

ساخت ADD-IN حرفه ایی برای اکسل

۱. توضیحاتی پیرامون Add-In

یک فایل Add-In برای اکسل فایلی است با پسوند .xla ، .xll ، ویا .xlam. که با شروع اکسل می توان آن را بارگزاری و مورد استفاده قرار داد. این نوع فایل ها شامل کدهای برنامه نویسی شده در زبان Visual basic for Application یا به اختصار VBA هستند که به منظور تعریف توابع یا انجام محاسباتی خاص و یا پیاده سازی و نمایش فرم و اهداف خاصی در پلتفرم اکسل بکار گرفته می شوند.

یک فایل Add-In معمولاً توسط افراد متخصص در زمینه ی برنامه نویسی و آشنا به اکسل تهیه می گردد و هدف از ایجاد چنین برنامه هایی آن است که از توانایی های اکسل برای ایجاد نرم افزار های مفید و کاربردی استفاده گردد به عبارت دیگر این نوع فایل ها یک راه میانبر برای تعریف و ایجاد نرم افزارهایی برای نمایش و کاربرد توانایی های نرم افزار اکسل بوده و روشی ایده ال برای توزیع و ترویج توابع و ماکروهای شخصی در اکسل می باشند. پس از نصب برنامه ی آفیس روی رایانه ، اکسل (و سایر نرم افزارهای مجموعه ی آفیس) حاوی تعدادی Add-In استاندارد و از پیش تعریف شده از قبیل (Analysis Toolpak) می باشند که در همان ابتدای امر و بنا بر نیاز کاربر ، قابل بارگزاری و استفاده خواهند بود.

برای استفاده از Add-In ها در نرم افزار اکسل ، ابتدا باید Add-In مورد نظر را در اکسل ، بارگزاری کنید. برای این کار از مسیر زیر استفاده کنید :
Developer => Com Add-Ins (چنانچه Tab مربوط به Developer در رایانه ی شما قابل مشاهده نیست به روش زیر آن را فعال کنید:

۲. فعالسازی تب Developer

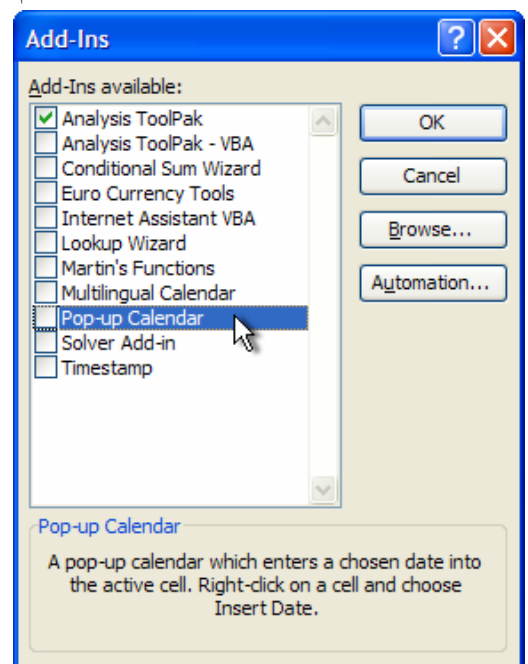
۱- از تب File گزینه ی Options را انتخاب کنید

۲- از گزینه های سمت چپ پنجره ی ظاهر شده ، Customize Ribbon را انتخاب کنید

۳- در گزینه های سمت راست پنجره ی ظاهر شده ، گزینه ی Developer را تیک بزنید

۴- دکمه ی OK را زده و از کل پنجره ها خارج شوید

با رفتن به مسیر Developer => Com Add-Ins پنجره ی مربوط به Add-In نمایان می شود. (تصویر شماره ۱) این پنجره لیست Add-In های ذخیره شده در رایانه را نمایش می دهد. اگر برخی از Add-In ها در فولدر یا جای دیگری مانند My Document یا روی شبکه قرار داشته باشند ، و در این لیست مشاهده نمی شوند، برای بارگزاری آنها لازم است از طریق دکمه ی Browse در همین پنجره آنها را یافته و بارگزاری نمایید.



تصویر شماره ۱ (لیست Add-In های موجود برای بارگزاری)

برای بارگزاری هر کدام از Add-In های موجود در این لیست کافیست آن را تیک زده و با زدن کلید OK از این بخش خارج شوید. پس از بارگزاری هر یک از Add-In ها از پنجره ی فوق ، هر بار که اکسل اجرا شود ، این برنامه نیز به همراه آن بارگزاری شده و قابل استفاده خواهد بود. چنانچه در هر زمان به حضور هر یک از Add-In ها نیاز نباشد ، می توان از پنجره ی مذکور تیک آن را برداشته و آن را غیر فعال نمود.

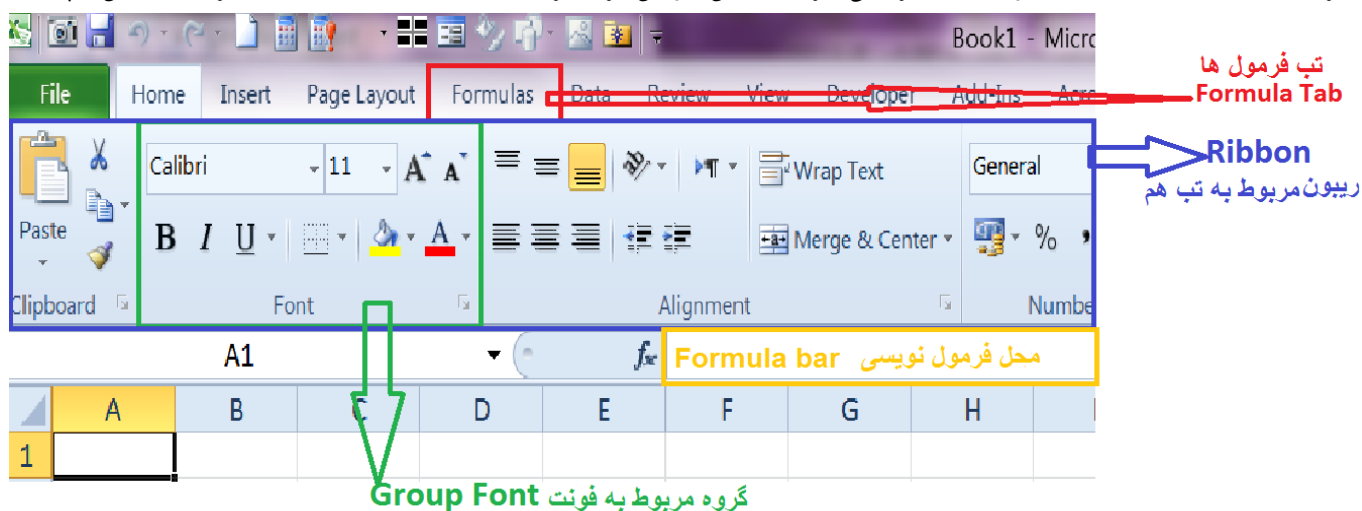
۳. ایجاد یک Add-In جدید در اکسل

همانگونه که گفته شد Add-In می تواند حاوی ماکرو و توابع تعریف شده توسط اشخاص باشد (UDF ها) که این توابع و ماکروها معمولاً "درون یک فایل اکسل (Workbook) قرار می گیرند که پسوند این فایل به صورت .xll یا .xla یا .xlam می باشد. برای یک Add-In می توان قابلیت های جدید اکسل ۲۰۱۰ همچون Tab و Ribbon و... اختصاصی تهیه و مورد استفاده قرار داد که در ادامه به آنها خواهیم پرداخت.

۴. توضیحات و مفروضات

قبل از اینکه از بحث تئوریک خارج شویم ، لازم است چند مورد گفته شوند تا در ادامه ی آموزش با مشکل مواجه نشویم : همانگونه که می دانید شرکت مایکروسافت در مجموعه ی نرم افزاری office 2007 ، به جای استفاده از منو، ساب منو، تولبار و کامندهای قدیمی از سیستم جدید خود شامل Group ، Ribbon ، Tab و... استفاده کرد و همین امر تا مدتها برای کاربرانی که با سیستم های office 2003 و ماقبل آن کار کرده و به منوهای آن عادت کرده بودند ، ایجاد مشکل کرده بود ولی به مرور زمان همه ی کاربران به استفاده از طرح جدید مایکروسافت و tab و ribbon و عادت کردند به هر حال در سیستم office 2003 و ماقبل آن برنامه نویسان از طریق دستورات VBA و متدهای موجود در آن می توانستند منوها، تولبارها و کلیدهای جدیدی به سیستم موجود اضافه و از آنها برای اجرای ماکروها و توابع شخصی خود بهره ببرند همین مراحل برای تهیه ی Add-IN نیز تکرار می شد ولی در سیستم جدید مایکروسافت (سیستم Ribbon) از طریق VBA توانایی تغییر و ایجاد Tab یا Ribbon و یا کلید جدید وجود ندارد لذا برای این کار لازم است از XML و دستورات آن بهره جست البته نگران نباشید **در این آموزش فرض ما بر این است که برای ساخت Add-In از هیچ نرم افزار اضافه ای استفاده نکنیم** و مفاهیم به گونه ای تعریف و اجرا شوند که **حتی افراد مبتدی و تازه کار با دنبال کردن مراحل و مثالی که برای این آموزش در نظر گرفته شده است ، بتوانند یک Add-In حرفه ای تولید و توزیع کنند.**

برای اینکه به یک درک مشترک از مفاهیم بکار رفته در این آموزش برسیم ، به تصویر شماره ۲ مربوط به ریبون اکسل، دقت کنید و کلمات و اصطلاحاتی را که برای Office 2007 و بعد از آن مورد استفاده قرار می گیرند و در این آموزش نیز به کرات از آنها استفاده شده است را یاد گرفته و به ذهن بسپارید.



تصویر شماره ۲ (معرفی اصطلاحات موجود در سربرگ های اکسل)

تا اینجا ی آموزش ، موارد تئوریک و پیش زمینه ، گفته شد در بخش بعدی ادامه ی کار را به صورت عملی دنبال می کنیم.

۵. آموزش عملی ساخت Add-In

۵.۱ مراحل آماده سازی Workbook

برای ادامه آموزش، یک مثال را مطرح کرده و در طول این آموزش از آن استفاده می کنیم تا نتایج ملموس تر و قابل مشاهده تر باشند، برای رسیدن به نتیجه ی دلخواه، تمام موارد و مراحل را به صورتی که در اینجا شرح می دهیم اجرا و با ما همراه باشید.

برنامه ی اکسل ۲۰۱۰ را اجرا کنید، به صورت پیش فرض فایلی به نام **Workbook 1** ساخته می شود و شما در سلول **A1** از **Sheet 1** آن قرار خواهید گرفت، اکنون بدون انجام هیچ کار خاص و اضافه ای، از تب **File** گزینه ی **Save As** را انتخاب نموده و از قسمت پایین پنجره، گزینه ی **Save As Type** را روی **Excel Macro-Enabled Workbook (*.xlsm)** قرار داده و در قسمت **File Name** نیز یک نام مناسب مثلاً "MyFirstAddIn" را نوشته و آن را روی **Desktop** (انتخاب محل ذخیره ی فایل به عهده ی شما) ذخیره نمایید. توجه داشته باشید که در صورت تنظیم نکردن نوع فایل به صورت **(*.xlsm)**، فایل به صورت معمولی ذخیره می شود و در این حالت برنامه ها و ماکروهای شما روی فایل ذخیره نخواهند شد.

۵.۲ آماده سازی محیط VBA برای نوشتن توابع و ماکروها

برای برنامه نویسی در محیط اکسل ابتدا باید تب **Developer** را فعال کنیم، اگر هنوز آن را فعال نکرده اید طبق فرایند زیر آن را فعال کنید:

- ۱- از تب **File** گزینه ی **Options** را انتخاب نمایید
 - ۲- از بخش سمت چپ پنجره ی باز شده، گزینه ی **Customize Ribbon** را انتخاب کنید
 - ۳- در بخش سمت راست این پنجره (قسمت **Main Tabs**) گزینه ی **Developer** را تیک زده و **OK** کنید.
- اکنون با انتخاب تب **Developer** می توانید **Ribbon** مربوط به آن را ملاحظه نمایید.
- از اولین **Group** در سمت چپ (به نام **Code**)، گزینه ی **Visual Basic** را انتخاب نمایید تا وارد محیط **VBA** شوید.
- نکته:** محیط **VBA** همچنان از سیستم قدیمی مایکروسافت، یعنی منو و تولبار و کامند استفاده می نماید و خبری از **Tab** و **Ribbon** نیست.
- ترفند:** می توان با زدن کلید های هم زمان **Alt + F11**، بدون فعال کردن تب **Developer** نیز وارد محیط **Visual basic** شد.

۵.۳ نوشتن تابع در محیط VBA

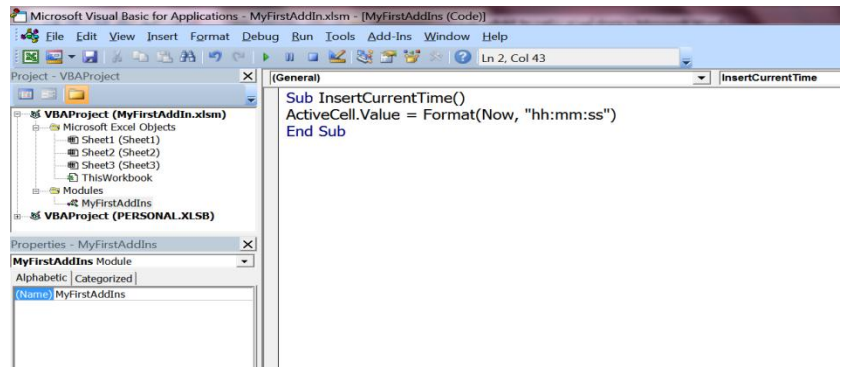
فرض کنید قصد نوشتن یک ماکروی ساده را داریم که در آن قرار است زمان جاری سیستم رایانه را به صورت ساعت، دقیقه و ثانیه، در سلول فعال فعلی، (سلولی از اکسل که هم اینک انتخاب شده و فعال است و نشانگر موس روی آن قرار دارد **Active Cell**) درج گردد.

برای این کار مراحل زیر را انجام دهید:

- ۱- به یکی از روش های بالا وارد محیط **VBA** می شویم (استفاده از تب **Developer** و یا فشردن همزمان **Alt + F11**)
- ۲- از منوی **Insert** گزینه ی **Module** را انتخاب می کنیم. (در قسمت **Project Explorer** که در سمت چپ قرار دارد، **Module1** اضافه می شود، اگر **Project Explorer** را مشاهده نمی کنید کلیدهای **Ctrl + R** را فشار دهید)
- ۳- **Module 1** از قسمت **Project Explorer** را انتخاب کنید و در قسمت **Properties Window** که در زیر آن قرار گرفته است نام آن را به **MyFirstAddIns** تغییر دهید (اگر بخش **Properties Window** مشاهده نمی شود کلید **F4** را فشار دهید)
- ۴- با دابل کلیک روی ماژول **MyFirstAddIns** وارد محیط برنامه نویسی آن می شویم.
- ۵- در محیط برنامه نویسی مربوط به ماژول **MyFirstAddIns** یک ماکرو (تابع) به شکل زیر بنویسید:

```
Sub InsertCurrentTime ()  
    ActiveCell.Value = Format(Now, "HH:MM:SS")  
End Sub
```

اکنون کار شما باید مانند شکل زیر (شماره ۳) شده باشد.



۶- از منوی File اولین گزینه یعنی Save را انتخاب کنید (می توانید از کلیدهای Ctrl + S نیز استفاده کنید)

۷- از منوی File آخرین گزینه یعنی Close and Return To Microsoft Excel را انتخاب نمایید (می توان از کلید های Ctrl + C نیز استفاده نمود)

بدین ترتیب ماکروی ما آماده شده است و در حال حاضر در محیط اکسل قرار داریم.

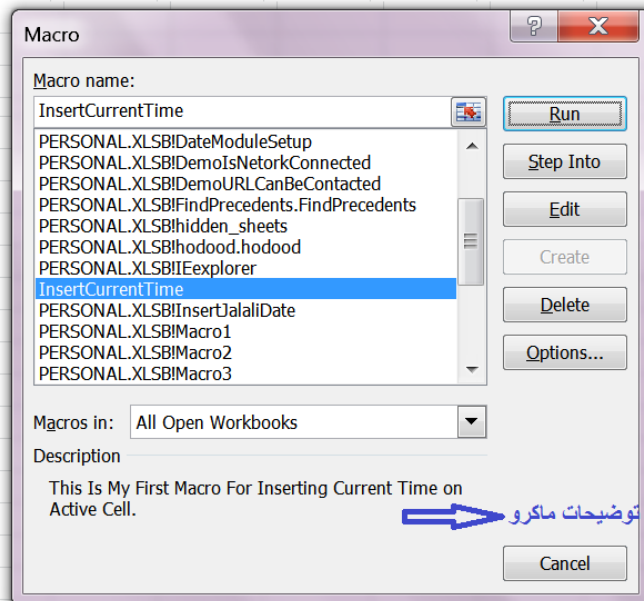
۵.۴ اضافه کردن توضیحات به ماکرو

به عنوان یک ایده ی خوب ، سعی کنید همیشه برای توابع و ماکروهایی که تهیه می کنید ، راهنما و توضیحاتی را اضافه کنید تا کاربران برای استفاده از آن بتوانند توضیحات مختصری از آن را مشاهده کنند. برای اضافه کردن توضیحات به ماکرو ، مراحل زیر را دنبال کنید :

۱- در تب Developer از اولین Group از سمت چپ (گروه Code) ، گزینه ی Macros را انتخاب نمایید تا لیست کل ماکروها نمایش داده شود.

۲- در لیست ماکروها ، ماکرویی به نام InsertCurrentTime را انتخاب نمایید(نام ماکرویی که در قسمت قبل تهیه نمودید) (تصویر شماره ۴)

۳- از سمت راست کلید Options... را انتخاب کنید (تا هنگامی که نام ماکرو را انتخاب نکرده باشید کلید Options... غیر فعال خواهد بود)



تصویر شماره ۴ (لیست ماکروها و توضیحات ماکروی InsertCurrentTime

۴- در قسمت Description ، توضیحاتی مربوط به این ماکرو و خلاصه ایی از عملکرد و خصوصیات آن را بنویسید. (بهتر است این توضیحات به انگلیسی نوشته شوند همانند تصویر شماره ۴)

۵- از پنجره ی فوق خرج شده و فایل اکسل را از مسیر File => Save ذخیره نمایید.

نکته: در این قسمت با زدن کلید Run می توان ماکرو را اجرا کرد و نتیجه ی آن را که درج شدن زمان جاری سیستم در سلولی که هم اکنون موس

روی آن قرار گرفته است را مشاهده کرد. (قبل از فشردن کلید Run توجه نمایید که کدام یک از ماکروها درحالت انتخاب قرار دارند)

نکته: برای استفاده از توابع اصلی اکسل، نام تابع مورد نظر را در قسمت Formula Bar نوشته و از آن استفاده می کنیم.

۶. سفارشی سازی ماکرو

حال نوبت به سفارشی کردن ماکرو رسیده است ، در این بخش قصد داریم برای ماکروی خود ابتدا یک **Shortcut** ساخته و سپس آن را به منوی راست کلیکک موس (**Right Mouse click**) اضافه کنیم و در نهایت میز یک **Ribbon** برای آن بسازیم پس با ادامه ی مثال همراه ما باشید.

۶.۱ ساختن کلید **Shortcut** برای ماکرو

۶.۱.۱ از این ماکرو فقط در رایانه ی شخصی استفاده می کنم

چنانچه این تابع را فقط برای استفاده در رایانه شخصی خود نوشته باشید ، می توان از طریق فرایند زیر برای آن یک **Shortcut** ساخت :

۱- از مسیر **Developer -> Macros** ، لیست ماکروهای موجود را باز کنید

۲- نام ماکروی مورد نظر را انتخاب کنید

۳- کلید **Options...** از سمت راست را فشار دهید

۴- در پنجره ی ظاهر شده ، بخش **Short cut Key** یکی از حروف الفبا را درج کنید مثلاً " حرف **T** (حرف **T** بزرگ) .

در کنار کادر مورد نظر ، کلید های **Ctrl + Shift** نوشته خواهد شد ، بدین معنی که برای اجرای ماکروی مورد نظر در محیط اکسل کافی است

کلید های **Shift** و **Ctrl** و حرف **T** بطور همزمان فشرده شوند و چنانچه حرف **t** (حرف **t** کوچک) استفاده شود ، در کادر مربوطه فقط کلید

Ctrl نوشته خواهد شد و نشان دهنده ی این است که برای اجرای این ماکرو در محیط اکسل فشردن همزمان کلیدهای **Ctrl** و حرف **t** منجر به

اجرای ماکرو خواهد شد

۵- کلید **OK** را جهت ذخیره ی تغییرات فشار دهید.

۶.۱.۲ از این ماکرو برای یک **Add-In** استفاده می کنم

از آنجایی که این آموزش جهت ساخت **Add-In** در اکسل می باشد لذا لازم است کلید **Shortcut** برای ماکروی مذکور را به گونه ایی تعریف کنیم ، که با بارگزاری **Add-In** روی هر رایانه ایی ، کلید **Shortcut** آن نیز برای ماکروی مذکور تعریف شده و غیر قابل تغییر باشد لذا در این مرحله باید از طریق **VBA** وارد عمل شد و برای آن از طریق برنامه نویسی کلید **Shortcut** مشخصی را تعریف کرد. ضمناً توجه داشته باشید که چون این آموزش به صورت مرحله ایی ارائه می گردد لذا در هر مرحله دستورات و کدهای جدید به برنامه اضافه یا از آن حذف خواهد شد ، پس همه مراحل به ترتیبی که در اینجا گفته می شوند ، باید توسط شما دوست عزیز دنبال گردد. یادآوری می گردد که چون آموزش برنامه نویسی **VBA** به صورت کامل و مفصل از حوصله ی این مختصر خارج است لذا در قسمت های بعدی که بیشتر با برنامه نویسی کار خواهیم کرد فقط بخشی از دستورات به صورت مختصر و صرفاً جهت آشنایی شما دوست عزیز توضیح داده خواهند شد ، پس در ادامه با ما همراه باشید:

۱- با زدن همزمان کلیدهای **Alt + F11** وارد محیط کدنویسی **VBA** شوید

۲- از قسمت **Project Explorer** روی گزینه ی **ThisWorkbook** دابل کلیک کنید تا به محیط برنامه نویسی آن وارد شوید. همانگونه که مشاهده می گردد در بالای محیط برنامه نویسی مذکور ، منوی باز شونده ی سمت راست با عنوان **(Declaration)** و منوی باز شونده ی سمت چپ آن با عنوان **(General)** به صورت پیش فرض ظاهر می گردد.

۳- از قسمت منوی باز شونده ی سمت چپ، گزینه ی **Workbook** و از منوی سمت راست گزینه ی **Open** را انتخاب نمایید ، بدین ترتیب **VBA** برای برنامه نویسی آماده شده است.

۴- اکنون دستورات را به شکل زیر تایپ کنید:

```
Private Sub Workbook_Open()  
Application.OnKey "+^{T}", "MyFirstAddIns.InsertCurrentTime"  
End Sub
```

توضیح این برنامه : کل برنامه های **VBA** با یک کلمه ی **Sub** و در پایان نیز با کلمه ی **EndSub** به پایان می رسند و کلیه ی دستورات برنامه باید بین این دو دستور قرار گیرند لذا در اینجا فقط یک خط برنامه نویسی شده است یعنی خط دوم. در این خط که خط اصلی برنامه است برای دستور **OnKey** ابتدا یک کلید به عنوان **Shortcut** تعریف شده و سپس مسیر یک تابع یا ماکرو جهت اجرا به آن معرفی شده است. برای کلید تعریف شده ، حرف **T** بین دو علامت آکولاد قرار می گیرد و علامت **+** به عنوان کلید **Shift** و علامت **^** به عنوان کلید **Ctrl** می باشد. پس در این دستور یک

- Shortcut تعریف کرده ایم که با فشردن هم زمان کلید های Shift + Ctrl + T عمل خواهد کرد همچنین مسیر اجرای تابع به این شکل تعریف شده که ابتدا نام ماژولی که تابع در آن قرار دارد یعنی MyFirstAddIn و سپس یک نقطه و بعد از آن نیز نام تابع (ماکرو) یعنی InsertCurrentTime .
- ۵- از محیط VBA خارج شوید
 - ۶- ابتدا فایل اکسل را Save کنید و سپس آن را بسته و از اکسل خارج شوید.
 - ۷- مجدداً فایل اکسل MyFirstAddIn را باز کنید
 - ۸- ضمن آنکه نشانگر موس در سلول A1 قرار گرفته است ، کلیدهای Shift + Ctrl+ T را به طور همزمان فشار دهید.
- همانگونه که مشاهده می شود ، زمان جاری سیستم در سلول A1 درج خواهد شد.
- بهتون تبریک می گم عالی بود.

۶,۲ ساخت Shortcut برای منوی راست کلیک موس

- در این قسمت قصد داریم برای ماکروی خود یک Shortcut در منوی راست کلیک موس بسازیم ، یعنی کاربر پس از راست کلیک در هر سلول، بتواند منوی مربوط به ماکروی شما را نیز بین گزینه های پیش فرض مشاهده کرده و در صورت انتخاب آن ، ماکروی شما اجرا شده و در سلول مورد نظر ، زمان جاری سیستم را درج کند. برای رسیدن به این خواسته ، مراحل زیر را دنبال کنید :
- ۱- وارد محیط VBA شوید (Alt + F11)
 - ۲- وارد محیط برنامه نویسی برای ThisWorkbook شوید.
 - ۳- تنظیمات را برای Workbook-Open انجام دهید. (در بخش قبل برای این مورد، با نوشتن یک خط دستور، Shortcut دائمی ایجاد کردیم و اکنون باید دستورات گفته شده را به آن اضافه کنیم.
 - ۴- دستورات را مانند زیر بنویسید:

Sub Workbook_Open()

```
Dim NewControl As CommandBarControl
Application.OnKey "+^{T}", "MyFirstAddIns.InsertCurrentTime"
On Error Resume Next
Application.CommandBars("cell").Controls("Insert Current Time").Delete
On Error GoTo 0
Set NewControl = Application.CommandBars("Cell").Controls.Add
With NewControl
.Caption = "Insert Current Time"
.OnAction = "MyFirstAddIns.InsertCurrentTime"
.BeginGroup = False
End With

End Sub
```

- ۵- اکنون در محیط VBA ، تنظیمات را برای Workbook-BeforeClose انجام دهید
- ۶- برنامه ی زیر را در آن بنویسید:

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
Application.OnKey "+^{T}", ""
On Error Resume Next
Application.CommandBars("cell").Controls("Insert Current Time").Delete
End Sub
```

نکته مهم : کلمات مربوط به نام ماژول و نام ماکرو به درستی نوشته شود ، مراقب حروف کوچک و بزرگ و فواصلی که بین کلمات قرار دارد باشید چرا که عدم رعایت این موضوع برنامه ی شما را با مشکل مواجه خواهد کرد و حل و فصل آن شما را خسته خواهد کرد.

- ۷- از محیط VBA خارج شوید
- ۸- ابتدا فایل اکسل را Save کنید و سپس آن را بسته و از اکسل خارج شوید.
- ۹- مجدداً فایل اکسل MyFirstAddIn را باز کنید
- ۱۰- ضمن آنکه نشانگر موس در سلول A1 قرار گرفته است، روی آن کلیک راست نمایید.

۱۱- اگر به انتهای منوی باز شده توجه کنید، گزینه ی جدیدی به نام **Insert Current Time** که مربوط به ماکرو ی شماسست به آن اضافه شده است، کافی است آن را انتخاب نمایید تا در سلول **A1** زمان جاری سیستم درج گردد.

نکته: حتماً به خط دوم این برنامه توجه کرده اید که همچنان کلید های **Shift + Ctrl + T** به عنوان **Shortcut** برای اجرای ماکرو تعریف شده و این موضوع به حالت خود باقی است یعنی علاوه بر منوی راست کلیک موس، فشردن همزمان کلیدهای شیفت+ کنترل + حرف **T** نیز آن را اجرا خواهد کرد

نکته: توجه داشته باشید که **Shortcut** موجود در منوی راست کلیک موس فقط برای فایل اکسل شما با نام **MyFirstAddIn** ساخته شده است و اگر در این حالت یک فایل اکسل دیگری در کنار آن باز کنید و منوی راست کلیک را باز کنید هرگز این **Shortcut** را نخواهید دید.

۶.۳ ساخت Ribbon اختصاصی برای ماکرو

این قسمت از آموزش مربوط به **Office 2007** و آفیس های بعد آن می باشد باید توجه داشت که مباحث مربوط به ساخت **Ribbon** و **Group** و... خارج از توانایی های محیط **VBA** بوده و مسائل مربوط به آن از محیط **VBA** فراتر رفته است لذا برای انجام این کار رویه های جدیدی برای برنامه نویسان و متخصصین این امر در نظر گرفته شده است.

بهتر است بدانید که موضوع **Ribbon** تا زمان تهیه این آموزش در بین دستورات **VBA** قرار نداشته و ممکن است در نسخه های جدید **VBA** که در آینده برای مجموعه ی **Office** تهیه خواهد شد این موضوع نیز لحاظ گردد. به هر حال جهت اعمال تغییرات در **Ribbon** ها لازم است دست به دامن **XML** شویم و موارد خواسته شده از قبیل ایجاد ریبین جدید، تغییر در یکی از ریبین های موجود، افزودن گروه جدید به یک تب، افزودن کلید جدید در یک گروه و تغییرات متعدد دیگر را در این قالب تهیه کرده و آن را به گونه ایی به خورد برنامه ی اکسل بدهیم. در همین جا متذکر می شوم که انجام تغییرات در **Ribbon** های اکسل، ورد، پراجکت و پاورپوینت مشابه می باشند ولی برای اوتلوک واکسس موضوع به گونه ی دیگری است و کلاً فرایند آن آسان تر می باشد.

۶.۳.۱ توضیحاتی پیرامون XML

از آنجایی که قصد آموزش **XML** را نداریم لذا فقط قسمتی از موضوع که احتمالاً در این آموزش کاربرد دارد را مطرح می کنیم. هر فایل در **Office 2007** دارای یک اینترفیس در قالب **XML** است که به سایر مشتقات فایل مذکور اتصال دارد و آن را به صورت **customUI** معرفی می کنند. در واقع تعریف نوع **Ribbon** ها در هر فایل بر عهده ی اینترفیس می باشد بنابراین برای تغییر در **Ribbon** یک فایل آفیس باید اینترفیس آن را تغییر و یا با اینترفیس جدیدی جایگزین نمود. در اینترفیس ها دستورات مختلفی در قالب **XML** وجود دارد که برخی از آنها با اهمیت تر بوده و باید با نگاه ویژه ی دیده شوند. در اینترفیس هر دستور به همراه پارامترها (در صورت نیاز) بین علامت های **</>** قرار می گیرد. معمولاً در دستورات مربوط به اینترفیس رعایت سلسله مراتب الزامی است و به ترتیب از تعریف یا تعیین ریبین و سپس تب و سپس گروپ و نهایتاً ن یک کلید تشکیل شده است.

برای هر یک از اعضای مذکور نیاز به یک کد شناسایی و یک **Label** و بعضاً یک تصویر و موقعیت مکانی در **Ribbon** داریم ضمناً برای کلیدها علاوه بر موارد بالا نیاز به کد رویداد داریم تا نسبت به کلیک شدن عملیات خاصی را انجام دهد.

کد شناسایی برای اعضا ی مورد نظر (ریبین، تب، گروه و کلید) دو حالت دارد:

- ۱- اگر بخواهیم یک عضو جدید تعریف کنیم که در این صورت کد شناسایی به صورت **ID** تعریف می گردد
 - ۲- اگر بخواهیم یکی از اعضای مذکور را که قبلاً وجود داشته است تغییر دهیم کد شناسایی به صورت **idMso** تعریف می گردد
- تذکر: حروف **Mso** همیشه برای شی موجود بکار می رود.

در مثال زیر دستوری نوشته شده است که قواعد گفته شده در بالا در آن به درستی لحاظ شده است:

```
<button id="myButton" onAction="MyButtonOnAction" />
```

توضیح: در این دستور کلید جدیدی به نام **myButton** تعریف شده است و چون این کلید از قبل وجود نداشته، کد شناسایی آن به صورت **id** تعریف شده است ضمناً برای کلیدهایی که تعریف می شوند باید یک رویداد نیز تعریف گردد که در اینجا برای رویداد **onAction** آن به یک ماکرو به نام **MyButtonOnAction** اشاره شده است که با فشردن این کلید اجرا خواهد شد.

۶.۳.۲ ساخت اینترفیس جدید

برای ساخت یک اینترفیس جدید به منظور جایگزین نمودن آن با اینترفیس پیشفرض دراکسل، به شکل زیر عمل می کنیم:

- ۱- یک فولدر جدید روی **Desktop** ساخته و نام آن را **customUI** بگذارید

۲- با استفاده از نرم افزار Ms Word یا هر Text Editor دیگری مانند NotePad یا WordPad کدهای XML زیر را بنویسید.

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <tabs>
      <tab id="CustomTab" label="My Tab">
        <group id="SimpleControls" label="My Group">
          <button id="Button1" imageMso="HappyFace" size="large"
            label="Large Button"
            onAction="MyFirstAddIns.InsertCurrentTime" />
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

۳- این فایل را با نام customUI و با پسوند Xml ذخیره کنید

۴- از فایل اکسل (MyFirstAddIns) یک کپی پشتیبان تهیه کنید و آن را در فولدر دیگری روی رایانه ی خود ذخیره کنید.

۶,۳,۳ جایگزین کردن ایتترفیس جدید

۱- پسوند فایل اکسل MyFirstAddIns.xlsm را از xlsm به zip تغییر دهید.

۲- روی فایل MyFirstAddIns.zip دابل کلیک کنید تا باز شود. (در محیط Winzip ویندوز)

مشاهده می کنید که درون این فایل zip شده ، فولدرهای _rels و docProps و xl به همراه یک فایل [Content_Types] از نوع XML وجود دارد.

۳- در حالی که فایل zip را مشاهده می کنید ، فولدر customUI را که روی دسکتاپ قرار دارد به درون این فولدر drag کنید.

۶,۳,۴ ایجاد ارتباط بین فایل اکسل و ایتترفیس جدید

۱- در حالی که فایل زیپ را باز کرده و مشاهده می کنید، فولدر _rels را از درون فایل zip به روی دسکتاپ Drag کنید و فایل درون آن را که به همین نام است ، بوسیله ی یک برنامه ی TextEditor مانند Notepad ویندوز برای انجام برخی تغییرات ، باز کنید.

۲- در انتهای آخرین خط از فایل rels.xml دستور </Relationship> قرار دارد و ما باید چند خط دستور را قبل از آن نوشته و تغییرات را Save کنیم به طوری که همچنان دستور </Relationship> آخرین خط دستور در فایل مذکور باشد.

۳- دستوراتی که باید طبق بند بالا به این فایل اضافه شوند به شرح ذیل می باشند :

```
<Relationship Type="http://schemas.microsoft.com/office/2006/
relationships/ui/extensibility" Target="/customUI/customUI.xml"
Id="customUIRelID" />
```

۴- فایل را پس از انجام تغییرات ، ذخیره کرده و آن را ببندید.

۵- اکنون باید این فایل را درون فولدر _rels از MyFirstAddIns.zip قرار دهید. لذا در حالی که فایل zip همچنان باز است ،فایل rels.xml را که تازه تغییر داده اید را Drag کرده و در فولدر _rels قرار دهید.

۶- فایل MyFirstAddIns.zip را ببندید.

۷- پسوند فایل MyFirstAddIns را از zip به xlsm تغییر دهید .

۸- فایل MyFirstAddIns.xlsm را باز کنید (باید به صورت اکسل باز شود)

۹- وارد محیط VBA شوید (Alt + F11)

۱۰- روی ماژول MyFirstAddIns دابل کلیک کنید.

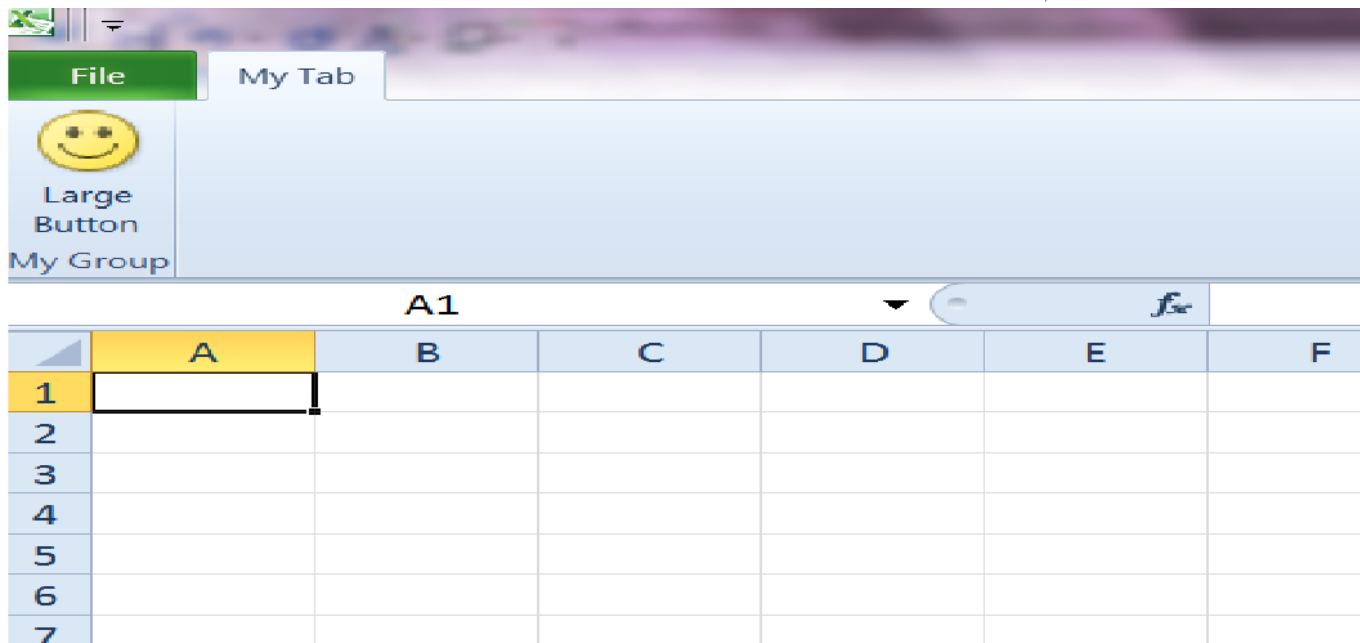
۱۱- تابع InsertCurrentTime را که قبلاً" نوشته بودیم به شکل زیر تغییر دهید (فقط در خط اول تغییرات داده شده است)

```
Private Sub InsertCurrentTime(ByVal Control As IRibbonControl)
ActiveCell.Value = Format(Now, "hh:mm:ss")
End Sub
```

۱۲- از محیط VBA خارج شده و فایل اکسل را Save کرده و از آن خارج شوید.

۶,۳,۵ آزمایش کردن فایل اکسل به همراه اینترفیس جدید (Ribbon جدید)

اگر همه ی مراحل بالا به خوبی انجام شده باشد ، فایل MyFirstAddIns همانند تصویر شماره ۵ ظاهر خواهد شد.



تصویر شماره ۵ (فایل اکسل با Ribbon جدید)

تبریک می گم ، همه چی تمام شد. هرچند که این قسمت آخر کمی پیچیده به نظر می رسد ولی شما موفق شدید.

۶,۳,۶ توضیحات بیشتر برای Ribbon جدید

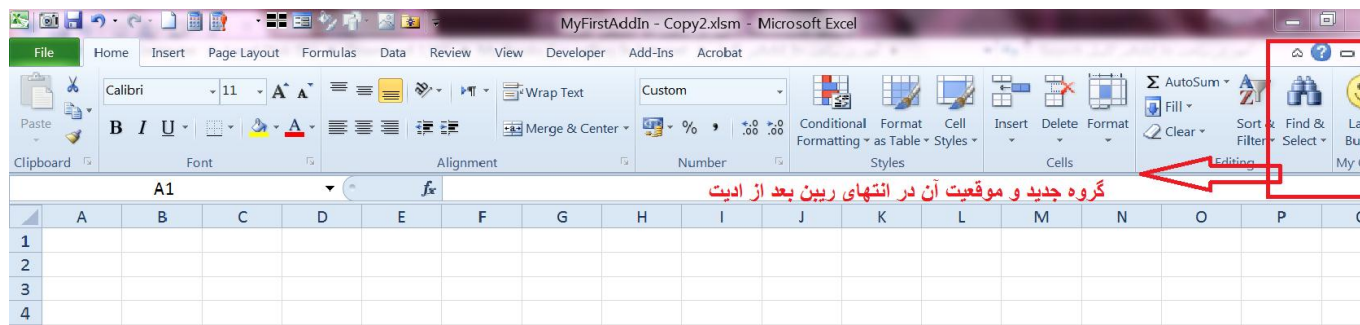
در فصل ۶,۳,۲ (ساخت اینترفیس جدید) که دستورات مربوط به customUI در قالب xml گفته شد ، می توان با انجام بعضی تغییرات ، اینترفیس های جدید و کاربردی را ایجاد کرد که در اینجا برخی تغییرات همراه با توضیحات لازم ارائه می گردد.

فرض کنید قرار است در همان Ribbon پیش فرض ، روی تب home یک گروه جدید ایجاد کرده و در این گروه جدید، یک کلید برای اجرای ماکرو ساخته شود برای این کار باید خطوط اولیه ی customUI را به شکل زیر تغییر داده و باقی مراحل مشابه با قسمت های گفته شده در بالا خواهد بود:

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon>
    <tabs>
      <tab idMso="TabHome">
        <group id="customGroup1" label="My Group" insertAfterMso="GroupEditingExcel" >
```

توضیحات : در خط دوم اعلام شده است که از همان Ribbon پیش فرض استفاده گردد. در خط ۴ نیز چون از تب Home که به صورت پیشفرض موجود بوده استفاده کرده ایم لذا از idMso استفاده شده است. در خط ۵ نیز چون گروه جدیدی تعریف کرده ایم کد شناسایی آن به صورت id تعریف شده است و نام گروه جدید را My Group گذاشته ایم و چون در نظر داشتیم که موقعیت آن بعد از گروه editing در تب home قرار گیرد لذا از دستور insertAfterMso و موقعیت گروه editing استفاده شده است

بدین ترتیب شکل Ribbon جدید با تغییر فوق همچون تصویر ۶ خواهد شد.



۶.۴ ساخت Add-In نهایی

اگر فایل MyFirstAddIns.xlsm را باز کنید ، فایل اکسل را ملاحظه خواهید کرد که دارای سه شیت خالی است حتی اگر خالی هم نباشند چندان فرقی نمی کند چرا که پس از تبدیل به فرمت Add-In عملاً هیچ کس قادر به دیدن محتویات شیت ها نمی باشد برای تبدیل فایل خود به فرمت Add-In مراحل زیر را انجام دهید :

- ۱- از تب File گزینه ی Save As را انتخاب کنید
 - ۲- نوع آن را Excel Add-In (*.xlam) انتخاب نموده و مسیر ذخیره شدن آن را مشخص نمایید.
 - ۳- کلید Ok را بزنید.
- تمام شد. اکنون شما یک Add-In ساخته اید با همه مشخصه های لازم و می توانید آن را بارگزاری کرده و امتحان کنید. مبارک است

چند نکته خیلی مهم :

- ۱- اگر فایل Add-In دارای یک ریبین اختصاصی باشد ، شما عملاً کاربران را در استفاد از ریبین اصلی اکسل با مشکل مواجه خواهید کرد لذا توصیه ی اکید می شود که از ریبین اختصاصی بجز در موارد خیلی نادر استفاده نکنید بلکه سعی کنید از ریبین های موجود استفاده کرده و یک گروه یا یک کلید به آن اضافه کنید
- ۲- در صورتی که با مشکل بالا مواجه شدید و راهی برای برگشت به ریبین اصلی نداشتید ، ساده ترین کار آنست که فایل Add-In که منجر به این مشکل شده است و شما نام و موقعیت آن روی رایانه ی خود را می دانید را یا تغییر نام دهید و یا آن را حذف نمایید تا با باز کردن مجدد اکسل این Add-In اجرا نشود و ریبین اصلی اکسل مشاهده گردد.
- ۳- چون مجموعه ی آفیس از حروف فارسی پشتیبانی می کند ، شما می توانید در اینترنت خود نام کلید ، گروه و تب را به فارسی بنویسید ولی به یاد داشته باشید که در VBA کار با کلمات فارسی فوق العاده سخت است و بهتر است که عطایش را به لقایش ببخشید.
- ۴- نکته ی آخر اینکه در صورت هر گونه مشکل یا داشتن پیشنهاد و انتقادی، بنده با کمال میل مایل به شنیدن سخنان شما هستم و برای رفع مشکلات مربوطه نیز در حد توان در خدمت دوستان خواهم بود.

به پایان آمد این دفتر ، حکایت همچنان باقیست

در این مختصر ، سعی گردید با ترفندهای دم دستی ولی کارگشا ، یک فایل Add-In ایجا کنیم همانگونه که ملاحظه فرمودید در قسمت تعریف تابع از یک تابع ساده استفاده کردیم ولی شما می توانید با تعریف توابع پیچیده و کاربردی ، Add-In های فوق العاده ایی تولید کنید ضمناً در سایر کشورها تولید Add-In یک شغل محسوب می شود و بسیاری از متخصصین این امر با خرید و فروش فایل های Add-In خود به ثروت رسیده اند و همانگونه که می دانید دنیای برنامه نویسی هیچ انتهایی ندارد و برای همه ی افراد جای فعالیت و پیشرفت وجود دارد لذا امید است این آموزش به سان جرقه ایی باشد تا توانایی های شما در این زمینه را تحریک و نقطه ی عطفی در پیشرفت و شکوفایی شما باشد.

پایان آموزش

موفق باشید.

تهیه کننده : حبیب منا Habib_mena@yahoo.co

با اعتقاد به اینکه هر نوشته ی دست بشر می تواند به سوی بهینه شدن حرکت کند لذا از نظرات و پیشنهادات شما بسیار خرسند خواهم شد.