الفصل الرابع
صناديق الإدخال والإخراج

الفصل الرابع

صناديق الإدخال والإخراج

صناديق الإدخال والإخراج هما من أهم قنوات الاتصال التي لا بد أن تتوافر بين المستخدم والبرنامج إذ أنه من الصعب أن تجد برنامج ما يخلو منهما مهما كان نوع هذا البرنامج .

والمقصود بصناديق الإدخال هي الرسائل التي تظهر للمستخدم من أجل الحصول منه على قيمة ما ، أما صناديق الإخراج فهى الرسائل المتعارف وهذه الرسائل يمكن أن تكون تحذيرية أو تنبيهية تظهر للمستخدم في حالة وقوعه في خطأ ما أو قد تكون تأكيدية مثل أن يقوم المستخدم بإنهاء البرنامج فتظهر له رسالة تأكيد قبل إنهاء البرنامج هل يريد بالفعل الخروج من البرنامج أم لا أو قد تكون غير ذلك ...

وصناديق الإدخال والإخراج أو الرسائل هذه هي عبارة عن Form صغيرة حجمها ثابت يمكن لك أن تقوم بتصميمها بنفسك ولكن الفيجوال بيسك توفر عليك عناء القيام بذلك وتتيح لك استخدام رسائل جاهزة بالمفاتيح التي يمكن أن تحتاج إليها وذلك من خلال دالة MsgBox ودالة InputBox وهذا هو ما سنتناوله من خلال هذا الجزء فهيا بنا ...

الدالة MsgBox :

تستخدم هذه الدالة في إظهار رسائل للمستخدم والصيغة القياسية لهذه الرسالة هي كما يلى :

```
MsgBox ( Prompt,[Buttons As VBMsgBoxSttyle],[Title] )
```

حيث تشير المعاملات الخاصة بالصيغة السابقة إلى ما يلى :

MsgBox : اسم الدالة المسئولة عن إظهار الرسالة .

Prompt : نص الرسالة التي ستظهر للمستخدم مع ملاحظة أن هذا النص يجب أن يكون بين علامتي تنصيص كهذا " نص الرسالة " .

Buttons As VBMsgBoxSttyle : يشير هذ المعامل إلى الشكل الذي سوف تظهر عليه الرسالة فعلى سبيل المثال يمكن من خلال هذا المعامل تحديد الأيقونة التي ستظهر داخل الرسالة مثل علامة استفهام أو علامة تعجب أو غير ذلك بما يتناسب مع مضمون الرسالة كذلك يمكن من خلال هذا المعامل تحديد نوعية وعدد المفاتيح التي ستظهر داخل الرسالة والوضع الافتراضي هو ظهور مفتاح واحد فقط هو مفتاح OK كما يمكنك من خلال هذا المعامل أيضاً تغيير اتجاه الرسالة لتتناسب مع البرامج العربية وهذا كله يتم من خلال مجموعة من الثوابت الخاصة باللغة سنتعرف عليها لاحقاً .

Title : يشير هذا المعامل إلى عنوان الرسالة الذي سيظهر داخل Titel Bar الخاص بالرسالة مع ملاحظة أن عنوان الرسالة هذا لا بد أن يكتب علامتى تنصيص كهذا " عنوان الرسالة " .

ملاحظة هامة :

ستجد أن بعض المعاملات الخاصة بالرسالة السابقة تظهر لك بين قوسين **[ ]** وهذا إشارة إلى أن هذه المعاملات اختيارية يمكن تجاهلها .

وفيما يلى سنقوم باستعراض بعض الأمثلة نتعرف من خلالها طريقة استخدام الرسائل داخل الفيجوال بيسك فقم بفتح مشروع جديد ثم قم بإضافة مفتاح CommandButton ثم قم بإضافة الكود التالي بداخل هذا المفتاح .

```
MsgBox "أتمنى لك التوفيق في دراسة البرمجة"
```

لتصبح نافذة الكود كما يلى :

حيث ستحتوى الرسالة على معامل واحد هو نص الرسالة فقم الآن بتشغيل البرنامج ثم أنقر فوق مفتاح CommandButton لتظهر لك الرسالة كما بالشكل التالي

كما يظهر لك بالشكل السابق أن الرسالة التي ظهرت لك كانت تحمل عنوان Project1 وهو اسم المشروع الذي نعمل عليه وذلك لأننا لم نقوم بكتابة عنوان للرسالة كذلك فإن الرسالة تحتوى على مفتاح واحد هو OK .

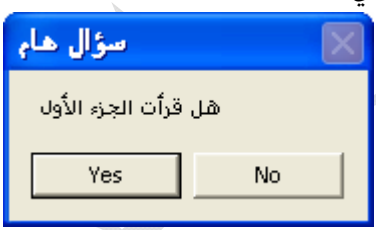
التحكم في المفاتيح التي تظهر داخل الرسالة :

فتعالى معى الآن نقوم بإظهار رسالة بها أكثر من مفتاح بما يتناسب مع محتواها من خلال الكود التالي :

```
MsgBox "هل قرأت الجزء الأول", vbYesNo, "سؤال هام"
```

لتصبح نافذة الكود كما بالشكل التالي :

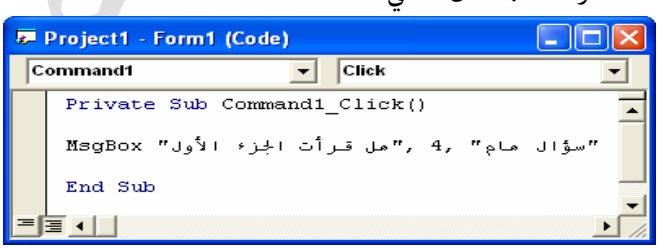
حيث يشير المعامل **vb YesNo** إلى الثابت الخاص بالمفاتيح التي ستظهر داخل الرسالة أما النص **"سؤال هام"** فهو عنوان الرسالة ، وفى هذه المرة ستظهر لك الرسالة كما بالشكل التالي :



لاحظ ... أنه يمكنك كتابة قيمة الثابت بدلاً من كتابة الثابت نفسه كما يلى :

MsgBox "هل قرأت الجزء الأول", 4, "سؤال هام"

لتصبح نافذة الكود كما بالشكل التالي :

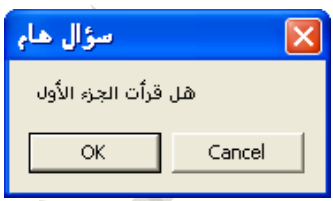


ويمكنك التعرف على أسماء الثوابت الخاصة بالمفاتيح التي تظهر داخل الرسالة والقيم الخاصة بكل منها من خلال الجدول التالي :

Value	Constant	Button
1	vbOKCancel	OK Cancel
2	vbAbortRetryIgnore	Abort Retry Ignore
3	vbYesNoCancel	Yes No Cancel
4	vbYesNo	Yes No
5	vbRetryCancel	Retry Cancel

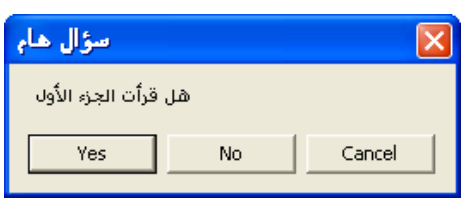
وفيما يلى أشكال المفاتيح التي ستظهر داخل الرسالة بناءاً على تلك الثوابت :

الثابت vbOKCancel :

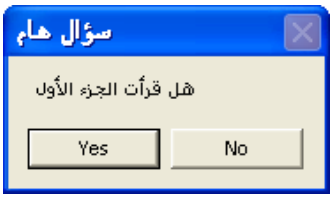


الثابت vbAbortRetryIgnore :

الثابت vbYesNoCancel :



الثابت **vbYesNo** :



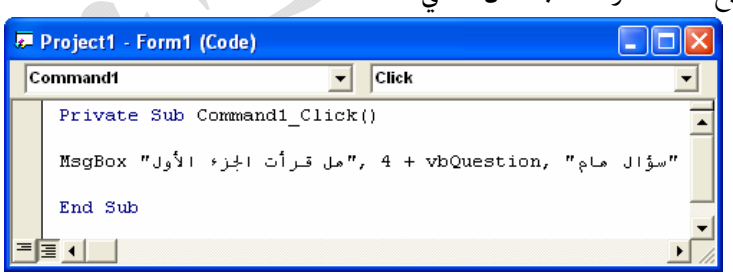
الثابت **vbRetryCancel** :

إظهار أيقونة داخل الرسالة :

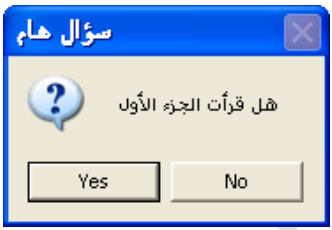
سنقوم الآن بإضافة أيقونة داخل الرسالة تتناسب مع محتواها ولإكسابها مزيد من الاحترافية وذلك من خلال الكود التالي :

```
MsgBox "هل قرأت الجزء الأول", 4 + vbQuestion, "سؤال هام"
```

لتصبح نافذة الكود كما بالشكل التالي :



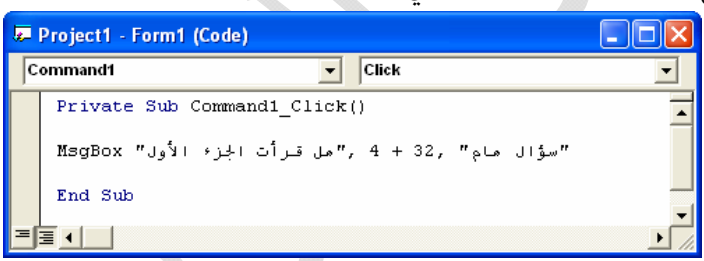
الجديد في الكود السابق هو الثابت **vbQuestion** حيث قمنا بإضافته بعد علامة + وهذا الثابت يقوم بإظهار أيقونة استفهام داخل الرسالة كما بالشكل التالي :



لاحظ ... أنه يمكنك كتابة قيمة الثابت vbQuestion بدلاً من كتابة الثابت نفسه وقيمة هذا الثابت هي 32 ليصبح الكود كما يلي :

```
MsgBox "هل قرأت الجزء الأول", 4 + 32, "سؤال هام"
```

وتصبح نافذة الكود كما بالشكل التالي :



ويمكنك التعرف على أشكال الأيقونات التي يمكن إظهارها داخل الرسالة وأسماء الثوابت الخاصة بتلك الأيقونات وقيم كل منها من خلال الجدول التالي :

Value	Constant	Icon
16	vbCritical	
32	vbQuestion	
48	vbExclamation	
64	vbInformation	

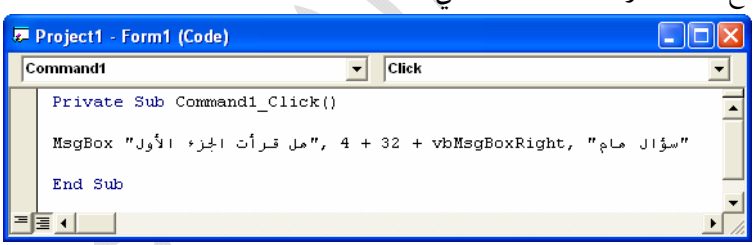
تغيير اتجاه الرسالة :

كما لاحظت من خلال ما سبق وبرغم ما قمنا به من تحكم في عناصر الرسالة المختلفة ومحتوياتها إلا أن الشكل العام للرسالة مازال غير مرضى وذلك لأن الوضع الافتراضي لهذه الرسالة أنها تتعامل مع برامج باللغة الإنجليزية وهذا ما يجعل الرسالة تظهر بهذا الشكل ، لذلك سنقوم هنا بالتعرف على كيفية تغير اتجاه الرسالة لتتناسب مع المظهر العام للغة العربية من خلال الثوابت الخاصة بذلك .

قم بإضافة الكود التالي :

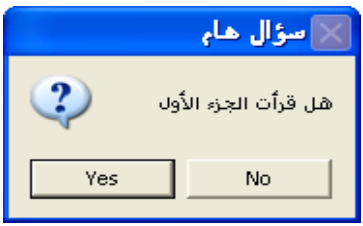
```
MsgBox "هل قرأت الجزء الأول", 4 + 32 + vbMsgBoxRight, "سؤال هام"
```

لتصبح نافذة الكود كما بالشكل التالي :



الجديد في الكود السابق هو الثابت **vbMsgBoxRight** الخاص بمحاذاة النص داخل الرسالة من الجهة اليمنى إلى الجهة ليسرى .

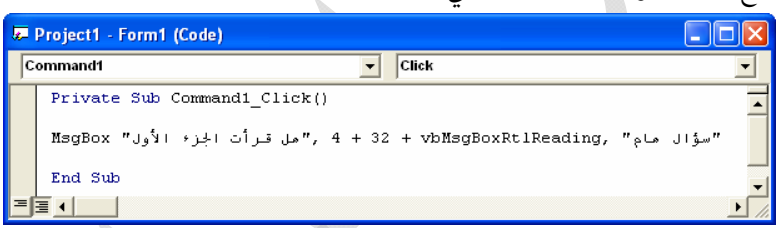
وفى هذه المرة ستظهر الرسالة كما بالشكل التالي :



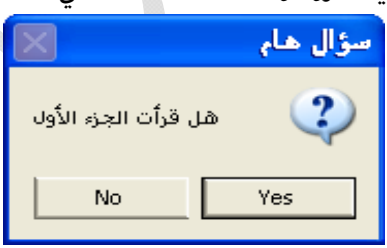
كما يمكنك استخدام الثابت **vbMsgBoxRtlReading** الخاص بتغير اتجاه الرسالة من الجهة اليمنى إلى الجهة اليسرى كما بالكود التالي :

```
MsgBox "هـــل قـــرأت الجـــزء الأول", 4 + 32 + vbMsgBoxRtlReading, "سؤال هام"
```

لتصبح نافذة الكود كما بالشكل التالي :



وهذا الثابت سيتسبب في ظهور الرسالة كما بالشكل التالي :



كيف نجعل البرنامج يتفاعل مع اختيار المستخدم :

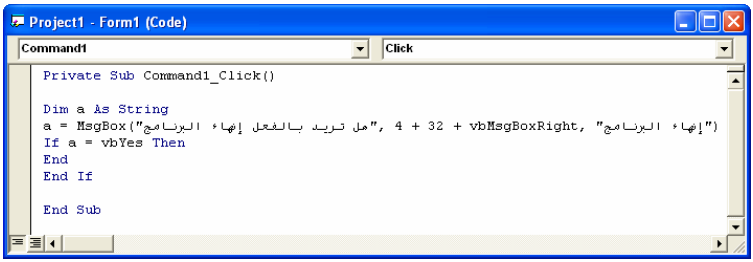
خلال جميع الأمثلة التي تناولنها سابقاً ستجد انك عند النقر فوق أي مفتاح من المفاتيح التي تظهر داخل الرسالة لا يوجد أي رد فعل من البرنامج تجاه ما قمت به سوى اختفاء هذه الرسالة وهذا قد يكون مقبولاً في حال إذا كانت الرسالة خاصة

بإظهار معلومة ما للمستخدم وتحتوى على مفتاح واحد هو OK ولكن إذا كانت الرسالة تحتوى على أكثر من مفتاح ومطلوب من المستخدم الاختيار بين أمرين أو أكثر ففى هذه الحالة سيختلف الوضع كلياً لأن اختيار المستخدم هذا يمكن أن ينتج عنه تغير في مسار عمل البرنامج ككل وعلى هذا الأساس سيتعين علينا هنا معرفة أي المفاتيح قام المستخدم بنقرها لنبدأ بعد ذلك تحديد رد فعل البرنامج تجاه ما قام به المستخدم فتعالى معى خلال الخطوات التالية نتعرف على كيفية القيام بذلك ...

سنفترض هنا أننا نريد عمل مفتاح إنهاء للبرنامج بحيث إذا نقر المستخدم فوق هذا المفتاح تظهر له رسالة تسأله إذا كان بالفعل يريد إنهاء البرنامج أم لا وهذه الرسالة ستحتوى على مفتاحين Yes و No فإذا نقر المستخدم مفتاح Yes في هذه الرسالة يتم إنهاء البرنامج بالفعل أما إذا نقر مفتاح No فسيتم تجاهل الأمر واستكمال البرنامج بشكل عادى وذلك سيكون من خلال الكود التالي :

```
Dim a As String
a = MsgBox("هل تريد بالفعل إنهاء البرنامج", 4 + 32 +
vbMsgBoxRight, "إنهاء البرنامج")
If a = vbYes Then
End
End If
```

لتصبح نافذة الكود كما بالشكل التالي :



شرح الكود :

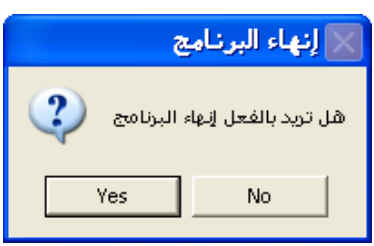
في السطر الأول قمنا بتعريف متغير باسم a من النوع String .

في السطر الثاني نقوم بإظهار رسالة للمستخدم للتأكد من أنه بالفعل يريد إغلاق البرنامج وهذه الرسالة تحتوى على مفتاحى Yes و No ثم قمنا بإسناد قيمة هذه الرسالة إلى المتغير a .

في السطر الثالث قمنا باختبار قيمة المتغير a عن طريق جملة If فإذا كانت قيمة المتغير a مساوية للثابت **VbYes** (أي أن المستخدم قام بنقر مفتاح Yes) فيتم تنفيذ الكود الموجود بالسطر الرابع وهو إنهاء البرنامج .

في السطر الخامس قمنا بإغلاق جملة If عن طريق عبارة End If .

قم بعد ذلك بتشغيل البرنامج ثم انقر فوق مفتاح إنهاء البرنامج لتظهر لك الرسالة التالية :

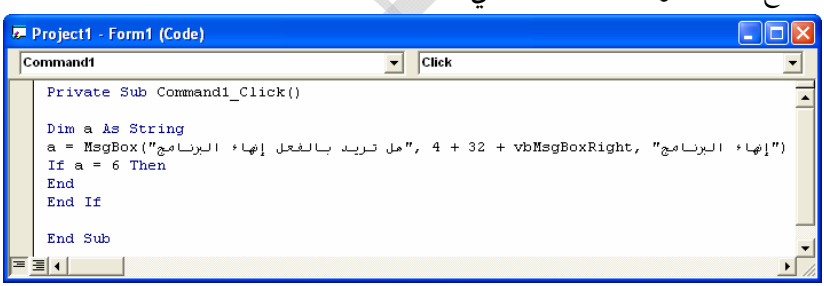


وعند النقر فوق مفتاح Yes سيتم إنهاء البرنامج أما إذا قمت بنقر مفتاح No فستختفي الرسالة ويستمر البرنامج في عمله .

لاحظ ... أننا في الكود السابق تمكنا من معرفة أن المستخدم قام بنقر مفتاح Yes عن طريق الثابت **VbYes** ولكن يمكنا تجاهل كتابة أسم الثابت وكتابة القيمة المعبرة عنه فقط وهى 6 ليصبح الكود كما يلى :

```
Dim a As String
a = MsgBox("هل تريد بالفعل إنهاء البرنامج", 4 + 32 + vbMsgBoxRight, "إنهاء البرنامج")
If a = 6 Then
End
End If
```

لتصبح نافذة الكود كما بالشكل التالي :



يمكنك الآن أن تقوم بمعرفة الثوابت الخاصة بأي مفتاح يقوم المستخدم بنقره أو القيم الخاصة بتلك الثوابت وذلك من خلال الجدول التالي :

Value	Constant	Button
vbOk	1	Ok
vbCancel	2	Cancel
vb Abort	3	Abort
vb Retry	4	Retry
vb Ignore	5	Ignore
vb Yes	6	Yes
vb No	7	No

الدالة Input Box :

الغرض من استخدام هذه الدالة هو استقبال قيمة ما من المستخدم والصيغة القياسية لهذه الدالة كما يلى :

```
InputBox (Prompt,[Title],[Default]
```

حيث تشير المعاملات الخاصة بالصيغة السابقة إلى ما يلى :

InputBox : اسم الدالة المسئولة عن إظهار صندوق الإدخال .

Prompt : الرسالة التي ستعرض للمستخدم ولا بد أن تكتب بين علامتي تنصيص

Title: عنوان صندوق الإدخال الذي سيظهر داخل Titel Bar لا بد أن يكتب علامتى تنصيص .

Default : القيمة الافتراضية ستظهر داخل صندوق الإدخال .

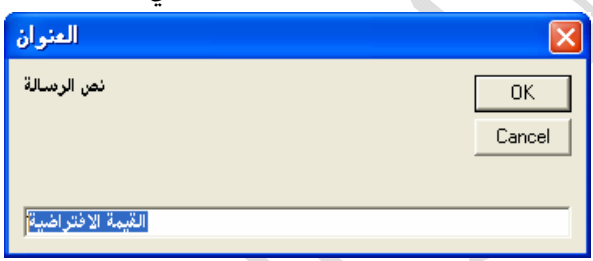
لاحظ ... أن التحكم في خصائص وعناصر هذه الدالة غير متاح بالصورة التي كانت عليه دالة MsgBox السابقة .

وللتعرف على هذه المعاملات بشكل عملى قم بكتابة الكود التالي داخل مفتاح CommandButton :

```
InputBox "نص الرسالة"، "العنوان"، "القيمة الافتراضية"
```

لتصبح نافذة الكود كما بالشكل التالي :

قم الآن بتجربة هذا الكود ليظهر لك صندوق الإدخال التالي :



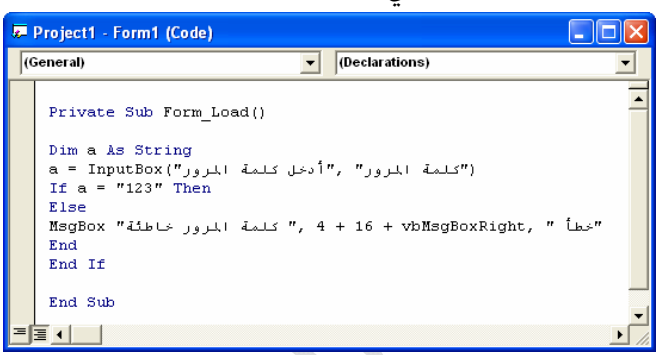
بعد أن تعرفت على هذه المعاملات عن قرب وعلمت ما هو المقصود بكل منها سنقوم الآن بالتعرف على كيفية استقبال القيمة التي يقوم المستخدم بإدخالها وكيف يمكن للبرنامج التفاعل مع تلك القيمة من خلال المثال التالي .

سنفترض أننا نريد عمل كلمة مرور للبرنامج 123 بحيث أنه عند بداية تحميل البرنامج يظهر صندوق نص للمستخدم ليقوم بإدخال كلمة المرور من خلاله فإذا كانت القيمة التي يقوم المستخدم بإدخالها صحيحة يتم تشغيل البرنامج وان كانت غير صحيحة يتم ظهور رسالة خطأ للمستخدم وإنهاء البرنامج وللقيام بذلك قم بإضافة الكود التالي داخل الحدث Load للفورم .

```
Dim a As String
a = InputBox("أدخل كلمة المرور"، "كلمة المرور")
If a = "123" Then
Else
```

```
MsgBox " كلمة المرور خاطئة"، 4 + 16 + vbMsgBoxRight,
" خطأ"
End
End If
```

لتصبح نافذة الكود كما بالشكل التالي :



شرح الكود :

في السطر الأول قمنا بتعريف متغير باسم a من النوع String ليتم استقبال القيمة التي ستعود بها الدالة InputBox من خلاله .

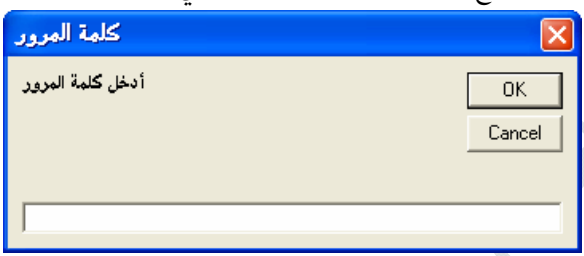
في السطر الثاني قمنا **باستخدام** الدالة InputBox لإظهار صندوق نص للمستخدم ليقوم بإدخال كلمة المرور من خلاله مع وضع القيمة التي تعود بها هذه الدالة داخل المتغير a .

في السطر الثالث باختبار القيمة التي قام المستخدم بإدخالها عن طريق جملة If فإذا كانت هذه القيمة هي 123 فمعنى ذلك أن كلمة المرور صحيحة وسيتم تشغيل البرنامج

في السطر الرابع في حالة إذا كانت كلمة المرور غير صحيحة سيتم إظهار رسالة خطأ للمستخدم وإنهاء البرنامج كما بالسطر الخامس والسطر السادس .

في السطر السابع نقوم بإنهاء جملة If عن طريق جملة End if .

قم الآن بتشغيل البرنامج ليظهر لك صندوق النص التالي :



عليك الآن أن تقوم بإدخال كلمة المرور ليتم تحميل الفورم وتشغيل البرنامج أما في حالة إذا قمت بإدخال كلمة مرور خاطئة فستظهر لك رسالة الخطأ التالية :