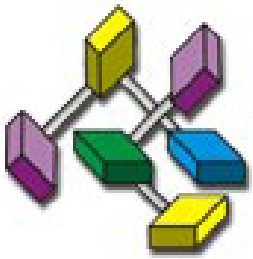
Microsoft
MSOffice 2000



آموزش
Microsoft Visual Basic for Applications

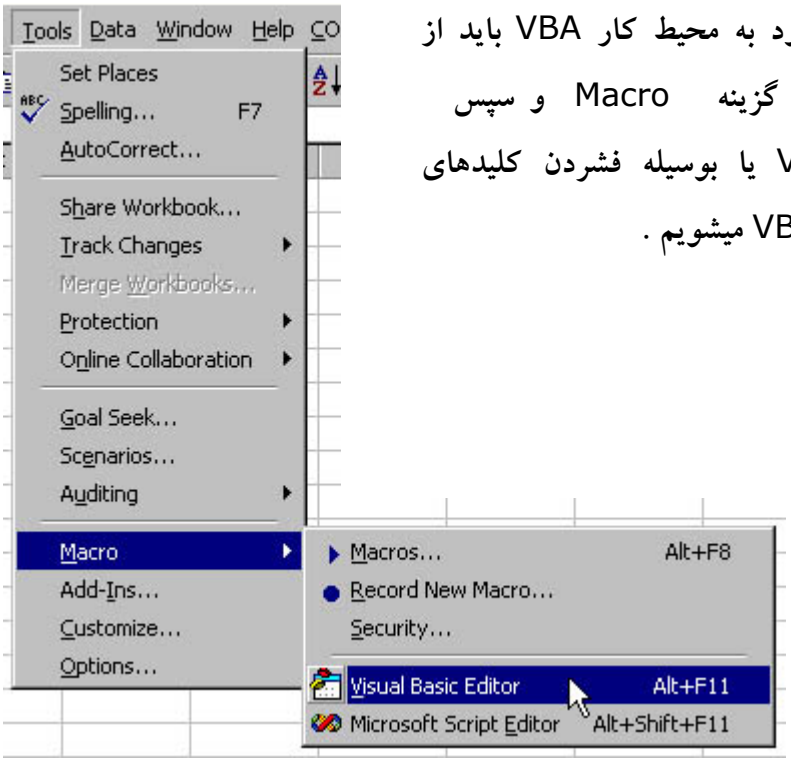
شامل :

- ✓ آموزش زبان برنامه نویسی ویژوال بیسیک
- ✓ کاربرد ویژوال بیسیک در نرم افزارهای Office
- ✓ حل تمرینهای مختلف و کاربردی VBA
- ✓ توابع و فرمولهای مهم VBA در MS Office
- ✓ آموزش زبان برنامه نویسی SQL
- ✓ کاربرد SQL در MS Access

مؤلف : علی رضا زمّرد

مقدمه :

مجموعه نرم افزارهای Office کاملترین و کاربردی ترین بسته نرم افزاری برای انجام امور مختلف اداری است . این Package بوسیله زبان Visual Basic و SQL پشتیبانی می شود ورفع نقاط ضعف و تکنیکهای خاص آن ازطریق این زبانهای برنامه نویسی ، انجام می پذیرد . در این آموزش کوتاه شما با محیط برنامه نویسی VBA (Visual Basic For Application) آشنا می شوید و اصول اولیه و تکنیکهای کاربردی این زبان برنامه نویسی را فرا می گیرید .



قبل از هر چیز برای ورود به محیط کار VBA باید از طریق منوی Tools گزینه Macro و سپس Visual Basic Editor یا بوسیله فشردن کلیدهای Alt+F11 وارد محیط VBA میشویم .

آشنایی با محیط برنامه نویسی VBA :

محیط برنامه VBA شامل قسمتهای مختلفی است که بجز نرم افزار Access بقیه نرم افزارهای Office 2000 محیطی شبیه به شکل صفحه بعد دارند . در زیر با قسمتهای مختلف محیط این برنامه آشنا می شوید :

Project Explorer : پنجره ای است که بصورت نمودار درختی Objectها ، Codeها ، Formها و دیگر ابزار بکار رفته در فایل مورد نظر را نشان می دهد .

Properties Window : پنجره ای است که بصورت جدول مشخصات و اطلاعات جزئی تری در مورد اشیاء و Object ها و Active X های بکار رفته دربرنامه را نشان می دهد .

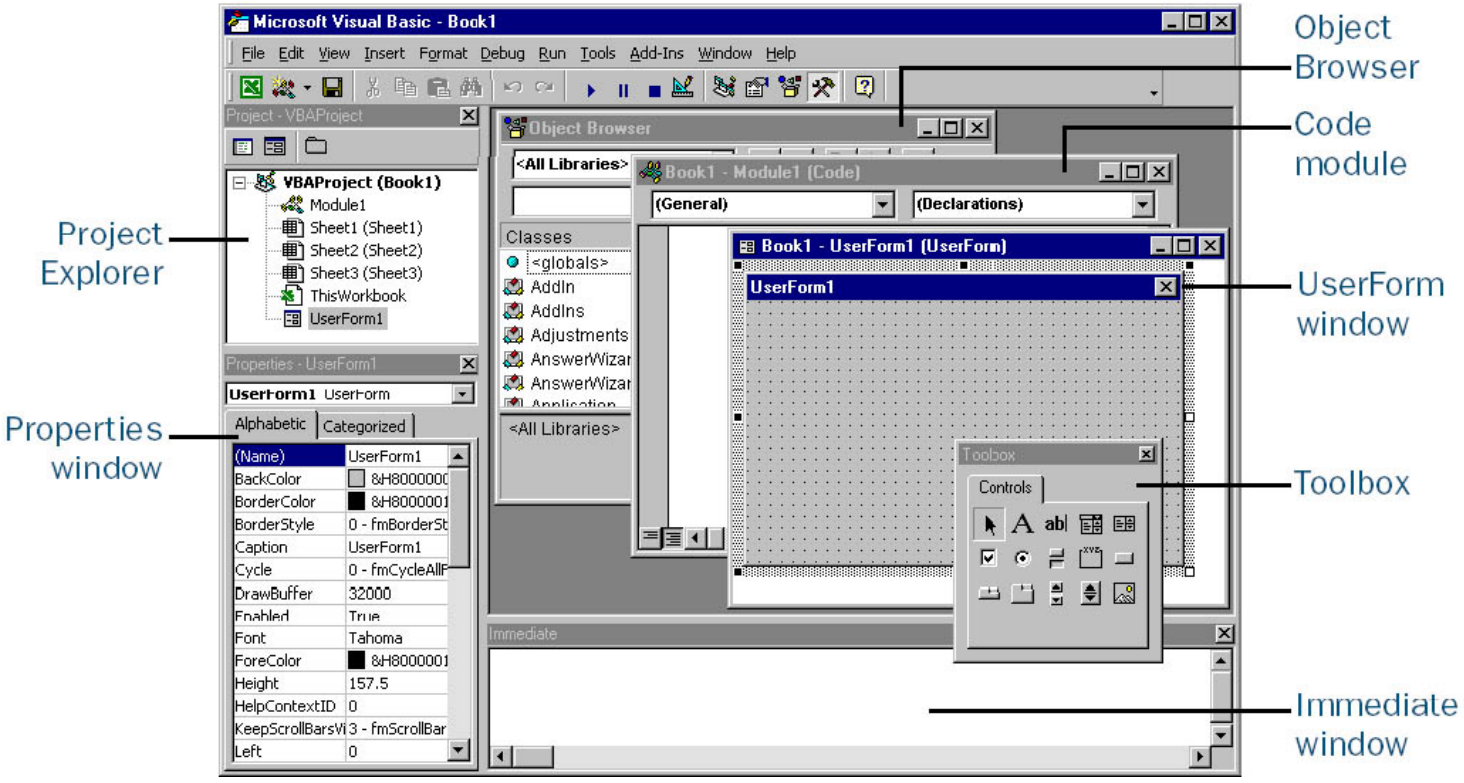
Object Browser : پنجره ای است که لیست کامل فرمانها (Commands) ، وقایع (Event) ، مشخصات (Properties) ، توابع (Active X) را با امکان جستجو و راهنمای استفاده و راهنمای نگارش در اختیار کاربران قرار می دهد .

Code Module : صفحه سفید رنگی است که محل کد نویسی و نوشتن برنامه ، غلط گیری (Debug) و نگارش متن برنامه می باشد .

UserForm Window : در صورتی که در برنامه ما Form و یا هرگونه Object وجود داشته باشد ، محیط طراحی آن را به ما نشان می دهد که در آن ما قادر به اعمال هرگونه تغییر در UserForm ها را داریم .

Toolbox : مجموعه کنترلها (Controls) و ابزارهای طراحی UserFormها است که برای نمایش آن و یا مخفی کردن آن Toolbox را از منوی View فرا می خوانیم .

Immediate Window : پنجره ای است که در آن شما می توانید یک خط از برنامه را اجرا کرده و نتیجه آن را ببینید . (کاربرد آن در زمان غلط گیری یا Debug گیری است .)



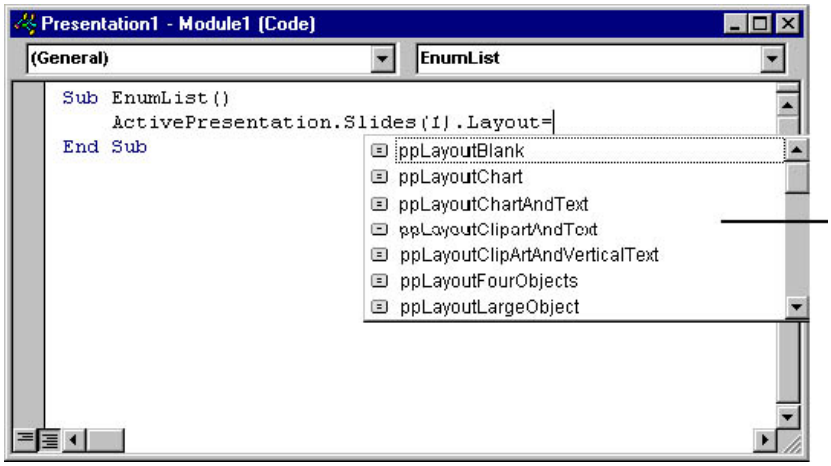
همانطور که یک keyboard از اجزاء کوچکتری به نام کلید تشکیل شده صفحه فرم ما از تعدادی Object یا شیء تشکیل شده است . زمانی که ما یک یک کلید را می فشاریم و حرفی نوشته می شود ، نوشته شدن حرف یک رخداد یا واقعه (Event) است . حال ممکن است وقایع مختلفی از قبیل حذف شدن (Delete) ، کپی شدن (Copy) و یا هر عمل دیگری رخ دهد . اینها وقایع یا Event هستند .

نوع اتفاقی که باعث بوجود آمدن Eventها یا وقایع میشود Method است مانند Click ، Mouse Over ، Mouse Down ، Key Press ، Dbl Click ، . . . بدین معنا که اگر روی Object کلیک کردیم و یا با ماوس (Mouse) روی آن حرکت کرده و یا حتی دکمه ای از کیبورد را فشردیم عمل یا اتفاق صورت پذیرد .

هر Object دارای مشخصاتی است که می توان از طریق کد نویسی آنها را کنترل کرده و تغییراتی را بر روی آنها انجام داد . مانند Font ، Font Size ، Height ، Border Style ، Caption ، Fill Color ، Fore Color ، . . . . برای اعمال کنترل های دلخواه بر روی این مشخصات (Properties) در برنامه نویسی پس از نام Object ، علامت " . " گذاشته و به آن مقدار می دهیم . مثلاً برای تغییر عنوان یک دکمه کد زیر را می نویسیم :

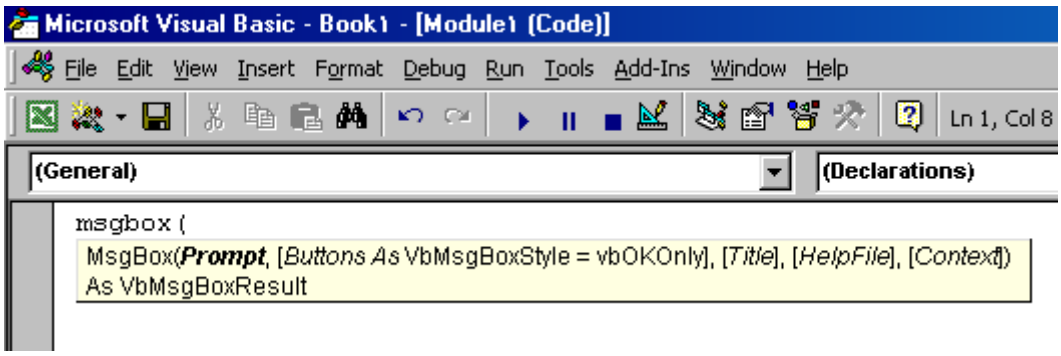
CommandButton.Caption = " Enter New Number!"

اگر دقت کنید می بینید زمانی که شما علامت " . "و یا " = " را تایپ می کنید پنجره ای بصورت Listbox در کنار عنوان شما پدیدار می شود که حاوی Properties و Method شیء یا Object مورد نظر شما است و با نگارش هر حرف از سوی شما بطور خودکار و مرتب شده صعودی لیست مشخصات و متد مورد نظر را جستجو می کند . کافی است روی عنوان دلخواه خود دکمه Space را بفشارید تا کد آن بطور خودکار نوشته شود .



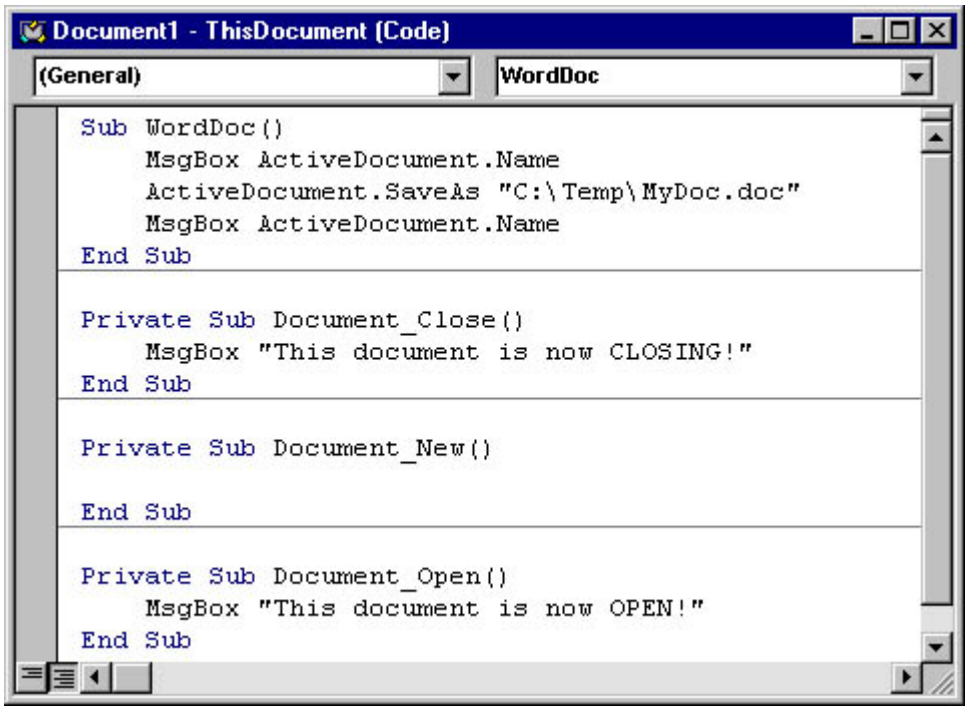
لیست Properties و Methods هر Object

اگر بخواهید تابعی را اجرا کنید و یا فرمانی بدهید که ورودی های آن چند آرگومان می باشد VBA به شما راهنمایی کرده و نوع آرگومانهایی را که باید وارد کنید به شما نشان می دهد .

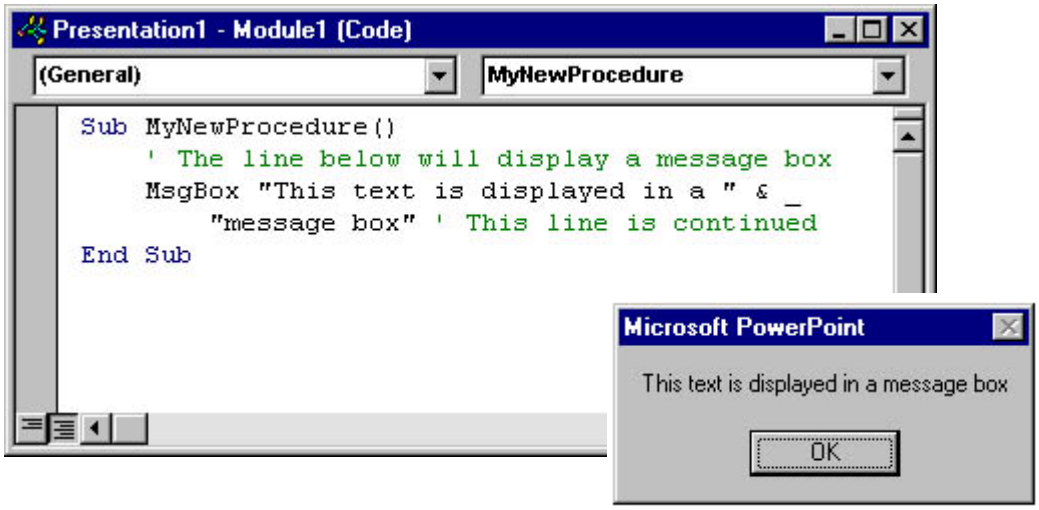


انجام چند تمرین ساده :

حال با کمی دقت برنامه های زیر را در محیط VBA نرم افزار MSWord بنویسید و نتیجه آن را ببینید :



کد زیر را وارد کنید و نتیجه آن را که نمایش پنجره پیغام است ببینید :



در محیط برنامه نویسی VBA اگر بخواهیم بدون اینکه در اجرای برنامه تأثیری گذاشته شود ، توضیحاتی در مورد هر خط از برنامه بدهیم باید در ابتدای آن علامت " ' " بگذاریم که در نتیجه آن سبزرنگ شدن متن بعد از علامت است که نشان می دهد این قسمت ، توضیحات است و کد آن قابل اجرا نیست .

کلیدهای میانبر (Shortcut Key) مهم در VBA :

توصیف آن	کلید
مرورگر Object	F2
نمایش Properties	F4
اجرای برنامه یا Compile	F5
شروع با Compile کامل	Ctrl+F5
اجرای برنامه خط به خط (For Debug)	F8
اجرا از روی	Shift+F8
تنظیم نقطه انفصال	F9
نگاه سریع (Quick Preview)	Shift+F9
نمایش Immediate Widow	Ctrl+G

برنامه نویسی Functions ، Procedures ،Modules

۱. Module

Module دارای یک صفحه اولیه به نام Declarations ویک یا چند تابع یا زیر برنامه است. از صفحه اولیه به برای تعریف متغیرها، ثابتها، و آنچه مورد نیاز زیر برنامه ها یا توابع آن Module است، استفاده می شود. از یک Module ممکن است فقط برای تعریف متغیرهای Global و ثابتهایی که در تمام توابع و زیر برنامه های بانک اطلاعاتی به کار می روند، استفاده می شود.

Procedures

وقتی یک بخش از برنامه توسط برنامه دیگر خوانده می شود به آن زیر برنامه گفته می شود. زیر برنامه خود می تواند یک زیر برنامه دیگر را اجرا کند. وقتی یک زیر برنامه در یک برنامه دیگر خوانده می شود، فرمانهای آن زیر برنامه یکی یکی اجرا شده و در پایان ادامه کار به برنامه اصلی بر می گردد. هر زیر برنامه با کلمات شناخته شده Sub (در ابتدا) و End Sub (در انتها) مشخص می شود. فرمت صفحه بعد یک برنامه را در VBA نشان می دهد:

نام زیر برنامه
Sub
شروع فرمانهای زیر برنامه
. . . . .
. . . . .
. . . . .
پایان فرمانهای زیر برنامه
End Sub

برای اشاره به یک زیر برنامه می توانید فقط از نام آن استفاده کنید. در VBAمی توانید با استفاده از فرمان Call و بعد از آن نام زیر برنامه، صریحاً به یک زیر برنامه اشاره کنید. با این روش نام اشاره شده در دنباله فرمان Call برابر با نام زیر برنامه تلقی می شود تا یک متغیر حافظه ای.

Functions

توابع یک نوع زیر برنامه هستند که با اشاره به نام آنها یک داده یا نتیجه ارائه می کنند.

توابـع عمدتـاً در Module سـاخته می شوند. برای ایجاد یک تابع در صفحه اولیه Module کلمـه Function سـپس یـک نـام دلخواه را تایپ و کلید Enter را بزنید. با این عمل یک صفحه جدید باز می شود که شبیه فرمت زیر است.

Function نام تابع (**[پارامترهای تابع]**)

شروع فرمانهای تابع

. . . .

. . . .

. . . .

پایان فرمانهای تابع

End Function

توابـع ممکن است دارای پارامتر نباشند. از توابع در Event فرم یا کنترلهای آن و همچنین در گـزارش اسـتفاده می شود. برای این منظور بایستی در مقابل Event علامت مساوی سپس نام تـابع را بنویسـید. اگر تابع پارامتر ندارد، بعد از نام تابع فقط علامت پرانتز باز و بسته، در غیر این صورت بایستی پارامترهای آنرا به ترتیب داخل پرانتز وارد کرد .

عـلاوه بـر ایـن در VBAانـواع مختلف عناصر بانک اطلاعاتی نرم افزار Accessهم به عنوان متغـیر حافظـه ای شناخته می شوند. جدول زیر این متغیرها و معادل آنها در بانک اطلاعاتی را نشان می دهد:

متغیر حافظه ای	عنصر معادل در بانک اطلاعاتی
Database	بانک اطلاعاتی در اکسس
Form	فرم و زیر فرم
Report	گزارش و زیر گزارش
Control	کنترلهای روی فرم و گزارش
QueryDef	طراحی سوال
Recordset	دسترسی به رکوردهای جدول یا نتیجه سوال در اکسس بیسیک
Table	جدولهای بانک اطلاعاتی
Dynaset	نتیجه سوال که قابل تغییر است
Snapshot	نتیجه سوال که قابل تغییر نیست

متغیرهای حافظه ای در VBA

متغـیرهای حافظه ای در VBA مانند فیلدهای جدول بانک اطلاعاتی متغیرها دارای نام هستند، بـا این تفاوت که در اینجا نام گذاری متغیرها بدون فاصله و علائم ویژه است. تنها می توان از خـط زیـر (_) اسـتفاده کرد. ضمناً ازکلمات شناخته شده دراکسس هم نمی توان به عنوان نام متغیر حافظه ای استفاده کرد.

در VBAمی توان یک متغیر را به سادگی و تعیین مقدار همزمان با تعریف آن ایجاد کرد:

```
VarName=1234
```

در این مواقع نوع متغیر بستگی به داده ای دارد که به آن تخصیص یافته است (Variant) اگر بخواهید آنرا کاراکتری تعریف کنید عبارت بالا بصورت زیر نوشته می شود:

```
VarName%=1234
```

گـرچه روش بـالا سـاده است لیکن توصیه می شود قبل از دادن مقدار به متغیر، آنرا با Dim تعریف و نوع آنرا مشخص کنید. سپس داده مناسب با نوع تعیین شده به آن بدهید. برای تمرین متغیر بالا با روش جدید، بصورت زیر تعریف می شود:

```
Dim VarName As Integer
VarName=1234
```

ایـن روش تعـریف متغیرهای حافظه ای را Explicit می گویند. برای اینکه در یک Module فقـط با این روش متغیر تعریف شود، بایستی در صفحه اولیه Module ازOption Explicit اسـتفاده کنـید. بـه این ترتیب چنانچه نام یک متغیر را اشتباهاً چیز دیگری وارد کنید سیستم با پـیغام خطـا، مـانع تعریف آن متغیر می شود. در غیر این صورت ممکن است نام یک نتغیر را اشـتباه تایپ کرده و برنامه را اجرا کنید. در این صورت سیستم نام غلط متغیر را به عنوان یک متغیر جدید برداشت کرده و رفع اشکال از برنامه را بسیار مشکل می کند.

متغیرهای حافظه ای در VBAدر ٤ سطح به شرح زیر استفاده می شوند:

متغیرهای محلی :

ایـن متغیرها در سطح زیر برنامه معتبر هستند و در سایر زیر برنامه ها یا توابع از آنها نمی توان اسـتفاده کرد. متغیر هایی که با Dim یا بدون آن تعریف می شوند، فقط همانجایی که تعریف شـده اند، شناخته شده اند و در دیگر زیر برنامه ها یا توابع شناخته شده نیستند. تا زمانیکه زیر بـرنامه یا تابع مربوطه فعال است، متغیر های آن نیز فعال است و با بسته شدن آن زیر برنامه یا تابع، متغیرهای مربوطه هم غیر فعال شده و از حافظه خارج می شوند.

متغیرهای Form و Report در Access

اگـر در صـفحه اولـیه Module یـک فرم یا گزارش متغیری را تعریف کنید آن متغیر فقط در همـان فرم یا گزارش شناخته شده است و با بستن آن فرم یا گزارش متغیرهای آن هم غیر فعال می شوند.

متغیرهای Module

اگر در صفحه اولیه یک Module مستقل، متغیری را تعریف کنید، آن متغیر در تمام زیر برنامه ها و توابع همان Module شناخته شده است و با بستن آن Module متغیرهای آن هم غیر فعال می شوند. برای تعریف این متغیرها از روش Dim در صفحه اولیه Module استفاده کنید.

متغیرهای عمومی یا Global

این متغیرها در تمام Moduleها، فرمها، توابع و زیر برنامه های یک بانک اطلاعاتی قابل استفاده هستند. برای تعریف این نوع متغیرها از روش Global در صفحه اولیه یک Module استفاده می شود. برای مثال، glVar بصورت زیر به عنوان متغیر عمومی تعریف می شود:

Global glVar As Integer : برای محفوظ نگه داشتن مقادیر متغیرهای محلی بین زیر برنامه ها یا توابع از Static به جای Dim استفاده می شود. متغیرهایی که با Static تعریف می شوند، در تمام طول برنامه شناخته شده هستند لیکن قابل استفاده بودن آنها بستگی به جایی دارد که تعریف شده اند. از این متغیرها برای شمارش تعداد رخداد یک حالت یا Event استفاده می شود. برای اینکه تمام متغیرهای یک تابع یا زیر برنامه از نوع Static شود، بایستی از کلمه Static قبل از Function یا Sub استفاده کنید.

متغیرهای حافظه ای با نوع دلخواه

یک متغیر می تواند یک یا چند نوع داده اکسس را در خود نگهداری کند. این نوع متغیرها با نوع دلخواه توسط برنامه نویس ساخته می شوند. برای ایجاد نوع متغیرها بین EndType... Type و با فرمت زیر استفاده می شود:

Type DupRec

DupRec یک نام دلخواه است

```
Field1 As Long
Field2 As String * 20
Field3 As Single
Field4 As Double
```

End Type

از ایـن نـوع داده زمانی استفاده می شود که بخواهید داده های یک یا چند رکورد با فیلدهای مـتفاوت را نگهـداری کنید. متغیرهای کاراکتری در این نوع داده ها با طول ثابت هستند. برای مـثال طـول Field2 در تمریـن قـبل ۲۰ کاراکـتر (مشـابه طول فیلد مورد اشاره در جدول) مـی باشد. برای تعریف این نوع متغیرها بایستی از همان کلمات Global,Dim یا Static به صورت زیر استفاده شود:

```
Dim Currentrec As DupRec
```

برای تخصیص داده به یکی از متغیرهای نوع بالا از نام متغیر و نقطه استفاده می شود:

```
Currentrec.Field1=2048
```

آرایه ها در VBA

برای نگهداری یک سری داده در یک متغیر از متغیرهای نوع آرایه استفاده می شود. هر داده از ایـن سری را یک جز یا عنصر آرایه می گویند. برای تعریف آرایه در VBAباید ار روش Dim استفاده کرده و پس از نام متغیر، داخل پرانتز تعداد عناصر آنرا قید کرد.

```
Dim NewArray(20) As String
```

درعـبارت بـالا یک متغیر آرایه با ۲۱ عنصر ساخته شده است. گرچه عدد ۲۰ در بالا قید شده لیکن اولین عنصر با عدد صفر شروع می شود. به همین دلیل تعداد عناصر آرایه، یکی بیشتر از آنچـه در پرانـتز قید شده در نظر گرفته می شود. برای تعداد عناصر ثابت و دقیق به روش زیر عمل می شود:

```
Dim NewArray(1 To 20) As String
```

در ایـنجا تعـداد عناصر آرایه ۲۰ خواهد بود. اگر عددهای تعداد عناصر را با علامت کاما جدا کنید، آرایه چند بعدی ایجاد می شود:

```
Dim NewArray (9,9,9) As Long
```

در ایـن عـبارت یک آرایه سه بعدی با ۱۰ عنصر در هر بعد ایجاد می شود. ار آنجاییکه نوع آرایـه Long اسـت ، مـیزان فضـایی کـه در ایـن آرایـه از حافظـه اشـغال می کند برابر با (۱۰×۱۰×۱۰×٤) چهار هزار بایت خواهد بود.

می توان آرایه را با تعداد عناصر شناور یا دینامیک تعریف کرد. برای این منظور از پرانتز خالی استفاده می شود. در تمرین زیر یک متغیر دینامیک ایجاد شده سپس با ReDim تعداد عناصر آن را می توان تشخیص داد:

```
Dim NewArray() As Long
```

در صفحه اولیه Module

```
ReDim Preserve NewArray(9,9,9)
```

در زیر برنامه، برای نگهداری مقادیر قبلی

```
ReDim NewArray(9,9,9)
```

در زیر برنامه، مقداردهی مجدد به تمام عناصر

برای اینکه همیشه بخش اعظمی از حافظه درگیر ضبط متغیر آرایه نباشد، می توانید با استفاده از ReDim مقادیر عناصر آنرا به صفر تبدیل کنید. با این عمل وقتی نیازی به مقادیر آرایه ندارید از حافظه تخلیه می شود. آرایه هایی که با روش Dim ایجاد می شوند، حداکثر می توانند تا ۸ بعد باشند. آرایه را می توان با ReDim بدون استفاده قبلی از روش Dim در زیر برنامه ایجاد کرد. در این صورت نوع آرایه محلی بوده و حداکثر می تواند تا ٦٠ بعد باشد.

نام عناصر بانک اطلاعاتی به عنوان متغیر حافظه ای

عناصر بانک اطلاعاتی که در اکسس ایجاد می کنید می توانند به عنوان متغیر حافظه شناخت و به آنها مقدار داد. برای مثال اگر داده یکی از کنترلهای روی یک فرم را بخواهید عوض کنید از روش زیر استفاده می شود:

```
Forms!Customers!Address=”123 Kaj Street“
```

در این روش ابتدا نام عنصر بانک اطلاعاتی (در اینجا Forms به عنوان فرم)، سپس نام فرم (در اینجا Customers) و پس ار آن نام فیلد یا کنترل روی فرم اشاره می شود. در صورتیکه نام کنترل یا فرم دارای فاصله خالی باشد، بایستی در علامت [] نوشته شود:

```
Dim txtContact As Control
Set txtContact=Forms!Customers![Contact Name]
txtContact =”Alizadeh“
```

ثابتها در VBA

ثابت پارامتری است که مقدار آن تغییر نمی کند. در VBA می توان این متغیرها را با کلمه Const در صفحه اولیه Module یا زیر برنامه و تابع تعریف کرد . برای مثال ، Const sigPi=3.1416 یک روش تعریف ثابت در VBAاست. برای اینکه یک ثابت در تمام Module های بانک اطلاعاتی استفاده شود، بایستی ازGlobal برای تعریف آن استفاده کرد:

```
Global Const sgnPi=3.1416
```

ثابتهای نوع Global فقط در صفحه اولیه Module قابل تعریف است.

اینها ثابت هایی هستند که توسط برنامه نویس ایجاد می شود. علاوه بر اینها در اکسس ۷ ثابت از پیش تعریف شده است که عبارتند از:

```
Null,True,False,Yes,No,On,Off
```

در بین این ثابتها فقط از Null,False,True می توان در VBAاستفاده کرد. بقیه در سایر عناصر بانک اطلاعاتی غیر از Module قابل استفاده هستند. یکی از کاربردهای این نوع ثابتها در عبارت شرط در Query می باشد.

علاوه بر ثابتهای بالا یک سری ثابتهای مخفی در VBA وجود دارد که برای برخی عملیات با _A و برای بانکهای اطلاعاتی با _DB شروع می شوند. برای آشنایی با آنها Constants را در راهنمای VBA (Help) جستجو و مطالعه کنید.

عبارتهای شرطی و تصمیم گیری در VBA

وقتی تصمیم گیری بین دو حالت مثبت و منفی مطرح است اغلب از

If...Else...Endlf استفاده میشود. فرمت این روش در VBAبه شرح زیر است:

If Mod(a,2)=1 **Then**

اگر متغیر a بر عدد ۲ تقسیم شود و باقیمانده داشته باشد.

MsgBox ”a is odd number“

ارسال یک پیغام مبنی بر فرد بودن متغیرa

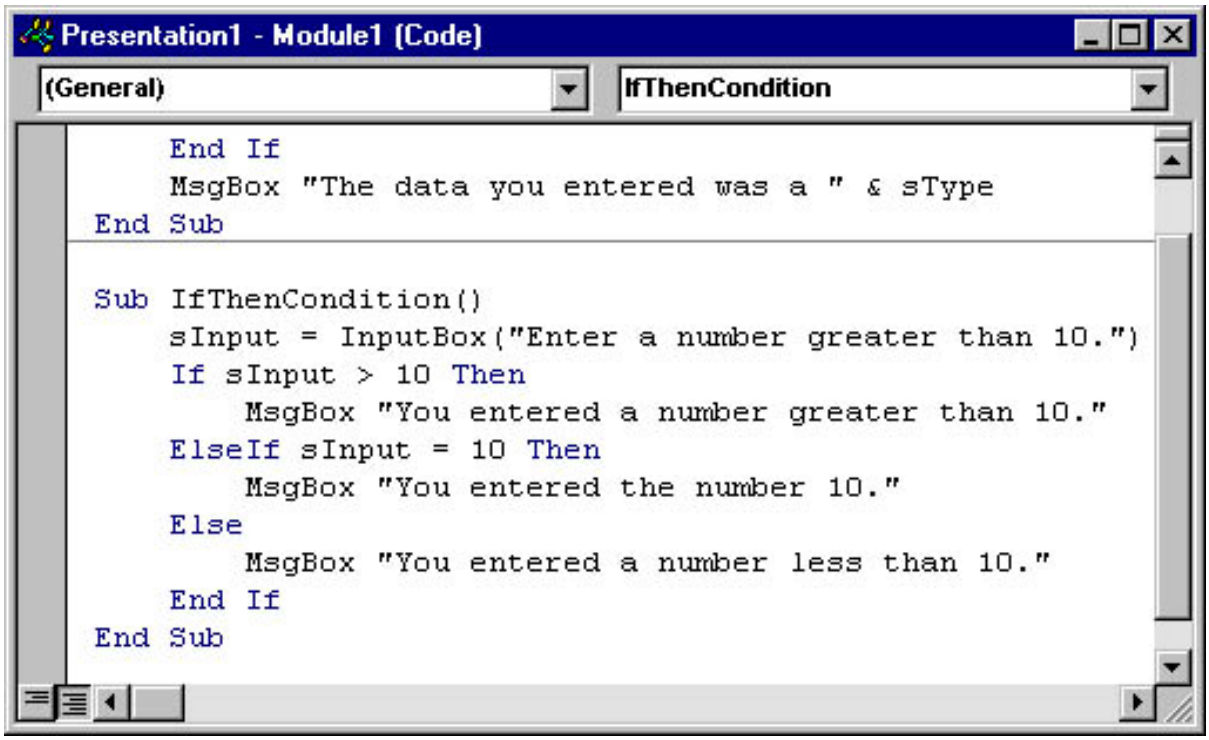
Else

در غیر این صورت

MsgBox ”a is even number“

ارسال یک پیغام مبنی بر زوج بودن متغیرa

End If



چنانچه تصمیم گیری بین چندین حالت متفاوت باشد و امکان استفاده از If نباشد از Select Case بصورت زیر استفاده می شود:

```
Sub IfThenCondition()
   sInput = InputBox("Enter a number greater than 10.")
   Select Case sInput
   Case Is > 10
      MsgBox "You entered a number greater than 10."
   Case Is = 10
      MsgBox "You entered the number 10."
   Case Is < 10
      MsgBox "You entered a number less than 10."
   End Select
End Sub
```

تمرین:

فـرض کنـیم بخواهید میزان فروش یا یک عدد مشابه آن را کلاس بندی کنید. ابتدا یک تابع با یک پارامتر و استفاده از روش Case به شرح زیر است:

```
Function Class(A As Double)
Select Case A
Case Is <10000
Class=0
Case 10000 To 49999.99
Class=1
Case 50000 To 100000
Class=2
Case Else
Class=3
End Select
```

بـرای استفاده از این تابع یک سوال از جدول فروش ایجاد و فیلدهای مورد نظر از جمله فیلد فـروش را در ایـن سـوال وارد کـرده و در قسـمت نـام فـیلد یـک سـتون جدیـد عـبارت Class([sales]) را وارد کنید.در اینجا[sales] اشاره به نام فیلد است که مقادیر فروش در آن وارد شده است.

ایجاد حلقه در VBA

در برخـی مـوارد لازم است یک سری عملیات یا فرمانها به تعداد معین یا بر اساس یک شرط خـاص تکـرار شـوند. به محض رسیدن به شرط تعیین شده یا انجام به تعداد مورد نظر تکرار عملیات یا فرمانها متوقف می شود. این عمل در برنامه نویسی به ایجاد حلقه یا Loop معروف است. برای ایجاد حلقه روشهای متفاوتی وجود دارد که در اینجا به آنها اشاره می شود.

For...Next

ازاین روش برای حلقه های تعدادی استفاده می شود فرمت کلی آن بصورت زیر است:

For<متغیر شمارنده=<عدد شروع> **To**<عدد پایان>[Step<مقدارفاصله>]

فرمانهایی که باید در این حلقه تکرار شوند

[شرط پایان حلقه، قبل از اینکه به تعداد بالا برسد]

Exit For

Next <متغیر شمارنده>

در فرمـت بـالا، جاییکه از علامت <> استفاده شده یعنی این بخش از فرمت اجباری است و آنهایـیکه در علامـت [] وارد شـده یعنـی اختـیاری است. برای مثال اگر بخواهید یک فرمول ریاضـی با ده مقدار X به صورت سری محاسبه شود و در یک آرایه ضبط شود به صورت زیر عمل می شود:

```
Dim a(1 To 10) As Double
Function fun()
For X=1 To 10 Step 1
A(X)= 2*X^2+3*X-3
Next X
```

Do While...Loop

از ایـن روش برای ایجاد حلقه شرطی استفاده می شود. فرمانها یا عملیات مورد نظر تا زمانیکه شرط برقرار است، تکرار می شوند. به محض عدم برقراری شرط یا برقراری شرط داخلی (اگر شـرط داخلی منظور شده باشد) حلقه متوقف می شود و فرمانهای بعد از حلقه اجرا می شوند. فرمت کلی به صورت زیر است:

Do While شرط[=True]

فرمانهایی که باید در این حلقه تکرار شوند

Exit Do
Loop

در اینجا VBAبا فرض مثبت بودن حالت شرط (True) کار می کند که نیازی به نوشتن آن نیست. برای حالت های منفی نباید از False استفاده کرد. برای قید حالت منفی در شرط حلقه از Not استفاده می شود.

Do Until...Loop

یک روش دیگر برای ایجاد حلقه شرطی استفاده از روش بالا است. فرمانها یا عملیات مورد نظر تا زمانیکه شرط برقرار نیست، تکرار می شوند. به محض برقراری شرط یا برقراری شرط داخلی (اگر شرط داخلی منظور شده باشد) حلقه متوقف می شود و فرمانهای بعد از حلقه اجرا می شوند. فرمت کلی بصورت زیر است:

Do Until شرط <> **True**

فرمانهاییکه باید در این حلقه تکرار شوند

[شرط پایان حلقه، یا یک شرط دیگر]

Exit Do
Loop

دسترسی به عناصر بانک اطلاعاتی در اکسس بیسیک

عناصر دسترسی به داده ها یاDAO (Data Access Objects) در اکسس عناصری هستند که با Jet Engine ایجاد شده و نگهداری می شوند. بطور خلاصه برای دسترسی به لیست داده ها در اکسس سه مرحله لازم است:

دسترسی به محیط کار یا Workspace

دسترسی به بانک اطلاعاتی یا Database

دسترسی به لیست داده ها یا Recordset

در اینجا انواع عناصر مربوط به لیست داده ها به شرح زیر توضیح داده می شود:

TableDef: اشاره به جدول ضبط شده در یک بانک اطلاعاتی است.

QueryDef: اشاره به سوال ضبط شده در یک بانک اطلاعاتی است.

Parameter: اشاره به پارامتر یک سوال است.

Field: اشاره به فیلد در یک جدول، ایندکس، سوال لیست رکورد است.

Index: اشاره به ایندکس در یک جدول است.

Relation: اشاره به ارتباط بین فیلدهای جدول یا سوال است.

Property: اشاره به مشخصات یک عنصر است.

Recordset: اشاره به لیست رکوردها در یک جدول یا سوال است.

Group: اشاره به یک گروه کاربر که در حفاظت اکسس تعریف شده است.

User: اشاره به یک کاربر که در حفاظت اکسس تعریف شده است.

متغیر حافظه ای برای عناصر

در VBA نیز مانند سایر محیط های برنامه نویسی امکان تعریف انواع متغیرها وجود دارد. برای تعـریف هـر متغیر ابتدا با استفاده از دستورالعمل زیر یک متغیر، متناسب با نوع آن فضای لازم را رزرو می کند.

Dim <نام متغیر> **As** <نوع>

انواع متغیرها در VBA عبارتند از عددی کوتاه (Integer)، عددی بلند (Long)، عددی شناور(Double)، کاراکتری(String)، تاریخی(Date).

مرحله بعد تخصیص مقدار به متغیر است. برای این منظور از روش زیر استفاده می شود:

نام متغیر = داده

عـلاوه بـر ایـنها یک سری متغیر بسیار مهم و پر مصرف هم برای ضبط عناصر بانک اطلاعاتی وجـود دارد. ایـن نـوع متغیرها هم برای تعریف دو مرحله دارند: تعریف و تخصیص عنصر به آنها. فرمانهای زیر هر دو مرحله تعریف و تخصیص عناصر را نشان می دهد.

```
Dim db As Databases
Dim rst As Recordset
Dim frm As From
Dim ctl As Control
```

مـرحله دوم تخصیص عنصر مناسب به هر یک از متغیرهای بالا است. در اینجا از عبارت set و علامت مساوی استفاده می شود:

```
Set db=DBEngine.Workspaces(0).Databases(0)
Set rst=db.Openrecordset ("نام جدول یا سوال")
Set frm=Form! <نام فرم>
Set ctl= frm(0)
```

کار با لیست داده ها

در بیشتر مواقع در برنامه نویسی(VBA) لازم است با داده های جدول یا سوالها کار کنید ، VBA روشهای خاصی برای دسترسی به داده ها و مشاهده، ویرایش، حذف و اضافه کردن آنها ارائه می کند. دسترسی به داده ها در برنامه نویسی غیر از فعال کردن ظاهری آنها با اجرای جدول یا سوال است. در اینجا لیست داده ها را Recordset می گویند و به ظاهر هیچ چیزی مشاهده نمی شود بلکه از طریق برنامه لیست داده ها در دسترس قرار می گیرد تا برنامه نویس هر عملی که نیاز باشد روی آنها انجام دهد. در اکسس بیسیک سه نوع Recordset برای کار با داده ها وجود دارد که بنا به منبع داده ها (جدول یا سوال) و روشهای کار با آنها از یکی از این روشها استفاده می شود.

دسترسی به لیست داده ها به روش Table

مجموعه ای از رکوردهای یک جدول در بانک اطلاعاتی است. در این روش برای جستجوی سریعتر می توان از ایندکس استفاده کرد. داده ها نیز قابل تغییر خواهند بود. این روش فقط در مورد جدولهای محلی نه الحاق شده صادق است.

دسترسی به لیست داده ها به روش Dynaset

مجموعه ای ازpointer یا Bookmark برای اشاره به داده های جدولها یا سوالها در یک بانک اطلاعاتی است. با این روش می توان به داده های چندین جدول محلی یا الحاق شده دسترسی پیدا کرد. این داده ها ممکن اسن نتیجه یک SQL باشند که قابل ویرایش نیز هستند. در اینجا از روش ایندکس برای جستجوی رکوردها نمی توان استفاده کرد.ضمناً در برخی موارد لیست داده ها با این روش قابل ویرایش نیست. برای جستجوی رکوردها در این نوع لیست داده ها از روش Find استفاده می شود.

دسترسی به لیست داده ها به روش Snapshot

این روش کپی یا تصویر مجموعه ای از رکوردهایی است که هنگام ایجاد Snapshot وجود دارنـد. بطـور اختـیاری فقط می توان روی آنها به جلو حرکت کرد در نتیجه عملیات سریعتر روی آنهـا انجـام می شود. در این روش داده ها قابل ویرایش نیست. قبل از برگشت کنترل به بـرنامه تمـام رکوردهـای منـبع داده ها خوانده می شود. تغییرات را در داده هایی که در یک محـیط چـند کاربـره ایجـاد شـده انـد، مـنعکس نمـی شود. در اینجا از روش ایندکس برای جستجوی رکوردها نمی توان استفاده کرد.

در VBA بـا یـک دستورالعمل و یک نوع عنصر می توان به هر سه روش بالا به لیست داده ها دسترسی داشت:

```
Dim rst As Recordset
Set rst=db.Openrecordset("نام Table یا Query" [ , Type [ , Options  ]]
```

لیسـت داده هـا ممکـن است از یک جدول موجود، نتیجه یک سوال، یا اجرای عبارت SQL باشد. همانطور که در دستورالعمل بالا مشاهده می کنید نوع دسترسی به داده ها به اختیار قابل تعریف است.

```
Set rst=db.Openrecordset("نام Table یا Query"Db_Open_Table)
```

برای دسترسی به لیست داده ها به روش Table

```
Set rst=db.Openrecordset("نام Table یا Query"Db_Open_Dynaset)
```

برای دسترسی به لیست داده ها به روش Dynaset

```
Set rst=db.Openrecordset("نام Table یا Query"Db_Open_Snapshot)
```

برای دسترسی به لیست داده ها به روش Snapshot

اگـر نوع دسترسی به لیست داده ها تعیین نشود VBA نزدیکترین روش مناسب با منبع داده ها را اسـتفاده مـی کـند. چه در صورت تعیین یک روش غلط، هنگام اجرای برنامه با پیغام خطا روبرو می شوید.

فرمانهای دسترسی به لیست داده های یک Table یا Query

همـزمان در یـک بـرنامه می توان به داده های چند جدول یا سوال مختلف دسترسی داشت و روی آنها کارهای مختلفی انجام داد.

```
Dim db As Databases
Dim rec1 As recordset
Dim rec2 As Recordset

Set db= DBEngine.Workspaces(0).Databasees(0)
Set rec1= db.Openrecordset (”Table1”)
Set rec2= db.Openrecordset (”Table2”)
```

پـس از فـرمانهای مـربوط بـه انجـام عملیات روی این لیست ها سعی کنید لیست داده ها را ببندید. برای تکمیل مثال بالا از روش زیر برای بستن لیست استفاده می شود:

```
rec1.Close
rec2.Close
```

چند نمونه از Property ها ، Method ها :

Active Method :

برای فعال کردن Workbook ، Worksheet ، Chart ، Window ، Range و OLE استفاده می شود .

[Object.] Activate

Exp: Sheets (“Sheet2”). Activate

Active Cell Property :

برای استفاده از صفات یک سلول استفاده می شود .

Exp : Activecell.value

Active Sheet Or Workbook Property :

برای ارجاع به اطلاعات یک Worksheet و یا Workbook استفاده می شود .

[Application.]Activesheet
[Application.]Activeworkbook

Exp: Activesheet.Name
Activeworkbook.Name

Add Method :

برای اضافه کردن Object ها و دیگر اشیاء و موضوعات کاربرد دارد .

Set My Object = Object.Add
Exp : Application.Workbook.Add
Seta = Activeworkbooks.Sheets.Add

Auto filter method :

تنظیم Autofilter بصورت رفت و برگشتی.

Rang.Autofilter([[[field,criteria1],operator],criteria2])
Operatorبرای عملگرهای منطقی and و Or استفاده می شود.

موارد Operator :

x1top10percent,x1top10items
x1orx1bottom 10 percent,x1 and x1 bottom 10 items
Exp: selection.autofilter field:=,criteria1:=”1/1/97”
Selection.autofilter.toggles autofilter off

Cells method

برای دسترسی سطری و ستونی به اطلاعات یک سلول .

Sheetname.cells(row,col)
Exp: ActiveSheet.cells(10,8).value

Close Method

برای بستن یکwindow و یا workbook.

[object.]close([savechanges,filename,routeworkbook])
Exp: Activeworkbook.close(true,”c:\change.xls”).
Activewindow.close

Color index Property

رنگ مربوط به محدوده یا سلول را تغییر می دهد.fill یا فونت.

[object.]colorindex=indexnum
Exp: selection.interior.colorindex=6

Copy method

محتویات object جاری را به Clipboard می فرستد .

Object . copy
Object . copy (destination)
Object . copy (befor,after)
Exp : selection . copy
Selection . copy (cells(5,5))

Delete method

Object ویا محدوده را حذف می کند .

[object.] Delete
[object.] Delete (shift) ----------▶ **shift : xlshift to left**
xlshift up

Exp : cells(1,1) . Delete
cells(1,1) . EntireRow . Delete

Display Alerts Property

صفتی که تعیین می کند آیا هشدارهای تبادلی کد نمایش داده شود یا نه.

[Application .]display Alerts = turn / false
Exp : display Alerts =false

Fontstyle property

خاصیت استیل فونت را نگهداری می کند .

Font . fontStyle = stelestring
Exp : selection , fontstyle = Bold Italic

Formula property

مشخصات فرمول یک سلول را بازیابی می کند .

Cell. Formula=Formula
Exp : Activecell . Formula = " =A1+10"

FormulaRlCl property

تنظیم مشخصات فرمول سلول در قالب سطر و ستون

Rang . Formula RlCl = Formula
Exp : Activecell . formulaRlCl = " =Rc[-1]+10"

Insert method

برای درج سلولها در work book یا کاراکترهای موجود قبل از رشته .

[object .] . Insert (shift)
[object .] . insert (insertstring)
Exp : selection . Insert (x1shiftdown)
Activecell . EntiveRow . Insert

Move method

برای حرکت دادن sheet ها .

sheet . move (befor,after)
Exp : Active sheet . move, sheets ("sheet3")

Name property

نام object را تغییر می دهد .

object . Name = string
Exp : Active workbook . name
Sheets ("sheet1") . Name = "my sheet"
Active sheet . name . name = "my sheet"

Number format property

فرمت نمایش lable ها و سلولها را نمایش می دهد .

Object . number format = stringval
Exp : Activecell . Value =12
Activecell . number format = "General"
Activecell . number format = "hh:mm:ss m/d/yy"
Activecell . number format =-"($# # #,# #0.00-)I[Blue]($# # #,# # 0.00)"

Range Object

برای دسترسی به یک یا چند سلول .

Exp :
Range (“A1”) . value = 10
Range (“A2”) =10
Range (“A3”) =”Hello”
Range (“A1 : A8”) . formula = “Rand()”

Screen Updating property

تعیین اینکه آیا ارتقاها روی صفحهٔ نمایش داده شوند یا نه .

Application . screenupdating = true/false

Select method

برای انتخاب سلول و یا object یا worksheet .

Object . select ([replace])
Exp : sheets (“sheet3”) . select
Rang (“A1 : A8”) . select

Action Property :

تعیین کننده عملی است که هنگام وقوع یک رویداد انجام می گیرد .

ActionSetting.Action = ActionType
Exp: ActivePresentation.Slides(1).Shapes(1).ActionSettings (ppMouseOver) . Action = ppActionNextSlide

Hyperlink Property :

اتصال به یک سایت HTTP یا یک صفحه Web بوسیله یک ارجاع از طریق شی

Hyperlink صورت می پذیرد. این قابلیت در تمامی برنامه های Office وجود دارد .

Object.Hyperlink
Exp:
ActivePresentation.Slides(1).Shapes(1).ActionSetting(ppMouseClick) .Action = ppActionHyperlink
ActivePresentation.Slides(1).Shapes(1).ActionSetting(ppMouseClick) .Hyperlink.Address = "http://www.cvisual.com"

Entry Effect Property :

تاثیر متحرک سازی روی ورودی به اسلاید جاری را تعیین می کند .

EntryEffect = _effectType
Exp :
ActivePresentation.Slides(2).Shapes.Title.AnimationSetting.EntryEffect = ppEffectWipeLeft

GoTo Slide Property :

نما را به شماره اسلاید مشخص تغییر می دهد .

View.GoToSlide(index)
Exp : Windows(1).View.GotoSlide 2

Address Property :

آدرس پست الکترونیکی e_mail دریافت کننده خاص را نگهداری می کند .

Recipient.Address = e_mail
Exp : Set myitem = Application.CreateItem(0)
' OlMailitem
Set myrecipient = myitem.Recipients.Add "Dan Rahmel"
Msgbox myrecipient.Address

تمرین _ ارسال یک پست الکترونیکی e_mail بوسیله VBA تحت OutLook :

در محیط VBA یک فرم به نام Refrences و با ظاهری شبیه محیط Outlook ساخته و دکمه ای به نام cmdoutlookmail ایجاد کرده و در Event – Click آن کد زیر را بنویسید :

```
Private Sub cmdoutlookmail_Click()
  Dim objobjectouylook As Object
  Dim objitem As MailItem
Set objoutlook = CreateObject("Outlook.Application")
  ' Create OlMailItem
Set objItem = objoutlook.CreateItem(OlMailItem)
With objitem
        .Subjct = " VB Prog Ref "
        .To = darn@cvisual.com
        .CC = darn@Coherntdata.com
        .Body = " I found this book useful "
        .Attachments.Add " C:\autuexec.bat "
        .Send
End With
  Set objitem = Nothing
  Set objoutlook = Nothing
End Sub
```

Value property

مقداری را برای object خاص نگهداری می کند .

Object . value = value

چند نمونه از علامتها و عبارتهای کلیدی در VBA :

`

فرمان نشانه گذاری . هر فرمان با آن ، نادیده گرفته میشود .

’ Comment

!

عملگر فیلد بانک اطلاعاتی و علامت هر Sheet در Excel .

Sheet3**!**C3
Recordset **!** field
Recordset از نوع dynaset ، table یا snapshot

#

عملگر متغیر را بر اساس double تنظیم می کند .

a #
Exp : a **#** = 57

$

نوع متغیر یک رشته را تنظیم می کند

a **$** = Dim x as string

%

نوع متغیر را معادل عدد صحیح intejer تنظیم می کند .

A **%** = 5/14

&

عملگر ترکیب یا الحاق رشته ای .

0 **&** 1.33 & # 1/2/97 #

Dim

متغیری راتعریف می کند.

Dim [shared]name [**as**[new]type][,name[as[new]type]]
Dim E **as** (long,currencey,single,string,…)
Dim E **as** integer

End

اجرا را متوقف کرده یا یک رویداد مثل تابع if را پایان می دهد.

Eof

شرط پایان فایل را برمی گرداند.

Eof(file_number)
Exp : **Open** "c:\test.txt" For **input as** #1 : ? **Eof**(1) : **b** #1

Do...Loop

ایجاد حلقه برای انجام فرمانها .

Do [While i until condition][Statements][**Exitdo**]
[Statements]**Loop**

برای آزمایش:

Do [statements]**Loop**[While i Until condition]

Array

آرایه در حافظه ایجاد می کند که شامل مقادیری است که در لیست پارامترها گذر داده شده اند.

Array(Arglist)
Exp: myarray =**Array** ("Mcclane","Gennero",…)

Beep

بیپ بلندگو

Beep (*.wav) , (*.mid)

Close & Open

کلیه فایلهای باز یا مشخص شده توسط شماره فایل را می بندد.

Close [#][filenumber%][,#]filenumber%]

Exp:

Open "c:\vbtest.txt" for output as #1 :-

Print #1,"Hello World!" : close #1

Const

مقداری را بعنوان یک ثابت اعلان می کند.

[Global] const name=expression [,name=expression]

چند نمونه از توابع کاربردی و معمول VBA (توابع سیستمی) :

دو گروه از توابع به نام توابع عددی و توابع رشته ای وجود دارد . توابع عددی به همراه ثابتها ، متغیرها و عبارتهای عددی بکار می روند ، توابع رشته ای بر روی متغیرهای رشته ای کار کرده ، آنها را به توابع عددی و برعکس تبدیل می کنند . تعدادی از توابع پر استفاده در زیر ذکر شده اند :

Cstr

عبارت دا ده شده را به یک رشته تبدیل می کند.

Cstr(expression)

Exp:

Cstr (12)

Cvar

عبارت مشخص را در انواع گوناگون برمی گرداند.

Cvar(expression)

Exp:

Cvar ("Hello")

Cvar (int(512))

تابع قدر مطلق – ABS

این تابع قدر مطلق عدد را به دست می آورد که در واقع مقدار واقعی یک عدد بدون توجه به علامت آن است .

شکل کلی این تابع به صورت زیر است :

```
ABS (X)
```

که در آن **X** می تواند یک ثابت و یا متغیر عددی و یا یک عبارت محاسبه ای باشد .

مثال ۱ :

```
A= ABS(x)
```

مقدار ذخیره شده در **A** برابر ۱۳ می باشد اگر **x** برابر ۱۳ – باشد .

مقدار ذخیره شده در **A** برابر ٦ می باشد اگر **x** برابر ٦ باشد .

مقدار ذخیره شده در **A** برابر صفر می باشد اگر **x** صفر باشد .

مثال ۲ :

```
Y= ABS (10*x + 2 )
```

مقدار ذخیره شده در **y** برابر ۱۲ است اگر **x** برابر یک باشد . مقدار ذخیره شده در **y** برابر ۲۸ است اگر **x** برابر ۳- باشد .

این تابع برای یافتن اختلاف بین دو عدد بدون توجه به مثبت یا منفی بودنشان مفید است .

تابع INT

تابع INT برای بدست آوردن نزدیکترین عدد صحیح به یک مقدار عددی بکار می رود .

این تابع بزرگترین عدد صحیحی را که از دست داده شده بزرگتر نباشد به دست می دهد .

شکل کلی این تابع به صورت زیر است :

```
INT(x)
```

که در آن **x** می تواند یک ثابت یا متغیر عددی و یا یک عبارت محاسبه ای باشد .

مثال ۱ :

```
A = INT (x)
```

مقدار ذخیره شده در **A** برابر ۱۲ است اگر **x** برابر ۱۲/۳۵۷ باشد .

مقدار ذخیره شده در **A** برابر ٥- است اگر **x** برابر ٤/٥٩- باشد .

مقدار ذخیره شده در **A** برابر ۲٦ است اگر **x** برابر ۲٦/۹۹۹ باشد .

مثال ۲ :

Y = INT (2*A-4)

مقدار ذخیره شده در **y** برابر ٦ است اگر **A** برابر ٥/٤ باشد .

تابع INT کاربردهای زیادی دارد . یکی از آنها روند کردن یکعدد به نزدیکترین عدد صحیح است . با افزودن ۰/۵ به هر عدد و سپس بکارگیری این تابع عدد صحیح مورد نظر خود را به دست خواهید آورد .

کاربرد دیگر تابع INT برای این است که معین کنیم آیا یک عدد صحیح است یا غیر صحیح .برای این کار از این حقیقت استفاده می شود که اگرN یک عدد صحیح باشد آنگاه N=INT(N) .

به بیان دیگر می توانیم با مقایسهٔ یک عدد با INT همان عدد ، تعیین کنیم که عدد صحیح است یا خیر.

ضمناً می توانید با استفاده از تابع INT قسمتهای پس از دو رقم اعشار یک عدد را قطع کرده و یا عددی را تا دو رقم اعشار Round کنید .

اگر می خواهید قسمتهای اضافه عدد را تا دو رقم اعشار قطع کنید از INT(100*x)100 استفاده کنید . بعنوان مثال به ازاء :

X = 475.5836

حاصل کار این تابع برابر است با :

100*x = 47558.36
INT(100*x) = 47558
INT(100*x) /100 = 475.58

اگر می خواهید عددی را تا دو رقم اعشار روندکنید از تابع به شکل زیر استفاده کنید :

INT(100*x = 0.5)/100

بعنوان مثال اگر :

X = 27.67815

آنگاه :

100*x = 2767.815
100*x + 0.5 = 2768.315
INT(100*x + 0.5) = 2768
INT(100*x + 0.5)/100 = 27.68

تابع RND

این تابع به برنامه نویس اجازه می دهد که اعداد تصادفی بین صفر و یک را تولید کند . راهی وجود ندارد که بدانیم تابع RND دقیقاً چه نتیجه ای می دهد . این تابع ممکن است ۰/٦۸۳ ، یا ۰/۰۰۸٤۷ یا ۰/۰۰۰۹۸ باشد – یعنی هر عددی بین صفر و یک به استثنای خود اعداد صفر و یک .

شکل کلی این تابع به صورت زیر است :

RND(X)

که درآن **X** می تواند یک ثابت یا متغیر عددی و یا یک عبارت محاسبه ای باشد .

مقدار موجود در **X** مسیر اعداد تصادفی تولید شده را کنترل می کند . در یک نگارش BASIC اگر مقدار **X** صفر باشد باعث می شود که RND هر دفعه عدد تصادفی جدیدی بین صفر و یک ایجاد کند . در نگارش دیگری از BASIC اگر **X** بزرگتر از صفر باشد باعث ایجاد اعداد تصادفی شده و اگر **X** صفر باشد آخرین عدد تولید شده قبلی را پس می دهد.

شما می توانید با انجان برخی عملیات محاسباتی اعداد تصادفی بزرگتر از یک تولید کنید . فرض کنید می خواهید عددی تصارفی بین صفر و ده را در **y** ذخیره کنید . به شکل زیر می توانید به کامپیوتر بگویید تا این کار را انجام دهد :

```
Y = 10*RND(J)
```

از آنجایی که مقدار RND بین صفر و یک است مقدار **y** بین صفر تا ده خواهد بود .

اگر عددی بین ۱ و ۱۰ می خواهید از Y=9*RND(1)+1 استفاده کنید .

اغلب ممکن است بخواهید یک عدد صحیح تصادفی ایجاد کنید . چون RND همیشه اعداد اعشاری تولید می کند برای این کار باید توابع RND و INT را باهم ترکیب کنید .

مثال ۱ :

```
N=INT(500*RND(1)+1)
```

N عددی صحیح بین ۱ و ۵۰۰ را خواهد داشت (به استثنای ۵۰۰) .

مثال ۲ :

```
N=100*INT(5*RND(1)+1)
```

N عددی صحیح بین ۱ و ۵۰۰ را خواهد داشت .

برای تولید اعداد غیر قابل پیش بینی و یا اعدادی مثل اعداد بازیهای شانسی ، **RND** مفید می باشد .

تابع علامت –SGN

این تابع شما را قادر می سازد که بدانید عدد ذخیره شده در یک متغیر و یا نتیجه یک عبارت محاسباتی مثبت است یا منفی .

شکل کلی این تابع به صورت زیر است :

SGN(X)

که در آن **X** حاوی یک مقدار عددی است .

اگر **X** کوچکتر از صفر باشد **SGN(X)** برابر منفی یک خواهد بود .

اگر **X** صفر باشد **SGN(X)** صفر خواهد بود .

اگر **X** بزرگتر از صفر باشد مقدار **SGN(X)** برابر یک خواهد بود .

تابع ریشه دوم SQR

از این تابع برای یافتن ریشهٔ دوم یک عدد مثبت استفاده می شود . ریشه دوم یک عدد منفی عددی حقیقی نبوده و با این تابع قابل محاسبه نمی باشد .

شکل کلی تابع **SQR** به صورت زیر است :

SQR(X)

که در ریاضی به صورت **X**√ نشان داده می شود .

توابع SIN,COS,TAN

BASIC برای حل مسائل ریاضی شما سه تابع مثلثاتی فوق را ارائه داده است .

شکل کلی این توابع به صورت زیر است :

SIN (X)
COS (X)
TAN (X)

که در آنها **X** حاوی زاویه بر حسب رادیان می باشد .

توابع LOG و EXP

این دو تابع با لگاریتم طبیعی سر و کار دارند . رابطه بین توان و لگاریتم در زیر داده شده است :

در ریاضی : $log(Y)=X$ و : $Y=ex$

شکل کلی این توابع در **BASIC** به صورت زیر است :

LOG(x)
EXP(X)

توابع CHR$ و ASC

پیش از یادگیری کاربرد این دو تابع باید در مورد کدهای عددی مورد استفاده در کامپیوتر جهت نشان دادن کاراکترها مطالبی بدانید. در پی تلاش جهت استاندارد نمودن نمایش داده ها، کامپیوترها را به گونه ای طراحی نموده اند که یکی از دو سیستم کددهی EBCDIC یا ASCII را بکار برند . EBCDIC مخفف کلمات

'Extended Binary Coded Decimal Interchange Code'

و ASCII مخفف کلمات :

'American Standard Code for Information Interchange'

می باشد.

تابع ASC اولین کاراکتر یک متغیر رشته ای را به کد اسکی آن تبدیل می کند . تابع CHR$ دقیقاً عکس تابع ASC عمل می کند . این تابع یک کد اسکی را به یک کاراکتر تبدیل میکند .

شکل کلی تابع ASC به صورت زیر است :

ASC(X$)

فرض کنید متغیر رشته ای X$ حاوی کلمه Computer است . در جدول ضمیمه می بینیم که کد اسکی برای **c** برابر ٦٧ است . بنابراین دستور A=ASC(X$) عدد٦٧ را در **A** قرار خواهد داد .

شکل کلی تابع CHR$ به صورت زیرست :

CHR$(X)

X یک ثابت عددی و یا متغیر عددی است که حاوی یک کد اسکی است که به یک کاراکتر تبدیل می شود . بعنوان مثال کد اسکی علامت کوتیشن (") برابر ٣٤ است . دستور PRINT CHR$(34) سبب می شود که علامت کوتیشن (") روی صفحه تصویر نشان داده شود .

بخاطر داشته باشید که ASC(X$) در یک متغیر عددی ذخیره شده و CHR$(X) همیشه در یک متغیر رشته ای ذخیره می گردد .

تابع CHR$ برای نشان دادن کاراکترهای خاص و یا کاراکترهای گرافیکی بر روی صفحه تصویر بسیار مفید می باشد .

تابع ASC در مواقعی که برنامه باید با کاراکترهای غیر معمول سرو کار داشته باشد مفید است . مثلاً صفحه کلید بیشتر کامپیوترها دارای کلید Esc می باشد که به منظور خاصی به کار می رود . وقتی این کلید فشرده شود یک کاراکتر معمولی را ایجاد نمی کند بلکه کد اسکی ۲۷ را ایجاد میکند . اگر می خواهید فشرده شدن این کلید را در برنامه بررسی کنید می توانید دستورات زیر را مورد استفاده قرار دهید :

```
I F    ASC (x $ )    = 27    THEN  ....
```

پس از اجرا کاراکتر رشته ورودی را تست کرده و بررسی می کند که آیا برابر کد اسکی Esc که ۲۷ است می باشد یا خیر .

توابع LEFT$ و MID$ و RIGHT$

این توابع به شما اجازه میدهند که یک متغیر رشته ای ره به رشته های کوچکتر یا فرعی تقسیم کنید .

تابع LEFT$ به کامپیوتر می گوید که به تعداد معینی از کاراکترها که از انتهای چپ رشته کاراکتری شروع می شوند رجوع نماید .

تابع RIGHT$ شما را آماده می کند که به تعداد معینی از کاراکتدها که از سمت راست رشته شزوع می شوند دستیابی پیدا کنید .

شکل کلی تابع LEFT$ به صورت زیر است :

LEFT$ (X$,N)

X$ متغیر رشته ای است که می خواهید آنرا تقسیم کنید .

N تعداد کاراکترهایی است که از سمت چپ متغیر X$ شروع می شوند .

مثال :

X$ = "GOOD EVENING"

LEFT$(X$,4) حاوی کاراکترهای **"GOOD "** است .

LEFT$(X$,2) حاوی کاراکترهای **"GO "** است .

LEFT$(X$,8) حاوی کاراکترهای **"GOOD EVE"** است .

شکل کلی تابع **RIGHTS** به صورت زیر است :

RIGHT$(X$,N)

X$ متغیر رشته ای است که می خواهید آن را تقسیم کنید .

N نشان دهنده تعداد کاراکترهایی است که از سمت راست X$ شروع می شوند .

چند برنامه نمونه برای تمرین در محیط Excel VBA :

برنامه مقایسه دو بانک **Excel** و یافتن مقادیر یک بانک در بانک دیگر و جاگذاری فیلدهای موردنظر:

```
Sub find ()
For i=1 to 200
  For j=2 to 100
     If cells(i,1).value= cells (j,10).value then
        Cells(j,11).select
        Cells (i,6).value=selection
        Exitfor
     Endif
   Next j
Next I
Endsub
```

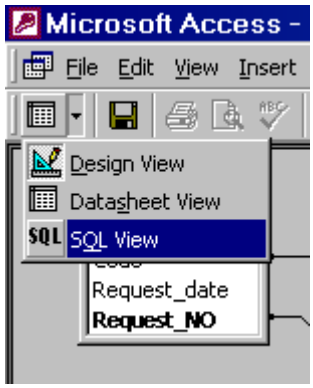
برنامه حذف ده ردیف :

```
Sub earase()
For i=2 to 10
   i=i-1
       astr=cstr(i) + ":" + cstr(i)
        Rows(astr).select
        Selection.delete shift :=xlup
 Next I
Endsub
```

برنامه نویسی SQL در VBA و Access :

SQL یک زبان عمومی و بسیار رایج برای دسترسی به داده ها در انواع کامپیوترهای امروزی است . این زبان یک روش پیشرفته و قوی برای ایجاد انواع Query و همچنین برنامه نویسی در محیط VBA می باشد . Ms Access این زبان را پشتیبانی کرده و امکان کار با دستورالعملهای آن را در خود فراهم کرده است . در مجموع سه نوع استاندارد SQL وجود دارد : SQL-86 (رایجترین استاندارد در حال حاضر) ، SQL-89 (با جزئی تغییر) ، SQL-92 (با تغییرات عمده) . Ms Access ، SQL-89 و برخی عناصر SQL-92 را پشتیبانی می کند .

در Access برخلاف سایر برنامه ها که SQL را پشتیبانی می کنند ، دارای خط فرمان یا محیط مشابه برای نوشتن مستقیم فرمانهای مستقیم SQL و زدن کلید Enter برای اجرای آنها ندارد . برای نوشتن یا مشاهده فرمانهای SQL از محیط مربوطه در طراحی Query استفاده می شود . برای مشاهده نتیجه عبارتهای SQL از فرمان اجرای Query یا نمایش لیست داده های آن استفاده می شود .



برای مشاهده محیط برنامه نویسی SQL در Access ، پس از طراحی Query در محیط Design گزینه SQL را از میان آیتمهای دکمه Veiw بالای صفحه انتخاب کنید .

فرمان SELECT :

SELECT در SQL یعنی همه چیز . با فراگیری این عبارت و تمام پارامترهای آن ، در واقع می توان گفت SQL را فرا گرفته اید . این عبارت رکوردهای داده ها را انتخاب و به صورت یک لیست از نوع Daynaset نمایش می دهد . دستورالعمل اصلی عبارت elect S به شرح زیر است :

SELECT نام فیلدها

FROM نام جدول

[WHERE شرط**]**

[ORDER BY ترتیب بندی**]**

همانگونه که مشاهده می کنید ، عبارتهای SELECT و FROM اجباری و بقیه اختیاری است . در اینجا هریک از پارامترهای SELECT به شرح زیر ارائه می شود .

عبارت SELECT

با این عبارت فیلدهای نمایشی در سئوال مشخص می شود . مانند روش QBE در اینجا هم از ستاره یعنی تمام رکوردها می توان استفاده کرد .

SELECT * **یعنی تمام رکوردها**

SELECT lastname یعنی فقط فیلد فایل لیست شود

یا :

SELECT [Customer #] ,LastName,Firstname

در آخرین عبارت به دلیل استفاده از علامت ویژه # همراه با نام فیلد از علامت **[ ]** استفاده شده است . البته تمام نام فیلدها را می توان با علامت **[ ]** همراه نمود .

برای تغییر عنوان فیلد هنگام نمایش از روش زیر استفاده می شود :

SELECT[Customer#] As ID ، [Lastname] &" "& [Customer Name];

اگر از بیش از یک جدول می خواهید استفاده کنید لازم است از نام جدول ، همراه با نام فیلد استفاده کنید . عبارت **SQL** که بطور خودکار توسط اکسس ساخته می شود نام فیلد همراه با نام جدول است . برای مثال اگر بخواهید فیلد Customer # با نام جدول یعنی tblOrder نوشته شود بصورت زیر خواهد بود :

SELECT tblOrder . [Customer #]

عبارت FROM

با این عبارت نام جدول یا سئوالی که رکوردهای آن انتخاب می شود مشخص می گردد . اگر از بیش از یک جدول نام برده می شود بایستی نوع ارتباط آنها هم مشخص شود . در این ارتباط توضیحات لازم در ادامه همین فصل آمده است . در اینجا برای سادگی کار از یک جدول نام برده می شود .

```
SELECT *
FROM tblOrder ;
```

تمام رکوردهای جدول **tblOrder** انتخاب می شوند .

```
SELECT  [Order #] , [Order Date]
FROM tblOrder ;
```

در اینجا فقط دو فیلد از جدول بالا لیست می شود .

مانند **SELECT** در اینجا هم می توان برای نام جدول یا جدولها یک نام کوتاه یا موقت تعریف کرد .

```
SELECT  [Order #] , [Order Date]
FROM tblOrder As T1;
```

عبارت WHERE

برای وارد کردن شرط در سئوال از WHERE استفاده می شود . این عبارت اختیاری است و SELECT بدون WHERE تمام رکوردها را نمایش می دهد . عبارت WHERE شبیه ردیف شرط و ردیف بعدی (OR) در QBE است . فرمت این بخش از SELECT به شرح زیر است :

WHERE شرط ۱ **[AND / OR** شرط ۲ **[,...]]**

برای مثال اگر بخواهید به مثالهای قبلی یک شرط اضافه کنید به روش زیر عمل می شود :

```
SELECT  [Order #] , [Order Date]
FROM tblOrder
WHERE  [Order Taker #] = 2;
```

برای آشنایی بیشتر با این بخش ازSELECT به مثالهای زیر توجه کنید :

```
WHERE  [Customer #] = 4
WHERE Sex =" Female" AND Age BETWEEN 21 AND 29
WHERE LastName IS NOT NULL OR (LastName IS NOT NULL AND
FirstName = " joe" )
WHERE OrderDate > DateAdd("yyy", -1 , Date0)
```

گرچه آنچه مشاهده کردید شبیه روش شرط در QBE است لیکن برخی موارد خاص را به شرح زیر بخاطر بسپارید :

- عبارت نوع متن را در علامت " یا ' وارد کنید .

```
WHERE LastName " Jones "
```

- عبارت نوع تاریخی رادر علامت # وارد کنید.

```
WHERE OrderDate > # 4/15/95 #
```

- وقتی به علامت ستاره یا ? در عبارت شرط اشاره می کنید ، از Like استفاده کنید .

```
WHERE FirstName LIKE "P*"
```

عبارت ORDER BY

برای مرتب سازی رکوردها از این عبارت در SELECT استفاده می شود . این عبارت نیز اختیاری است . برای مرتب سازی صعودی از کلمه ASC و مرتب سازی نزولی از DESC استفاده می شود . فرمت این عبارت بصورت زیر است :

ORDER BY ستون ۱ **[ASC/ DESC][,** ستون ۲ **[ASC/ DESC][,...]]**

برای آشنایی بیشتر به یک مثال بشرح زیر توجه کنید :

```
SELECT *
FROM tb/Customer
ORDER BY LastName , FirstName
```

جدولهای مرتبط (Related Tables)

هنگام استفاده از بیش از یک جدول در سئوال بایستی ارتباط (Relation) آنها را نیز در عبارت SQL تعریف کرد . علاوه بر این در اشاره به نام فیلدها ، بایستی نام جدول مربوطه را هم تایپ کنید . فرمت عمومی برای استفاده از چند جدول و اشاره به ارتباط آنها بشرح زیر است :

لیست فیلدها SELECT

FROM table1 INNER / LEFT [OUTER] / RIGHT [OUTER] JOIN table2
ON table1 . column1 = table2.column2;

مقادیر و رکوردهای تکراری

بعد از کلمه SELECT می توان با استفاده از عبارتهای زیر وضعیت نمایش رکوردهای نتیجه را مشخص کرد :

- **ALL**

فرمت این مشخصه فیلدها SELECT ALL است . با این فرمان تمام رکوردهای مطابق شرط نمایش داده می شوند . این روش شبیه مشخصات Uniquevalues > NO و UniqueRecords > NO در پنجره مشخصات می باشد .

- **DISTINCT**

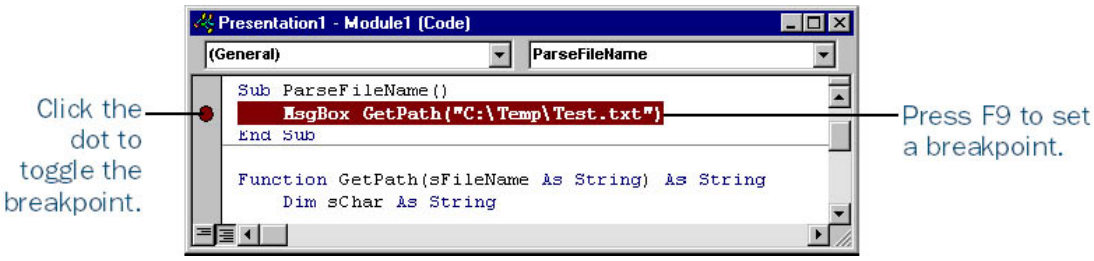
فرمت این مشخصه فیلدها SELECT DISTINCT است . با این فرمان از تکرار رکوردها جلوگیری می شود . شرط تکرار در اینجا داده های فیلدهای منظور شده در فرمان است . اگر بیش از یک ستون در فرمان منظور شده باشد ، تکرار در تمام ستونها تشخیص داده شده و از تکرار رکوردها جلوگیری می شود . نتیجه این نوع سئوال قابل ویرایش نیست و به همین دلیل فقط در موارد خاص باید از آن استفاده کرد . استفاده از این فرمان شبیه مشخصه UniqueRecords > YES در پنجرا مشخصات سئوال است .

- **DISTINCTROW**

فرمت این مشخصه فیلدها SELECT DISTINCTROW است . با این فرمان از تکرار رکوردها جلوگیری می شود . شرط تکرار در اینجا داده های تمام فیلدهای جدول مبدأ است . اگر یک یا بیش از یک ستون در فرمان منظور شده باشد ، تکرار در تمام فیلدهای جدول مبدأ تشخیص داده شده و از تکرار فیلدها جلوگیری می شود . نتیجه این نوع سئوال قابل ویرایش است . استفاده از این فرمان شبیه مشخصه UniqueRecords > YES در پنجره مشخصات سئوال است . در بیشتر مواقع از این فرمان استفاده می شود .

کنترل برنامه ها از طریق خطاگیری و اشکال زدائی (Debugging)

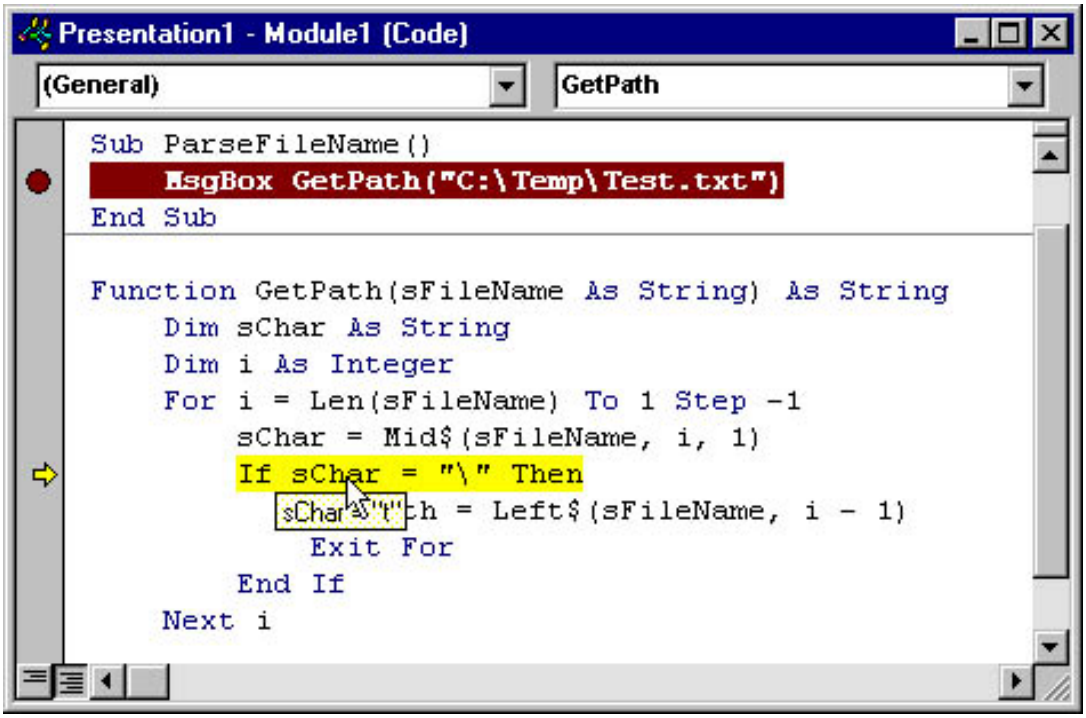
هر برنامه نویسی نیاز دارد تا اشکال گیری و چگونگی یافتن خطاها را در برنامه ها و کد نویسی های خود در محیط VBA بیاموزد . برای این کار راههای مختلفی وجود دارد . روش Break Point یکی از بهترین و ساده ترین متدهای Debug کردن در محیط Visual Basic است .

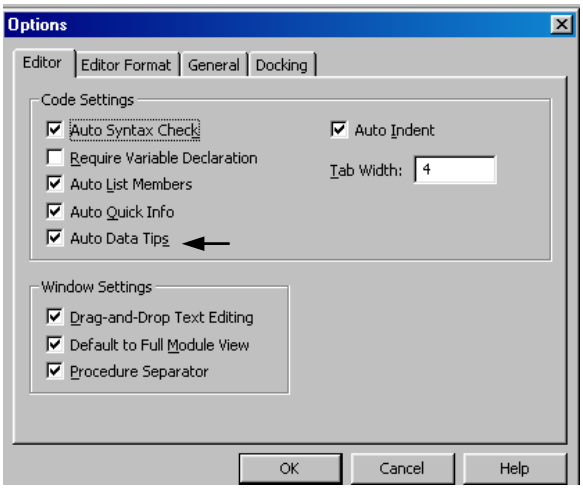


روش نقطه توقف (Break Point) :

وقتی ویژوال بیسیک ، برنامه شما را اجرا می کند شما می توانید نتیجه اجرای هر خط از فرمانهای خود را ببینید و مقداردهی متغیرهای خود را ارزیابی کنید . اگر شما دکمه F5 را فشار دهید ، کد برنامه شما اجرا می شود و ویژوال بیسیک در حالت Run Mode قرار می گیرد ولی اگر کلید F8 را فشار دهید می توانید خط به خط اجرای برنامه را کنترل کنید ، در این حالت ویژوال بیسیک در حالت Break Mode می باشد . اگر در جریان اجرای برنامه بخواهید کنترل خط به خط برنامه را ببینید ، باید کلید F9 را بفشارید (Toggle BreakPoint) . در صورت HighLight شدن خط برنامه شما می توانید با MouseOver کردن بر روی هر کدام از متغیرها ، مقدار فعلی آن را ببینید .

روشهای بسیاری برای مشاهده مقادیر متغیرها در برنامه ها وجود دارد که در حالت Break Mode اگر خط مورد نظر روشن شود (HighLight)با حرکت موس بر روی متغیرها مقادیر فعلی آنها در Data Tips Window می توانیدببینید . در این حالت شما می توانید با کشیدن (Drag) موس به خط بعدی و فشردن کلید F5 ادامه برنامه را دنبال کنید . (می توانید با فشردن کلید F8 نیز به کنترل خط به خط ادامه دهید)





نکته :

اگر پنجره Data Tips را مشاهده نکردید می توانید از مسیر زیر :

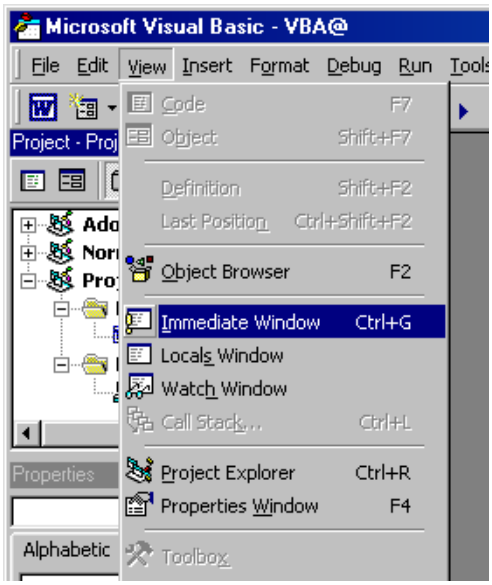
Tools / Option / Editor (Tab)

گزینه Auto Data Tips را فعال کنید .

VBA سه پنجره دیگر را نیز جهت نمایش مقادیر فعلی متغیرها در زمان اجرای برنامه فراهم آوده است که عبارتند از :

پنجره های Watch و Locals و Immediate .

برای استفاده از Watch Window شما می توانید در حالت Break Mode متغیر را از محیط برنامه انتخاب کرده و با موس به پنجره Watch Window , Drag کنید . پنجره Locals نیز شبیه پنجره Watch می باشد از این نظر که مقادیر درون آن با جریان برنامه بصورت اتوماتیک Update می شود .



شما می توانید مقادیر متغیرهای مختلف Module های مختلف را در یک محیط OutLine (در کنار هر عنوان علامت (+)ای برای نمایش زیر عنوانهای آن وجود دارد) در صورت عدم مشاهده این پنجره از منوی View آن را فعال کنید .

پنجره سوم Immediate است که امکانات بیشتری در زمینه کنترل برنامه و خطایابی در اختیار کاربر قرار می دهد . شما می توانید با Copy و Paste کردن متغیر به درون پنجره مقدار آن را ببینید و یا کل خط جاری را Copy کرده و پس از Paste در Immediate Window در انتهای آن کلید Enter را بفشارید تا نتیجه آن را ببینید .

برای نمایان کردن تمامی Sheet های پنهان شده در برنامه می توانید کد زیر را در یک Module بنویسید :

```
Sub Un_Hide_All ()
Dim sh As Worksheet
For Each sh In Worksheets
Sh.Visible = True
Next
End Sub
```

برای یافتن و انتخاب یک محدوده از تاریخهایی که بر اساس روز و ماه و سال فرمت بندی شده باشند کد زیر را در یک Module بنویسید :

```
Sub Find Dates()
On Error GoTo errorHandler
Dim startDate As String
Dim stopDate As String
Dim startRow As Integer
Dim stopRow As Integer
startDate = InputBox("Enter the Start Date : (mm/dd/yy)")
If startDate = "" Then
End
stopDate = InputBox("Enter the Stop Date : (mm/dd/yy)")
If stopDate = "" Then
End
startDate = Format(startDate , :mm/dd/yy")
stopDate = Format(stopDate , :mm/dd/yy")
startRow =
Worksheets("Table").Columns("A").Find(startDate,lookin:=xlVal
ues,lookat:=xlWhole).Row
stopRow =
Worksheets("Table").Columns("A").Find(stopDate,lookin:=xlValu
es,lookat:=xlWhole).Row
Destination:=Worksheets("Report").range("A")
End
ErrorHandler:
MsgBox " There has been an error : " & Error() & Char(13) &
"Ending Sub …….Please Try Again",48
End Sub
```