

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

دروس في

VBA Excel

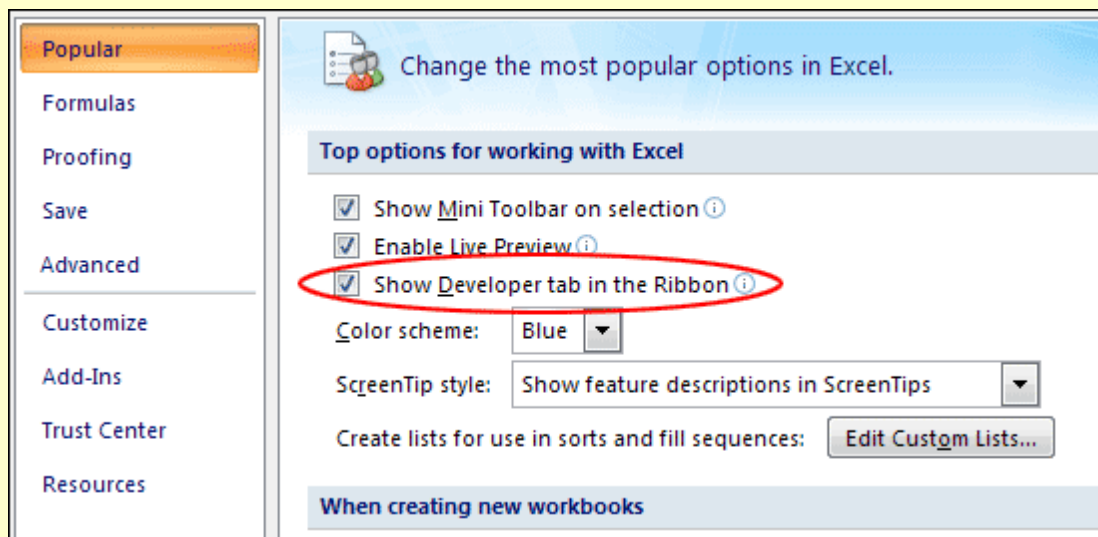
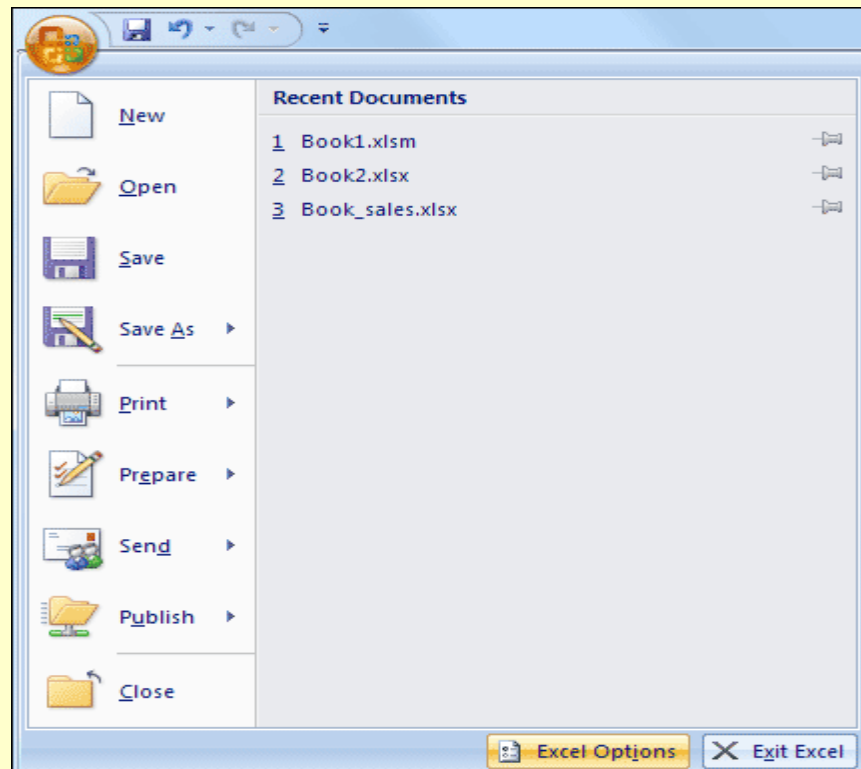
هذه الدروس نتيجة جهد مستفاد من شروحات

وأعمال السادة أساتذتي

أسأل الله عز وجل لهم أن تكون في ميزان حسناتهم

الدرس الأول

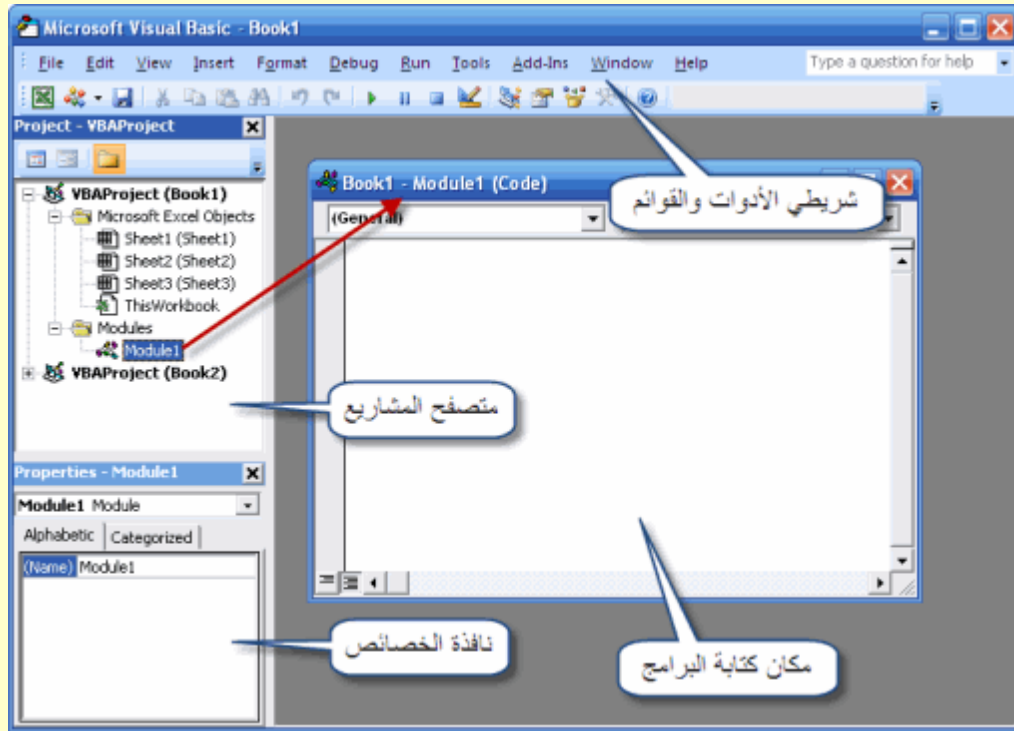
قبل أن نبدأ، دعونا نتأكد من أن الأدوات التي نحتاجها موجودة.
إذا كنت تستخدم (Excel 2007) أو إصدار أعلى
click ". Office button, then click Excel Options بالاسفل



سيتم إضافة علامة تبويب جديدة وهي Developer



سوف نحتاج في العمل مع التعليمات البرمجية ل VBA ، محرر، يتم تثبيته بشكل افتراضي. يمكنك فتحه عن طريق الضغط على تركيبة مفتاح الاختصار " Alt+ F11 : او انقر على



الدرس الثاني

تسجيل الماكرو

سنقوم بعمل المثال التالي معا للتعرف على تفاصيل خاصة بالكود والماكرو وأتمنى أن يتم تنفيذه من قبل الدارس أو الراغب في التعلم

المثال

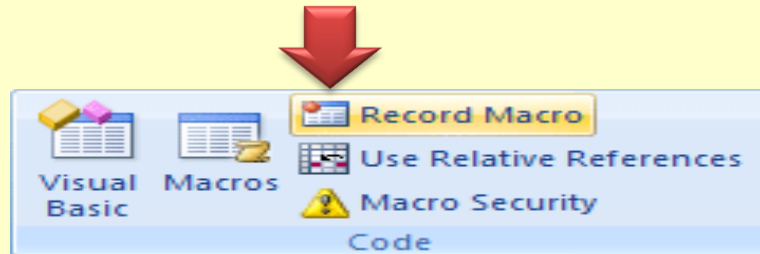
- حذف محتويات الأعمدة A و C
- نقل محتويات العمود B الى العمود A
- نقل محتويات العمود D الى العمود C

قم بتعبئة محتويات الأعمده السابقة بأيه بيانات

قم بالضغط على "Record Macro" "Ok"
قم بتنفيذ المطلوب يدويا الى أن تنفذ جميع الخطوات بالترتيب
لأن كل ما تفعله يتم تسجيله من قبل اداه تسجيل الماكرو
ثم إضغط على Stop Recording

الخطوات بالصور لتسهيل تنفيذ المطلوب

بعد تعبئة البيانات قم بالضغط على Record Macro



تظهر لنا نافذة

١ - نقوم بكتابه اسم الماكرو
نلاحظ أن اسم الماكرو لا يحتوى على ايه مسافات

ومن الممكن أن نضع علامه (_) بين كلمتين لو اردنا تسمية الماكرو بأسم مكون من كلمتين مثلا

٢ - نجد ان هناك كلمه + Ctrl وبجوارها مربع

وتعنى أننا يمكن وضع أى حرف داخل هذا المربع ويكون وسيله لإستدعاء الماكرو فيما بعد لتنفيذ المطلوب وهذا امر غير مطلوب حاليا سيتم توضيح ذلك فيما بعد ان شاء الرحمن بالتفصيل

٣ - هناك مستطيل به سهم رأسه الى الأسفل

وتعنى اختيار من متعدد وهنا نقوم بإختيار الملف المراد تسجيل به الماكرو فى حاله ان هناك مثلا عدد (٢) ملف اكسل مفتوحين

٤ - وهناك مربع آخر وهو الوصف المختصر لهذا الماكرو الذى سيتم تسجيله وحاليا هذا الأمر غير مطلوب

٥ - نقوم بالضغط على زر OK

Record Macro

Macro name: نقوم بكتابه اسم الماكرو او نتركه كما هو موضح

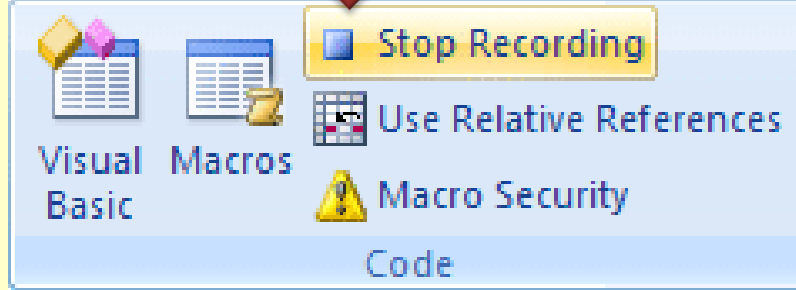
Shortcut key: Ctrl+ نقوم بكتابه اى حرف للاختصار لإستدعاء الماكرو وهو غير مطلوب حاليا

Store macro in: اختيار الملف

Description: الوصف المختصر

OK Cancel

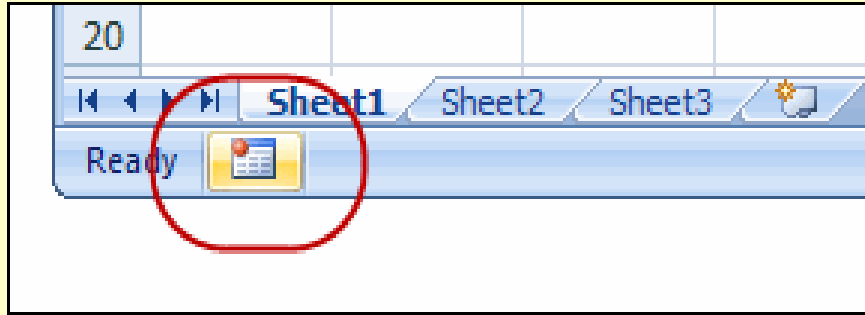
بعد الانتهاء من تنفيذ المطلوب بالكامل
نقوم بالضغط على ايقاف التسجيل
كما بالصورة التالية



ملاحظات

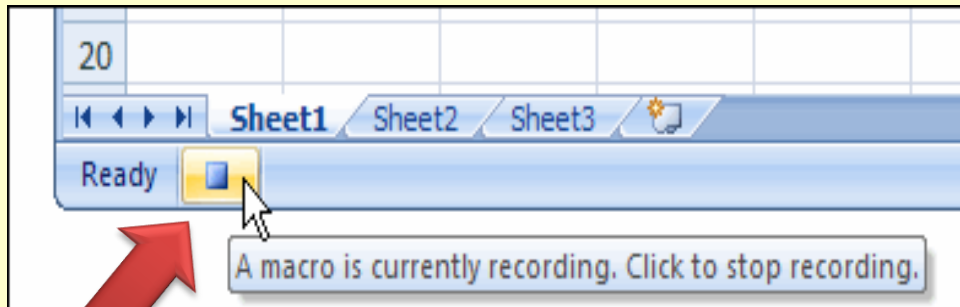
قد تظهر علامه تسجيل الماكرو بأسفل الشيت نقوم بالضغط عليها لتسجيل الماكرو

كما بالصورة



ثم بعد الانتهاء من تنفيذ المطلوب نقوم بالضغط على ايقاف التسجيل

كما بالصورة



النتائج والملاحظات

لقد تم تسجيل الماكرو
ونبدأ الآن في معرفة تفاصيل الكود الذي قمنا بعمل خطواته من خلال عملية
التسجيل ومعرفة بعض الأمور الأساسية والهامة للكود البرمجي

للدخول الى الصفحة التي بها الكود البرمجي والذي تم تسجيله تلقائيا في موديول
نقوم بالضغط على ALT+F11 معا

تفتح لنا صفحة تسمى صفحة محرر الأكواد
كما بالصورة

الكود بشكل أوضح الذي تم تسجيله كالتالي

الملاحظات :

١ - يبدأ الكود بكلمه Sub ثم اسم الماكرو ثم هذين القوسين المغلقين () الكود دائما يبدأ بذلك الأمر وخاصة إذا كان داخل موديول وهناك أكواد تكتب ولا تسجل عن طريق الماكرو تبدأ بخلاف ذلك وتلك التي توجد داخل حدث الصفحة (الشيت) أو داخل حدث الملف
This Workbook
 وسيأتى الحديث عن ذلك بالتفصيل فيما بعد ان شاء الرحمن داخل تلك الدروس

٢ - هناك داخل الكود علامه ((')) وهى دائما تكون مصبوغة باللون الأخضر وتسمى علامه اقتباس أحادية تشير الى تعليق يمكنك كتابته يدويا بأى لغة
الفائدة منها : وهذا التعليق يكون خاص إما بشرح جزء من الكود أو تشير الى تعليمات برمجيه معينه
 يتجاهلها الكود تماما أثناء عملية تنفيذ الكود وليس لها أى تأثير مطلق على الكود

٣ - ينتهى الكود بجمله End Sub وهذه الجملة هامه جدا لإنهاء عمل الكود
 ٤ - يمكننا هنا ونحن داخل الموديول الذى به الكود البرمجى الذى تم تسجيله تلقائيا بناء على أننا قمنا بتنفيذ الخطوات الخاصة بالمثال يدويا

- تغيير اسم الكود الى أى اسم .. أى نقوم بتغيير كلمه Module1 الى أى كلمه أخرى قد تكون معبرة عن طبيعة عمل الكود أو حتى يمكننا وضع اسمنا باللغة الإنجليزية طبعاً دون ترك مسافات ويمكننا الاستعانة بوضع تلك العلامه (_) التى تسمى Under Scour
- يمكننا التعديل والإضافة داخل الكود البرمجى ايضا فى تغيير اسماء الأعمده والمدى وخلافه أو أى أمر آخر كما يتراء لنا وسيأتى فيما بعد ذكر هذا الأمر فى الدروس القادمة بشيء من التفصيل

الدرس الثالث

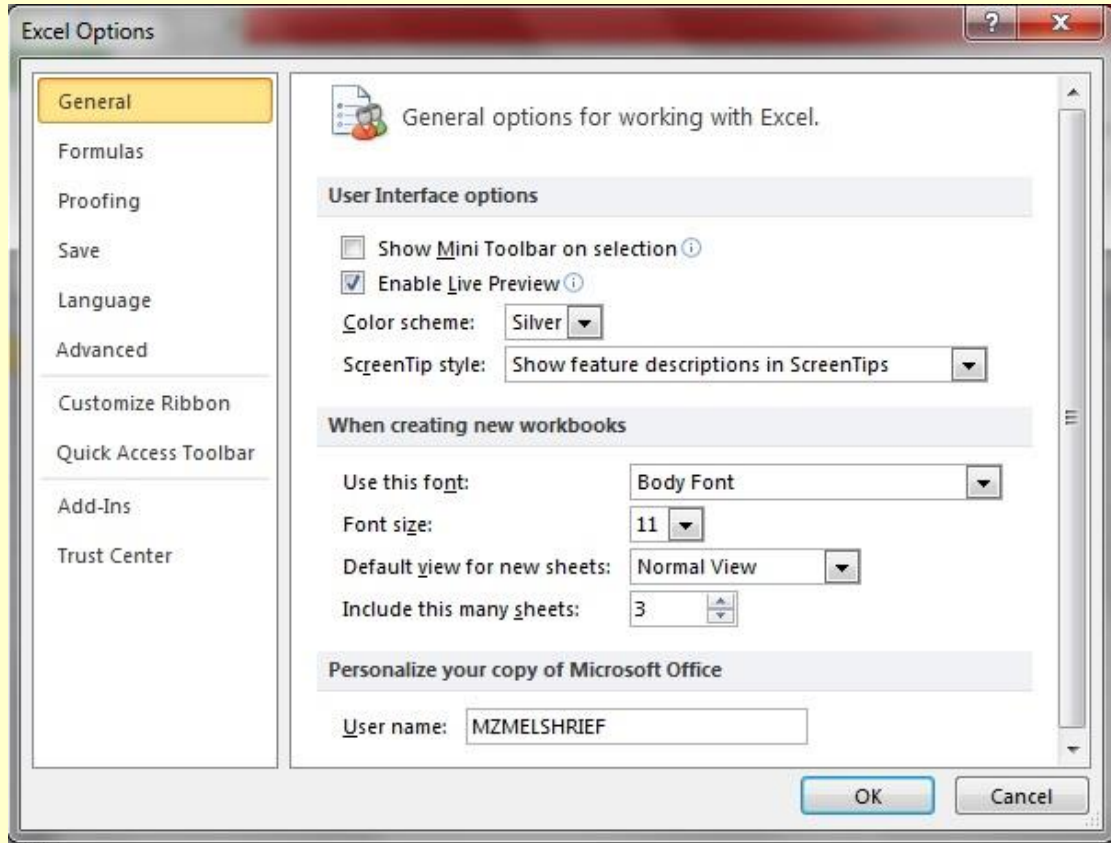
إعدادات الأمان

لغة VBA هي لغة برمجة وهذا يعنى أنه يمكن لبعض المبرمجين أن يستغلها استغلالا سيئا فى إنشاء برامج ضاره تضر بالحاسب الآلى الذى يستخدم هذا البرنامج وعليه قررت شركة مايكروسوفت إضافة خاصية الأمان لحماية المستخدم وللدخول إلى خيارات الأمان نتبع الخطوات التالية :

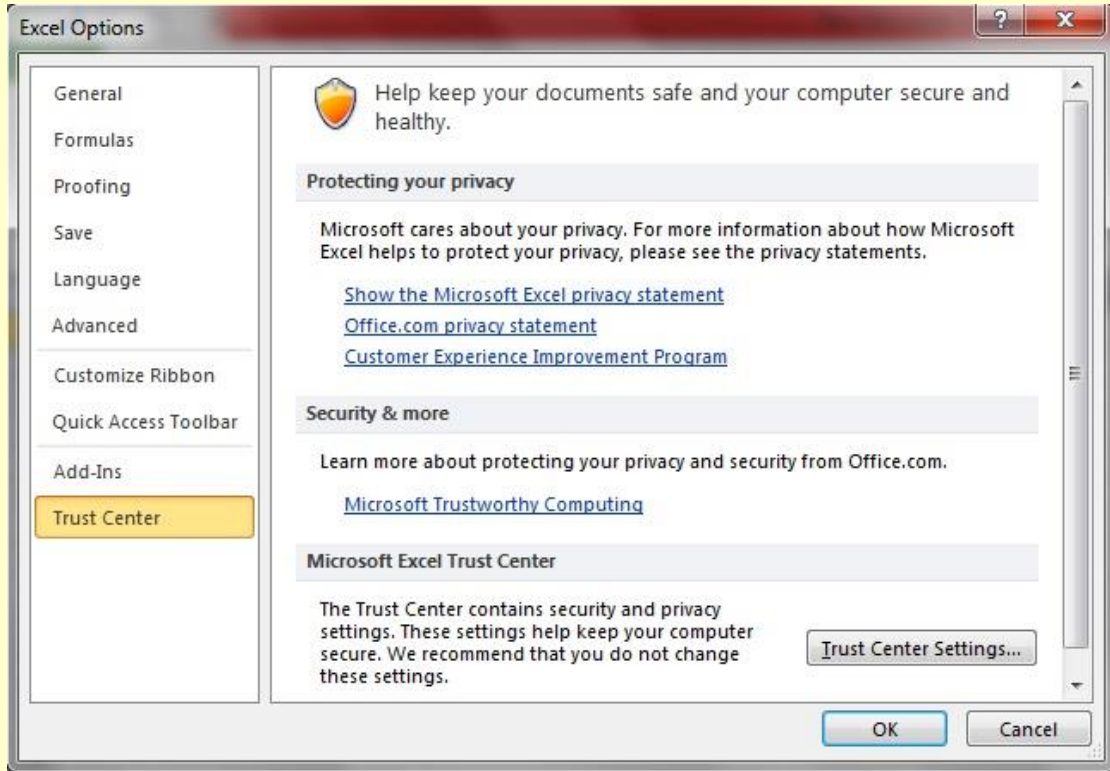
بأوفيس 2007 وما فوق
نضغط على زر برنامج الأوفيس كما بالصورة



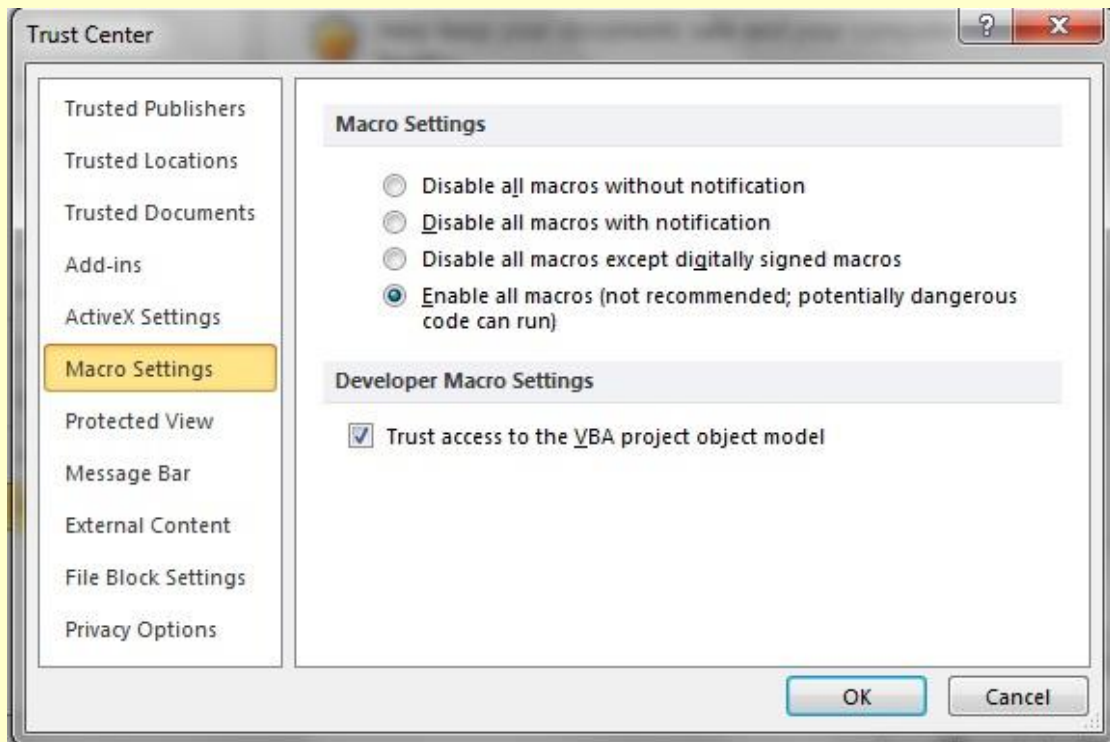
ثم نضغط على Excel Option تظهر لنا نافذة كما بالصورة



ثم نختار Trust Center من القائمة تظهر لنا نافذة أخرى كما بالصورة



ثم نضغط على Trust Center Settings تظهر لنا نافذة كما بالصورة

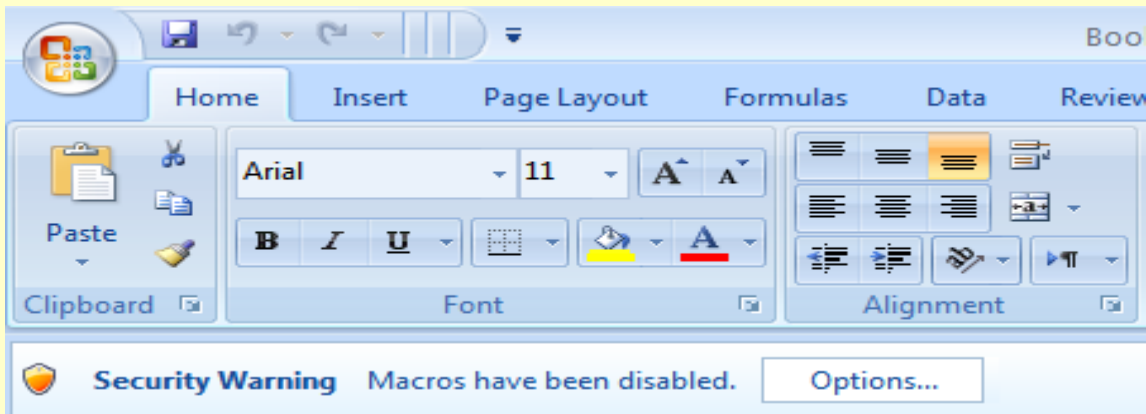


نجد هنا قسمين بالنافذة الأعلى (الصورة السابقة)

الجزء الأول به عدد (٤) اختيارات
ومعناهم كالتالى :

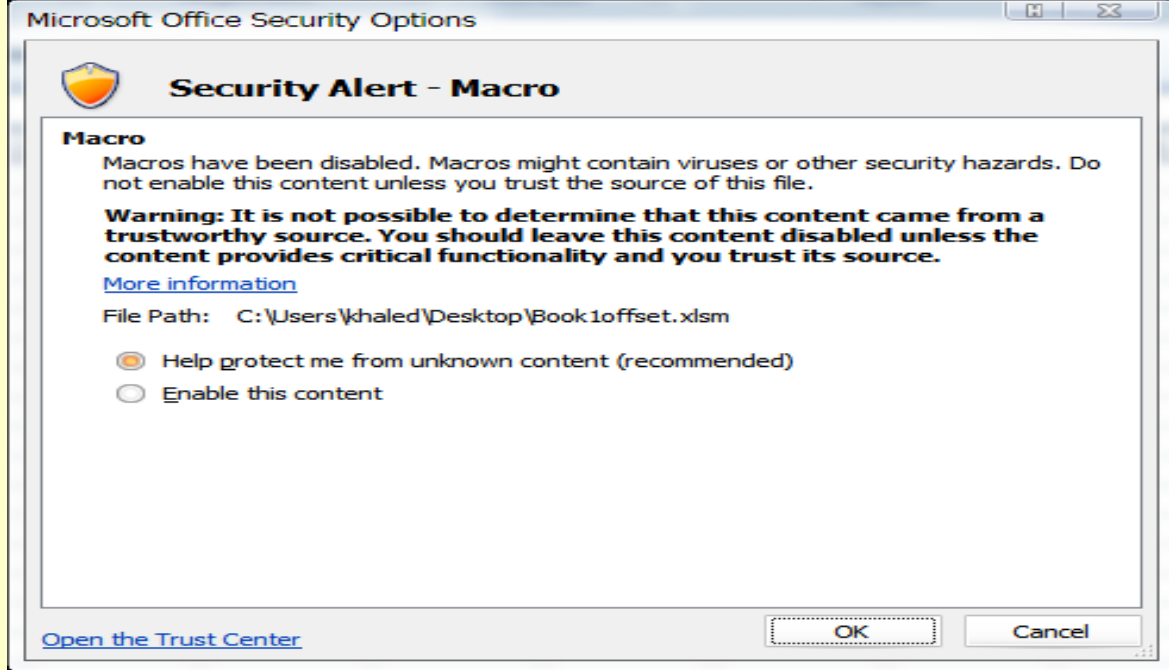
Disable all macros without notification	منع اكسل من تشغيل برامج VBA الموجوده بالملف
Disable all macros with notification	تنبيه المستخدم بوجود برامج VBA ويسمح للمستخدم بتشغيلها وسيلى شرحها بالتفصيل
Disable all macros except digitally signed macros	لا يسمح بتشغيل برامج VBA لا تحمل توقيعاً (كلمة توقيع تعنى ان النسخة الخاصة بالأوفيس نسخة أصلية وتحمل رقم) وان كان البرنامج يحمل توقيعاً يعطى تنبيهها مثل الخيار السابق الذى سيلى شرحه واذا سمح المستخدم لبرنامج ما يحمل توقيعاً يسمح له تلقائياً فيما بعد بتشغيل جميع البرامج التى تحمل نفس التوقيع
Enable all macros	السماح باستخدام جميع البرامج دون تنبيه او سؤال المستخدم

اختيار الخيار الثانى او الثالث
عندما يقوم المستخدم بفتح البرنامج تظهر نافذة الاكسيل كما بالصورة
التالية



وبالضغط على زر Options

تظهر لنا نافذة كما بالصورة التالية



فإن كنا نثق بالبرنامج والمصدر نضغط على الخيار الثانى وهو

Enable this content

ونضغط على **OK**

الدرس الرابع

VBA من خلال الرسائل



نتحدث هنا عن استخدام بعض التقنيات التي تجعل استخدامنا
للماكرو_ (برمجه اكسل) _ أكثر مرونة .. وذات طابع احترافي
وأكثر جمالاً

وذلك من خلال استخدام الرسائل من خلال كتابة بعض الأكواد
البسيطة
السؤال ؟

أين تكتب تلك الأكواد ؟

تكتب تلك الأكواد داخل حدث الشيت

تكتب داخل حدث المصنف

تكتب داخل موديول

تكتب داخل أكواد اليوزر فورم

أى تكتب حسب الحاجة لها على أن تكون داله على الهدف منها
سواء كان تحذير من شيء أو طرح سؤال بسيط

مثال (١)
نريد أن نتأكد من أن الطابعة في وضع التشغيل وإعطاء المستخدم
فرصة للتأكد من ذلك مع توقف الماكرو عن العمل الى أن يقرر
المستخدم الضغط على زر موافق أو OK

التالى نص الرساله التى ستظهر
كما بالصورة

MsgBox "Please make sure that the printer is switched on"



جدير بالذكر
عند الملاحظة الأولى لشكل الرساله كما بالصورة السابقة
نجد نص الرساله
ثم زر OK

اذا الإستنتاج هنا يدفعنا الى ذكر سؤال هام لمعرفة كيفية ظهورها
بهذا الشكل

س : مما تتكون الرسالة ؟ وكيف تكتب ؟

ج : تتكون الرسالة من هذه التركيبة التالية

MsgBox (prompt [, buttons] [, title] [, helpfile, context])

السؤال الذى يطرح نفسه هنا

س : ماذا تعنى هذه التركيبة ؟

ج : كالتالى

كلمة Prompt تعنى ان تكون الرسالة سريعة أى عند حدوث أمر ما تظهر الرسالة سريعا

Buttons	تعنى الأزرار .. وهو أمر اختياري .. يشمل تركيبة الزر وشكل الأيقونة المصاحبه له التى تظهر داخل نافذة صندوق الحوار .. وهو تعبير رقمى سنتحدث عنه لاحقا
Title	تعنى العنوان الذى يظهر أعلى صندوق الحوار وهو اختياري ايضا ويعبر عنه بكلمات تدل على الهدف من الرسالة واذا لم يكتب تلقائيا يتم وضع اسم التطبيق فى شريط العنوان
Helpfile	تعنى ملف التعليمات ويعبر عنه بكلمات تهدف الى تقديم المساعدة للمستخدم وهو ايضا اختياري
context	تعنى السياق الخاص بالتعليمات التى يسجلها المبرمج ويعبر عنه رقميا وسنتحدث عنه لاحقا

هناك أيضا أمور يجب ذكرها :

١ - طول الرسالة يتكون من ١٠٢٤ حرف

٢ - واذا كانت الرسالة تتكون من عدة أسطر يجب الفصل بين كل سطر بعلامه

(& _)

٣ - هناك معيار هام في الرسائل عندما مثلا تريد تحذير المستخدم الى ان الطابعه تعمل مع التأكيد على ذلك بزر OK
هنا نستخدم الفواصل وهو امر ضرورى كمثل الرساله التاليه

MsgBox “Is the printer on?”, , “Caution!”

وكما الصورة



القيم الرقمية

التي سبق الإشارة عنها

Constant	Value	Description
<i>vbOK</i>	1	OK
<i>vbCancel</i>	2	Cancel
<i>vbAbort</i>	3	Abort
<i>vbRetry</i>	4	Retry
<i>vbIgnore</i>	5	Ignore
<i>vbYes</i>	6	Yes
<i>vbNo</i>	7	No

Constant	Value	Description
vbOKOnly	0	
vbOKCancel	1	
vbAbortRetryIgnore	2	
vbYesNoCancel	3	
vbYesNo	4	
vbRetryCancel	5	
vbCritical	16	
vbQuestion	32	
vbExclamation	48	
vbInformation	64	
vbDefaultButton1	0	First button is default
vbDefaultButton2	256	Second button is default
vbDefaultButton3	512	Third button is default
vbDefaultButton4	768	Fourth button is default
vbApplicationModal	0	يجب أن يستجيب المستخدم للرسالة قبل متابعة العمل
vbSystemModal	4096	تعطيل كافة التطبيقات حتى يستجيب المستخدم للرسالة
vbMsgBoxHelpButton	16384	إضافة زر تعليمات في الرسالة
vbMsgBoxSetForeground	65536	تحديد إطار الرسالة كإطار أمامي
vbMsgBoxRight	52488	محاذاة النص الى اليمين
vbMsgBoxRtlReading	1048576	إظهار الرسالة من اليمين الى اليسار كالمعمول به في الأنظمة العربية (تعريب)

فمثلا إذا كنت تريد القيمة 4 والقيمة 32

يكون النص الذي يكتب فى الكود هكذا

MsgBox Prompt:="Delete this record?", Buttons:=36

وتظهر الرساله كما بالصورة التالية



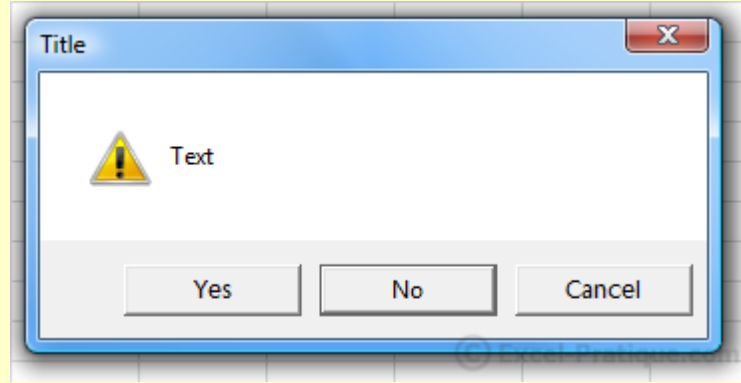
أو يكتب النص هكذا

**MsgBox (Prompt:="Delete this record?",
Buttons:=vbYesNo + vbQuestion)**

وإذا أردنا كتابة الرسالة بالترتيب الخاص بها كما ذكرنا بكيفية كتابة
الرساله فتكون هكذا

**MsgBox("Text", vbYesNoCancel +
vbExclamation + vbDefaultButton2, "Title")**

فتظهر الرسالة كما بالصورة التالية



هنا السؤال يطرح نفسه

هل يمكن كتابة كل هذه الأزرار فى الرسالة ؟

الإجابة بالقطع ((لا))

- لمجموعة القيم من (1:5) تمثل الأزرار التى ستظهر فى الرسالة وعلى حسب ما تريد وتختار
- ومجموعة القيم (16,32,48,64) تمثل نوع الرسالة من كونها استفهام أو تعجب .. الخ وعلى حسب ما تختار
- ومجموعة القيم (0,256,512,768) تعنى أى الأزرار YES أو NO أو CANCEL تريدها أن تكون الافتراضية للمستخدم أى المضيئة

يمكن التعبير عن الرسالة فى الكود بالقيمة مثل وتظهر الرسالة كما بالصورة السابقة

```
MsgBox("Text", 3 + 48 + 256, "Title")
```

ملحق الدرس الرابع الخاص بالرسائل

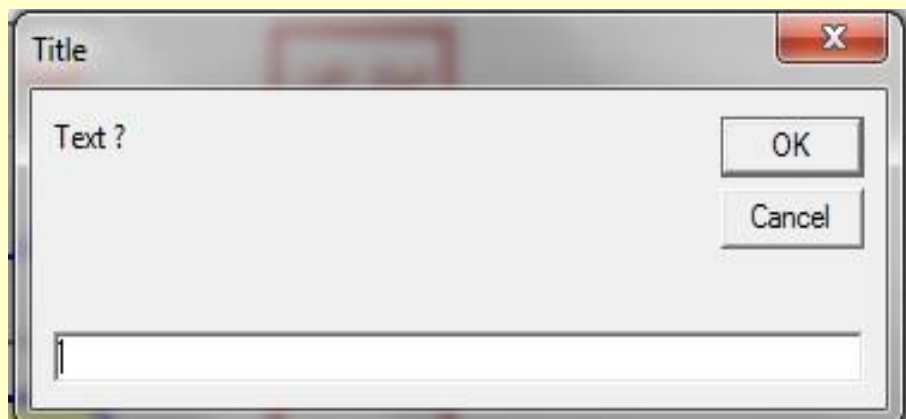
نستعرض هنا الرسالة التي تحتوى على مربع ادخال بيان
قد يكون اسم ما
وقد يكون كلمة سر للدخول

فيما يختص بالجزئية الأولى

يكون كود الرسالة بهذا الشكل
وهذا الكود على سبيل المثال

```
Sub example()  
  Dim result As String  
  result = InputBox("Text?", "Title")  
  
  If result <> "" Then  
  End If  
End Sub
```

فتكون الرسالة بهذا الشكل
كما بالصورة التالية



وإذا افترضنا أننا نريد أن يجعل الأكسل يظهر الاسم أو الكلمة التي ستكتب داخل هذا المربع

فيكون الكود بهذا الشكل

```
Sub example_1 ()
Dim result As String

result = InputBox("Text ?", "Title", "write the name")

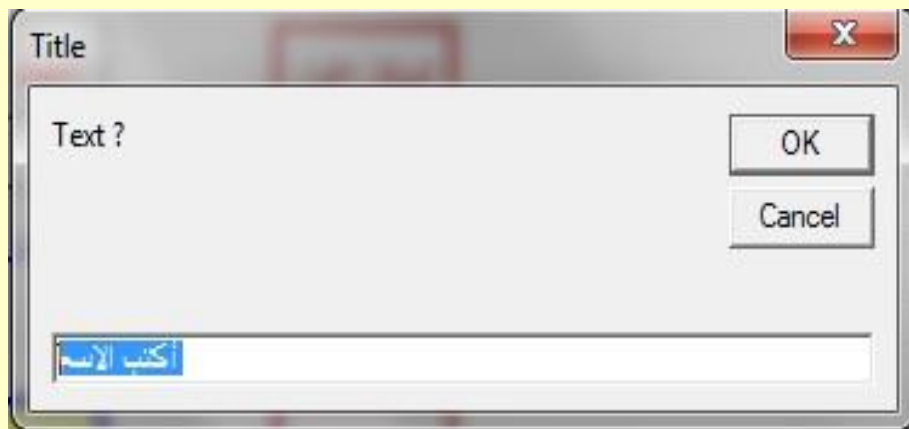
If result <> "" Then
MsgBox result
End If
End Sub
```

الاختلاف بين الكودين في اضافة هذه الجملة

Msgbox result

وهي تعنى ظهور رساله تاليه بها الجملة التي تم ادخالها بالمربع

فتظهر الرساله
كما بالصورتين التاليتين



بعد كتابه الاسم على سبيل المثال
تظهر الرساله بالشكل التالي



نأتى لجزئية كلمة السر

لقد جعلت الكود خاص بالمثل العملى المرفق
يختص بالدخول الى الشيت الثانى
وقد سجلت الكود بمحرر الأكواد الخاص بحدث الشيت الثانى
أى داخل الصفحة نفسها من خلال الفيچوال بيسك

وان شاء الرحمن سوف نتطرق لأمر الأكواد بشىء من التفصيل فى
الدروس القادمة وما هو الفرق بين الكود داخل موديول أو بحدث الشيت
أو بحدث المصنف بالكامل

والكود الخاص برساله كلمه السر
كالتالى

```

Private Sub Worksheet_Activate()

xx:

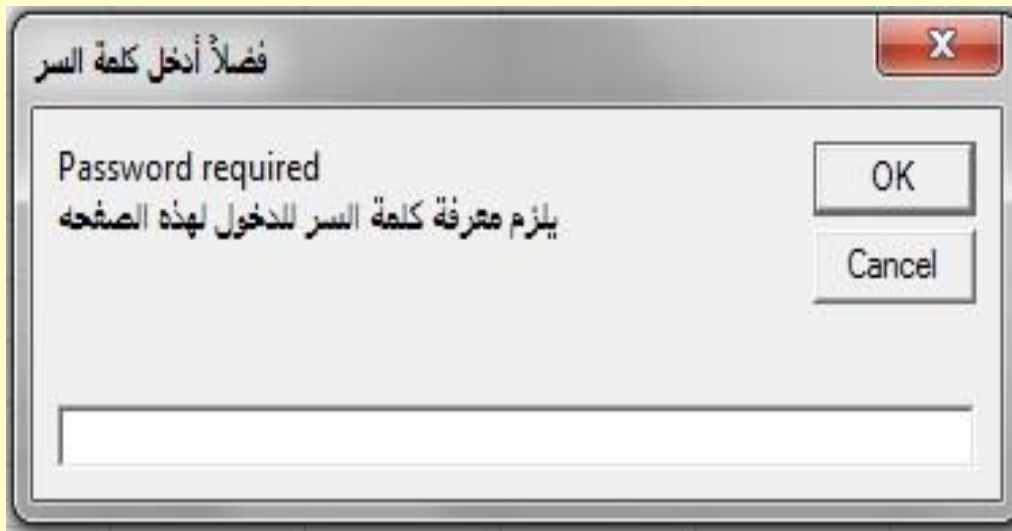
Dim x
x = InputBox("Password required" & Chr(13) & "يلزم معرفة كلمة السر للدخول لهذه الصفحة", "فضلاً أدخل كلمة السر")
If IsNull(x) Or x = "" Then GoTo xx

If x = "بسم الله" Then
MsgBox "كلمة السر تم قبولها تفضل لتنفيذ العملية"
Else
MsgBox "Wrong Password" & Chr(13) & "!! عفواً لم تدخل كلمة السر الصحيحة سيتم العوده بك للصفحة الرئيسيه !!", vbOKOnly
Sheets("sheet1").Activate
End If

End Sub

```

ف عند محاولة الدخول للشيت الثاني تظهر لنا الرساله كما بالصورة التالية



وعند كتابة كلمة السر وهي
بسم الله
 بين الكلمتين مسافة واحده فقط
 تظهر لنا رساله بهذا الشكل



وبالضغط على زر OK يتم الدخول الى الشيت الثانى

فى حال كوننا لم ندخل كلمة السر الصحيحة
 تظهر لنا رساله توضح ذلك
 وتنتقل مباشرة الى الشيت الأول
 والرساله كما بالشكل التالى



مقدمة عن البرمجة بلغة VBA في برنامج إكسيل

يتميز برنامج EXCEL بالواجهة الرسومية شانه شأن كافة البرامج التي تعمل تحت نظام التشغيل Windows وهذه الواجهة الرسومية تخفى خلفها الأوامر التي أصدرها للحاسب والتي تكون عادة مكتوبة بلغة من لغات الحاسب التي تعتمد على ما يسمى بالبرمجة الحديثة (أو البرمجة الموجهة بالأحداث) Events Programming تعتمد على حدوث حدث بواسطة المستخدم – مثل الضغط بزر الماوس الأيسر على زر تخيلي على الشاشة وبناء على هذا الحدث يتم تنفيذ مجموعة من الأوامر كما هو الحال في وحدات الماكرو التي سبق الإشارة إليها .

والواقع ان هذه الأزرار تخفى خلفها التعليمات التي تعطى للحاسب ، من خلال لغة الـ Visual Basic حيث يتم كتابتها بطريقة محددة وفقا لقواعد اللغة وكل مجموعة وهذه التعليمات تسمى الاكواد Codes وكل إجراء يرتبط تنفيذه بحدث معين يحدث من الأكواد او التعليمات تسمى إجراء Procedure

من قبل المستخدم مثل الضغط بزر الماوس الأيمن او الأيسر على زر او تحريك الماوس او فتح كائن أو إغلاقه

وتعتبر وحدات الماكرو احد الأدوات المستخدمة بواسطة Excel لكي يقوم المستخدم بتخزين مجموعة من التعليمات او الاكواد وتنفيذها من خلال حدث معين دون الحاجة إلى كتابة هذه التعليمات بنفسه بواسطة لغة Visual Basic ودون الحاجة الى تعلم قواعدها.

ونتناول في هذا الفصل كيفية التعامل مع أوامر لغة Visual Basic من خلال برنامج Excel وذلك بهدف التعرف على كيفية تصميم برامج تطبيقية قوية باستخدام Excel تتمتع بخواص وإمكانيات متقدمة يسهل من خلالها إدخال البيانات والتعامل معها.

نبدأ على بركة الله تعالى
في شرح بعض الأمور الخاصة بالفيجوال بيسك اكسل
وهي في غاية الأهمية
كونها توضح لنا العديد من الأمور المستخدمة في الأكواد
وبرجاء قرائتها بتأني وبدقه لإستيعاب ما بها من نقاط هامه
جدا

التعامل مع محرر الكود في الفيچوال بيسك اكسل

١ – فتح Visual Basic من خلال برنامج الأكسل :

سبق الإشارة الى كيفية فتح محرر الأكواد راجع درس
الماكرو

٢ – إنشاء الإجراء Procedure :

الإجراء Procedure هو مجموعة الجمل أو الأوامر التي
يتم كتابتها لتنفيذ مهمة معينة .
ولإنشاء الإجراء يستخدم الأمر sub ويعنى ذلك إنشاء
إجراء جديد ويعقب ذلك اسم الإجراء ويكون من اختيار
المبرمج أو المستخدم فنجد أن البرنامج يكتب تلقائيا جملة
End sub
على أن يتم كتابة الأوامر الخاصة بالإجراء بين جملتي
الإعلان والنهاية .

٣ – تنفيذ الإجراء :

يتم تنفيذ الإجراء من خلال ورقة العمل بأحد طريقتين:

الأولى : من خلال تنفيذ أمر أدوات / ماكرو / وحدات ماكرو واختيار الإجراء الذى تم حفظه ويتم حفظ الإجراء ضمن وحدات الماكرو بالاسم الذى تم اختياره والضغط على زر تشغيل .

الثانية : عمل زر خاص لتنفيذ الماكرو باستخدام أشرطة أدوات النماذج واختيار الأمر إنشاء زر وتخصيص الزر للماكرو المحدد ويتم تنفيذ الماكرو بالضغط على هذا الزر .

٤ – المتغيرات Variables :

يقصد بالمتغير مكان يتم حجزه فى ذاكرة الحاسب يتم فيه تخزين قيمة او مجموعة من القيم لكى يتم استخدامها داخل البرنامج .

ولابد من إعطاء اسم لكل متغير حتى يتسنى فيما بعد استخدام هذا المتغير فى الوصول الى النتائج المطلوب تحقيقها من البرنامج .
وهناك شروط محددة يجب توافرها فى اسم المتغير وإلا تم رفضها من قبل البرنامج .

شروط تسمية المتغيرات :

- ١ - يجب ألا يبدأ اسم المتغير برقم وأن يبدأ بحرف .
- ٢ - الحروف التى تلى الحرف الأول يمكن أن تكون حروف أو أرقام أو الاثنين معا .
- ٣ - لا يجوز أن يحتوى اسم المتغير على مسافات أو نقطة أو رموز خاصة مثل (؟ ، * ، / ، الخ) ولكن يمكن استخدام الشرطه السفليه (_) .
- ٤ - ألا يكون اسم من الأسماء المحجوزة مثل

(dim, double, Else, if , الخ)
 ٥ - يجب ألا يكون مستخدماً من قبل.
 بمعنى لو هناك كود بإسم MZM فلا يجوز كتابه كود
 آخر داخل نفس المصنف بنفس الاسم .

وتنقسم المتغيرات إلى الأنواع الآتية :

نوع البيان	معناه	حجمه في الذاكره
Byte	عدد صحيح صغير	واحد بايت
Short	عدد صحيح	٢ بايت
Integer	عدد صحيح	٤ بايت
Long	عدد صحيح طويل	٨ بايت
Single	عدد حقيقي صغير	٤ بايت
Double	عدد حقيقي طويل	٨ بايت
Date	تاريخ	٨ بايت
Variant	صحيح أو كسر أو عمله	١٦ بايت
Boolean	قيمة منطقية	٢ بايت
Char	حرف واحد فقط	٢ بايت
String	سلسلة من الحروف	١٦ بايت + بايت لكل حرف

أمثلة على أسماء صحيحة للمتغيرات :

– mynameisyasser – Mahmoud – Salma
 Y2011 - Howareyou

أمثلة على أسماء غير صحيحة للمتغيرات :

- Howareyou? - Good Morning - Ali Gomaa 1210

٥ - الثوابت Constants :

عرضنا فيما سبق ما يخص المتغيرات وأنواعها وشروط تسميتها ، والثوابت هي مثل المتغيرات والفرق الوحيد بينهما هو أن الثوابت لا تتغير قيمتها أثناء عمل البرنامج وأيضا نجد أن شروط تسمية الثوابت هي نفس شروط تسمية المتغيرات وأيضا أنواع الثوابت هي نفس أنواع المتغيرات .

٦ - الإعلان عن المتغيرات :

الإعلان عن المتغير هو أن نخبر لغة visual basic عن اسم المتغير ونوع البيان وهناك طريقة للإعلان عن المتغيرات وهي بكلمة dim .

وهناك أساليب مختلفة لعملية الإعلان وهي كالتالي :

١ - الإعلان عن متغير واحد فقط في كل سطر من سطور الإعلان
مثل :

Dim a as integer

Dim b as single

٢ - الإعلان عن أكثر من متغير في سطر الإعلان مثل :

Dim a, b as integer

٣ - الإعلان عن متغير وتخصيص قيمة له في سطر الإعلان :

Dim C as integer = 7

٤ - الإعلان عن متغيرات ليست من نفس النوع وإعطاء قيمة لها في نفس السطر

Dim X as integer = 9 , Y as string = "Mr. Mzm EL-Shrief "

٥ - الإعلان عن متغير وتخصيص قيمة له من ناتج معادلة رياضية في نفس السطر

Dim X as integer = a + b

٧ - تخصيص القيم للمتغيرات :

عند تخصيص قيمة للمتغير فإن هذه القيمة تأخذ أحد الأشكال التالية :

- ١ - قيمة مجردة $X = 3$
- ٢ - متغير آخر $X = Y$
- ٣ - عملية حسابية $X = a + b$

ملاحظات :

• عند الإعلان عن متغير ولم يتم وضع قيمة له فإنه يتم افتراض قيمة له كالاتي:

- ١ - المتغيرات الرقمية تأخذ الرقم صفر.
- ٢ - المتغير المنطقي يأخذ القيمة False .
- ٣ - المتغيرات الحرفية تأخذ قيمة فارغة " " .
- ٤ - المتغير date يأخذ التاريخ ١٨٩٩/ ١٢ / ٣١ .

- القيمة الحرفية "" لا تساوى القيمة الحرفية " " حيث أن الأولى قيمة فارغة أما القيمة الثانية فهي عبارة عن مجموعة من المسافات والمسافة لها قيمة.
- يمكن جمع الأرقام وأيضا النصوص وذلك باستخدام علامات الربط & أو علامة +

مثال

Dim X , Y , Z as string

X = " Deiaa "

Y = " Mahmoud "

Z = X + Y

وعلى ذلك تكون قيمة Z هي

Deiaa Mahmoud

الأوامر والدوال فى لغة الـ Visual Basic

سنركز فى هذا الدرس على عدد من الجمل والدوال
ومن أهم هذه الجمل والدوال :

- ١ – جملة If – then
- ٢ – جملة Select Case
- ٣ – جملة For – next
- ٤ – جملة Do while loop
- ٥ – جملة Do loop while
- ٦ – دالة صندوق الرسائل Message Box وقد سبق
الحديث عنها بالدروس السابقة
ولكننا سنتحدث أيضا عنها بهذا الدرس .
- ٧ – دالة صندوق الإدخال Input Box
- ٨ – إنشاء الدوال الخاصة

الجملة الشرطية If – then

تستخدم هذه الجملة فى حالة وجود احتماليين منطقيين
كنتيجة لوجود شرط معين .
وتأخذ هذه الدالة الأشكال التالية:

١ – جملة IF ذات السطر الواحد

أمر Then الشرط IF

مثال :

If x = 5 then y = x

٢ – جملة IF ذات السطور المتعددة

IF الشرط Then

الأوامر

End IF

مثال :

If x = 5 then

Y = x

End IF

نلاحظ الآتى :

- لا يتم كتابه اى شىء بعد Then
- أنه تم اغلاق جملة IF بكلمة End IF

٣ – جملة If – then – else

IF الشرط Then

مجموعة الأوامر

Else

مجموعة الأوامر

End IF

مثال :

If x = 5 then
Y = x
Else
Y = z
End If

نلاحظ الآتى :

- لا يتم كتابة أى شىء بعد **Then**
- إذا تحقق الشرط يتم تنفيذ الأوامر الموجودة فى (السطر / السطور) التى تلى كلمة **Then**
- فى حالة عدم تحقق الشرط يتم تنفيذ الأوامر الموجودة فى (السطر / السطور) التى تلى كلمة **Else**
- يتم إغلاق جملة **IF** بكلمة **End IF**

٤ – جملة If – Then - Else if – Else

IF الشرط **Then**
 مجموعة الأوامر
Else IF
 مجموعة الأوامر
Else
 مجموعة الأوامر
End IF

نلاحظ التالى :

ذكرنا سابقا أن جملة **IF** تستخدم في حالة وجود احتماليين منطقيين كنتيجة لوجود شرط معين ولكن هذه الحالة الرابعة سوف تحتل أكثر من شرط وبالطبع كل شرط له احتماليين إما أن يتحقق أو لن يتحقق وسوف نلاحظ الآتي في هذه الحالة:

- لا يتم كتابة أى شىء بعد كلمة **Then**
- إذا تحقق الشرط يتم تنفيذ الأوامر الموجودة فى (السطر / السطور) التى تلى كلمة **Then**
- فى حالة عدم تحقق الشرط لا يتم تنفيذ الأوامر التى تلى كلمة **Then** ويتم الانتقال لاختبار الشرط الثانى المكتوب بعد كلمة **Else IF** ويلى الشرط مباشرة كلمة **Then** أيضا .
- إذا لم يتحقق الشرط الثانى ووجد شرط ثالث يتم كتابته بعد كلمة **Else IF** أيضا بنفس الطريقة السابقة فى الخطوة السابقة .
- وفى حالة عدم تحقق الشروط جميعا يتم تنفيذ الأوامر الموجودة فى (السطر / السطور) التى تلى كلمة **Else**
- تأتى كلمة **Else** فى نهاية كافة الشروط وهى آخر احتمال فى الجملة حيث يتم إغلاق الجملة بعدها نهائيا بكلمة **Else IF** ويجب ملاحظة ألا يسبق سطر **Else**

أى سطر به كلمة Else IF وإلا تسبب فى خطأ جملة IF بالكامل .

- يجب ملاحظة أننا فى إطار جملة IF واحدة وأن كلمة Else ليست مثل Else IF وأيضا IF ليست مثل Else IF حيث أن كل هذه الكلمات تدخل ضمن جملة IF الرئيسية وبالتالي فإنه يتم إغلاق جملة IF بكلمة End IF واحدة فقط.

مثال :

```

Dim x as double
x = text1.text
If x <50 then
    MsgBox ( " راسب " )
Elseif x >=50 and x <65 then
    MsgBox ( " مقبول " )
Elseif x >=65 and x <75 then
    MsgBox ( " جيد " )
Elseif x >=75 and x <85 then
    MsgBox ( " جيد جدا " )
Else
    MsgBox ( " ممتاز " )
End if

```

٥ - جملة IF المتداخلة (Nested if)

```

IF الشرط Then
    IF الشرط Then
        مجموعة الأوامر
    Else
        مجموعة الأوامر

```

End IF

Else

مجموعة الأوامر

End IF

نلاحظ التالي :

هذه الحالة مختلفة لأنه يوجد جملي IF وسوف يتم إغلاقهما
بكلمتي End IF

نشرح هذه الحالة الخامسة بالمثال التالي : وهو مثال هام

إذا أردنا اختبار اسم المستخدم وكلمة المرور عند الدخول
للموقع أو البرنامج أو الإيميل الشكل البسيط جدا لذلك
هو إخراج رسالة للمستخدم تفيد التالي (هناك خطأ في اسم
المستخدم أو في كلمة المرور)
إلا أن هذا لا يليق بمبرمج محترف
حيث يجب أن يوضح للمستخدم سبب الخطأ
هل في اسم المستخدم أو في كلمة المرور
وبالتالي فإن الرسالة يجب أن تكون كالتالي
(هناك خطأ في كلمة المرور)
وهذا معناه أن المستخدم قد أدخل اسم المستخدم بشكل
صحيح والعكس أيضا.

ويكون الكود كالتالي :


```

Dim username, password as string
Username = text1.text
Password = text2.text

If username = " Ilham" then
    If password = "123456" then
        Form2.show
        Me.hide
    Else
        MsgBox ("خطأ فى كلمة المرور",,"يوجد خطأ فى كلمة المرور")
    End If
Else
    MsgBox ("خطأ فى اسم المستخدم",,"يوجد خطأ فى اسم المستخدم")
End If

```

Select Case الجملة الشرطية

المثال السابق وهو قبل الأخير الذى ذكرناه فى جملة
If - then - elseif – else والخاص بتقديرات الطلبة
 يمكن تطبيقه بشكل أفضل وأسهل بكثير باستخدام جملة

Select Case

سوف نقوم بتطبيق المثال مرة أخرى باستخدام جملة
Select case ثم نقوم بالتعليق على المثال وشرح
الملاحظات:

```

Start :
Dim x as double
X = text1.text
On Error GoTo mistake
Select Case x

    Case 0 To 49
        MsgBox ( " راسب " )
    Case 50 To 64
        MsgBox ( " مقبول " )
    Case 65 To 74
        MsgBox ( " جيد " )
    Case 75 To 84
        MsgBox ( " جيد جدا " )
    Case 85 To 100
        MsgBox ( " ممتاز " )

End Select
Exit sub

Mistake :
Msgbox ( "من فضلك أدخل قيمة عددية" )
Go to start

End sub

```

الشرح :

سنتحدث عن ما يخص جملة **Select case**

١ - بدأت الجملة بكلمة **Select case** ثم اسم المتغير الذي سيتم تتبع حالته .

٢ - الحالة الأولى قيمة المتغير التي تم إدخالها في

صندوق النصوص هي بين صفر ، ٤٩

- ٣ - ظهور صندوق رسالة يفيد أن الطالب راسب
- ٤ - يتم إدخال قيمة كل حالة وهكذا في جميع الحالات بنفس الطريقة

الملاحظات :

١ - تبدأ الجملة بكلمة **Select case** وتنتهى بجملة **End select**

٢ - تم وضع الجملة الآتية فى بداية الكود

On Error GoTo mistake

ومعناها

ومعناها أنه فى حالة وجود أى خطأ يتم الانتقال إلى العنوان **Mistake** وذلك تحسباً لقيام المستخدم بإدخال قيمة نصية وبالتالي يصعب اختبارها وفى هذه الحالة سوف يتم نقل الكود للخطوة الأخيرة والتي يظهر بها صندوق نص يفيد أن المستخدم لم يدخل قيمة عددية وعليه إعادة إدخال قيمة عددية ليتم اختبارها.

وفى هذه الحالة سيتم إعادته مرة أخرى للبداية إلى العنوان **Start**

٣ - يمكن تسمية العناوين بأى اسم
مثال

(X - Y - Z - Yasser) ولكن يفضل أن يكون الاسم معبرا .

٤ - نلاحظ أننا وضعنا مدى في قيم المتغير في خطوات الجملة وهذا ليس شرطاً بل يمكن وضعه قيمة قيمة ولكن سيكون ذلك تطويلاً لا طائلاً منه مثل التالي

Case 1

Case 2

Case 3

Case 100

الدالة Inputbox

تستخدم في إدخال بيانات بواسطة المستخدم من خلال مستطيل يطلب منه إدخال البيان المطلوب .

مثل

إدخال معدل الضريبة الكلي يتم حساب الضرائب التي تستقطع من العاملين أو إدخال الراتب الأساسي لموظف حساب صافي مستحقاته بعد إضافة الحوافز والمكافآت وخصم الضريبة والاستقطاعات

وتأخذ هذه الدالة الشكل التالي:

الشكل العام للدالة Inputbox

("عنوان الصندوق" , " نص الرسالة ") = Inputbox متغير

مثال :

```

Start :
Dim x as double
X = Inputbox ( "إدخال درجة طالب " , "من فضلك أدخل الدرجة لحساب التقدير " )
On Error GoTo mistake
Select Case x
    Case 0 To 49
        MsgBox ( " راسب " )
    Case 50 To 64
        MsgBox ( " مقبول " )
    Case 65 To 74
        MsgBox ( " جيد " )
    Case 75 To 84
        MsgBox ( " جيد جدا " )
    Case 85 To 100
        MsgBox ( " ممتاز " )
End Select
Exit sub

Mistake :
Msgbox ( "من فضلك أدخل قيمة عددية " )
Go to start
End sub

```

التعامل مع الكائنات

الكائنات Objects

الكائنات هي المكونات الأساسية لاي برنامج وهي الأدوات التي تحتوى على البيانات او تستخدم للتعامل مع البيانات مثال ذلك الجداول والنماذج والاستعلامات والتقارير فى

Access

ويتضمن Excel الكائنات التالية :

- ١ – التطبيق Application ويتمثل في برنامج الإكسيل نفسه .
 - ٢ – المصنف Work Book ويتمثل في كتاب العمل الذي يتم التعامل مع البيانات من خلاله.
 - ٣ - ورقة العمل Work Sheet وهى التى تحتوى على مجموعة البيانات
 - ٤ – النطاق Range ويتضمن مجموعة من الخلايا Cells وهى التى تحتوى على بيان واحد من البيانات سواء كان بيان نصى أو رقمى أو معادلة أو دالة ويكتب النطاق على النحو التالى
Range(a1:c5) أى نطاق الخلايا من a1 الى c5
- ولكل كائن من هذه الكائنات خصائص معينة Properties مثل الاسم Name والعنوان Caption ومجموعة من الأساليب Methods التى تستخدم فى تنفيذ أمر معين على الكائن كما أن الكائن قد يقع عليه حدث يؤثر فى سلوكه ورد فعله

كائن التطبيق Application

```
Sub JASME1()
Application.Caption = "JASME MAHMOUD ELSHRIEF"
End Sub
```

يقوم الكود بتوظيف الخاصية Caption على كائن التطبيق أى أن هذه الخاصية تقوم بتغيير الاسم العام للبرنامج فبدلاً من كلمة مايكروسوفت سيظهر

"JASME MAHMOUD ELSHRIEF"

```
Sub JANA2 ()
Application.DisplayFormulaBar = False
End Sub
```

تقوم هذه الخاصية **DisplayFormulaBar** بإخفاء شريط الصيغ عند استخدام **False** وبالمقابل إظهاره عند استخدام **TRUE**

```
Sub JOUMANA3 ()
Application.DisplayFullScreen = True
End Sub
```

تقوم هذه الخاصية **DisplayFullScreen** بجعل التطبيق في وضع ملء الشاشة وللرجوع طبعاً يجب ان نجعل القيمة **False**

```
Sub MZM4 ()
Application.DisplayRecentFiles = False
End Sub
```

تقوم هذه الخاصية **DisplayRecentFiles** بعدم عرض المستندات الأخيرة في كائن تطبيق اكسيل وإذا أردنا إستعراضها طبعاً نجعلها تأخذ القيمة **TRUE**

```
Sub MZM5 ()
Application.WindowState = xlMaximized
End Sub
```

تقوم هذه الخاصية **WindowState** بعرض البرنامج في حالة التكبير

```
Sub MZM6 ()
Application.WindowState = xlMinimized
End Sub
```

تقوم هذه الخاصية **WindowState** بعرض البرنامج في حالة التصغير

```
Sub MZM7 ()  
Application.WindowState = xlNormal  
End Sub
```

تقوم هذه الخاصية **WindowState** بعرض البرنامج
في حالة العرض العادي

Worksheet كائن ورقة العمل

```
Sub MZM8 ()  
Worksheets("sheet1").Visible = False  
End Sub
```

يقوم الكود السابق بإخفاء ورقة العمل المسماة **sheet1**
وإذا أردنا إظهار الشيت نجعل القيمة **TRUE**

```
Sub MZM9 ()  
Worksheets(1).Name = "sheet1"  
End Sub
```


يقوم الكود بإعادة تسمية الورقة ١ في ترتيب اكسيل ب sheet1

```
Sub MZM10 ()
Worksheets("sheet1").Delete
End Sub
```

يقوم الكود بحذف الورقة المسماه sheet1

```
Sub MZM11 ()
Worksheets(1).Activate
End Sub
```

يقوم الكود بتنشيط الورقة المسماه sheet1

```
Sub MZM12 ()
Worksheets.Add
End Sub
```

يقوم الكود بإضافة ورقة جديدة

```
Sub MZM13 ()
Worksheets(3).Copy
End Sub
```

يقوم الكود بنسخ الورقة الثالثة ضمن ترتيب ورقات المصنف

كائن المدى / النطاق Range

نستعرض كائن من نوع النطاق أو ما يسمى بالمدى أو المجال وهو الأكثر شيوعاً أي أن أغلب الأكواد تشير بكثرة لهذا النوع من الكائنات

```
Sub MZM14 ()
Range("d2:i10").Select
End Sub
```

يقوم الكود بتحديد المجال المشار إليه بين القوسين

```
Sub MZM15 ()
Range("d2:i10").Columns(2).Select
End Sub
```

يقوم الكود بتحديد العمود الثاني بالمجال المشار إليه بين قوسين

```
Sub MZM16 ()
Range("b10:f15").Columns(2).Value = 0
End Sub
```

يقوم الكود بكتابة قيمة صفر في العمود الثاني من المجال المحدد بين القوسين

```
Sub MZM17 ()
Range("c5:c10").Rows(1).Value = 100
End Sub
```

يقوم الكود بكتابة قيمة ١٠٠ في الصف الأول من المجال المحدد بين القوسين وهو هنا الخلية C5

```
Sub MZM18 ()
Range("d2:i10").Cells(2, 3).Select
End Sub
```

يقوم الكود بتحديد الخلية الواقعة في الصف الثاني من العمود الثالث داخل المجال المحدد بين القوسين وهي الخلية F3

```
Sub MZM19 ()
Range("f10:i15").Cells(3, 2).Value = 200
End Sub
```

يقوم الكود بكتابة قيمة ٢٠٠ بالخلية الواقعة في الصف الثالث من العمود الثاني داخل المجال المحدد بين القوسين وهي هنا الخلية G12

```
Sub MZM20 ()
Worksheets(3).Range("f1:h5").Value = 100
End Sub
```

يقوم الكود بكتابة قيمة ١٠٠ في المجال المحدد من الورقة الثالثة من أوراق المصنف

```
Sub MZM21 ()  
Range("d2:i10").Clear  
End Sub
```

يقوم الكود بمسح محتويات المجال المشار إليه بين قوسين

```
Sub MZM22 ()  
Worksheets(3).Range("a1:c10").Font.Bold = True  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف قم باختيار
خاصية **Bold** سماكه الخط

```
Sub MZM23 ()  
Worksheets(3).Range("a1:c10").Font.Italic = True  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف قم باختيار
خاصية **Italic** إمالة الخط

```
Sub MZM24 ()  
Worksheets(3).Range("a1:c10").Font.Underline = True  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف قم باختيار
خاصية **Underline** وضع خط تحت البيانات

```
Sub MZM25 ()  
Worksheets(3).Range("a1:c10").Font.Name = " Arabic Typesetting"  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف قم باختيار نوع
الخط **Arabic Typesetting**

```
Sub MZM26 ()  
Worksheets(3).Range("a1:c10").Font.Size = 18  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف قم باختيار حجم الخط ١٨

```
Sub MZM27 ()  
Worksheets(3).Range("a1:c10").Columns(1).Font.Size = 18  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف بالعمود الأول فقط قم باختيار حجم الخط ١٨

```
Sub MZM28 ()  
Worksheets(3).Range("a1:c10").Rows.Font.Bold = True  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف لجميع صفوف المجال قم باختيار خاصية **Bold**

```
Sub MZM29 ()  
Worksheets(3).Range("a1:c10").Rows(3).Font.Bold = True  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف بالصف الثالث فقط قم باختيار خاصية **Bold**

```
Sub MZM30 ()  
Range("f20:i25").Cells.Font.Italic = True  
End Sub
```

عند المجال المحدد من الورقة الثالثة في أوراق المصنف بجميع الخلايا قم باختيار خاصية **Italic**

```
Sub MZM31 ()  
ActiveCell.Formula = "=C1+D2"  
End Sub
```

عند الخلية الحالية النشطة اكتب المعادلة المذكورة

```
Sub MZM32 ()
Range("h10").FormulaR1C1 = "=r[-1]c[-1]+r[1]c[1]"
End Sub
```

لإدراج المعادلة بمقدار إزاحة أقل من صف وعمود لليمين + إزاحة أكبر من صف وعمود لليساار للخلية المختارة وبالتالي سيكون الناتج معادلة

G9+I11

```
Sub MZM33 ()
Range("h10").FormulaR1C1 = "=rc[1]+rc[2]"
End Sub
```

لإدراج المعادلة بمقدار إزاحة أكبر من صف وعمود لليساار + إزاحة أكبر من صفين وعمودين لليساار للخلية المختارة وبالتالي سيكون الناتج معادلة

I10+J10

```
Sub MZM34 ()
Worksheets(3).Range("H10").Offset(1, 2).Value = 100
End Sub
```

لإدراج القيمة ١٠٠ بمقدار إزاحة زائد عمودين لليساار وصف من الخلية المختارة وهي ستكون الخلية **J11**

لاحظ الاختصار في الأكسيل فبدلاً من الأسطر المتعددة يمكن الإشارة للمجال المختار بهذه الجملة **WHIT**

وهي جملة تستخدم لتخصيص المجال أي عند المجال المعين بالخاصية المطلوبة افعل كذا

```
Sub MZM ()
With Worksheets(3).Range("A1:h10").Font
.Bold = True
.Italic = True
.Underline = True
.Name = "ARIAL"
.Size = 20
End With
End Sub
```

استخدمنا خاصية WHIT لتخصيص المجال المحدد بالخاصية المطلوبة FONT لعمل ما يلي من أوامر بعد سطر التخصيص يجب إنهاء جملة التخصيص WHIT بجملة End With

```
Sub MZM ()
Worksheets(5).UsedRange.Font.Size = 16
End Sub
```

تستخدم هذه الخاصية عند وجود بيانات متقطعة فهي تشير لكامل المجال المستخدم

```
Sub MZM ()
Worksheets(3).Range("a1").CurrentRegion.Font.Size = 16
End Sub
```

تستخدم هذه الخاصية عند وجود بيانات متقطعة أي البيانات المتصلة فقط بالمجال المذكور هي التي سيتم تنفيذ المطلوب عليها

```
Sub MZM ()
Worksheets(3).Range("C2:G10").BorderAround ColorIndex:=5
End Sub
```

يقوم الكود بإحاطة المجال المختار بحدود باللون الأزرق

```
Sub MZM ()
Worksheets(3).Range("C2:G10").Interior.ColorIndex = 6
End Sub
```

يقوم الكود بتلوين المجال المحدد باللون الأصفر

وعلى ذكر ColorIndex قد يتساءل البعض كيف يمكن معرفة رقم اللون المختار؟

وفيما يلي جدول بالرقم الخاص بكل لون قدر المستطاع

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56

مثال

```
Sub MZM ()
With Cells(1, "D")
.Value = " منتديات أوفيسنا التعليمية "
.Font.Bold = True
.Font.Name = "Arial"
.Font.Size = 72
.Font.Color = RGB(0, 0, 255)
.Columns.AutoFit
.Interior.Color = RGB(0, 255, 255)
.Borders.Weight = xlThick
.Borders.Color = RGB(0, 0, 255)
End With
End Sub
```

بالكود السابق

استخدمنا الخاصية **Cells** للإشارة إلى الخلية الأولى من العمود **D**

شرح الكود :

- عند الخلية الأولى من العمود **D** أعطها قيمة نصية " مننديات أوفيسنا التعليمية"
- تغليظ الخط قم باستخدام نوع الخط **Arial**
- حجم الخط ٧٢
- اختيار اللون الأزرق
- استخدم خاصية الاحتواء المناسب لضبط النص داخل الخلية
- جعل الخلفية باللون التركوازي
- إحاطة الخلية برسم حد ذي حد غليظ
- اختيار للحد لوناً أزرق
- قم بإنهاء جملة **WITH**

ويمكن الإشارة هنا إلى أن أنواع الـ **Borders.Weight** هي كما يلي

خط رفيع جدا	xlHairline
خط رفيع	xlThin
خط متوسط	xlMedium
خط غليظ	xlThick


```

Sub MZM ()
With Range("A1:A5,H1:H5")
.Value = "MAHMOUD ELSHRIE"
.Font.Bold = True
.Columns.AutoFit
End With

```

بالكود أعلاه :

- تم الإشارة لمجالين مختلفين مع جملة WITH
- تم كتابة قيمة نصية MAHMOUD ELSHRIE
- تم اختيار نوع الخط الغليظ
- تم عمل احتواء مناسب للأعمدة
- ثم إنهاء جملة WITH

جميع ما سبق الحديث عنه بالنسبة للمدى كان محدد ومعلوم ، أي أن الأبعاد ثابتة ومحددة مثل الخلية A1 المدى A1 : H10

فماذا لو كان المجال المطلوب العمل ضمنه هو مجال متغير الأبعاد ؟

إذا يجب علينا التعرف على ما يلي

جمل التخصيص SET AND LET

كلاهما تستخدم لتخصيص قيم المتغيرات المعن عنها ككائنات والفرق بينهما في أن جملة **SET** تستخدم لتخصيص المتغيرات من نوع كائن

أما **LET** فتستخدم لتخصيص القيم الرقمية أو الحرفية للمتغيرات الأخرى

مثال على جملة **SET** لأنها هي التي تهتمنا الآن :

```
Sub MZM()
Dim R1 As Range
Dim R2 As Range
Dim MAHMOUD As Range
Set R1 = Range("B2:C5")
Set R2 = Range("F2:J5")
Set MAHMOUD = Union(r1, r2)
With MAHMOUD
.Value = "MAHMOUD ELSHRIEF "
.Font.Size = 20
.Font.Bold = True
.Font.Italic = True
.Font.ColorIndex = 5
.Borders.Weight = xlMedium
.Borders.Color = RGB(0, 255, 255)
.Interior.ColorIndex = 6
.Columns.AutoFit
End With
End Sub
```

السطر الأول والثاني والثالث إعلان عن المتغيرات على أنها من النوع **RANGE**

السطر الرابع والخامس تخصيص المجال المذكور للمجال الأول والثاني **R1،R2** وذلك باستخدام جملة **Set**

السطر السادس تخصيص المجال **MAHMOUD** الذي نريده أن يعبر عن دمج المجالين **R1،R2** وذلك باستخدام الأمر **Union**

والفائدة من ذلك كله تكمن في السطر السابع ، فعند تخصيص المجال
MAHMOUD كمتغير من نوع نطاق ليُعبّر عن المجالين **R2،R1**
 أمكن لنا أن نقرنه بالجملة **WHIT** ليأخذ الخصائص المطلوبة في الأسطر
 الباقية من الكود

مثال آخر :

```
Sub MZM ()
Dim Z As Worksheet
Set Z = Application.Worksheets(2)
Z.Name = "منتديات أوفيسنا التعليمية"
End Sub
```

هنا تم تخصيص الكائن من ورقة عمل للمتغير **Z** وبالتالي عند استخدام
 خاصية **Name** مع المتغير **Z** أمكن تغيير أسم الورقة

باختصار فإن المتغير **Z** بفضل الجملة **SET** يعبر عن كائن ورقة العمل
 المحددة

المدى الديناميكي

لا يزال الحديث متصلاً بكائن النطاق **RANGE** والذي استغرق منا تقريباً
 معظم الحالات

ولكن الملاحظ فيما سبق أننا قد نحتاج من حين لآخر لزيادة أو إنقاص حجم مساحة العمل أي المدى الذي نريد العمل ضمنه ، وبالتالي نضطر لتعديل أبعاد المدى من داخل الكود

فماذا لو كان مجال العمل المدى متحرك أي ديناميكي ؟

عندها يصبح الكود والعمل المنجز أكثر فعالية ومفعماً بالحيوية

فمثلاً قد نعد تنسيقات لجدول بحجم ضخم ولكن لا نستخدم منه إلا الجزء اليسير وينتج عن ذلك

إرهاق لموارد البرنامج وبطء ملحوظ في كفاءة ملف العمل

والكود التالي يلخص فكرة المثال السابق :

```
Private Sub Worksheet_Change(ByVal Target As Range)
With Range("MYRANGE")
.Font.Size = 16
.Font.Bold = True
.Font.ColorIndex = 5
.Borders.Weight = xlMedium
.Borders.ColorIndex = 3
.Interior.ColorIndex = 36
.Columns.AutoFit
.HorizontalAlignment = xlCenter
.VerticalAlignment = xlCenter
.RowHeight = 25
End With
End Sub
```

وجميع اسطر الكود تقريباً تم شرحها سابقاً فيما عدا أن الكود لا يبدأ بكلمة SUB المعهودة وذلك لأنه مقترن بحدث التغيير بورقة العمل عند المجال MYRANGE الديناميكي

كذلك سطري الأوامر

```
.HorizontalAlignment = xlCenter  
.VerticalAlignment = xlCenter
```

واللذان يعنيان توسط البيانات أفقياً وعمودياً

أيضاً سطر الأوامر

```
.RowHeight = 25
```

والذي يعني جعل الصفوف متساوية بارتفاع ٢٥

الكائن Window

لتغيير التسمية التوضيحية لأسم النافذة للمصنف النشط في الشريط العلوي من البرنامج

```
Sub MZM ()
ActiveWindow.Caption = "منتديات أوفيسنا التعليمية"
End Sub
```

ويمكن أيضا جعلها بدون أي تسمية توضيحية

```
Sub MZM ()
ActiveWindow.Caption = ""
End Sub
```

لتنشيط نافذة بعينها مفتوحة في الخلفية

```
Sub MZM ()
Windows("Book1").Activate
End Sub
```

يمكن ان يكون خيار التنشيط باختيار التالي

```
Sub MZM ()
ActiveWindow.ActivateNext
End Sub
```

كما يمكن ان يكون خيار التنشيط باختيار السابق

```
Sub MZM ()
ActiveWindow.ActivatePrevious
End Sub
```

```
Sub MZM ()
ActiveWindow.NewWindow
End Sub
```

لإخفاء إطار نافذة المصنف الحالي النشط ، طبعاً يمكن إظهاره القائمة إظهار أو اختيار TRUE إذا كنت داخل محرر الأكواد

```
Sub MZM ()
ActiveWindow.Visible = False
End Sub
```

تكبير حالة النافذة الحالية للمصنف النشط

```
Sub MZM ()
ActiveWindow.WindowState = xlMaximized
End Sub
```

تصغير حالة النافذة الحالية للمصنف النشط

```
Sub MZM ()
ActiveWindow.WindowState = xlMinimized
End Sub
```

الوضع العادي لحالة النافذة الحالية للمصنف النشط

```
Sub MZM ()
ActiveWindow.WindowState = xlNormal
End Sub
```

لإغلاق نافذة المصنف الحالي النشط

```
Sub MZM ()
ActiveWindow.Close
End Sub
```

إعطاء زووم بقيمة معينة لنافذة ورقة العمل الحالية

```
Sub MZM ()  
ActiveWindow.Zoom = 80  
End Sub
```

إخفاء علامات التبويب لأوراق العمل بنافذة المصنف النشط

```
Sub MZM ()  
ActiveWindow.DisplayWorkbookTabs = False  
End Sub
```

إخفاء رؤوس الأعمدة والصفوف بنافذة ورقة العمل الحالية

```
Sub MZM ()  
ActiveWindow.DisplayHeadings = False  
End Sub
```

إخفاء شريط التمرير الأفقي بنافذة المصنف النشط الحالي

```
Sub MZM ()  
ActiveWindow.DisplayHorizontalScrollBar = False  
End Sub
```

وللتحكم بشريط التمرير الأفقي من حيث الزيادة والنقصان يمكن التدرج بالشكل التالي

```
Sub MZM ()  
ActiveWindow.TabRatio = 1  
End Sub
```



```
Sub MZM ()  
ActiveWindow.TabRatio = 0.5  
End Sub
```

```
Sub MZM ()  
ActiveWindow.TabRatio = 0  
End Sub
```

إخفاء شريط التمرير العمودي بنافذة المصنف النشط الحالي

```
Sub MZM ()  
ActiveWindow.DisplayVerticalScrollBar = False  
End Sub
```

عرض جميع الصيغ بنافذة ورقة العمل النشطة

```
Sub MZM ()  
ActiveWindow.DisplayFormulas = True  
End Sub
```

إخفاء خطوط الشبكة بنافذة ورقة العمل النشطة

```
Sub MZM ()  
ActiveWindow.DisplayGridlines = False  
End Sub
```

إعطاء لون محدد لخطوط الشبكة بنافذة ورقة العمل النشطة

```
Sub MZM ()  
ActiveWindow.GridlineColor = QBColor(14)  
End Sub
```

لعمل معاينة لفواصل الصفحات للورقة النشطة حالياً

```
Sub MZM ()  
ActiveWindow.View = xlPageBreakPreview  
End Sub
```

وللعودة لوضع العرض العادي بالورقة النشطة حالياً

```
Sub MZM ()  
ActiveWindow.View = xlNormalView  
End Sub
```

لإخفاء القيم الصفرية بنافذة ورقة العمل النشطة

```
Sub MZM ()  
ActiveWindow.DisplayZeros = False  
End Sub
```

لقراءة اسم الورقة الحالية في شكل Message Box

```
Sub MZM ()  
MsgBox Windows(1).ActiveSheet.Name  
End Sub
```

لقراءة عنوان مدى محدد تم اختياره ليظهر العنوان في شكل Message Box

```
Sub MZM ()  
MsgBox ActiveWindow.RangeSelection.Address  
End Sub
```

لقراءة عنوان الخلية النشطة بنافذة المصنف تم اختياره
ليظهر العنوان في شكل Message Box

```
Sub MZM ()
MsgBox Windows("أوفيسنا").ActiveCell.Address
End Sub
```

لعمل انقسام داخل نافذة ورقة العمل النشطة

```
Sub MZM ()
ActiveWindow.Split = True
End Sub
```

لمعرفة عدد الأجزاء المنقسمة داخل نافذة ورقة العمل
النشطة واسترجاع الناتج في شكل مسج

```
Sub MZM ()
MsgBox ActiveWindow.Panes.Count
End Sub
```

لعمل تجميد الألواح داخل نافذة ورقة العمل النشطة

```
Sub MZM ()
ActiveWindow.FreezePanes = True
End Sub
```

workbook حدث الملف
worksheet أو حدث الصفحة

هناك أكواد يتم كتابتها داخل حدث الملف
وتعنى بتطبيق هذا الكود على كامل الملف مثل اكواد
الحماية أو عندما يتم فتح الملف تظهر رساله ما أو
فورم للدخول أو ... الخ

وهناك أكواد يتم كتابتها داخل حدث الصفحة
وتعنى بتطبيق هذا الكود على هذه الصفحة فقط أو تنفيذ
اجراء معين داخل هذه الصفحة

وهناك أكواد تكتب داخل موديول ويتم ربطها بكود يكتب
داخل حدث الصفحة

وأیضا هناك أكواد تكتب داخل موديول ويتم ربطها بكود
داخل حدث الفورم

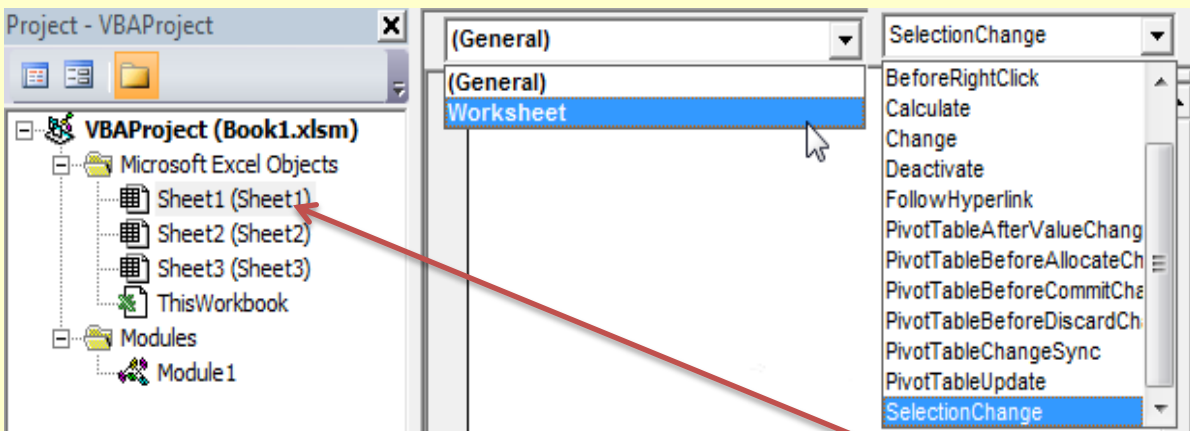
سنتحدث بهذا الدرس عن الأكواد التي تكتب داخل حدث
الصفحة

ولكن يجب أن نتعرف عليها وأین تكتب هذه الأكواد

عند الدخول الى محرر الأكواد نجد على اليسار اسماء
الشيئات أى الصفحات لو قمنا بالنقر عليها مرتين تفتح
لنا صفحة على اليمين هذه تسمى محرر الأكواد الخاص
بالصفحة نفسها أى حدث الصفحة كذلك مع حدث الملف

نجد بهذه الصفحة مستطيلين بهم رأسين لسهم اتجاه للأسفل بالضغط على السهمين تظهر لنا قائمة نختار منها رأس الكود بما يتناسب مع المطلوب أي المراد تنفيذه من خلال هذا الكود

كما بالصورة التالية



• بالضغط هنا مرتين تظهر الصفحة التي على اليمين

عند اختيار الحدث selectionchange تلقائياً يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

```
End Sub
```

ويعم تنفيذ الحدث عندما يتم تغيير الاحتيار أي مثلاً كلما قمنا بالانتقال من خلية الى خلية أخرى

على سبيل المثال

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

```
Static previous_selection As String
```

```
If previous_selection <> "" Then
```

```
Range(previous_selection).Interior.ColorIndex = xlColorIndexNone
```

```
End If
```

```
Target.Interior.Color = RGB(181, 244, 0)
```

ومعنى الكود أى كلما تم الإنتقال من خليه الى خلية اجعل
خلفية الخلية المختارة اى النشطه اى التى تم الانتقال إليها
خلفيتها باللون الأخضر ولا يتم احتفاظ الخليه بهذه الخلفيه
عندما يتم الانتقال الى خليه أخرى

عند اختيار الحدث Worksheet Activate
تلقائيا يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_Activate ()
```

```
End Sub
```

وهو خاص بعندما يتم تنشيط الورقة أى الدخول الى الورقة

عند اختيار الحدث Worksheet Deactivate
تلقائيا يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_Deactivate ()
```

```
End Sub
```

وهو خاص عندما يتم تنشيط ورقة عمل أخرى

عند اختيار الحدث

Worksheet BeforeDoubleClick

تلقائياً يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
```

```
End Sub
```

وهو خاص عندما يتم النقر مرتين على خلية ما

عند اختيار الحدث Worksheet BeforeRightClick

تلقائياً يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_BeforeRightClick (ByVal Target As Range, Cancel As Boolean)
```

```
End Sub
```

وهو خاص عندما يتم الضغط على كليك يمين على الورقة

عند اختيار الحدث Worksheet Calculate

تلقائياً يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_Calculate ( )
```

```
End Sub
```

وهو خاص عندما يتم حساب بيانات في ورقة العمل أو إعادة حساب

عند اختيار الحدث Worksheet Change تلقائياً يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_Change(ByVal Target As Range)
```

```
End Sub
```

وهو خاص عندما يتم تغيير محتويات الخلايا بورقة العمل

عند اختيار الحدث Worksheet FollowHyperlink تلقائياً يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink)
```

```
End Sub
```

وهو خاص عندما يتم النقر على ارتباط تشعبي

عند اختيار الحدث Worksheet FollowHyperlink تلقائياً يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink)
```

```
End Sub
```


وهو خاص عندما يتم النقر على ارتباط تشعبي

نتقل الى حدث الملف workbook

عند اختيار الحدث Workbook Open
تلقائيا يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Workbook_Open ( )
```

```
End Sub
```

ويتم تنفيذ الحدث عندما يتم تغيير الاختيار
أى مثلا كلما قمنا بالانتقال من خلية الى خلية أخرى

على سبيل المثال

```
Private Sub Workbook_Open ( )
```

```
MsgBox "منتديات أوفيسنا ترحب بكم"
```

```
End Sub
```

ومعنى الكود أى عند فتح الملف سيتم عرض مربع حوار فى
شكل رساله ترحب بك

عند اختيار الحدث Workbook BeforeClose
تلقائيا يتم كتابة رأس الكود ونهايته ويكون الكود هكذا

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
```

```
End Sub
```

وهو خاص بتنفيذ تعليمات مباشرة قبل اغلاق الملف

ومرفق مع هذا الدرس مثال عملي به الأكواد المذكوره أعلاه
وبعض الأحداث الخاصة بحدث الملف والتي لم تذكر بهذا
الدرس وأيضا تطبيقات أخرى
مثل التعرف على كيفية استخدام التاريخ والوقت

إنتهت أولى حلقات الدروس في VBA EXCEL
وتقبلوا منى وافر الاحترام والتقدير
محمود زكى الشريف