نصف العادي تقريبا	<b>FixedToolWindow</b> نافذة أدوات ثابتة الحجم
مثل السابق إلا أنه يمكن تغيير حجمه	<b>SizableToolWindow</b> نافذة أدوات متغيرة الحجم

يظهر تأثير تغيير **FormBorderStyle** أثناء تشغيل التطبيق فقط، أما أثناء التصميم ، فيمكن تغيير حجم النموذج بغض النظر عن قيمة الخاصية.

النمط الافتراضي للنافذة هو النافذة ذات الإطار المتغير الحجم كما يبدو في معظم برامج **Windows**. أيضا رغم أن الخاصية **FormBorderStyle** تتحكم بصورة عامة في ظهور أزرار التحكم على النافذة إلا أنه يمكن إظهار أو إخفاء أيا من هذه الأزرار على حدة باستخدام الخصائص **MaximizeBox** و **MinimizeBox** و **ControlBox** وذلك باختيار القيمة **True** أو **False** لأي منها. ومن الجدير بالذكر أيضا أن هذه الخصائص

تحدد قيمتها أثناء التصميم ولا يمكن تغيير قيمتها أثناء تشغيل التطبيق. أما الخاصية **Font** فتحدد الخط المستخدم في الكتابة عند استخدام الوظيفة **Print()** مع النافذة، ويلاحظ أن **Font** هو كائن بحد ذاته له العديد من الخواص مثل **Form1.Font.size** وهو حجم الخط. أيضاً تغيير خط النموذج يغير الخط الافتراضي لكل الأدوات التي توضع على النموذج بعد ذلك. خط النافذة لا يتحكم في خط شريط العنوان. لتغيير خط شريط العنوان استخدم لوحة التحكم **Control Panel** الموجودة داخل نظام التشغيل.

إذا لم تظهر كلمات عند استخدام **Print** مع النافذة، اضبط الخاصية **AutoRedraw** للنافذة حتى تعيد دائماً رسم محتويات النافذة الرسومية عند امتحانها لأي سبب مثل تغطيتها بنافذة أخرى. أيضاً لا بد من مراجعة قيمة الخواص **CurrentX** و **CurrentY** ربما كانا بقيم سالبة أو يقعا تحت إحدى الأدوات فلا يظهر الكلام.



آخر الخواص التي سنذكرها خاصية الوضع الابتدائي **StartPosition** وهي كما يظهر من اسمها تتحكم في وضع النموذج عند بدء ظهوره على الشاشة وهناك أربع قيم لهذه الخاصية نسردها في جدول ٨-٤ التالي.

جدول ٨-٤ قيم الخاصية **StartPosition**

القيمة	التأثير
<b>Manual</b> وضع يدوي	الموضع الابتدائي يتحدد بالخاصتين <b>X</b> و <b>Y</b>
<b>CenterScreen</b> مركزة على سطح الشاشة	تمركز النافذة على سطح المكتب
<b>WindowsDefaultLocation</b> الوضع الافتراضي لنظام التشغيل	يترك تحديد الموقع الابتدائي لنظام التشغيل

يتم تحديد الموقع الابتدائي تبعاً لحدود نظام التشغيل	<b>WindowsDefaultBounds</b> الحدود الافتراضية لنظام التشغيل
تتمركز النافذة في منتصف النافذة الأم	<b>CenterParent</b> منتصف النافذة الأم

تفيد نافذة تخطيط النموذج **Form Layout** عندما تعمل على دقة عالية للشاشة أثناء تصميمك لنافذة ستعمل على دقة أقل فهي تريك كيف سيبدو حجم النافذة بالنسبة لشاشة أقل دقة.



### الخصائص المشتركة

تنقسم خصائص الأدوات إلى خصائص مشتركة وخصائص خاصة. فالخصائص المشتركة هي الخصائص التي تشترك فيها معظم الكائنات (نقصد بالكائنات هنا الأدوات والنماذج) أما الخصائص الخاصة فهي الخصائص التي يستعملها كائن أو مجموعة معينة من الكائنات وإليك بعض نماذج من الخصائص المشتركة.

نصح بتجربة الخصائص التالية على الحاسب للتعرف على إمكانيات هذه الخصائص المشتركة.



#### خاصية النص **Text**

توجد هذه الخاصية داخل المجموعة **Appearance** بمربع الخصائص ويمكنك استخدامها لتغيير عنوان الأدوات أو النماذج حيث يظهر العنوان فوق الأداة. هذه الخاصية متوفرة مع الأدوات التي تحتاج لعنوان مثل النموذج وزر الأمر وزر الخيار والقوائم المنسدلة. كما أنها متوفرة أثناء تصميم وتنفيذ التطبيق.

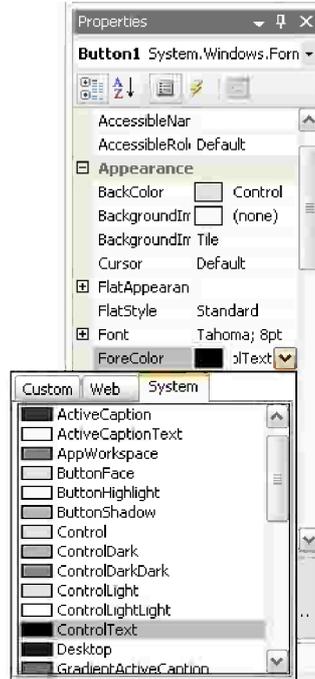
#### خاصية الاسم **Name**

توجد هذه الخاصية داخل المجموعة **Design** بمربع الخصائص وتستخدم في تحديد الاسم الذي تستخدمه للإشارة إلى الأداة أثناء كتابة الكود. فمثلاً يكون الاسم في الوضع الافتراضي لأول نموذج هو **Form1** والثاني **Form2** بينما يكون الاسم في الوضع

الافتراضي لأول زر أمر هو **Button1** والثاني **Button2** وهكذا ويمكنك أن تغيرها  
كيفما تشاء أثناء تصميمك لواجهة التطبيق. وتستخدم هذه الخاصية أثناء كتابة الكود  
والإجراءات للتعامل مع جميع الأدوات الموجودة في التطبيق. وتختلف هذه الخاصية عن  
خاصية النص فهي لا تظهر مطلقا أمام المستخدم وإنما تظهر بوضوح في نافذة الكود حيث  
يتم التعامل مع أى أداة فيها من خلال الاسم الموجود بخاصية الاسم.

### خصائص الألوان *Color Properties*

وتحتوى على خاصية لون الكتابة **ForeColor** وخاصية لون الخلفية **BackColor** والتي  
توجد بالمجموعة **Appearance** بمربع الخصائص حيث يمكنك الاختيار من بين الألوان  
المعرفة داخل نظام التشغيل **System** أو الألوان المستخدمة في تطبيقات الويب **Web**  
كما يمكنك إنشاء ألوان مخصصة من خلال التبويب **Custom**.



شكل ٨-٧ اختيار لون النص من مجموعة الألوان.

### خصائص الحجم والموقع *Top, Left, Height, Width*

تستخدم كل من X و Y داخل الخاصية **Location** لتحديد موقع النموذج داخل الشاشة (ولكن يتم الاستيعاض عن X و Y بالخاصيتين **Top, Left** داخل الكود) كما يستخدمان مع جميع الأدوات لتحديد موقع الأداة داخل النموذج أو الوعاء (**Container**) الذى يحتويها، حيث تستخدم **Top** للتحكم فى الطرف العلوى للأداة، بينما تستخدم **Left** للتحكم فى الطرف الأيسر للأداة. و تستخدم كل من **Height** و **Width** داخل الخاصية **Size** لتحديد حجم الواجهة داخل الشاشة أو لتحديد حجم الأداة داخل الواجهة، حيث تستخدم **Height** للتحكم فى ارتفاع الأداة، و **Width** للتحكم فى عرضها. وجميع هذه الخصائص موجودة داخل المجموعة **Layout** بمربع الخصائص.

وبرغم أن هذه الخصائص متوفرة أثناء التصميم والتنفيذ، إلا أنها مفيدة أكثر أثناء التنفيذ، لأن فى مرحلة التصميم تقوم أنت بتحريك الأداة أو النافذة إلى الموقع الذى تريده وتتحكم فى حجمها حسب ما تراه مناسباً باستخدام الفأرة. أما أثناء تشغيل التطبيق فانك تحتاج لكتابة الأمر الذى يتحكم فى موقع أو حجم الأداة أو النموذج. فمثلاً لنقل نافذة النموذج إلى أقصى اليسار العلوى من الشاشة استخدم الأمرين التاليين (حيث تشير **Me** إلى النموذج الحالى) :

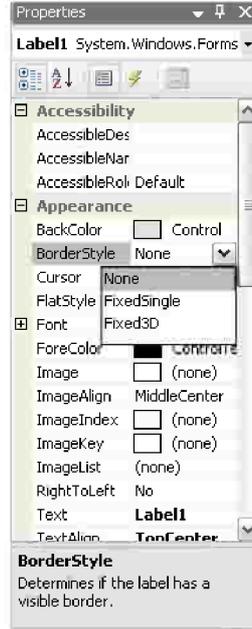
**Me.Top=0**

**Me.Left=0**

اسحب نافذة النموذج بالفأرة لنقلها من مكانها، تجد أن قيمة خاصيتي **Top, Left** تتغير حسب الموضع الجديد. وأيضا تغيير حجم أى أداة سيغير قيمة كل من **Height** و **Width** المخصصتين لها فى مربع الخصائص.

**BorderStyle** خاصية نمط الحد

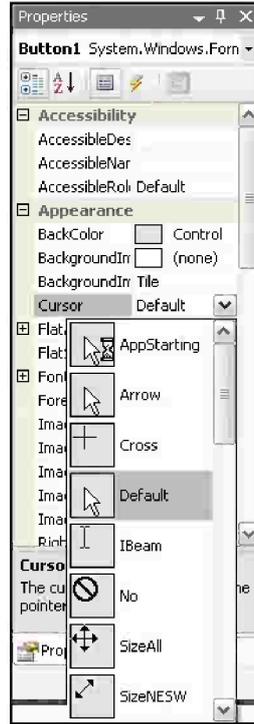
وهى خاصية يمكن من خلالها إظهار/إخفاء الحد **Border Style** أو التحكم فى شكله (انظر مربع الخصائص فى شكل ٨-٨).



شكل ٨-٨ خيارات خاصية نمط الحد.

### خاصية شكل المؤشر *Cursor*

يتيح لك **Visual Basic** تغيير شكل المؤشر تبعاً للأداة التي يمر فوقها، ويمكنك اختيار واحد من ٢٨ شكل مختلف للمؤشر (انظر شكل ٨-٩). وهذه الخاصية متوفرة أثناء التصميم والتنفيذ.



شكل ٨-٩ أشكال مؤشر الفأرة.

### خاصية التمكين *Enabled*

توجد هذه الخاصية داخل المجموعة **Behavior** بمربع الخصائص وتستخدم لتحديد إمكانية التعامل مع الأداة. فعندما تكون قيمتها **True** (الوضع الافتراضي) يمكنك أن تستخدمها، كما أنها تستجيب لجميع الأحداث **Events** التي نوقعتها عليها. أما إذا غيرتها إلى **False** فلن تستجيب لأي حدث يقع عليها. فمثلاً تغيير خاصية **Enabled** لزر أمر إلى **False** يتسبب في ظهور زر الأمر خافتاً دلالة على تعطيل استجابته لأي حدث.

### خاصية الظهور *Visible*

وتوجد داخل المجموعة **Behavior** أيضاً حيث تظهر عادة الأدوات التي تضعها على النموذج أثناء تشغيل التطبيق فيما عدا أداة المؤقت  وأدوات مربعات الحوار الشائعة.

إذا أردت إخفاء أداة أو نموذج أثناء تنفيذ التطبيق غير هذه الخاصية من **True** إلى **False**. ورغم أن هذه الخاصية متوفرة أثناء التصميم والتنفيذ، إلا أنك إذا غيرتها إلى **False** أثناء التصميم فلن يظهر تأثيرها إلا عند تشغيل التطبيق.

#### خاصية من اليمين إلى اليسار *Right to Left*

توجد هذه الخاصية داخل المجموعة **Appearance** بمربع الخصائص. والوضع الافتراضي لاتجاه الكتابة داخل الأدوات هو من اليسار إلى اليمين (القيمة **False**) غير أنه بإمكانك تغيير اتجاه الكتابة ليصبح في الاتجاه من اليمين إلى اليسار بتخصيص القيمة **True** هذه الخاصية .

### الوظائف والأحداث **Methods & Events**

الوظائف **Methods** (بترجمة غير حرفية) هي ما يستطيع الكائن تأديته من وظائف برمجية وهي دوال مبنية داخله نستدعيها من خلال اسم الكائن والوظيفة معا من خلال ما يسمى بالتسمية المنقوطة **dot-notation** مثل (**Form1.Show()** والتي تعني حثك الكائن **Form1** على تأدية الوظيفة (**Show()** والتي يدرك الكائن تفاصيل القيام بها. يساعدك **Visual Basic 2008** عند كتابتك للكود إذا كتبت اسم كائن ثم نقطة بأن يستعرض (في محل الكتابة) الوظائف والخصائص المتاحة لهذا الكائن ( جرب ذلك بأن تكتب **Form1** في أي مكان من نافذة الكود).

#### الاستجابة لأفعال المستخدم بالأحداث

فضلا عن الوظائف يمكن للكائنات الاستجابة للأحداث. مثلا عندما يضغط المستخدم على زر الفأرة الأيسر يطلق حدث **Click** لزر الأمر. من أمثلة الأحداث أيضا اختيار عنصر من قائمة خيارات، أو تعديل محتويات مربع نص. تقع الأحداث أيضا عندما يغلق المستخدم نافذة أو يتحول إلى نافذة أخرى، ويقوم **Visual Basic 2008** بتشغيل الإجراء الحداثي المرتبط بهذا الحدث عند وقوعه، لذلك إذا أردت تحديد استجابة محددة فعليك بوضع كود في الإجراء المراد.

### العلاقة بين الخصائص والأحداث والوظائف

كما قلنا تحدد الخصائص مظهر الكائن بينما تحدد الوظائف مهامه التي يمكنه القيام بها والأحداث هي المثيرات التي تستحث الكائن على القيام بفعل ما، في الحقيقة هذه الثلاثة لا تعمل منفصلة بل هناك علاقة وثيقة بين الثلاثة. مثلا عند استدعائك لوظيفة ما لكائن قد تتغير تبعاً لذلك بعض خصائصه، أيضا عادة ما نستدعي وظائف معينة استجابة لبعض الأحداث (فيما يسمى البرمجة المسيرة بالأحداث Event-Driven).

من الأمثلة الجيدة للعلاقة بين الخصائص والوظائف الخاصية **Visible** والوظيفتان **Show()** و **Hide()** حيث أن إظهار الكائن يمكن أن يتم إما بالوظيفة **Show()** أو بتغيير قيمة الخاصية **Visible** إلى **True** والعكس بالنسبة للوظيفة **Hide()**.

