ماکرو نویسی و برنامه نویسی در اکسل

به روش من

به زبان *VBA* - ویژوال بیسیک در اکسل - *VB in excel*

مؤلف : مهندس علی فاتحی

رشته تحصیلی: مهندسی کامپیوتر- از شهید بهشتی تهران

کارشناسی ارشد مهندسی صنایع - مهندسی سیستمهای اقتصادی اجتماعی- از امیرکبیرتهران

مطالب پیش روی شما در واقع جزوه ای است که بنده در کلاسهای ماکرونویسی و برنامه‌نویسی اکسل ارایه می نمایم و در وبلاگ *http://www.alifatehi.persianblog.ir* نیز آمده است.

مطالب خیلی خلاصه و کاملا کاربردی ارایه شده اند و نقطه شروع بسیار خوبی برای ماکرونویسی و برنامه نویسی می باشند.

مطالب بیشتر و کامل تر را می توانید از وبلاگ بنده و کتاب کامل تری که بزودی به تالیف اینجانب منتشر می‌شود دنبال کنید.

نظرات و پیشنهادها و نیازهای آموزشی و مشاوره‌ای خود را از طریق ایمیل ali_fatehi@yahoo.com مطرح فرمایید.

علی فاتحی
ali_fatehi@yahoo.com

جلسه اول: برنامه نویسی در اکسل - مقدمه و ایجاد ماکرو

مقدمه

VBA که مخفف عبارت ***Visual Basic For Applications*** یک زبان برنامه‌نویسی است که توسط شرکت نرم افزاری مایکروسافت طراحی شده است. ***VBA*** در ***Excel*** و سایر نرم افزارهای آفیس گنجانده شده است.

روباتی را در نظر بگیرید که تسلط کافی بر اکسل دارد و قدرت محاسباتی بالا داشته و اکسل را با دقت و سرعت تمام انجام می دهد. اگر شما بخواهید این روبات بجای شما با اکسل کار کند می بایست لیست کارهایی را که در نظر دارید، تهیه کرده و بصورت کدهای خاصی درآورده و به روبات بدهید. روبات نیز دستورات شما را به ترتیب انجام می دهد. ***VBA*** نیز مانند همین مثال عمل می کند. درواقع ***VBA***زبان خاصی(زبان کد نویسی) برای ارتباط با اکسل است.

VBA با ***VB (Visual Basic)*** تفاوت دارد. اگر چه این دو شباهاتهایی هم با هم دارند ولی کاربردهای آنها متفاوت است. مهارت در ***VB*** به معنای مهارت در ***VBA*** نمی باشد اگر چه کار را برای یادگیری و پیشرفت در ***VBA*** هموار می کند. از سوی دیگر هر قدر در نرم افزارهای آفیس مهارت بیشتری داشته باشد در ***VBA*** توانایی و مهارت بیشتری خواهید داشت.

موارد استفاده از VBA

خودکارسازی(***automating***) کارهای دستی و طولانی

خودکارسازی(***automating***) امور تکراری

سفارشی کردن و ایجاد رابط با کاربر(با استفاده از دگمه ها و منوها)

ایجاد توابع جدید که در اکسل وجود ندارد

ارتباط با سایر برنامه های ***office*** مانند ***Word*** و ***Access*** و ***Powerpoint***

ایجاد ماکرو

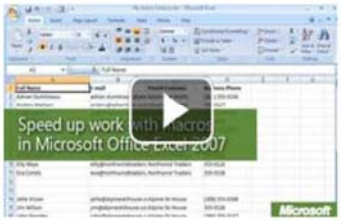
اگر در اکسل(یا هر یک از نرم افزارهای آفیس) بخواهیم کارهای ثابتی را پشت سر هم انجام دهیم از ماکرو استفاده می کنیم. ماکرو این کارهای ثابت و پشت سر هم را ضبط می کند و بعدا میتوان از آن استفاده کرد.

Macro recorderمانند یک ضبط عمل می کند. در واقع از زمانیکه دگمه ***record*** زده میشود شروع به ضبط ریز به ریز کلیه عملیات نموده و تا زمانیکه دگمه ***Stop***زده شود اینکار را ادامه میدهد.

قبل از اینکه کار با ماکروها را شروع کنیم با استفاده از یک فیلم آموزشی مروری سریع بر اینکار خواهیم داشت و در ادامه با مثالهای متعددی به جزییات کار خواهیم پرداخت.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

فیلم آموزشی زیر مراحل کار را خیلی ساده و سریع نمایش می‌دهد.



http://alifatehi.persianqig.com/video/ExcelMacros.wvx

تشریفات و مقدمات- کمي صبور باشید

اضافه کردن تب Developer به اکسل

قبل از اینکه برنامه نویسي در اکسل را شروع کنیم باید مقدمات و تشریفات اینکار را انجام دهیم. براي اینکار لازم است که تغییري جزیي در اکسل ایجاد کنیم تا در بالاي تب هاي اکسل یک تب دیگر همانند شکل زیر که مخصوص برنامه نویسي است اضافه شود:**Dveloper**

در حالت عادی این تب فعال نیست و شما باید به اکسل بگویید که چگونه آنرا نمایش دهد. اضافه کردن آن بسیار ساده است و تنها کافیست یکبار اینکار انجام شود. برای اینکار مراحل و مسیرهای را دنبال کنید:

در اکسل ۲۰۰۷

۱- مسیر ***Office Button*** ---> ***Excel OPtion*** را دنبال کنید

شاید برای شما این سوال پیش بیاید که تب ***Office*** کجاست؟

مایکرو سافت در تمام برنامه های خود علامت یا شکلک (آیکونی) در **گوشه بالای سمت چپ** ایجاد کرده که ***Office Button*** نام دارد. پس از انتخاب آن منوهای آفیس نمایش داده می شود. (یا اینکه ***Alt+F*** را بزنید)

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

۲- تب ***excel option*** ----> ***Popular*** را انتخاب کنید

۳- قسمت زیر را تیک بزنید

Show Developer tab in Ribbon

۴-کلیک ***OK***

در اکسل ۲۰۰۳

در ***excel 2003*** از دو راه می توان پنجره ماکرو را باز نمود:

۱- ***Tools/macro***

۲- فعال کردن ***Visual Basic Toolbar*** و ***Recor New Macro***

به هر ترتیب آیکونهای مربوط به ماکرو از جمله ***Record Macro*** در برنامه اکسل ایجاد می شود که برای ایجاد ماکرو از آن استفاده می کنیم

مثال۱:برای شروع میخواهیم یک ماکرو ساده در اکسل ایجاد کنیم. فرض کنید صفحه ای در اکسل شامل اعداد، روبروی شماست. رئیس شما بالای سر شما ایستاده و مرتبا از شما میخواهد که برخی از سلولها را علامتگذاری کنید. مثلا فونت آنرا ***Arial*** سایز ۱۶ و ***Bold***و رنگ زمنیه سلول را قرمز کنید. اگر بتوانیم یکبار اینکار را انجام دهیم و مرتبا از آن استفاده کنیم سرعت و دقت کارمان افزایش می یابد. ایجاد و بکارگیری یک ماکرو یک راه حل بسیار سریع وساده برای اینکار است.

Record Macroرا از تب ***Developr*** انتخاب می کنیم.

به محض فشردن آیکون، ***Record Macro***پنجره ای باز می شود

این پنجره حاوی اجزای یک ماکرو است. اجزای ماکرو را که شامل موارد زیر است تعیین می کنیم:

۱.نام ماکرو : این نام***ChangeColor*** را برای این ماکرو انتخاب می کنیم.

۲.کلید میانبر(***Shortcut key***) (اجباری در تعیین کلید میانبر نیست.) که بصورت ترکیبی از کلید ***Ctrl*** و یک حرف می باشد: در اینجا***Ctrl + T*** را انتخاب می کنیم.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

۳.توضیحات ماکرو(***Description***)

۴.محل ذخیره ماکرو(بصورت پیش فرض در یک درفایل جاری (***Thisworkbook***) می باشد)

حال عملیات مورد نظر را انجام می دهیم. یعنی سلول مورد نظر را ***Bold*** و رنگ زمینه آن را قرمز میکنیم. با زدن دگمه ***Stop*** عملیات ضبط ماکرو خاتمه می یابد.

۲. اجرا ی ماکرو(Playing Macro)

برای اجرای ماکرو روشهای متعددی وجود دارد

فشردن کلیدهای ***ALT+F8***

فشردن آیکون ***Run Macro***

استفاده ار کلید میانبر ماکرو

بطور خلاصه: از پنجره ماکرو گزینه ***record*** را انتخاب می کنیم عملیات مورد نظر را انجام میدهیم ودر نهایت با زدن دکمه ***stop*** آن را می بندیم. با اجرای ماکرو(با دگمه ***PLAY*** یا استفاده از کلید میانبر) کلیه کارهای مورد نظرمان انجام می شوند.

مثال ۲- میخواهیم ماکرویی ایجاد کنیم که محتویات سلولی که در آن واقع هستیم را به گوشه سمت چپ و بالای اکسل منتقل کند(***A1***).

از تب ***Developer*** دگمه ***Record Macro*** را می‌زنیم. کلید میانبری هم تعیین می کنیم. حال شروع کرده و محتوای سلول را ***Cut*** کرده و در سلول ***A1*** آن را ***Paste*** می‌کنیم و در نهایت دکمه ***Stop Recording*** را می زنیم.

حال هر کجای اکسل که باشیم با فشردن کلید میانبر سلول جاری به ***A1*** منتقل می شود.

حال هر کجای اکسل که باشیم با فشردن دکمه میانبر سلول جاری به ***A1*** منتقل می شود.

پشت پرده ماکرو های اکسل-پیش به سوی برنامه‌نویسی- شیطنت و ناخنک به ماکرو

تا این مرحله یاد گرفتیم که چگونه یک ماکرو را ایجاد و چگونه آنرا اجرا کنیم. همانطور که پیش از این هم گفته شد ماکرو یک برنامه است که توسط ***VBA*** نوشته شده است. بنابراین ***VBA*** برای هر ماکرو کدهایی تولید می نماید. شاید شما کنجکاو هستید و می‌خواهید بدانید این کدها چه هستند!!برای دیدن کدهای ماکرو روشهای متعددی وجود دارد. ولی با توجه به کنجکاوی و اشتیاق شما سعی می شود که کوتاه ترین راه گفته شود.

مسیر ***Developer->Code->Macros*** را انتخاب کنید یا اینکه بطور مستقیم هم می توانید ***ALT+F8*** را بزنید.

بلافاصله پنجره محاوره ای ***Macro*** باز می شود.

ماکروی ***ChangeColor*** را از این پنجره انتخاب می کنیم. و سپس دگمه ***Edi t*** را می‌زنیم.

وارد محیط جدیدی می شویم که با محیط اکسل متفاوت است.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

کدهای تولید شده توسط ***VBA*** نمایش داده می‌شود.

برای دیدن کدهای ماکرو ایجاد شده در پنجره Macro دکمه EDIT را می زنیم.

اگر کدهای نوشته شده در این برنامه را ببینید، به قدرت جادویی ***Recorder Macro*** پی می برید. حتما از خودتان خواهید پرسید: "یعنی من این کدها رو ایجاد کردم؟ من که چیزی از***VBA*** و برنامه‌نویسی نمی‌دونم !!" ما بدون اینکه از برنامه‌نویسی چیزی بدانیم کدهایی جادویی را می‌بینیم که توسط ***Macro Recorder*** تولید شده‌اند.

```
Sub ChangeColor()
'
' ChangeColor Macro
این اولین ماکرو من است'
'
' Keyboard Shortcut: Ctrl+t
'
    With Selection.Font
        .Name = "Arial"
        .Size = 16
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ThemeColor = xlThemeColorLight1
        .TintAndShade = 0
        .ThemeFont = xlThemeFontMinor
    End With

    With Selection.Interior

        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 192
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
```


http://www.alifatehi.persianblog.ir
ali_fateh1@yahoo.com

علی فاتحی ali_fateh1@yahoo.com

```
    Selection.Font.Bold = True
End Sub
```

بخشی از کدها گیج کننده و بخشی هم جالب هستند.

هر چند که هنوز ما اطلاعاتی در رابطه با کدهای ***VBA*** و برنامه نویسی نداریم ولی با کمی دقت ملاحظه می شود به نظر میرسد بخشی از کدها هم قابل درک و شناسایی باشند. مثلا کد زیر را در نظر بگیرید:

Size = 16.

به نظر می‌رسد این همان سایز فونت است. **بیایید ناخنکی به این کدها بزنیم**. البته یک مقدار احتیاط کنید. عدد ۱۶ را به ۲۰ تغییر بدهید و دوباره ماکرو را اجرا کنید.(اگر عجله دارید می توانید با فشردن کلید ***Alt+F11*** اینکار را بکنید).

می‌بینید که همان کار قبلی انجام می‌شود ولی سایز فونت سلول به ۲۰ تغییر می‌کند.

می‌توانید بخشهای دیگر را هم تغییر دهید. و نتیجه را امتحان کنید.

فکر کنم که شروع خوبی داشتیم!!

آزمایش کنید و اجرا کنید. نتیجه جالبی خواهید دید

مثال ۳: در این صفحه ملاحظه می شود که یک برنامه ***VBA***چگونه نوشته می شود و می توان در آن تغییراتی ایجاد نمود.

فرض کنید که صفحه اکسلی با ۵۰۰۰۰ (پنجاه هزار) سطر اطلاعات داریم. میخواهیم سطرهای ۱۰۰۰۰(ده هزار) تا ۲۰۰۰۰(بیست هزار)را حذف کنیم.

برای این منظور ابتدا یک ماکرو ساده در شیت دیگری ایجاد میکنیم. بعد به سراغ ماکرو ایجاد شده می رویم تا ببینیم اکسل چه کدهایی ایجاد کرده است.

کدهای زیر دیده می شود.

```
SubMacro1()
'

' Macro1Macro
' Macro recorded 2010/01/13 byali.fatehi
'
    Rows("3:5").Select
    Selection.Delete Shift:=xlUp

End Sub
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

به نظر میرسد کد ***Rows("3:5").Select*** سطرهای ۳ تا ۵ را انتخاب می کند.اگر این کد را به صورت زیر تغییر دهیم مساله حل می شود.

Rows("10000:20000").Select

البته چون در این ماکرو اشاره ای به نام هیچ شیتی نشده بنابراین هرجا که آنرا اجرا کنیم صحیح عمل خواهد کرد. پس ابتدا به شیت موردنظرمی رویم و سپس آنر اجرا می کنیم.

آزمایش کنید و اجرا کنید. نتیجه جالبي خواهید دید

مثال ۴: انتقال اطلاعات از اینترنت به اکسل:

از منوی دیتا آدرس زیر را انتخاب می کنیم:

Data/import external data/new web query

را انتخاب کرده و در صفحه ای که ایجاد شده است آدرس سایتی را می دهیم. از کنار هر کدام از قطعات نوشته شده که حاوی فلش زرد رنگ است را انتخاب کرده و به داخل اکسل Import می کنیم. می توان تعیین نمود که اطلاعات هر چند دقیقه یکبار بروز شوند. میتوان همه روزه با ***REFRESH*** کردن در گزینه دیتا اطلاعات را بروز کرد.

برای این عملیات ماکرویی ایجاد می کنیم. و همه روزه با اجرای این ماکرو اطلاعات مورد نظر را از اینترنت دریافت می کنیم.

آزمایش کنید و اجرا کنید. نتیجه جالبي خواهید دید

تمرین:

ستونهای ***A*** تا ***E*** از یک شیت اکسل حاوی اطلاعات است. ماکرویی ایجاد کنید که برای هر ستون یک شیت ایجاد کرده و اطلاعات آن ستون را به شیت جدید کپی کند.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه دوم:عمومی نمودن ماکرو

ماکروهایی که در اکسل ایجاد میشوند تنها در همان فایل قابل اجرا می باشند. برای اینکه بتوان یک ماکرو را همواره و همه جا در اکسل اجرا نمود روشهای ساده و مقدماتی مختلفی وجود دارد. یک روش ایجاد ماکرو به شکل "ماکرو عمومی" و روش دیگر "تخصیص دگمه یا آیکون" برای ماکرو است.

در اکسل ۲۰۰۷ و ۲۰۱۰

ایجاد ماکروی عمومی

اگر میخواهیم یک ماکرو همواره و همه جا در اکسل قابل استفاده باشد در هنگام ذخیره کردن ماکرو در زیر گزینه ***Store macro In*** گزینه ***Personal Macro Workbook*** را انتخاب کنید. با انتخاب این گزینه اکسل یک فایل مخفی(***hidden***) که حاوی این ماکرو است ایجاد میکند.بدین ترتیب هر زمان که شما اکسل را باز کنید این ماکرو نیز قابل اجرا خواهد بود.

نکته :البته با توجه به اینکه این ماکرو در یک فایل مخفی است برای اینکه بتوانید هر گونه تغییری در آن ایجاد کنید یا حتی آنرا حذف کنید لازم است در ابتدا آنرا از حالت ***hiden*** خارج کنید. اینکار از مسیر ***view>unhide*** امکانپذیر است. بعد از انجام تغییرات از همان مسیر ***hide*** می شود.

Workbook مخفی ایجاد شده با نام ***Personal.xlsb*** در پوشه ***XLStart*** قرار دارد. مسیر این پوشه در ویندوزهای متفاوت با هم فرق میکند :

مسیر در window vista :

C:\Users\user name\AppData\Local\Microsoft\Excel\XLStart

مسیر د ر window xp:

C:\Documents and Settings\user name\Application Data\Microsoft\Excel\XLStart

تمام ***workbook*** هایی که در پوشه ***XLStart***باشند بطور اتوماتیک با باز شدن اکسل باز می شوند.

اجراي ماكرو از طريق دگمه يا آيكون

در اكسل شما ميتوانيد يك دگمه يا كليد را براي اجراي يك ماكرو در نظر بگيريد بطوريكه با فشردن آن دگمه ، ماكرو اجرا شود.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

براي اينكار به مسير زير برويد:

Microsoft Office Butto> Excel Options> Customize> Choose commands from>Macros

در اينجا ليست تمامي ماكروهايي كه باز هستند داده ميشود.

ماكروهاي اين ليست عبارتند از:

ماكروهاي فايل جاري

ماكروهايي كه در *Personal Macro Workbook* هستند

ماكروهايي كه در *add-ins* ها هستند.

بديهي است براي اينكه يك دگمه بتواند همواره و همه جا در اكسل عمل كند مي بايست ماكرو در *Personal Macro Workbook* يا *add-ins* باشد.

سپس ماكرو را انتخاب و*add* مي كنيم.

با استفاده از تب *modify* مي توان آيكون و نام مناسبي براي آن ايجاد كرد.

دگمه اي كه بدين ترتيب ايجاد ميشود به ليست *Quick Access Toolbar* ،اضافه مي شود.

در اکسل ۲۰۰۳

گاهي اوقات لازم است براي اجراي يک ماکرو از آيکون استفاده کرد.

براي اينکه به يک ماکرو يک آيکون اختصاص دهيم مراحل زير را طي ميکنيم:

۱. انتخاب آيکون

مسير زير را طي مي کنيم:

button Tools→customize→commands→macros→custom

علامت را گرفته و به Toolbars انتقال ميدهيم(Drag مي کنيم).

۲- تنظيمات آيکون و تخصيص ماکرو به آن

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

بر روی ***modify selection*** که فعال شده کلیک می کنیم. گزینه های مربوط به این آیکون مانند نام، شکل و سایر مشخصات را تنظیم می کنیم. برای تخصیص یک ماکرو به این آیکون از گزینه ***Assign macro*** استفاده می کنیم. این ماکرو می تواند از ***workbook*** جاری یا سایر ***workbook*** های موجود باشد.

نکته: هنگام نامگذاری ماکرو توجه شود که علامت & قبل از هر حرفی که در نام آیکون بکار گرفته شود آن حرف را به شکل HotKey تعریف می کند. به عنوان مثال ve&mo باعث میشود که نام آیکون به شکل move نمایش داده شود. یعنی Alt +v هم یک کلید میانبر برای این آیکون است.

ali_fatehi@yahoo.com علي فاتحي

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه سوم: شروع برنامه نویسی در اکسل

روشهای مختلفی برای شروع و ایجاد یک برنامه با ***VBA*** وجود دارد. یک روش ساده ایجاد یک ***command button*** بر روی صفحه گسترده محیط اکسل و شروع برنامه نویسی با کلیک بر روی آن است. روش دیگر نوشتن توابع در داخل ویرایشگر ***VB*** یا همان ***VBE*** می باشد. کار را با روش اول شروع می کنیم

روش اول : ایجاد یک برنامه با استفاده از ***command button***

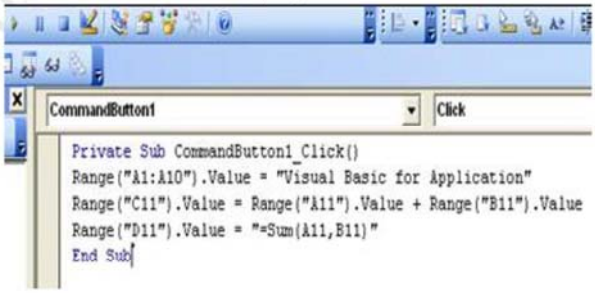
برای قرار دادن ***command button*** بر روی محیط اکسل مسیر زیر را طی می کنیم:

منوی ***View*** →***Toolbar***→***Control ToolBox***

(البته اگر قبلا نوار ابزار ***Visual Basic Editor*** نیز فعال شده باشد میتوانید ***Control ToolBox*** را از آن انتخاب کنید.)

بعد از طی مسیر فوق ***Control ToolBox*** ظاهر میشود. ***command button*** را انتخاب و آنرا بر روی صفحه گسترده محیط اکسل اکسل قرار میدهیم. یک دگمه در محیط اکسل ظاهر میشود.

با کلیک کردن بر روی ***command button*** بلافاصله ***Visual Basic Editor*** یا همان ***VBE*** ظاهر می شود. بلافاصله میتوانید دستورات خود را همانند شکل تایپ کنید.



http://www.alifatehi.persianblog.ir
ali_fateh1@yahoo.com

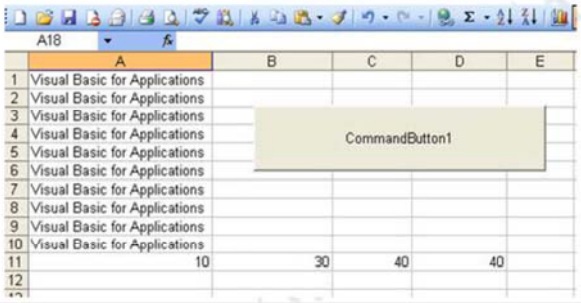
با دقت در این شکل ملاحظه میشود که:

دستور اول در نظر دارد مقدار(***Value***) سلولهای ***A1*** تا ***A10*** را با عبارت ***"Visual Basic for Application"*** پر کند.

دستور بعدی نیز حاصلجمع سلولهای ***A11*** و ***B11*** را در سلول ***C11*** قرار میدهد.

دستور آخر نیز مقدار سلول ***D11*** را نیز با روش دیگری برابر با حاصلجمع ***A11*** و ***B11*** قرار میدهد. با اندکی دقت متوجه می شویم که این دستور با دستور قبلی متفاوت است.

خروجي و رابط برنامه



برای اجرای این برنامه به محیط اکسل برمی گردیم. (میتوان از ***ALT + F11*** نیز استفاده کرد).

با فشردن آیکن ***Exit Design Mode*** از ***Control Toolbox*** از حالت طراحی خارج می شویم. پس از آن با کلیک بر روی ***command button*** برنامه اجرا می شود که خروجی آن در شکل نشان داده شده است.

ایجاد یک برنامه ساده در ***VBE***

برای نوشتن یک برنامه در محیط برنامه نویسی اکسل مراحل زیرانجام می شود:

۱- ورود به محیط برنامه نویسی یا همان ***Visual Basic Editor***

برای اینکار چند روش وجود دارد

- استفاده از کلیدهای میانبر: ***ALT + F11***
- استفاده از منوی اکسل: ***Tools-> Macro-> Visual Basic Editor***
- استفاده از ***ToolBar*** با همان نوار ابزار ***Visual Basic*** و فشردن آیکون ***Visual Basic Editor***

به هر حال با یکی از روشهای فوق وارد محیط برنامه نویسی می شویم

۲- ورود به منوی ***Insert*** و انتخاب گزینه ***Module***

۳- نوشتن فرمان (کلمه کلیدی) ***Sub*** و سپس نام برنامه

۴- فشردن دگمه ***Enter***

۵- بلافاصله بطور اتوماتیک فرمان ***End Sub*** در یک خط جدید اضافه می شود.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

۶- کدهای برنامه را خط به خط بین فرمانهای ***Sub*** و ***End Sub*** می نویسیم.

به عنوان مثال:

```
Sub MyProgram()
Range("A1:A10").Value = "Visual Basic For Applications"
Range("A11").Value=10
Range("B11").Value = 20
Range("C11").Value = "=A11+B11"
End Sub
```

ملاحظه می شود که در سطر ***C11*** ارزشهای موجود در ***A11*** و ***B11*** جمع زده می شود.
به علامت = در داخل کوتیشن توجه شود.

اجرای برنامه : برای اجرای برنامه چندین روش وجود دارد

- فشردن کلید ***F5***
- فشردن آیکن ***Rub Sub/User Form*** از ***Toolbar***
- بازگشت به محیط ***Excel*** و استفاده از ***ALT + F11***

نکته: در صورتی که از علامت ' در ابتدای یک خط برنامه نویسی در محیط ***VB*** استفاده شود آن خط سبز شده و عملیاتی بر روی آن خط اجرا نخواهد شد. این خط اصطلاحا ***Comment*** (توضیحات)نامیده می شود.
Comment حالت اجرایی ندارد و از آن برای ارایه توضیحات استفاده میشود. ***Comment*** ها به خوانایی برنامه بسیار کمک میکنند.

فرمانهای ***Select , Value , Range***

فرمان ***Range*** برای مشخص کردن یک ناحیه در اکسل استفاده می شود.

با انجام دستور ***Select*** سلهای مورد اشاره انتخاب می شود. می توان بعد از ***Rrange*** از این دستور استفاده نمود.

دستور زیر را وارد کنید:

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
Range("F1:F10").Select
```

و دکمه ***RUN*** را بزنیم، ملاحظه می شود که سطرهای مورد از ***F1***تا ***F10*** اشاره های انتخاب میشوند.
می توان به جای : از علامت , استفاده نمود که در این حالت سلولهای ***F1*** و ***F10*** می انتخاب می شوند.

VALUE=
برای دادن مقدار از این دستور استفاده می کنیم.

مواردی که در بالا اشاره گردید خلاصه ای از سه فرمان ***RANGE VALUE, SELECT***,بود که اشاره ای مختصر به آن گردید. در اینجا مقدمه ای بر برنامه نویسی ارایه گردید و در جلسات بعدی با برنامه نویسی بیشتر آشنا خواهیم شد.

ali_fatehi@yahoo.com
علی فاتحی

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه چهارم- کار با سلولها و ناحیه ها(Ranges objects Cells and)

از رایج ترین و پرکاربردترین کارها در اکسل کار با سلولها و ناحیه ها است. انتخاب یک *cell* یا *Range*، وارد نمودن فرمول، تغییر رنگ و قلم و وارد نمودن یک عدد از جمله کارهایی است که می‌توان در مورد آنها انجام داد. معمولا در ابتدا *Range* را مشخص و سپس یکی از ویژگیهای(*Propertires*) آن را تنظیم کرده یا یک متد(*Method*) را بر آن اعمال می‌کنیم.

یک *Range* می‌تواند شامل تنها یک سلول یا چندین سلول باشد. این سلولها می‌توانند در کنار یکدیگر یا بطور پراکنده باشند. در ادامه متداول ترین روشهای تعریف و کار با شی *Range* ارایه می‌شود

۱.۱. استفاده از نماد گذاري *A1*

با استفاده از روش نمادگذاري *A1* ميتوان يک *Range* را مشخص کرد. برنامه زير سلول‌هاي موجود در ناحيه *A1* تا *D5* را *Bold* مي‌کند. براي اينکار ابتدا اين ناحيه بصورت *Range("A1:D5")* مشخص و سپس ويژگي مورد نظر را تغيير مي‌دهيم.

```
Sub FormatRange()

Range("A1:D5") .Font.Bold = True

End Sub
```

براي انتخاب يک *Range* ، ابتدا آنرا مشخص و سپس متد *Select* را بر آن اعمال مي‌کنيم.

- کد زير سلول‌هاي همسايه در ناحيه *A1* تا *D5* را انتخاب مي‌کند:

```
Range("A1:D5").Select
```

- ميتوان نقاط غيرهمسايه و پراکنده را به عنوان *Range* معرفي کرد. کد زير به ناحيه اي که شامل سلول *A8*، ناحيه *B2* تا *C5* و سلول *K2* اشاره مي کند:

```
Range(" A8, B2:C5 , K6")
```

ها در رابطه با نواحي مستطيلي است که *Range*يک روش بسيار پرکاربرد در معرفي با دادن دو سر قطر يک ناحيه مستطيلي آنرا سلولهاي آن نيز همجوار هستند. مي توان ناحيه اي مثل مشخص کرد. که روش جايگزين براي معرفي است: ***A1:D5***

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

Range(" A1","D5")

را نشان مي‌دهد *A1* نماد گذاري جدول بعدي برخي از آدرس دهي ها با شيوه.

عبارت	مفهوم
Range(" A1")	سلول *A1*
Range(" A1:B5")	سلولهاي *A1* تا *B5*
Range("C5:D9,G9:H16 ")	شامل دو ناحيه : ناحيه *C5* تا *D9* و ناحيه *G9* تا *H16*
Range("A:A")	ستون *A* (*Column A*)
Range("1:1")	سطر ۱ (*Row 1*)
Range("A:C")	ستونهاي *A* تا *C*
Range(" 1:5")	سطرهاي ۱ تا ۵
Range(" 1:1,3:3,8:8")	سطر ۱ و سطر۳ و سطر ۸
Range(" A:A,C:C,F:F")	ستون *A*، ستون *C* و ستون *F*

بطور كلي ويژگي *Range*بصورتهاي زير بكار مي‌رود:

Range("range1,range2,....rangeN")

يا

Range("ناحيه اول ,ناحيه دوم ,... , ناحيه آخر")

Range(Cell1,Cell2)

***Cell1* و *cell2* نقاط دو سر قطر يک ناحيه مستطيلي است**

كه ميتوان با استفاده از مختصات دو سر قطر يک ناحيه مستطيلي *Range* را مشخص كرد.

۱.۲ ويژگي *Cells* و روش آدرس دهي مختصاتي

با استفاده از ويژگي *Cells* ميتوان يک سلول را با استفاده از آدرس سطر و ستون آدرس دهي و مشخص كرد. اين ويژگي يک شي از نوع *Range* را به ما ميدهد كه اين شي تنها يک سلول دارد.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

ویژگي *Cells* بصورت کلي زیر نوشته مي‌شود:

*Cells(*شماره ستون ,شماره سطر*)*

در مثال بعدي `Cells(6,1)` سلول *A6* را مشخص و ویژکي *Value*را برابر با *alifatehi.persianblog.ir*مي‌کند. این بدین معناست که در سلول *A6* عدد *10* قرار مي‌گیرد.

```
SUB EnterValue
"Cells(6, 1).Value = "alifatehi.persianblog.ir
End Sub
```

میتوان بجاي شماره سطر و یا ستون از اسامي‌متغیرها نیز استفاده کرد.

```
Sub EnterValue()
 i=cells(1,1).Value
 j=6
"Cells(i, j).Value = "alifatehi.persianblog.ir
End Sub
```

مي‌توانیم روش آدرس دهي مختصاتي را با روش قبلي ترکیب و ناحیه مستطیلي *A1:D5* را مشخص کنیم:

```
Range(Cells(1, 1), Cells(4, 2))
```

- براي اینکه بصورت ***همزمان*** و در یک لحظه تغییر ویژگیها(*properties*) یا انجام یک متد بر روي یک ناحیه صورت پذیرد از ویژگي *Range* استفاده کنید.

۱.۳ نحوه ارجاع به سطرها و ستونها

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

با استفاده از ویژگي *Rows* و *Columns* میتوان به راحتي با سطرها و ستونها کار کرد. این ویژگي ها نیز یک شي *Range* به ما مي‌دهد که شامل ناحیه اي از سلولهاست. در مثال بعدي *Rows(1)* ستون *1* را مشخص مي‌کند. ویژگي ***Bold*** از شي ***Font*** براي این ناحیه را با مقدار *True* تنظیم مي‌کند.

```
()Sub RowBold
Worksheets("Sheet1").Rows(1).Font.Bold = True
End Sub
```

جدول بعدي برخي از آدرس دهي هاي سطرها و ستونها را با استفاده از ویژگي هاي ***Rows*** و ***Columns*** نشان مي‌دهد.

عبارت	مفهوم
	Rows(1) سطر *1*
Rows	تمامي‌سطرهاي *Sheet*
	Columns(1) ستون یک
	Columns("A") ستون *A*
Columns	تمامي‌ستونهاي *Sheet*
	Rows("1:5") سطرهاي *1* تا *5*
	Columns("B:D , F,K") ستونهاي *B* تا ***ستون*** *D* و *F* و ***ستون*** *K*

۱.۴ ارجاع به *Range*هاي نامگذاري شده

فرض کنید که در شیت جاري ناحیه اي با نام *MyRange* وجود داشته باشد . براي پاک کردن محتواي این ناحیه ابتدا با دستور *Select* آن را انتخاب و با اعمال متد *ClearContents* محتوایش را پاک مي‌کنیم.

```
Sub ClearRange()
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
Range("MyRange").Select
Selection.ClearContents
End Sub
```

اگر ناحیه اي در اکسل در *workbook* یا *Sheet* جاري نباشد و به عنوان مثال دریک *workbook* به نام *"MyBook.xls."* با نام *"MyRange"* نامگذاري شده باشد مي‌توان این ناحیه را به شکل زیر مورد استفاده قرار داد:

```
Sub FormatRange()
Range("MyBook.xls!MyRange").Font.Italic = True
End Sub
```

ارجاع به تمامي‌سلولهاي یک شیت 1.5

براي اینکه تمامي‌سلولهاي یک شبت را مشخص کینم از ویژگي *Cells* بدون هیچ سطر و ستون و به شکل خالي استفاده مي‌کنیم. مثال بعدي محتواي کلیه سلولهاي موجود در *Sheet1* از *workbook* جاري را پاک مي‌کند.

```
Sub ClearSheet()
Worksheets("Sheet1").Cells.ClearContents
End Sub
```

ali_fatehi@

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه پنجم- متغیرها(variables)

۱. مفهوم متغیر

متغیر شبیه یک جعبه پستی است. محتوای هر متغیر هر لحظه می تواند تغییر کند. همانگونه که محتوای یک جعبه پستی هر لحظه می تواند تغییر نماید. در *VBA* متغیرها بخش هایی از حافظه کامپیوتر هستند که برای نگهداری داده ها مورد استفاده قرار می گیرند. درست شبیه صندوق پستی هر متغیر دارای یک نام است. برای هر متغیر دو چیز در نظر گرفته می شود:

۱- نام متغیر

۲- نوع داده اي(*Data Type*)

۱.۱ نامگذاری متغیر

برای نامگذاری یک متغیر قواعد زیر می بایست رعایت گردند:

نام هر متغیر کمتر از ۲۵۵ کاراکتر می باشد

فاصله خالی(***space***) مجاز نمی باشد

نقطه مجاز نمی باشد

نام متغیر با عدد شروع نمی شود

برخی کاراکترها مانند ! ، @ ، # ، $ ، ٪ ، ^ ، & ، × ،) ، (، + نمی توانند مورد استفاده قرار گیرند

مثال براي نامهاي معتبر و نامعتبر

نام معتبربراي يک متغير	نام نامعتبر براي يک متغير
My_Car	***My.Car***
ThisYear	***1NewBoy***
Long_Name_Can_beUSE	***He&HisFather*** ****& is not acceptable***
Group88	***Student ID*** **** Spacing not allowed***

۱.۲ نوع داده اي(***Data Type***)

نوع داده ای در دو شکل کلی دسته بندی می شوند

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

الف- نوع داده ای عددی(*Numeric Data Type*) : این شکل از داده ها قابلیت محاسباتی دارند و می توانند توسط عملگرهایی نظیر چهار عمل اصلی مورد استفاده قرار گیرند. در *VBA* این گروه شامل ۷ نوع داده مطابق جدول زیر است:

Numeric Data Types		
Data Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values 1.401298E-45 to 3.402823E+38 for positive values.
Double	8 bytes	-1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Decimal	12 bytes	+/- 79,228,162,514,264,337,593,543,950,335 if no decimal is use +/- 7.9228162514264337593543950335 (28 decimal places).

ب- نوع داده اي غير عددي (Noun-Numeric Data Type)

این نوع داده در جدول زیر بطور خلاصه آمده است. این نوع داده قابلیت محاسباتی ندارد و معمولا از آن برای ارجاع به اشیا یا انبوهی از اطلاعات استفاده می شود.

Noun-Numeric Data Types		
Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False
Object	4 bytes	Any embedded object
Variant(numeric)	16 bytes	Any value as large as Double

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

Variant(text)	***Length+22 bytes***	***Same as variable-length string***

در **VBA** برای معرفی متغیر نیاز است تا نام و نوع داده ای به آن تخصیص یابد.

۱.۳ معرفی متغیرها

برای معرفی یک متغیر لازم است نام متغیر و نوع داده ای آن را مشخص کنیم. (البته می توان از نوع داده ای صرفنظر کرد که در اینصورت نوع داده ای ***Variant*** فرض میشود) برای این منظور از دستور ***Dim*** به شکل زیر استفاده می شود:

```
Dim variableName as DataType
```

مثال:

```
Dim password As String Dim yourName As String
Dim firstnum As Integer
Dim secondnum As Integer
Dim total As Integer
Dim BirthDay As Date
```

مي توان معرفي متغيرها را به شكل زير تركيب نمود

```
Dim password As String, yourName As String, firstnum As Integer.
```

متغير هاي تاريخ در داخل # # و متغيرهاي حروفي در داخل " " گذارده مي شود.

در مثال بعدي سه نوع متغير رشته اي ، تاريخي و ارز آورده شده است.

```
Private Sub CommandButton1_Click( )
Dim YourName As String
Dim BirthDay As Date
Dim Income As Currency
YourName = "Sadeghi"
BirthDay = #1 April 1980#
Income = 10000000
Range("A1") = YourName
Range("A2") = BirthDay
Range("A3") = Income
End Sub
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

مثال :فرض کنیم سه شیت داریم که در ستون اول هر سه شیت اطلاعات داریم. مي خواهيم اطلاعات ستونهاي اول مربوط به هر سه شیت را در ستون اول شیت یک قرار دهد.

براي اين منظور ابتدا ماکرويي را ايجاد مي کنيم و ايده هاي اصلي را به بهره گيري از کدهاي اين ماکرو مي گيريم و براي نوشتن برنامه استفاده مي نماييم.

توضيح و يا دآوري:

فرمول***count***در اکسل تعداد رديفهاي پرشده در درستون ***a***را مي دهد.

پس از نوشتن ماکرو وارد محيط ***VBA*** مي شويم و مي بينيم که اطلاعات زير نوشته شده است:

```
Sub dll()

' www.alifatehi.persianblog.ir


Range("A1:A32").Select
Selection.ClearContents
Sheets("Sheet2").Select
Range("A1:A30").Select
Selection.Copy
Sheets("Sheet1").Select
Range("A1").Select
ActiveSheet.Paste
Sheets("Sheet3").Select
Range("A1:A70").Select
Application.CutCopyMode = False
Selection.Copy
Sheets("Sheet1").Select
Range("A31").Select
ActiveSheet.Paste
End Sub
```

مساله موجود اين است که در انتهاي اطلاعات وارد شده در شيت اول(پس از دليت کردن) اطلاعات شيت سوم را وارد نمايد.براي اين منظور بايد تغييراتي در ماکرو نوشته شده در اکسل ايجاد کرد و با استفاده از اطلاعات ستون ***c*** و تعريف دو متغير ***x1 ,x2*** و اضافه نمودن در سطر هاي برنامه نوشته شده به اين هدف رسيد. بخش اصلي براي حل اين مساله **استفاده از روش آدرس دهي مختصاتي براي معرفي ناحيه ها** است . زيرا نواحي که انتخاب مي شوند تغيير ميکنند و مي بايست از دستور ***cells(i,j)*** استفاده کنيم

```
Sub rep()
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
Range("A:a").Select
Selection.ClearContents
Sheets("Sheet2").Select

Dim x1, x2, x3

x1 = Cells(1, 3)
Range("A1", Cells(x1, 1)).Select
Selection.Copy
Sheets("Sheet1").Select
Range("A1").Select
ActiveSheet.Paste
Sheets("Sheet3").Select
x2 = Cells(1, 3 )
Range("A1", Cells(x2, 1)).Select
Application.CutCopyMode = False
Selection.Copy
Sheets("Sheet1").Select
x3 = x1 + 1
Cells(x3, 1).Select
ActiveSheet.Paste
Range("B1").Select

End Sub
```

علي
ali_fatehi@

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه ششم- دستور شرطی if then else در محیط برنامه نویسی اکسل

برنامه هايي كه تا اينجا نوشته شدند همگي داراي كدهايي بودند كه پشت سر هم انجام ميشدند. در برخي از برنامه ها در صورت وجود شرطي بخشهايي از برنامه انجام ميشوند. به عبارت ديگر در اينگونه برنامه ها تصميم گيري صورت مي گيرد. ساختارهاي شرطي يا ساختارهاي تصميم بخش بسيار مهمي از برنامه نويسي را شامل ميشوند كه در ادامه به توضيح آنها می پردازيم

ساختار تصميم يا شرطی *IF*

- ساختار اوليه وساده:

if شرط ***then***
دستورات
end if

در اين ساختار ابتدا شرطي بررسي ميشود و در صورت وجود آن شرط دستو يا دستورات مورد نظر انجام ميشوند.

مثال: در سلول ***A1*** عددي قرار دارد كه سن شخصي را نشان ميدهد. ميخواهيم با توجه به سن در سلول ***B1*** پيغامي بنويسيم

```
Sub grade()

If Range("a1") >= 17 Then
      Range("b1") = "good"
End If

End Sub
```

- ساختار كامل دستور شرطي ***If***

If شرط۱ ***then***
دستورات

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

Elseif 2 شرط ***then***
دستورات

Elseif شرط آخر ***then***
دستورات

Else
دستورات
Endif

در این شکل از دستور شرطي در صورت وجود شرط اول دستوراات مربوط به آن، در صورت وجود شرط بعدي دستورات مربوط به آن و در نهایت براي بقیه حالتها هم دستورات مربوطه انجام میشوند.

مثال:برنامه اي بنویسید که براي نمرات بالاي ۱۷ عبارت ***A*** بین ۱۷ تا ۱۴ عبارت ***B*** و کمتر از ۱۴ عبارت ***C***را تایپ نماید.

جواب:

```
Sub grade()

If Range("a1").Value >= 17 Then
        Range("b1").Value = "A"

ElseIf Range("a1").Value < 17 And Range("a1").Value >= 14 Then

        Range("b1") = "B"

Else

        Range("b1").Value = "C"

End If

End Sub
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه هفتم- ساختار شرطی Select Case

اگر دربرنامه با توجه به مقادیرمشخصي که یک متغیر دارد تصمیم گیري شود از ساختار ***SELECT CASE*** استفاده مي کنیم. این دستوربا کاربردي مانند دستور ***IF***مي باشد.
براي استفاده ار این دستور شروط مورد نظر را بر روي یک متغیر اعمال مي کنیم.

شکل کلي دستور *Select Case*

Select Case **نام متغیر**

Case **حالت اول**

دستورات

Case **حالت دوم**

دستورات

.
.
.

Case Else

دستورات

End Select

در زیر برنامه اي نوشته شده است که براي مقادیر مختلف موجود در سل ***A1*** یک شیت اکسل عبارات ***A , B, C, ...*** را در سل مقابل آن مي گذارد.

```
Sub level( )
x = Range("a1").Value
Select Case x
Case 17 To۲۰
Range("b1").Value " = A"
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
Case 14 To۱۷
Range("b1").Value " = B"
        Case 12 To۱۴
Range("b1").Value " = C"
Case 10 To۱۲
Range("b1").Value " = D"
        Case 0 To۱۰
Range("b1").Value " = E"
Case Else
Range("b1").Value" = false"
End Select
End Sub
```

ساختار حلقه ای FOREach - NEXT

حلقه ها دستوراتی هستند که می توان جهت انجام دستورات تکراری از آنها استفاده نمود. حلقه ***FOREach*** برای کار بر روی عضوهای(***members***) یک مجموعه(***collection***) استفاده می شود. مجموعه شامل تعدادی شیء یکسان است. به عنوان مثال ***Range("A1:A10***") یک ***collection*** است زیرا تعدادی عضو به نام سلول است که همگی از یک جنس هستند. ***worksheet*** های یک فایل اکسل مجموعه ای به نام ***Worksheets*** تشکیل می دهند. جارتهای یک فایل اکسل مجموعه ای به نام ***Charts*** تشکیل می دهند.

الگوی استفاده از یک حلقه ***FOREach*** در برنامه نویسی به صورت زیر می باشد:

```
For Each عضو In مجموعه
دستورات
NEXT
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

در برنامه ای که در زیر نوشته شده عملیات مورد نظر بر روی ردیفهای ۱ تا ۱۰ ستون ***A*** انجام می شود.برنامه به گونه ای نوشته شده است که اعداد کمتر از ۱۰موجود در ستون اول، را ***Bold*** کند

حال با استفاده و ترکیب دو دستور ***FOR-NEXT***به سادگی برنامه مورد اشاره به شرح ذیل نوشته می شود:

```
Sub range_leve()

For Each c In Range("a11:h20")

  If x < 10 Then c.Font.Bold = True

Next

End Sub
```

استفاده از فرمان ROW در برنامه:

برنامه بالا برنامه مناسبی بود اما در فاز بعدی برنامه نویسی ما خواهان این نکته هستیم برنامه ای نوشته شود که به ازای مقادیر مختلف در یک ستون در اکسل عباراتی متناسب با آن اعداد را در مقابل آنها بنویسد.

تنها نکته موجود در این برنامه وجود دستور ***ROW*** است که از دسته دستوراتی مانند ***VALUE*** می باشد که در اینجا ***ROW*** به مفهوم ردیف مورد نظر می باشد.و برای نیل به هدف بالا می بایست دو متغیر تعریف نمود.

متغیر اول برای عدد مورد نظر که هدف برنامه آن است و متغیر دوم ردیف عدد مود نظر که برای اینکه در مقابل آن عدد باید عبارت مورد نظر برنامه تایپ گردد، مورد نیاز است.مع الوصف برنامه مورد نظر به شکل زیر نوشته می شود:

```
Sub range_level()
 Dim c As Range
 For Each c In Range("a1:a10")
 x = c.Value
i = c.Row

Select Case x
Case 17 To 20
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
            Cells(i, 2) = "A"
        Case 14 To 17
            Cells(i, 2) = "B"
        Case 12 To 14
            Cells(i, 2) = "C"
        Case 10 To 12
            Cells(i, 2) = "D"
        Case 0 To 10
            Cells(i, 2) = "E"
        Else Case
            Cells(i, 2) = "ERROR"
        End Select
        Next
        End Sub
```

ali_fatehi@yahoo

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه هشتم: Message Box

یک پنجره پیغام(*Message Box*) از سه بخش به شرح ذیل ساخته شده است:

عنوان پیغام , شرح پیغام گزینه هاي پیغام

Message box هايي که در طول يک برنامه به آنها نياز داريم شامل دو گروه مي باشد:

الف- پيغامهايي که صرفا اطلاعاتي بوده و به کاربر مفهومي را بدون آنکه عملياتي بر روي داده ها انجام دهد انتقال مي دهد.مانند پيغامهاي خوش آمد گويي.

ساختار ساده اينگونه پيغامها به صورت زير در محيط ***VBA*** به صورت زير است:

```
Sub message()
    MsgBox "hello my friends"
End Sub
```

ب- پيغامهاي عملياتي که بر روي داده هاي موجود در بانک اطلاعاتي يا برنامه محاسباتي کاربر، تغييراتي انجام مي دهد. در اينجا ساختار اصلي يک پيغام، که شامل هر سه قسمت يک پيغام است(عنوان، شرح و گزينه) مشاهده مي شود
در زير مثالي آورده شده است با شرحي که آيا مايل هستيد ادامه دهيد و سه گزينه انتخابي بله، خير و کنسل را شامل مي گردد. سپس به ازاي انتخاب گزينه بله تعدادي از داده ها را دليت نموده و به ازاي گزينه خير، همان اطلاعات را انتخاب مي نمايد.
ساختار چنين برنامه اي به شکل زير نوشته مي شود:

```
Sub message()

x = MsgBox("do you want to continue?", vbYesNoCancel,
"message box"(

If x = vbYes Then

   Rows("1000:2000").Select

   Selection.Delete Shift:=xlUp

ElseIf x = vbNo Then

   Rows("1000:2000").Select
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

End If
End Sub

همانطور که در خط دوم برنامه ملاحظه مي گرددساختار کلي يک کادر پيغام شامل سه بخش عنوان، شرح و گزينه هاي پيغام مي باشد که به علامت , از هم جدا مي شود. به عبارت ***VBA*** در قسمت مياني فرمول توجه کنيد.

نکته: اطلاعات کامل در مورد *Message Box* در ***HELP*** اکسل آورده شده است .

ali_fatehi@yahoo.com علي فاتحي

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه نهم - حلقه های FOR -NEXT در محیط برنامه نویسی اکسل

حلقه ها

جهت انجام کارهای تکراری از حلقه ها استفاده می شود. در ابتدا با حلقه های ***For***شروع میکنیم .
ساختار کلی این حلقه به شکل زیر ا ست:

For *counter= start* To *end* [Step *step*]

دستورات

Next [*conter*]

مثال : برنامه ای می نویسیم که سلولهای ناحیه A1 تا A10 را به صورت یکی درمیان پر نماید:

جهت نیل به این هدف از فرمول زیر استفاده می شود:

```
For i=1 to 10 step۲

Cells(1,i)=i

Next i
```

به همین ترتیب و با استفاده از تکنیک ***step*** در حلقه ها می توان از اعداد زوج با ***step 2*** و اعدادفرد با ***step*** یک یا سه بهره جست.

حلقه هاي تو در تو

با استفاده از چندین حلقه ***for*** می توان امور تکراری پیچیده تر و بیشتری را انجام داد.
در مثال زیر تلاش ما بر این است که یک جدول ضرب ۱۰×۱۰ در محیط اکسل ایجاد نماییم.
برای رسیدن به این منظور به راحتی و با استفاده از ۲ حلقه می توان این برنامه را به شکل زیر نوشت:

```
Sub ZARB()

For i = 1 To۱۰

  For J = 1 To۱۰

Cells(i, J) = i * J
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
Next J

Next i

End Sub
```

استفاده از ساختارهاي شرطي در حلقه ها

حال مي خواهيم كه در همين جدول مضارب ۵ را با تغيير فونت مشخص نمايد:

براي اين منظور بايد بعد از سطر چهارم برنامه خط زير را نوشت:

```
i=10 or j = 10 Then If i = 5 Or j = 5 Or

۲۵ = Cells(i, j).Font.Size

End If
```

می توان در برنامه موجود کار پیغامی مبنی بر اینکه آیا مایل به ذخیره نمودن برنامه هستید یا خیر، پس از ***if*** برنامه ایجاد نمود.

برخي تكنيكها درحلقهه ها

شمارنده(COUNTER)

در جلسات گذشته دیدیم که برای جمع نمودن تعداد داده های موجود در یک سطر یا یک ستون از فرمول ***COUNT***=( ) در یک سل از سلهای اکسل استفاده می شود. در این مرحله می‌خواهیم با استفاده از متغیری تحت عنوان شمارنده (یا ***COUNTER***) عمل شمارش را انجام دهیم.

بدین منظور ابتدا باید مقدار شمارنده را برابر صفر گذاشته و در مرحله بعدی پس از گذاردن شرط برنامه، به شمارنده یک واحد یا هر مقداری که لازم باشد اضافه یا کم می کنیم.

مثال : نمرات دانش آموزان یک کلاس در ستون اول یک شیت وجود دارد. می خواهیم برنامه ای بنویسیم که تعداد افرادی که قبول شده اند و تعداد افرادی که قبول نشده‌اند را در یک سطر اکسل نوشته و خود نمرات را نیز برحسب قبولی و یا رد شدن افراد تغییر رنگ دهد.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

برای نیل به این هدف ازیک متغیر به نام ***c*** به عنوان شمارنده استفاده می کنیم. سپس از یک حلقه ***FOR*** و یک شرط ***IF*** استفاده نمایم. لذا برنامه به شکل زیر نوشته می شود:

```
Sub example1()

sum=0

For i = 1 To 20

Then ۱۰ =< Value.(۱ ,If Cells(i

۱ + sum = sum

Font.ColorIndex = 5.(۱ ,Cells(i

Else

Font.ColorIndex = 3.(۱ ,Cells(i

If End

i Next


Value = c.(۱ ,Cells(21

Value = 20 - c.(۱ ,Cells(22


Sub End
```

همانطور که ملاحظه شد، برنامه در ابتدا برای مقدار متغیر ***c*** عدد صفر را در نظر دارد. همانطور که هر سطر جدول با شرط خط چهارم بررسی می شود و چنانچه واجد این شرط بود(اعداد بزرگتر از ۱۰) یک شماره به شمارنده اضافه می گردد و همانطور تا اتمام برنامه این عملیات بروزرسانی شمارنده ادامه پیدا می کند.

انباره (accumulator)

در این مرحله می خواهیم پروسه جمع نمودن اعداد یک سطر یا یک ستون و یا اعدادی که مورد توجه برنامه می باشند را مد نظر قرار دهیم. به عنوان مثال می خواهیم برنامه ای بنویسیم که اعداد فرد ۱ تا ۱۰۰ را جمع نموده و نتیجه را در سلول ***B1*** تایپ نماید.

برای این منظور می بایست متغیری مانند ***Sum*** را در نظر گرفت و حاصلجمع را مرتبا در آن انبار کنیم برای نیل به این منظور می بایست برنامه ای به شرح زیر نوشت:

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
Sub aaa()

Sum = 0

For i = 1 To 100 Step 2

Cells(i, 1 = (i

Sum = Sum + i

Next


Cells(1, 2 = (Sum

End Sub
```

نکته: به عبارت ***sum=sum+i*** توجه کنید. در نوشتن یک شمارنده ما به مقدار قبلی شمارنده یک (۱) اضافه می کردیم. ولی در این برنامه با توجه به شرایط بوجود آمده مقدار دیگری مانند ***i*** همواره به ***sum*** اضافه می شود.

ali_fatehi@yahoo.com

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه دهم- برنامه نویسی اکسل - حلقه های شرطی

ساختار حلقه های شرطی(*Do-Loop*)

حلقه های شرطی حلقه هایی اند که تا زمانی که شرط برقرار باشد (و یا شرطی برقرارنباشد) مورد استفاده قرار می گیرد.

ساختار کلی یک حلقه شرطی در ***VBA*** که تحت عنوان حلقه های ***DO-LOOP*** نیز مورد استفاده قرار می گیرند به شکل زیر است:

روش اول: در این روش شرط در ابتدا بررسي مي شود

DO

دستورات

LOOP WHILEیاUNTIL شرط مورد نظر

روش دوم: در این روش شرط در انتها بررسي مي شود

DO WHILEیاUNTIL شرط مورد نظر

دستورات

LOOP

به این مفهوم که:

انجام بده این کار را تا زمانی که بتونی انجام بدی....

یا

انجام بده این کار را تا زمانی که نتونی انجام بدی....

بنابراین

- وقتي از ***WHILE*** استفاده مي شود مادامیکه شرط برقرار باشد دستورات انجام میشوند

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

- وقتي از ***Until*** استفاده مي شود حلقه به محض وقوع شرط حلقه خاتمه مي يابد(به عبارتي حلقه تا زمانيكه شرط برقرار نيست ادامه پيدا مي كند)

بنابراین در مجموع ۴ نوع حلقه شرطی ***DO-LOOP*** وجود دارد.
با ذکر چند برنامه ساده اینگونه حلقه ها توضیح داده خواهد شد
مثال اول :
در ستون اول از یک شیت اعدادی بصورت پشت سر هم داده هایی وجود دارند .میخواهیم برنامه ای بنویسیم که تعداد داده های موجود در این ستون را پیدا کند. (بین داده ها این ستون سلول خالی وجود ندارد.)

```
Sub do_loop()
 i = 0
  Do
    i = i + 1
  Loop While Cells(i, 1)<>""
Cells(1, 2) = i – 1
End Sub
```

ملاحظه می شود که برنامه اعداد ***i*** را در ستون مذکور تا جایی که به سلول خالی برسید با هم جمع می کند.
در مثال قبلی برای بررسی شرط می توان از دستور ***IsEmpty*** در حلقه شرطی ***DO-LOOP***استفاده کنیم
در آنجا دیدیم که از خط برنامه زیر در برنامه جهت رسیدن به این هدف استفاده نمودیم:

Loop While Cells(i, 1)<>""

به جای این عبارت در صورتی که از حلقه شرطی ***DO-LOOP***استفاده می کنیم می توانیم از عبارت زیر استفاده نماییم:

Loop Until IsEmpty(Cells(i, 1))

لذا برنامه به شکل زیر اصلاح می گردد:

```
Sub do_loop()
i = 0
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
        Do
            i = i + 1
        Loop Until IsEmpty(Cells(i, 1))
    Cells(1, 2) = i - 1
End Sub
```

مثال دوم: می خواهیم برنامه ای بنویسیم مجموع مقادیر ۱ تا ۱۰۰ را بدون آنکه در سلهای جدول این اعداد را نشان دهد، در یک ***MASSAGE BOX***نشان دهد.برنامه به شکل زیر نوشته می شود:

```
Sub do_loop2()
    i = 1
    Sum = 0
        Do
            Sum = Sum + i
            i = i + 1
        Loop While i <= 100
    MsgBox Sum
End Sub
```

مثال سوم:تعدادی عدد (حداکثر ۱۰۰۰ عدد)در سلهای یک ستون اکسل وجود دارد که برخی سلهای همان ستون خالی و فاقد عدد است. برنامه ای بنویسید که آخرین عدد آن ستون را در سطر ***B1***بنویسد.

برنامه مورد اشاره به شکل زیر نوشته خواهد شد:

```
Sub do_loop3()
    i = 1000
        Do
            i = i - 1
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
        Loop While IsEmpty(Cells(i, 1))

    MsgBox i

End Sub
```

علي فاتحي ali_fatehi@yahoo.com

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه یازدهم- برنامه نویسی در اکسل - ایجاد function

تا این مرحله ملاحظه نمودیم چگونه یک برنامه ساده در محیط ***VBA*** ساخته و پرداخته می شود. اکنون زمان آن فرا رسیده است تا فراگیریم چگونه می توان این برنامه های ساده را به عنوان تابعی از توابع موجود در اکسل به کار برد. می توان برنامه های ساده یا پیچیده را که به عنوان تابع در اکسل تعریف نشده اند را نوشته و به عنوان توابع جدید معرفی و استفاده نمود. مثلا می توان برنامه ای نوشت که یک عدد را به صورت حروف بنویسد و سپس این برنامه را در اکسل به عنوان تابعی تعریف نمود.می توان برنامه ای نوشت که تاریخهای میلادی را در برنامه های مورد استفاده به شمسی و بالعکس تبدیل نماید و... تمامی این برنامه ها و یا هر برنامه دیگر را می توان به عنوان توابع در اکسل تعریف نمود.اما چگونه؟

فرمت کلی تعریف یک تابع به صورت زیر است:

FUNCTION نام تابع (, , ,) AS نوع داده ای خروجی تابع

دستورات و محاسبات

محاسبات= نام تابع

دستورات و محاسبات

END FUNCTION

کلمه کلیدی ***Function*** در ابتدا و کلمه کلیدی ***End Function*** در انتها می آید. بخشهای اصلی یک تابع به صورت زیر است:

نام تابع: لازم است برای یک تابع نام تعیین شود. قواعدی که در نامگذاری یک تابع می بایست رعایت شود همان قواعدی است که در تعیین نام یک متغیرلحاظ میگردد.

ورودي هاي تابع و نوع داده اي مربوط به آنها: مشخص شود که ورودي Integer، string و ... است.

خروجي تابع: در واقع نام تابع خروجي تابع نيز هست. بعد از كلمه كليدي AS نوع داده اي مربوط به خروجي تابع مي آيد.

محاسبات تابع: بين عبارات Function و End Function محاسبات تابع نوشته مي شود.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

نکته بسیار مهم: حداقل یکبار مي بایست عملیات انتصاب به نام تابع درمحاسبات آمده باشد. به عبارتي مي بایست با استفاده از عملیات مساوي حداقل یکبار مقداري به نام تابع اختصاص داده شده باشد.

نکته : در صورتي که نوع داده اي ورودي هاي تابع یا خروجي آن اشاره نشود نوع داده اي Variant براي آن درنظرگرفته مي شود.

مثال : مي خواهیم تابعي بنویسیم که درآمد، هزینه و درصد مالیات را دریافت نموده و خالص در آمد را محاسبه نماید:

```
Function netprofit(income, cost, tax) As Variant
t = 1 - tax
netprofit = (income - cost) * t
End Function
```

حال مي توانیم در هر سلول اکسل مانند سایر توابع از این تابع استفاده کنیم. حال براي اجراي این برنامه دو حالت پیش روي ماست.

الف- استفاده از منوي insert

وارد محیط ***excel*** می شویم. از گزینه ***insert*** حالت ***function*** را انتخاب می کنیم.در پنجره باز شده از کمبو باکس مقابل عبارت ***select a category or*** گزینه ***difind user*** را انتخاب می کنیم.ملاحظه می کنیم که تابع جدید ما در اینجا وجود دارد.

دوم- نوشتن فرمول در یکي از سلهاي اکسل: وارد یکي از سلهاي اکسل شویم در اینجا مانند هر برنامه دیگري مي نویسیم:

=NETPROFIT(A1;B1;C1)

بدیهی است ***A1;B1;C1*** در اینجا به ترتیب در بر گیرنده در آمد، هزینه و درصد مالیات می باشد.

مثال:مي خواهیم تابعي بنویسیم که سن افراد را بگیرد و چنانچه زیر ۱۸ سال است بگوید خیلي جوان، بین ۱۸ تا ۶۵ را مناسب و بالاي ۶۵ سال را به عنوان خیلي پیر معرفي نماید.ساختار چنین تابعي به شکل زیر مي باشد:

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
Function check_old(old)
        Select Case old
          Case Is <= 18
              check_old = "too young"
          Case 18 To 65
              check_old = "ok"
          Case Is > 65
              check_old = "too old"
      End Select
End Function
```

مثال دوم:می خواهیم برنامه ای بنویسیم که در ابتدا بر حسب وزن وقد افراد ***BMI*** هر فرد را محاسبه و نشان دهد.سپس برنامه دیگری که با توجه به این عدد نشان دهد که این اندازه ***BMI*** نشان دهنده کدامیک از وضعیتهای کم وزنی، نرمال و یا اضافه وزن می باشد؟

چنین برنامه اي به شکل زیر نوشته مي شود:

```
Function BMI(weight, height)
        BMI = weight / (height ^ 2)
End Function
Function check_weight(BMI)
        Select Case BMI
                Case Is <= 15
                        check_weight = "under weight"
                Case 15 To 25
                        check_weight = "normal"
                Case Is > 25
                        check_weight = "over weight"
        End Select
End Function
```

ADD-INSها

http://www.alifatehi.persianblog.ir
ali_fateh1@yahoo.com

تابعی که ما دراین جلسه معرفی نمودیم تنها در همان ***WORKBOOK*** ای که برنامه در آن نوشته شده است کاربرد دارد. چنانچه ما بخواهیم یک ***FUNCTION*** یا چندین ***Function*** به عنوان توابع قابل استفاده درتمام ***workbook*** های دیگر باشد، می بایست از روش زیر استفاده نمود:

- **نوشتن توابع:** تابع یا توابع مورد نظر را در یک workbook ایجاد مي كنيم.

ایجاد ADD-IN:workbook شامل توابع را با فرمت ADD-INS ذخيره مي كنيم. براي اين منظوردر محيط اكسل از گزينه FILE منوي AS SAVE را انتخاب مي كنيم. در پنجره موجود از كمبو باكس زير صفحه آخرين گزينه را يعنيMICOSOFT EXCELL AD-IN را انتخاب كرده و در هر جايي كه مايليم چنين برنامه هايي SAVE شوند، ذخيره مي نماييم.

افزودن ADD-IN به اكسل: از منوي TOOLS گزينه ADD-INS را انتخاب مي كنيم و در پنجره موجود كنار برنامه اي كه نوشته شده است را تيک مي زنيم.از هم اكنون توابع جديد نوشته شده ما به عنوان توابع ثابتي در كليه workbook ها قابل استفاده است.

ali_fatehi@yahoo.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

جلسه دوازدهم - کار با رشته ها(STRING)-

پایان دوره مقدماتی برنامه نویسی در اکسل

رشته (***STRING***) یکی از انواع داده ای پرکاربرد می باشد. برای کاربا رشته ها لازم است که عملیات و توابعی را که در این زمینه کاربرد دارند بهتر بشناسیم. ترکیب (چسباندن) چند رشته، جدا کردن بخشهایی از رشته و عملیات جستجو از اینگونه عملیات می باشند که در ادامه به معرفی برخی ها از آنها خواهیم پرداخت.

الف- عملگر &

از این عملگر براي ترکیب دو یا چند رشته استفاده مي شود

مثال: برنامه اي خواهیم نوشت که در یک MASSAGE BOX دو عبارت visual basic و for aplcation را به هم چسبانده و نمایش دهد:

```
Sub aaa()

    s1 = "visual basic"

    s2 = " for aplcation"

    s = s1 & s2

    MsgBox a

End sub
```

ب- عبارت VbNewline

این عبارت معرف کد Enter مي باشد . به عبارتي این کد معادل فشرده شدن کلید ENTER مي باشد در انتهاي یک رشته مي باشد. به این معني که رشته دوم در سر بعدي رشته اول نشان داده خواهد شد.

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

مثال:مي خواهيم عبارتVBA this class name is به صورتي كه عبارت VBA در زير عبارت قبلي قرار گيرد، نمايش داده شود.

```
Sub example1()
   s="This class name is " & VbNewline & "VBA"
    MsgBox s
End Sub
```

ج – تابع LEN

اين تابع طول يک رشته را بر مي گرداند(مي دهد).

ساختار كلي تابع LEN به شكل زير مي باشد:

Len(رشته)

كه در داخل پرانتز رشته مورد نظر يا نام آن قرار خواهد گرفت.

مثال: برنامه زير تعداد كاركترهاي(حروف) موجود در رشته alifatehi.persianblog.ir را محاسبه و نمايش مي دهد.

```
Sub example()
      n = Len("alifatehi.persianblog.ir")
      MsgBox n
End Sub
```

د- توابع left و right

اين توابع تعدادي كاراكتر را از سمت چپ يا راست رشته جدا نموده و برمي گرداند. ساختار كلي اين تابع به شكل زير مي باشد:

Left(string,i)يا Right(string , i)

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

که در داخل پرنتز ابتدا رشته مورد نظر و سپس تعداد کاراکتري که مي خواهيم از اين رشته جدا شود مي آيد.

مثال: برنامه اي بنويسيد که چهار رقم سال ۱۳۸۸ در تاريخ ۱۳۸۸/۱۱/۱۵را جدا نموده و نمايش دهد.

```
Sub example2()

        s = "1388/11/15"

        t = Left(s, 4)

      MsgBox t

End Sub
```

۵- توابع TRIM و LTRIM و RTRIM

تابع ***TRIM***:اين تابع فضاهاي خالي (کاراکترهاي خالي) ابتدا و انتهاي يک رشته را پاک مي کند و نتيجه را برمي گرداند

تابع ***TRIM***:اين تابع فضاهاي خالي (کاراکترهاي خالي) سمت چپ يک رشته را پاک مي کند و نتيجه را برمي گرداند

تابع ***RTRIM*** :اين تابع فضاي خالي (کاراکترهاي خالي) سمت راست يک رشته را پاک مي کند و نتيجه را برمي گرداند.

مثال:مي خواهيم در رشته اي مانند" VBA " تنها کلمه VBA نمايش داده شود. ساختار چنين برنامه اي به شکل زير خواهد بود:

```
Sub example3()

        s = "          VBA  "

        t = Trim(s)

    MsgBox t

End Sub
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

و- تابعMID

این تابع به ما می گوید:

رشته را در نظر بگیر، از فلان جا شروع کن، این تعداد کاراکتر را به ما نشان بده. ساختار این تابع شکل می باشد:

MID(string,start,length)

بدین معنا که در رشته string از کاراکتر ردیف start شروع کرده و به تعداد length جدا نموده و برمی گرداند.

مثال: می خواهیم برنامه ای بنویسیم که در عبارت visual basic تنها کلمه basic نمایش داده شود.برنامه به شکل زیر نوشته خواهد شد:

```
Sub example4()
        s = "visual basic"
        t = Mid(s, 8, 5)
    MsgBox t
End Sub
```

ز- تابع INSTR

تابع INSTRبرای جستجوی یک رشته در رشته دیگر بکار میرود. شکل کلی این تابع به صورت زیر است:

INSTR(START, STRING1 , STRING2)

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

بدین معنی که از کاراکتر ردیف START در رشته STRING 1 به دنبال رشته STRING2 می گردد و محل آنرا بر می گرداند.

مثال: مي خواهيم برنامه اي بنويسيم كه دررشته اي مانند alifatehi.persianblog.ir به ما بگويد كه حرف p چندمين كاراكتر اين رشته مي باشد.

```
Sub example4()
        s = "alifatehi.persianblog.ir"
        i = InStr(1, s, "p")
  MsgBox i
End Sub
```

مثال:يک تاريخ مشخص مانند تاريخ ۱۳۸۸/۱۱/۲۵ را كه با مميز"/" از هم جدا شده اند را در نظر مي گيريم. برنامه اي بنويسيد كه سال، ماه و روز را جدا نموده و در سه كادر پيغام نمايش دهد.

```
Sub hhh()
   s = "1359/12/22"
   i = InStr(1, s,"/")
  j = InStr(i + 1, s,"/")
   y = Mid(s, 1, i – 1)
  m = Mid(s, i + 1, (j - 1) – i)
   d = Mid(s, j + 1)
    MsgBox y
     MsgBox m
     MsgBox d
End Sub
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

مثالهای کاربردی

ماکرویی برای حذف سطرهای خالی

در بسیاری مواقع لازم است که سطرهای خالی از یک شیت یا یک ناحیه از یک شیت اکسل حذف شوند. ماکرو زیر اینکار را انجام می دهد. قبل از اجرای این ماکرو لازم است ناحیه مورد نظر را انتخاب کنید. بدیهی است برای اینکه سطرهای خالی یک شیت حذف شوند تمام یک شیت را انتخاب می کنیم.

برای اینکه بتوانید این ماکرو را اجرا کنید مراحل زیر را دنبال کنید

۱- ماکرو زیر را کپی کنید

۲- به محیط اکسل بروید و دگمه ***Alt*** و ***F11*** را بزنید

۳- از منوی ***Insert*** بخش ***Module*** را انتخاب کنید تا یک ماژول ایجاد شود و ماکرو را کپی کرده و به آنجا منتقل کنید(***paste*** کنید)

۴- حال به شیت برگردید. بخشی را که میخواهید سطرهای خالی اش را پاک کنید انتخاب کنید.

۵- دگمه های ***Alt*** و ***F8*** را بزنید تا فهرست ماکروهای موجود را ببینید

۶- ماکروی ***DeleteBlankRows*** را انتخاب و اجرا کنید.

```
Sub DeleteBlankRows()

'Deletes the entire row within the selection if the ENTIRE row contains no data.

'We use Long in case they have over 32,767 rows selected.

Dim i As Long

 'We turn off calculation and screenupdating to speed up the macro.

 With Application

  .Calculation = xlCalculationManual

  .ScreenUpdating = False


'We work backwards because we are deleting rows.
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
For i = Selection.Rows.Count To 1 Step -1
 If WorksheetFunction.CountA(Selection.Rows(i)) = 0 Then
  Selection.Rows(i).EntireRow.Delete
 End If
Next i

 .Calculation = xlCalculationAutomatic
 .ScreenUpdating = True
 End With
End Sub
```

پیدا کردن آخرین سلول یک سطر یا ستون

در اینجا دو تابع براي پيدا كردن آخرين سلول غير خالي در يک سطر يا ستون برايتان آورده ايم.

تابع ***LASTINCOLUMN*** همانطور که از نامش پیداست آخرین سلول در اولین ستون ناحیه مورد نظر را به میدهد.

تابع ***LASTINROW*** همانطور که از نامش پیداست آخرین سلول در اولین سطر ناحیه مورد نظر را به میدهد.

به عنوان مثال فرمول زیر آخرین سلول غیر خالی در ستون ***B*** را میدهد.

=LASTINCOLUMN(B5)

و فرمول زیر هم آخرین سلول سطر ۷ را میدهد:

```
=LASTINROW(C7:D9)

Function LASTINCOLUMN(rngInput As Rang)
  Dim WorkRange As Range
  Dim i As Long, CellCount As Long
  Application.Volatile
  Set WorkRange = rngInput.Columns(1).EntireColumn
  Set WorkRange = Intersect(WorkRange.Parent.UsedRange, WorkRange)
  CellCount = WorkRange.Count
  For i = CellCount To 1 Step -1
    If Not IsEmpty(WorkRange(i)) Then
      LASTINCOLUMN = WorkRange(i).Value
      Exit Function
    End If
```


http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
  Next i
End Function

****************************

Function LASTINROW(rngInput As Range) As Variant
  Dim WorkRange As Range
  Dim i As Long, CellCount As Long
  Application.Volatile
  Set WorkRange = rngInput.Rows(1).EntireRow
  Set WorkRange = Intersect(WorkRange.Parent.UsedRange, WorkRange)
  CellCount = WorkRange.Count
  For i = CellCount To 1 Step -1
    If Not IsEmpty(WorkRange(i)) Then
      LASTINROW = WorkRange(i).Value
      Exit Function
    End If
  Next i
End Function
```

تغییر رنگ زمینه یک سلول یا ناحیه

ویژگی Interior و دستور RGB

برای تغییر رنگ زمینه یک سلول یا ناحیه از ویژگی *Interior* استفاده می کنیم . به عنوان مثال عبارت زیر زمینه سلول ***A1*** را تغییر می دهد:

```
Range("A1").Interior.Color = 8421504
```

این عبارت رنگ زمینه سلول ***A1*** را به رنگ خاکستری تغییر می دهد. کد رنگ زمینه از ۰ تا ۱۶۷۷۷۲۱۵ قابل تنظیم است. راه دیگر برای تعیین رنگ استفاده از دستور *RGB* است. این دستور مخفف سه کلمه ***Red*** ، ***Green*** و ***Blue*** می باشد که در واقع سه رنگ اصلی هستند.این تابع سه ورودی دارد که به سه رنگ قرمز و سبز و آبی اشاره می کنند و مقادیر ۰ تا ۲۵۵ را میگیرند. مثالهای زیر نشان می دهند که چگونه با استفاده از این تابع رنگ سلول ***A1*** را تغییر پیدا می کند:

```
Range("A1").Interior.Color=RGB( 0, 0 ,0) 'black

Range("A1").Interior.Color=RGB( 255, 0 ,0) 'red

Range("A1").Interior.Color=RGB( 0, 0 ,255) 'blue
```

اکسل برای رنگهای استاندارد ثابتهای عددی زیر را در نظر گرفته است:

vbBlack , vbRed, vbGreen , vbYellow , vbMagenta, vb Cyan یا ***vbWhite***

به عنوان مثال عبارت زیر سلول ***A1*** را به رنگ زرد در می آورد:

http://www.alifatehi.persianblog.ir
ali_fatehi@yahoo.com

```
Range("A1").Interior.Color= vbYellow
```

نسخه های قبلی اکسل تنها ۵۶ رنگ متفاوت را پشتیبانی می کردند ولی اکسل ۲۰۰۷ بیش از ۱۶ میلیون رنگ را پشتیبانی می کند و ویژگی ***Themes*** را هم در خود جای داده است. که البته در بسیاری موارد استفاده از رنگهای مختلف هم سرگرم کننده است و هم گیج کننده.

روش میانبر

روش میانبر استفاده از ماکرو برای تغییر رنگ زمینه یا الگوی یک سلول، تغییر فونت و اندازه فونت و ... است. بدین ترتیب که با استفاده از ماکرو رکوردر کلیه تغییرات مربوط به رنگ و فونت و زمینه و الگوی یک سلول یا ناحیه را ضبط می کنیم. بعد این کدها را با کمی تغییر و ویرایش در داخل برنامه خودمان بکار میگیریم.

ali_fatehi@yahoo.com