

الان من می خوام خیلی ساده شروع کنم و یک چیزهای پایه رو با وی بی ا مرور کنم
این الان یک ساب پشت یک تکه است

```
Private Sub CommandButton1_Click()
```

```
MsgBox 'salam'
```

```
End Sub
```

خروجی سلام رو با یک دیالوگ نشون میدی (به این فسل که بالا میاد میگن دیالوگ)
حالا اگر بخوام همین کار رو می تونم تو یک ساب دیگه بکنم و از این ساب اون ساب رو کال کنم
ببخشید قبلی یک اشکال کوچیک داشت
باید دابل کوتیشن باشه

```
Private Sub CommandButton1_Click()
```

```
MsgBox "salam"
```

```
End Sub
```

ساب همون مخفف ساب روتینه
یک چیز دی
دیگر هم هست به نام فانکشن
که خیلی شبیه ساب روتینه فرقی اینه که فانکشن می تونه یک چیزی برگردونه ولی ساب نمی تونه
الان نشونتون می دم که منظورم چیه
می خوایم یک ساب روتین بنویسیم که یک عدد رو به توان دو برسونه میشه این

```
Private Sub CommandButton1_Click()
```

```
Dim number As Long
```

```
number = 10
```

```
number = number * number
```

```
MsgBox "tavan do =" & number
```

```
End Sub
```

خط اول متغیر تعریف می کنه به نام نامبر که توش ۱۰ هست
خط بعدی اون رو توان دو می رسونه
خط بعدی اش هم طی یک دیالوگ یا مسیج باکس به کاربر نشون میدی
بجز یک مسیج باکس یک راه بهتر هم برای نمایش خروجی هست که به نظرم بهتره از اون استفاده کنیم و ان پرینت
تو لاگ خروجی هست
از این نظر بهتره که مسیج باکس رو همیشه چندتا باز کرد یکی رو باز می کنی دیگه همیشه چیزی نشون داد
پرینت می تونه چند بار انجا بشه

برای امتحان پرینت کافیه این کد رو ببینید

```
Private Sub CommandButton1_Click()  
    Debug.Print "salam agha reza"  
End Sub
```

اگر این رو اجرا کنید نمی بینید که چه اتفاقی افتاده
برای اینکه کنسول رو ببینید کافیه که از مسیر زیر کنسول رو فعال کنید
تو منوی بالا بخش
view
گزینه immediate view
حالا یک خروجی پایین نشون داده میشه
میتونید به جای اون کار کنترل و G رو هم باهم بزنید تا نشون داده بشه
خب حالا برگردیم که ساب روتین محاسبه مربع عدد
با پرینت بنویسیم میشه این

```
Private Sub CommandButton1_Click()  
    Dim number As Long  
  
    number = 10  
    number = number * number  
  
    Debug.Print "tavan do =" & number  
End Sub
```

برای اجرا هم که یادتون هست
باید موس رو کرسر روی ساب روتین باشه بعد تکمه مثلث سبز رو بزنید یا اینکه برین تو خود فرمتون تکمه رو فشار
بدین تا اجرا بشه
حالا یک نکته ای
بیاین همین کد رو دیباگ کنیم
دیباگ به این درد می خوره که همیشه مراحل مختلف اجرای یک کد رو دید و خط به خط اجرا کرد و حین اجرا دید که
متغیر ها با چی پر میشن
رو خط
number = number * number

نوار خاکستری کنارش رو کلیک کنید تا یک دایره زرشکی دست بشه به این می گن بریک پوینت break point یعنی
جایی که اجرای برنامه اونجا خواهد ایستاد
حالا اگه اجرا کنید اجرای برنامه اونجا می ایسته
برنامه که وایمیسته اون خط زرد میشه حالا اگر موس رو هم کدوم از متغیر ها نگاه دارین نشون میده توش چیه
حتی می تونید یک متغیر یا یک بخشی از کد رو انتخاب کنید بعد با موس راست کلیک کنید و گزینه add watch رو
بزنید بعد اوکی کنید یک جایی باز میشه که مقدار بخش انتخاب شده رو نشون میشه

اگر بخواین خط به خط یک برنامه را اجرا کنید کافیست که تو منو گزینه

debug

رو گزینه step over رو بزنی یعنی یک گام به جلو

خط زرد یک یک خط میره جلو و یک خط اجرا میشه

اگر خط خودش ساب روتین دیگری باشه با گزینه

step into

میتونی بری توش

هر وقت هم خواستین کلن اجرا رو از سر بگیرد کافیست دوباره تکه مثلث سبز رو بزنی

شورت کی هاش هم به درد می خوره

اجرا برنامه f5 =

اجرا یک خط f8+shift =

رفتن به داخل خط f8 =

خب اینا رو خوب تمرین کنید که تو برنامه نویسی دیباگ خیلی به درد می خوره

حالا برگردیم سراغ کد قبلی یعنی محاسبه مربع عدد

یعنی این

```
Private Sub CommandButton1_Click()
```

```
Dim number As Long
```

```
number = 10
```

```
number = number * number
```

```
Debug.Print "tavan do =" & number
```

```
End Sub
```

من می خوام این برنامه رو داخل یک ساب روتین دیگه بنویسم و از داخل این ساب روتین اون رو فراخوانی کنم کدش

میشه این

میشه این

```
Private Sub CommandButton1_Click()
```

```
Call newSubroutine
```

```
End Sub
```

```
Private Sub newSubroutine()
```

```
Dim number As Long
```

```
number = 10
```

```
number = number * number
```

```
Debug.Print "tavan do =" & number  
End Sub
```

می بینید چقدر ساده است

یک سابروتین دیگر به نام newSubroutine تعریف شده و در خط

```
Call newSubroutine
```

فراخوانی شده

حالا فکر کنید که می خوام عددی که قراره مربعش حساب بشه رو از بیرین به ساب روتین دوم بدم میشه این

```
Private Sub CommandButton1_Click()
```

```
Call newSubroutine(12)
```

```
End Sub
```

```
Private Sub newSubroutine(number As Long)
```

```
number = number * number
```

```
Debug.Print "tavan do =" & number
```

```
End Sub
```

در ساب دوم دیگه خط

```
Dim number As Long
```

رفته و به جاش ساب روتین تو تعریفش داخل پارانتز اونو گرفته

به این متغییر که داخل تعریف یک تابع یا ساب روتین میشه میگن پارامتر

یعنی ساب روتین دوم یک پارامتر به نام نامبر از نوع لاگ داره که توس ساب روتین اول موقع فراخوانی بهش پاس میشه

دقت کنید که توی کال ساب روتین این بار با پارانتز است و یک ورودی دارد چون دیگر ساب روتین مثل قبل بدون پارامتر نیست

```
Call newSubroutine(10)
```

حالا همین مثال رو می خوام با فانکشن یا تابع انجام بدم

گفتم که فرق تابع با ساب روتین اینه که تابع می تونه مقداری رو هم برگردونه ولی ساب روتین چیزی بر نمیگردونه یا اصلا void بر می گردونه

void

تو انگلیسی یعنی هیچی

خب برین همون رو با فانکش بنویسیم

```
Private Sub CommandButton1_Click()
```

```
Dim result As Long  
result = mehdiFuction(10)  
Debug.Print "result=" & result
```

```
End Sub
```

```
Private Function mehdiFuction(number As Long)
```

```
number = number * number  
mehdiFuction = number  
End Function
```

خیلی شبیه ساب روتینه ولی دو تا فرق داره

اسم mehdiFuction هست شما می تونستین هر چیزی بگذارین

یک فرقی سطر اخره

mehdiFuction = number

این یعنی اینکه متغیر number رو به عنوان خروجی تابع

mehdifunction

برگردون

فرق دیگر حالا تو فراخوانی یا کالش هم

result = mehdiFuction(10)

یعنی اینکه تابع فلان رو با ورودی ۱۰ کال کن و خروجی اش رو برین تو متغیر result

به همین سادگی این شد ساب روتین و تابع

دقت کنید که فراخوانی تابع دیگه

call

نداره

ما برای اینکه کد ساده تر بشه و خوانا تر بشه کد رو به صورت مجموعه از ساب روتین ها و فانکشن ها می نویسیم

و تا همدیگر رو کال کنن و هر کدوم کار خاص خودشون انجام بدن

این طوری میشه کد رو بهتر فهمید و هم راحتتر دیباگ کرد

همین طور میشه یک تیکه کد رو یک بار نوشت و چندبار استفاده کرد

مثل تابع هایی که من برای تقویم برا شما نوشتم میشه چندبار ازش استفاده کرد

حالا بیاین کمی عمیقتر حالا کد ها رو بررسی کنیم

برین سراغ تعریف متغیر

Visual Basic type	Common language runtime type structure	Nominal storage allocation	Value range
Boolean	Boolean	Depends on implementing platform	True or False
Byte	Byte	1 byte	0 through 255 (unsigned)
Char (single character)	Char	2 bytes	0 through 65535 (unsigned)
Date	DateTime	8 bytes	0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
Decimal	Decimal	16 bytes	0 through +/- 79,228,162,514,264,337,593,543,950,335 (+/- 7.9...E+28) [†] with no decimal point; 0 through +/- 7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/- 0.000000000000000000000000000001 (+/- 1E-28) [†]
Double (double-precision floating-point)	Double	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324 [†] for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 [†] for positive values
Integer	Int32	4 bytes	-2,147,483,648 through 2,147,483,647 (signed)
Long (long integer)	Int64	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18 [†]) (signed)
Object	Object (class)	4 bytes on 32-bit platform 8 bytes on 64-bit platform	Any type can be stored in a variable of type Object
SByte	SByte	1 byte	-128 through 127 (signed)

Short (short integer)	Int16	2 bytes	-32,768 through 32,767 (signed)
Single (single-precision floating-point)	Single	4 bytes	-3.4028235E+38 through -1.401298E-45 [†] for negative values; 1.401298E-45 through 3.4028235E+38 [†] for positive values
String (variable-length)	String (class)	Depends on implementing platform	0 to approximately 2 billion Unicode characters
UInteger	UInt32	4 bytes	0 through 4,294,967,295 (unsigned)
ULong	UInt64	8 bytes	0 through 18,446,744,073,709,551,615 (1.8...E+19 [†]) (unsigned)
User-Defined(structure)	(inherits from ValueType)	Depends on implementing platform	Each member of the structure has a range determined by its data type and independent of the ranges of the other members
UShort	UInt16	2 bytes	0 through 65,535 (unsigned)

یک عکس رو ببینید
توش انواع متغیر ها هست
معروف ترین شون
برای اعداد صحیح
integer
هست برای بازه

-2,147,483,648 through 2,147,483,647 (signed

و اعداد صحیح بزرگ هست

long

-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18 +) (signed)

بهتره همیشه برای اطمینان از لانگ استفاده کنید

برای متغیر منطقی یعنی درست و نادرست

Boolean

که توش فقط میشه این ها رو ریخت

True or False

برای اعداد اعشاری

Double

که بازه زیر رو پشتیبانی می کنه

-1.79769313486231570E+308 through -4.94065645841246544E-324 + for negative values;

4.94065645841246544E-324 through 1.79769313486231570E+308 + for positive values

برای رشته ها یا همون متن

String

که یونیکد یا همون فارسی هم میشه توش ریخت

0 to approximately 2 billion Unicode characters

اعداد اعشاری یک اعداد اعشاری دیگر هم داره

Single

که دقتش کمتر از دابله

خب حالا برین کمی برنامه بنویسیم

راستی یک تایپ

Date

هم هست برای تاریخ میلادی

بازه زیر رو پشتیبانی می کند

0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999

نکته جالب اینکه اسم date در واقع datetime هست و علاوه بر تاریخ زمان هم پشتیبانی می کنه

string

رو میشه با &به هم چسبونند مثل مثال های بالا

Private Sub Form_Load()

Dim FirstName, LastName As String

Dim FullName As String


```
FirstName = "Francis "  
LastName = "Pottelson"  
FullName = FirstName & LastName  
Text0 = FullName  
End Sub
```

این ساب نام و نام خانوادگی رو به هم می چسبونه
جمع و منها و ضرب هم که مشخص
فقط تقسیم یک نکته ای داره
/میشه تقسیم اعشاری و \ میشه تقسیم صحیح
مثلا کد زیر رو ببینید

```
Private Sub CommandButton1_Click()
```

```
Dim number  
Dim other  
  
number = 3  
other = 7  
Debug.Print other + number  
Debug.Print other - number  
Debug.Print other * number  
Debug.Print other / number  
Debug.Print other \ number
```

```
End Sub
```

خروجی اش میشه این

10
4
21
2.33333333333333
2

راستی اگر immediate view پر شد می تونید کلش رو پاک کنید
انتخاب کنید بعد delete رو رو کیبرد بزنید یا

ctrl+A

بعد

delete

یعنی کنترل وA

برای انتخاب همه بعد دپلیت

یک چیز دیگه هم هست برای محاسبه باقی مانده

که یک اپراتورش همیشه

mod

12 mod 5

همیشه باقی مانده ۱۲ بر ۵ یا همون ۲

برای توان رساندن هم همیشه این

y^x

که یعنی

y^x

برای علامت ^ تو انگلیسی بهش می گن هت یا کلا شیفت رو بگیرد و ۶ رو بزدین تا بیاد

پس کد زیر

```
Private Sub CommandButton1_Click()
```

```
Dim number
```

```
Dim other
```

```
number = 3
```

```
other = 7
```

```
Debug.Print other Mod number
```

```
Debug.Print other ^ number
```

```
End Sub
```

خروجی اش همیشه

1

343

راستی کامنت که می دونید چیه

بخش هایی از کد که تنها برای توضیح هستند و نباید اجرا بشن

تو وی بی ا کافیه که single quotation بگذارید بعد از این هر چیش بیاد کامنته

```
Private Sub CommandButton1_Click()
```

```
Dim number
```

```
Dim other
```

```
'this is comment
```

```
number = 3
```

```
other = 7
```

```
Debug.Print other Mod number 'this is comment too
```

```
Debug.Print other ^ number
```

End Sub

تو ادیتور هم سبز می کنه

'this is comment

'this is comment too

این ها کامنتن

تو اولویت اپراتور ها همیشه ضرب و تقسیم به جمع و تفریق اولویت داره

$8 + 3 * 5$

بر آورد میشه

$8 + (3 * 5)$

که حاصلش ۲۳ است

حالا مثال

این تابع شاخص توده بدنی رو حساب می کنه

Private Function BMI(Height As Single, weight As Single) As Double

BMI = weight / Height ^ 2

End Function

راستی یک نکته ای

می بینید بعد از پارانتز های تعریف این تابع یک عبارت

as double

هست این یعنی تابع یک دابل بر می گردونه

پس تعریف تابه میشه این شکلی

private functionname(parameter as [parameter_type]) as [return_value_type]

همش توی یک سطره اینجا می بره تو دو تا سطر

یا این تابع ارزش آینده پول رو بر اساس تعداد سال و نرخ سود سالانه محاسبه می کنه

Private Function arzesheAyande(arzeshefeli As Single, nerkheSood As Single, tedadSal As Integer) As Double

arzesheAyande = arzeshefeli * (1 + nerkheSood / 100) ^ tedadSal

End Function

حالا بریم کار با استرینگ رو ببینیم

این کد رو ببینید

Private Sub CommandButton1_Click()

Dim strValue As String

```
Dim iLen As Integer
```

```
strValue = "eide shoma mobarak"
```

```
iLen = Len(strValue)
```

```
MsgBox "The length of "" & strValue & "" Is " & _  
CStr(iLen) & " characters."
```

```
End Sub
```

توش تابع len

یک استرینگ می گیره و طولشو بر می گردونه

تابع CStr یک integer می گیره و استرینگ تبدیل می کنه

هر چند که تبدیل عدد به استرینگ با همون & هم می شد

این تابع ها کجا تعریف شدن ما که تعریفشون نکردیم

یاں حا توابع buildin یا از پیش آماده هستند

هر زبانی از این تابع ها داره که به درد می خوره

لیست فانکشن های آماده اکسس اینجاست

<http://www.techonthenet.com/access/functions/>
Techonthenet

MS Access: Functions - Listed by Category

For easy reference, we have provided a list of all Microsoft Access functions. The list of Access functions is sorted into the type of function based on categories such as string, numeric/mathematical, date/time, advanced/logical, domain aggregate, data type conversion, group by, and file/directory functions.

تو هر کدوم توضیحش هست مثلن برای استرینگ این ها معروفاشه

مثلا کد زیر یک استرینگ رو از تکرار چنتا دیگه میسازه

یک کاراکتر رو ببخشید

```
Private Sub CommandButton1_Click()
```

```
Dim strValue As String
```

```
strValue = String(18, "f")
```

```
Debug.Print strValue
```

```
End Sub
```

میشه

ffffffffffffffffffff

رضا

آقا این دستور پرینت اجرا نمیشه اصلا immediate view را پیدانکریم

تو بخش اکسس کنترل و زدین؟

تو تب

view

برین

جایی که کد vba

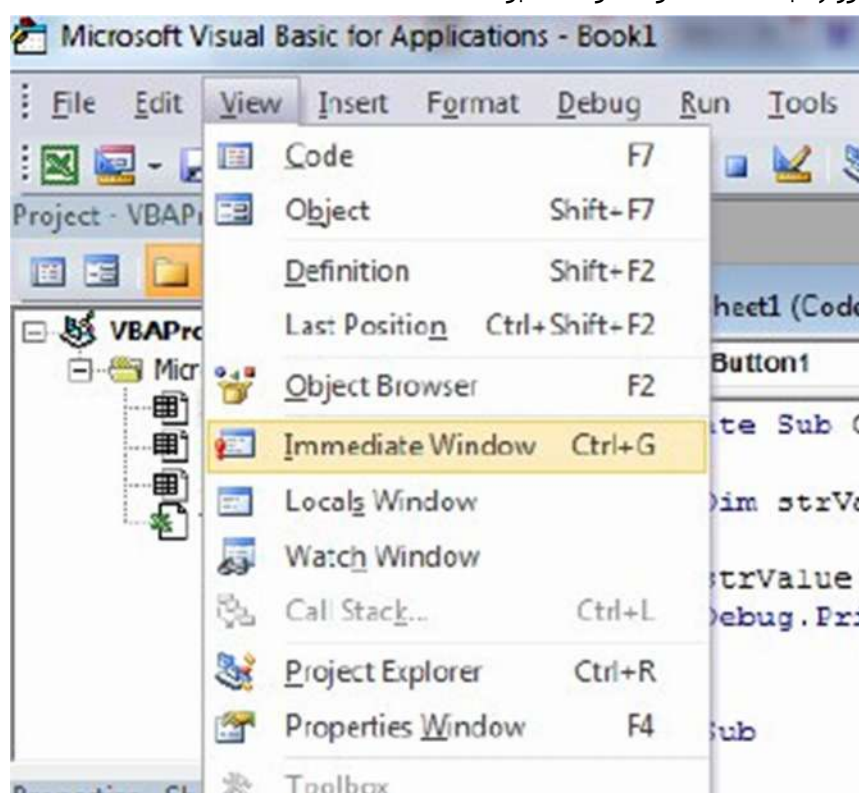
می نویسین

بزینین روی

immediate view

پیدا کردین؟

بیخشید منظورم تب view همون منو view بود



دیدین

تو اکسل هست به نظرم اکسس هم باید همون باشه

پیدا شد؟

شد؟

حالا مثل اینکه من اشتباه گفتم

msg box

هم چندتا پشت سر هم اجرا میشه

حالا اینم یک راهه
فقط لطفن برین از اول بخونین
از سلام
خصوصا اون جاها برای دیباگ خیلی به دردتون می خوره بعدن
اینکه تابع چیه ساب روتین چیه
انواع داده چیه
همه رو نوشتن
اونا رو اول ببیند خیلی خوبه
حالا من می رم جلو بعدن بیاین انجا رو بخونین
خب این تابع یک استرینگ رو بر عکس می کنه

```
Private Sub CommandButton1_Click()
```

```
Dim strValue As String  
Dim strRev As String
```

```
strValue = "salam salam satta salam"  
strRev = StrReverse(strValue)
```

```
MsgBox strValue  
MsgBox strRev
```

```
End Sub
```

یک تابع به درد بخور دیگه هم trim هست
این تابع میاد از اول و اخر هر استرینگ space ها رو بر میداره
اینو اجرا کنین می فهمین چی کار می کنه

```
Private Sub CommandButton1_Click()
```

```
Dim strValue As String  
Dim strRev As String
```

```
strValue = " salam "  
strRev = Trim(strValue)
```

```
MsgBox "|" & strValue & "|"  
MsgBox "|" & strRev & "|"
```

End Sub

دو تا تابع هم هست که استرینگ انگلیسی رو حرفشو بزرگ و کوچک می کنه

Private Sub CommandButton1_Click()

Dim strValue As String

Dim strRev1 As String

Dim strRev2 As String

strValue = " salam Jahan AA "

strRev1 = UCase(strValue)

strRev2 = LCase(strValue)

MsgBox strValue

MsgBox strRev1

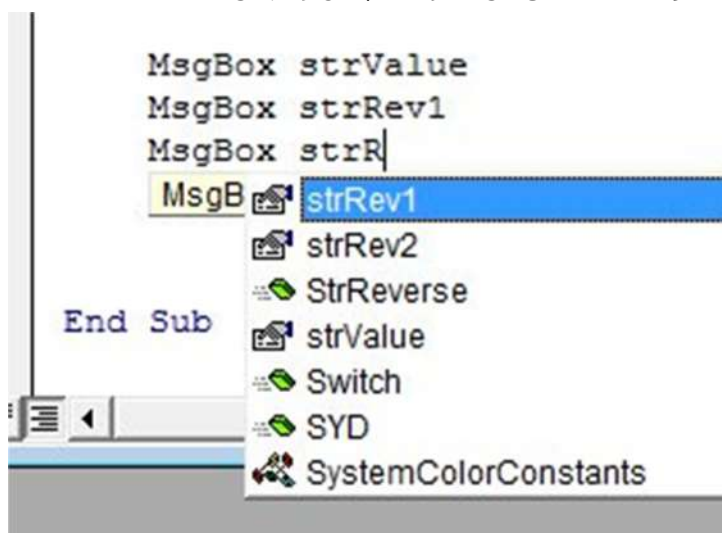
MsgBox strRev2

End Sub

راستی یک نکته خیلی به درد بخور

کلن تو محیط برنامه نویسی کافیه یکه یک متغیر رو اسمشو نصفه بنویسید و بعد ctrl+space بزنین تا خودش بهتون پیشنهاد بده

الان وسط مثال قبل من کنترل اسپیس زدم این امد



راستی می گم دفعه پیش کنترل G می زدین اون ویو نمی امد شاید کنترل و G رو همزمان نمی زدین

الان هم باید کنترل و اسپیس رو همزمان بزنین

همیشه عادت کنید تند تند بزنیند هم پیشنهادات باعث میشه سرعت کد زدنتون بیشتر بشه و هم اینکه لازم نیست اسم تابع ها رو حفظ کنید اول یادتون باشه می تونید پیدا کنید

تازه چیزهایی جدید رو هم می بینید و یاد می گیرید

یک چیز به درد بخور دیگه ساب استرینگه اینکه یک بخشی از یک استرینگ رو در بیارین و تو یک استرینگ دیگه بریزین

این ها رو ببینید

```
Private Sub CommandButton1_Click()
```

```
Dim strValue As String
```

```
Dim strLeft As String
```

```
strValue = "salam salam setare"
```

```
strLeft = Left(strValue, 10)
```

```
Debug.Print strValue
```

```
Debug.Print strLeft
```

```
Debug.Print Right(strValue, 10)
```

```
End Sub
```

خروجی اش میشه این

salam salam setare

salam sala

lam setare

از وسط هم میشه برید

```
Private Sub CommandButton1_Click()
```

```
Dim strValue As String
```

```
Dim strLeft As String
```

```
strValue = "salam salam setare"
```

```
Debug.Print strValue
```

```
Debug.Print Mid(strValue, 15)
```

```
Debug.Print Mid(strValue, 15, 3)
```


End Sub

Mid(strValue, 15)

این از کاراکتر ۱۵ به بعد رو می‌ده و این

Mid(strValue, 15, 3)

از کاراکتر ۱۵ ام سه تا بعدی اش یعنی ۱۵و۱۶و۱۷ ام رو می‌ده

خروجی میشه

salam salam setare

tare

tar

میشه یک استرینگ رو داخل یک استرینگ دیگه گشت

خیلی به درد می‌خوره

Private Sub CommandButton1_Click()

Dim strValue As String

Dim iPos As Integer

strValue = "Republic of South Africa"

iPos = InStr(1, strValue, "South")

Debug.Print "In "" & strValue & """, " & " South can be found at " & CStr(iPos)

End Sub

تابع

InStr([start,]string1, string2[, compare])

خروجی میشه این

In "Republic of South Africa", South can be found at 13

تو فرمت تابع که بالا نوشتم داخل کروشه یعنی اینکه این پارامتر اختیاریه

InStrRev()

هم از آخر به اول می‌گرده

اینم برای جایگزینی داخل یک استرینگ یا رشته

Replace(expression, find, replace[, start[, count[, compare]]])

رشته منظورم همون استرینگه

خب حالا برین سراغ شرط و حلقه

اول متغیر منطقی یا همون بولین

گفتم که متغیر بولین توش می‌تونه درست یا نادرست باشه همین

مثالان این برنامه رو اجرا کنید

```
Private Sub CommandButton1_Click()

    Dim IsFullTime As Boolean

    Debug.Print "Is Employee Full Time= " & IsFullTime

    IsFullTime = True
    Debug.Print "Is Employee Full Time= " & IsFullTime

End Sub
```

خروجی همیشه این

Is Employee Full Time= False

Is Employee Full Time= True

به همین سادگی

دقت کنید که متغیر منطقی وقتی ساخته میشه اول توش False پر میشه

بعدن True شده

برای بررسی مساوی بودن دو تا چیز از=

برای عدم تساوی <>

برای بزرگتر <

برای کوچکتر >

```
Private Sub CommandButton1_Click()
```

```
    Dim result As Boolean
```

```
    Dim num1
```

```
    Dim num2
```

```
    num1 = 12
```

```
    num2 = 25
```

```
    result = (num1 = num2)
```

```
    Debug.Print result
```

```
    result = (num1 <> num2)
```

```
    Debug.Print result
```

```
    result = (num1 > num2)
```

```
    Debug.Print result
```

```
    result = (num1 < num2)
```

```
    Debug.Print result
```

```
End Sub
```

خروجی همیشه

False

True

False

True

کوچکتر مساوی $\text{Value1} \leq \text{Value2}$

بزرگتر یا مساوی $\text{Value1} \geq \text{Value2}$

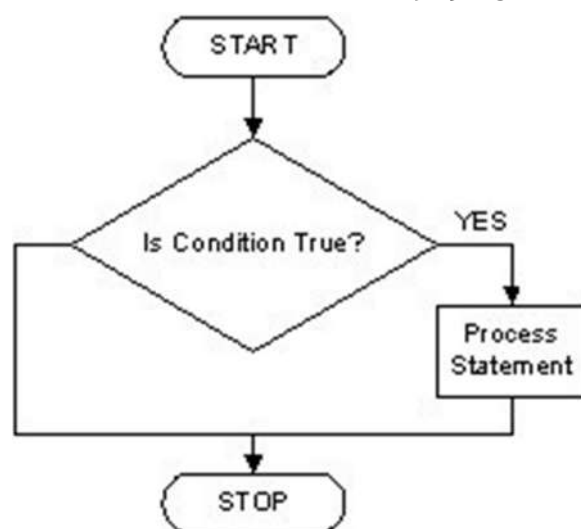
هم داریم که بدیهه

حالا می‌خوایم بگیم اگر این شرط برقرار بود این کارو بکن

همیشه این

If Condition Then Statement

یعنی این فلوچارت



این تابع یک شرط داره

```
Private Function myfunc(num As Long)
```

```
If num > 22 Then MsgBox "The number is greater than 22"
```

```
End Function
```

اگر داخل if بیش از یک سطر باشه باید end if داشته باشه

```
If Condition Then
```

```
Statement
```

```
End If
```

قبله همیشه این

```
Private Function myfunc(num As Long)
```

```
If num > 22 Then
    MsgBox "The number is greater than 22"
End If
```

End Function

چند سطر هم حالا همیشه داخل if nhaz داشت

```
If Condition Then
    Statement1
    Statement2
    Statement_n
End If
```

مثلا

```
If dAge >= 21 Then
    Caption = "Membership Category: Adult"
    MsgBox "The number is greater than 22."
    MsgBox "The member will be considered an adult."
End If
```

یک حال if else هم هست
که else یعنی موقع که که شرط if برقرار نشد

```
If ConditionIsTrue Then
    Statement1
Else
    Statement2
End If
```

مثلا

```
If Gender = "Male" Then
    Detail.BackColor = 16772055
Else
    Detail.BackColor = 13034239
End If
```

یعنی اگر شرط برقرار شد بالایی و گرنه داخل else
دقت کنید که کدهایی مثل این که کپی کردم کامل نیستند برای آشنایی کپی کردم
عینش رو کپی کنید کار نمی کنه مثلا برای کد بالا باید متغیر

gender

تعریف بشه

یک حالت هم هست که چندتا شرط باشه

مثلا

```
If Condition1 Then
    Statement1
ElseIf Condition2 Then
    Statement2
ElseIf Condition_n Then
    Statement_n
End If
```

این رو مثلا می زدم یک تابع می خوایم که یک عدد بین ۱ تا ۷ بگیره و روز هفته رو چاپ کنه

```
Private Function myfunc(num As Long)
```

```
If num = 0 Then
    MsgBox "shange"
elif num = 1
    msbox "yekshanbe"
elif num = 2
    msbox "doshanbe"
```

```
End If
```

```
End Function
```

حالا من کامل ننوشتم

میشه else اخر هم داشت برای حالتی که هیچ کدوم از if ها برقرار نشه

```
If Condition1 Then
    Statement1
ElseIf Condition2 Then
    Statement2
ElseIf Condition3 Then
    Statement3
Else
    Statement_False
End If
```

میشه برای یک شرط از نقیض استفاده کرد یعنی برعکسش

برای این کار باید not نوشت

مثل

```
If Not firstname = "mehdi" Then
    MsgBox "other"
```

```
End If
```

این یعنی اگر متغیرfirstname

مساوی مهدی نبود

معادل اینه

```
If firstname <> "mehdi" Then  
    MsgBox "other"
```

```
End If
```

حالا تنها چیزی که باقی موند حلقه است

که اونم ساده است

حلقه ها برای تکرار استفاده میشن که با وجودی که چند مدل دارن ولی عملا خیلی ساده اند

اولین حلقه فور هست

فور ساده این شکلیه

```
For Counter = Start To End  
    Statement(s)  
Next
```

برای مثال

```
Private Sub CommandButton1_Click()  
    Dim counter As Integer
```

```
    For counter = 0 To 4  
        Debug.Print counter  
    Next
```

```
End Sub
```

خروجی اش میشه این

0
1
2
3
4

یعنی یک تکرار رو یک شمارنده از صفر تا ۴ دقت کنید که خود صفر و چهار جزش است

این حلقه هر گام شمارنده یکی افزایش پیدا می کنه

برای افزایش چندتایی میشه این طوری

```
For Counter = Start To End Step Increment  
    Statement(s)  
Next Counter
```

مثال

```

Private Sub CommandButton1_Click()
Dim counter As Integer

For counter = 0 To 10 Step 3
    Debug.Print counter
Next

End Sub

```

خروجی همیشه

0
3
6
9

یک حلقه دیگر هم است که می‌گه تا زمانی که یک شرط برقراره یک کاری رو تکرار کن

```

Do While Condition
    Statement(s)
Loop

```

مثال

```

Private Sub CommandButton1_Click()
Dim counter As Integer
Do While counter < 20
    Debug.Print counter
    counter = counter + 6
Loop

End Sub

```

خروجی همیشه

0
6
12
18

یک مدل خیلی شبیه به این می‌گه یک کاری رو بکن تا زمانی که یک شرط برقرار بشه

```

Do
    Statement(s)
Loop While Condition

```

مثال

```

Private Sub CommandButton1_Click()
Dim counter As Integer
Do

```

```

Debug.Print counter
counter = counter + 6
Loop While counter < 20

```

End Sub

خروجی همیشه

0
6
12
18

فرقش با قبلی اینه که اگر از اول شروع به کار در اولی شرط برقرار نشه یک بار هم داخل حلقه اجرا نمیشه ولی در دومی اول داخل یک بار اجرا میشه بعد شرط چک میشه

فارسی اش میشه اینکه اولی یعنی

تا زمانی که این شرط برقراره این کارو بکن

دومی میشه این کار رو بکن تا زمانی که این شرط برقرار هست

دومی یک بار حداقل اجرا میشه

تموم شد

تقریباً میشه گفت که وی بی ای همین بود

الان تقریباً همش رو مرور کردیم

خب این ها هم چندتا مثال برای اینکه چطور تو وی بی ای با جدول دیتا بیستون کار کنید

```

private sub showTableData

```

```

dim db as dao.database

```

```

dim rs as dao.recordset

```

```

set db = currentDb

```

```

set rs = db.OpenRecordSet("myTable") 'myTable is a MS-Access table created previously

```

```

'populate the table

```

```

rs.movelast

```

```

rs.movefirst

```

```

do while not rs.EOF

```

```

    debug.print(rs!myField) 'myField is a field name in table myTable

```

```

    rs.movenext 'press Ctrl+G to see debuG window beneath

```

```

loop

```

```

msgbox("End of Table")

```

```

end sub

```


این کد دایره از دیتا بیس جدول myTable رو سطر به سطر می خونه
فقط دقت کنید که از هر سطر دایره ستونی که اسمش
myField

هست رو چاپ می کنه

برای اینکه بخواهیم یک کوئری رو دیتا بیس بزنیم یعنی مثلاً همه رکورد ها رو بر نداریم بازم می تونیم
دقت کنید که منظور من از کوئری یک sql روی دیتا بیس نه الزاماً کوئری خود اکسس
این کد رو ببینید

```
private sub showQueryData
```

```
dim db as dao.database  
dim rs as dao.recordset  
dim sqlStr as string
```

```
sqlStr = "SELECT * FROM customers as c WHERE c.country='Brazil'"
```

```
set db = currentDb  
set rs = db.openRecordset(sqlStr)
```

```
rs.movefirst
```

```
do while not rs.EOF  
    debug.print("cust ID: " & rs!id & " cust name: " & rs!name)  
    rs.movenext  
loop
```

```
msgbox("End of customers from Brazil")
```

```
end sub
```

یک sql

به صورت زیر

```
SELECT * FROM customers as c WHERE c.country='Brazil'
```

انجام شده

اگر کوئری بلد باشیم این یعنی اینکه از جدول customers هر جایی که ستون country برابر برزیل هست رکورد ها
رو انتخاب کن

بعد تو سطر

```
debug.print("cust ID: " & rs!id & " cust name: " & rs!name)
```

ستون های id , name چاپ شدن

بیاین خط به خط توضیح بدم

```
dim db as dao.database
```

یعنی یک متغیر از نوع dao.database تعریف کن

این نوع داخل خودش تعریف شده و یک دیتا بیس رو نشون میده

```
dim rs as dao.recordset
```

یک متغیر از نوع ریکورد ست تعریف کن

ریکورد ست یک چیزی شبیه به همون جدول که میشه تو کد ازش استفاده کرد

کویری که قبلن توضیح دادم رو تو ی یک متغیر استرینگ تعریف کن

```
dim sqlStr as string
```

```
sqlStr = "SELECT * FROM customers as c WHERE c.country='Brazil'"
```

```
set db = currentDb
```

یعنی متغیر db رو بگذار دیتابیس جاری

در واقع currentDb دیتا بیس جاری که توش داره برنامه نوشته می کنه رو اشاره می کنه

```
set rs = db.openRecordset(sqlStr)
```

حالا خروجی کویری که نوشتیم رو بگذار داخل ریکورد ست

یکورد سرت رو فکر کنید یک جدول به یک اشاره گر داره که به یک سطر اشاره می کنه

کافیه ما سطر به سطر بخونیم و اشاره گر رو جلو ببریم

فکر کنید این جدول

ProductID	ProductName	Quantity	UnitPrice
P001	Coca Cola	12	\$0.50
P002	Pepsi	6	\$0.30
P003	7Up	24	\$0.40
P004	Season	6	\$0.50

اشارگر اول رو سطر ۱- هست



ProductID	ProductName	Quantity	UnitPrice
P001	Coca Cola	12	\$0.50
P002	Pepsy	6	\$0.30
P003	7Up	24	\$0.40
P004	Season	6	\$0.50

بعد با دستور rs.movefirst میره عنصر اول



ProductID	ProductName	Quantity	UnitPrice
P001	Coca Cola	12	\$0.50
P002	Pepsy	6	\$0.30
P003	7Up	24	\$0.40
P004	Season	6	\$0.50

بعد با while قدم به قدم توش پیش میریم تا زمانی که

do while not rs.EOF

rs.EOF

به آخر جدول اشاره می کنه

یعنی تا زمانی که تا آخر جدول نرفتی هر بار گام به گام برو جلو

rs.movenext

با همین دستور

حالا دخل وایل میشه کاری که با هر رکورد می خوایم انجام بدیم

برای خواندن هر ستون اسمش رو بعد از ! می اریم

rs!name

میشه ستون name داخل رکورد

بهتر آخر کار با رکورد ست با دستور

```
rs.Close 'Close the recordset
Set rs = Nothing 'Clean up
```

اونو ببینید و متغییر رو آزاد کنیم

این برای خوندن بود

میشه موقع کار با هر رکورد اونو تغییر هم داد

این کد چنین مثالی توشه

ولی فعلن شما رو همون خوندن تمرکز کنید

```
Dim rs As DAO.Recordset
Set rs = CurrentDb.OpenRecordset("SELECT * FROM Contacts")
```

```
'Check to see if the recordset actually contains rows
If Not (rs.EOF And rs.BOF) Then
    rs.MoveFirst 'Unnecessary in this case, but still a good habit
    Do Until rs.EOF = True
        'Perform an edit
        rs.Edit
        rs!VendorYN = True
        rs("VendorYN") = True 'The other way to refer to a field
        rs.Update

        'Save contact name into a variable
        sContactName = rs!FirstName & " " & rs!LastName

        'Move to the next record. Don't ever forget to do this.
        rs.MoveNext
    Loop
Else
    MsgBox "There are no records in the recordset."
End If
```

```
MsgBox "Finished looping through records."
```

```
rs.Close 'Close the recordset
Set rs = Nothing 'Clean up
```

فرق اصلی این تیکه است

```
rs.Edit
    rs!VendorYN = True
    rs("VendorYN") = True 'The other way to refer to a field
    rs.Update

rs.Edit
```

برای باز کردن برای ویرایشه و rs.Update برای اعمال تغییرات است