

برنامه نویسی با اکسل









, hj

تذكر

به موجب ماده قانون حمایت از حقوق مولفان و مصنفان مصوب ۱۳۴۸/۱۰/۱۱ کلیه حقوق این کتاب برای مهندسی با اکسل (mohandesibaexcel.com) محفوظ میباشد و هیچ شخص حقیقی و حقوقی حق چاپ و برداشت تمام یا قسمتی از اثر را به صورت چاپ ، فتوکپی ، جزوه و حتی دستنویس ندارد و متخلفین به موجب این قانون تحت پیگرد قرار می گیرند.

برنامەنويسى با اكسل

مبانی

مهراج محمدي هندواري

تابستان ۹۸

ويرايش اول آذر ۹۹

فهرست

ندمه	io
پالبی در رابطه با مطالعه کتاب ۲	lo
متویات کتاب	6
سل اول	فر
بط ماکرو با اکسل ۹	ò
٩٩	مة
۱-۱. اضافه کردن زبانه Developer به ریبون اکسل۹	
۲-۱. مراحل ضبط ماکرو با اکسل	
۱–۳. مثالی از نحوه ضبط ماکرو با اکسل۱۷	
۱۹ VBA ورود به دنیای VBA	
۱-۴-۱. مواردی که ابزار ضبط ماکرو قادر به انجام آن نیست۱۹	
۲-۴-۱. دسترسی به محیط ویرایشگر ویژوال بیسیک	
۲-۴-۲. آناتومی محیط ویرایشگر ویژوال بیسیک۲۱	
۲-۴-۱. محل قرارگیری کدهای VBA ماکروی ایجاد شده۳۰	
۵-۱. بررسی کدهای ماکروی ضبط شده۳۲	
۱-۵-۱. مبانی کدهای VBA ماکروی ایجاد شده	
۶-۱. چند نکته در خصوص نحوه یادگیری برنامه نویسی با اکسل٤٢	
۱–۲. کدهای آماده VBA را کجا قرار دهیم	

٤٣	۱-۷-۱. تفاوت پنجره کد ماژول و پنجره کد شیء
٤٣	۱-۷-۱ ایجاد ماژول و اضافه کردن کد به پنجره کد آن
٤٧	۱-۷-۳. اضافه کردن کد به پنجره کد شیء
٤٧	۱–۸. ذخیره فایل اکسل حاوی ماکرو
٤٩	فصل دوم
٤٩	امنیت ماکرو ها
٥٠	مقدمه
٥٠	۲-۱. چرا ماکروهای اکسل به صورت پیش فرض فعال نیست؟
٥١	۲-۲. چگونه می توان تنظیمات عمومی امنیت ماکرو را تغییر داد
٥٢	۲-۲-۱. مرحله۱: دسترسی به Trust Center اکسل
٥٤	۲-۲-۲. مرحله۲: انتخاب یکی از گزینه های فعال یا غیرفعال کردن ماکروها
זר	۲-۳. فعال سازی ماکرو برای برخی از فایل های اکسل
٦٣	۲–۳–۱. مراحل ایجاد پوشه امن
٦٨	۲-۳-۲. مواردی که هنگام استفاده از پوشه های امن نباید انجام دهید
٦٨	۲-۴. فعال کردن ماکرو برای یک بار
۷۱	فصل سوم
۷۲	۱۰ اصطلاح مهم برنامه نویسی با اکسل
۷۲	مقدمه
۷۳	۷BA .۱-۳ چیست؟
٧٤	۲-۳. ماکرو و تفاوت آن با VBA
۷٥	۳-۳. کد VBA. که ۳.
٧٦	۳-۴. ماژول (Module)
vv	۵-۳. فرآیند (Procedure) ، زیرمجموعه (Subroutine) و رویه (Routine)
vv	۵–۵–۳. فرآیند Function
٧٨	۲-۵-۳. فرآیند Sub.

٧٩	۳-۵-۳. ساختار / ترکیب یک فرآیند
٧٩	۳-۵-۴. ساختار / ترکیب یک فرآیندSub
۸۱	۵-۵-۳. نام گذاری یک فرآیند VBA
Λ٢	۳-۵-۶. تعیین دامنه فرآیند VBA
۸٦	۳-۵-۷. اجرا و فراخوانی فرآیندهایSub
۱۲۱	۶-۳. ساختار اصلی دستورهایVBA
١٢٢	۳-۷ شیء (Object)
١٢٣	۲-۲-۳. مدل اشیاءVBA
١٢٤	۳-۲-۲. چرا مدل اشیاء VBA اهمیت دارد
١٢٤	۳-۷-۳. مدل اشیاء (سلسله مراتب اشیاء) VBA اکسل
١٢٧	۴-۲-۳. مجموعه اشياء (Object Collection)
١٢٧	۳-۷-۵. مجموعه ها و مدل اشیاء اکسل
۱۲۸	۳-۷-۶. ارجاع به یک شیء
ודז	۳-۷-۷ دستکاری اشیاء
۱۳۷	۸-۳. ویژگی اشیاء (Property)
۱۳۹	۳-۸-۱. کار با ویژگی اشیاء
١٤٠	۳-۸-۳. ساختار ویژگی اشیاءVBA
١٤٩	۳-۸-۳. دسترسی به ویژگی های یک شیء
100	۳-۹. شيوه عمل (Method)
۲٥١	۳-۹-۱. کار با شیوه های عمل اشیاء
۱۵۸	۳-۹-۲. دسترسی به شیوه های عمل یک شیء
אדו	۳-۱۰. مولفه ها (Parameters)
סדו	۳-۱۰-۱۰. نحوه کار با مولفه های شیوه عمل شیء در VBA .
۱۷۰	۳-۱۱. ساختار دستورات VBA (به روز رسانی)
ىء	۳-۱۱-۱. ساختار دستورات VBA با مولفه های شیوه عمل ش
۱۷۲	۲-۱۱-۳. ساختار دستورات VBA با مولفه های ویژگی شیء

١٧٣	۱۲-۳. کلاس ها (Classes)
۱۷۵	فصل چهارم
۱۷٥	نوع داده های VBA
างฯ	مقدمه
างา	۲-۴. نوع داده های VBA
۱۷۸	۱-۱-۴. نوع داده Variant
אז	۲-۱-۴. نوع داده Byte
ነለግ	۳-۱-۴. نوع دادهBoolean
אז	۴–۱–۴. نوع داده Currency
١٨٧	۵-۱-۴ نوع داده Date
۱۸۸	۲-۱-۴. نوع داده Decimal
١٨٩	۲-۱-۴. نوع داده Double
١٨٩	۸-۱-۴. نوع داده Integer
۱۹۰	۹-۱-۴. نوع داده Long
۱۹۰	۱۰-۱-۴. نوع داده Object
۱۹۰	۱۱-۱-۴. نوع داده Single
۱۹۰	۲-۱-۴. نوع داده String
191	۴–۱–۱۳. سایر نوع داده های VBA
197	۴-۲. انتخاب نوع داده مناسب
۱۹٤	۲-۴-۱. مشخص کردن نوع داده فرعی متغیرهای Variant
۱۹۹ TypeNar	ne -۲-۲-۴. مشخص کردن نوع داده فرعی متغیر با استفاده از تابع
۲۰۰ varTy	pe ج-۲-۴. مشخص کردن نوع داده فرعی متغیر با استفاده از تابع
۲۰۳	فصل پنجم
۲۰۳	متغیرها در VBA
۲۰٤	مقدمه

۲۰٤	۵-۱. چرا از متغیر استفاده می کنیم؟
۲۰۵	۵-۲. چرا باید متغیر ها را به صراحت اعلان و تعریف کرد؟
۲۰٦	۵-۲-۵. استفاده از ویژگی AutoComplete
۲۰۷	۵-۲-۲-۲. آشکار شدن خطا های احتمالی
۲۰۸	۵-۲-۳. بهبود خوانایی کد
۲۱۰	۵-۳. معایب اعلان صریح متغیر های VBA
۲۱۰	۵-۴. یادآوری VBA برای اعلان متغیرها
۲۱۵	۵-۵. اعلان متغیر در VBA
רוז	۵-۵-۱. اعلان متغیرهای VBA با استفاده از کلیدواژه Dim
۲۱Λ	۵-۵-۲. اعلان متغیرهای VBA با استفاده از کلیدواژه tatic
Private و Private و Private	۵-۵-۳. اعلان متغیرهای VBA با استفاده از کلیدواژه های ۲
۲۲۰	۵-۶. دامنه متغیرهای VBA
۲۲۰	۵-۶-۱. متغیرهای سطح فرآیند (محلی)
٢٢٤	۵-۶-۲. متغیرهای سطح یک ماژول (خصوصی)
۲۲۸	۵-۶-۳. متغیرهای سطح فایل (عمومی)
۲۳۲	۵-۷. طول عمر متغیر
۲۳۳	۵-۸. نام گذاری متغیرهای VBA
۲۳٦	۵-۹. اختصاص ابزار بیان به متغیر VBA
ن اعلان آنها)	۵-۱۰. اعلان متغیرهای VBA به طور ضمنی (ایجاد متغیرها بدو
٢٤٢	۵-۱۱. متغیرهای شیء
۲٤٣	۵-۱۱-۱ اعلان متغیرهای شیء
٢٤٣	۵–۱۱–۲. تخصیص مقدار به متغیر شیء
٢٤٥	۵–۱۲. ثابت ها در VBA
٢٤٦	۵–۱۳. عملگرها
٢٤٦	۵-۱۳-۱ عملگرهای ریاضی
٢٤٦	۵–۱۳–۲. عملگرهای مقایسه

٢٤٧	۵–۱۳–۳. عملگرهای منطقی
٢٤٨	۵–۱۳–۴. عملگرهای اتصال
٢٤٨	۵-۱۳-۵. عملگر تخصیص
٢٤٩	فصل ششم
٢٤٩	آرایه ها در VBA
۲٥٠	مقدمه
۲٥٠	۶−۱. آرایه VBAچیست؟
۲۵۱	۶-۲ اعلان یک آرایه VBA
۲٥٣	۶-۲-۱. اندازه آرایه ، نوع داده ها و الزامات حافظه
۲٥٤	۶-۳. آرایه یک بعدی و چند بعدی
٢٥٦	۴-۶. آرایه های ثابت و پویا
٢٥٦	۶-۴-۱. اعلان یک آرایه یک بعدی ثابت
۲٥٨	۶-۴-۲. اعلان یک آرایه چند بعدی ثابت
709	۶-۴-۳. اعلان آرایه های پویا
٢٦٤	۶-۵. کران پایین آرایه و دستور Option Base
דדז	۶-۶ ارجاع به عناصر آرایه
דזז	۶-۶-۱. ارجاع به عنصری در آرایه یک بعدی
۲٦٨	۶-۶-۲. ارجاع به یک عنصر در آرایه چند بعدی
۲۷۱	۶-۷. چگونه داده های یک آرایه (یا خود آرایه) را پاک کنیم
۲۷۳	فصل هفتم
٢٧٣	روش های رایج ارجاع به شیء Range
τνε	مقدمه
τνε	۱-۷. شیء Range چیست
۲۷٥	۲-۷. ارجاع به شیء Range
۲۷٥	۲-۷. مقدمه ای بر ارجاع به شیء Range

777.	۲-۳-۲. مقدمه ای بر آدرس کامل شیء Range در مدل اشیاء اکسل
777.	۷-۳-۲. ارجاع به یک شیء از یک مجموعه
۲VV.	۷-۳-۳. ساده سازی آدرس کامل ارجاع به اشیاء
۲۷۸.	۲-۲. ارجاع به شیء Range با استفاده از ویژگی Range
۲۷۹.	۱-۴-۷. ترکیب ویژگی Range
۲۸۱.	۲-۴-۷. ویژگیWorksheet ، Range و ارجاع به یک سلول واحد برگه
۲۸۲.	۳-۴-۷. ویژگیRange ، Range و ارجاع به یک سلول واحد مربوط به یک محدوده
۲۸۳.	۴-۴-۷. ویژگیWorksheet ، Range و ارجاع به یک ستون یا سطر کامل
۲۸۳.	۵-۴-۷. ویژگی Worksheet ، Range و ارجاع به سلول های مجاور
٢٨٤.	۴-۴-۲. ویژگی Worksheet.Range و ارجاع به سلول های غیر مجاور
٢٨٥.	۷-۴-۷. ویژگی Worksheet ، Range و ارجاع به تقاطع دو محدوده
۲۸٦.	۸-۴-۷. ویژگی Worksheet . Range و ارجاع به محدوده های نامگذاری شده
ΥΛΥ.	۹-۴-۷ ویژگی Worksheet ، Range و ارجاع به سلول های ادغام شده
۲۸۸.	۲-۴-۷. میانبر ارجاع به شیء Range با استفاده از ویژگیRange
۲۹۰.	۵-۷. ارجاع به شیء Range با استفاده از ویژگی Cells
791.	۲-۵-۲. ترکیب ویژگی Cells
T9T.	۲-۵-۲ ویژگی Worksheet ، Cells و ارجاع به یک سلول واحد
T9T.	۳-۵-۲ ویژگی Worksheet.Cells و ترکیب جایگزین ارجاع به سلول واحد
۲۹٤.	۴-۵-۲. ویژگیRange . Cells و ارجاع به سلول واحد مرتبط با یک محدوده
۲۹٤.	۵-۵-۲. ویژگی Range ، Cells و ترکیب جایگزین ارجاع به سلول واحد مرتبط با یک محدوده
T9V .	۶-۵-۲ ارجاع به یک محدوده سلولی با استفاده از ویژگی های Range و Cells
۲۹۸.	۷-۵-۷. ویژگی Worksheet ، Cells و ارجاع به تمام سلول های یک برگه
۲۹۸.	۸-۵-۷. چرا باید یاد گرفت که از ویژگی Cells برای ارجاع به شیء Range استفاده کنیم
۲۹۸.	۶-۷. ویژگی Range و ارجاع به شیء Range Range
799 .	۲-۶-۲. ویژگی Range ، Offset و ارجاع به یک سلول واحد
۳•۱.	۲-۶-۲ ویژگی Range ، Offset و ارجاع به محدوده ای از سلول ها

۳۰۲	۲-۶-۷. چرا باید از ویژگی Range ، Offset برای ارجاع به شیء Range استفاده کرد	
۳•۲	۷-۷. ویژگی Range و ارجاع به یک شیء Range	
۳•٦	فصل هشتم	
۳۰٦	سبک R1C1 و ویژگی Range.FormulaR1C1 در VBA	
۳•٧	مقدمه	
۳•۸	۸-۱. سبک R1C1 و A1 : بررسی اجمالی	
۳•۹	۱-۱-۸. سبک A1.	
۳۰۹	۲-۱-۸. سبکR1C1. سبک	
۳۱۱	۲-۸. چرا ویژگی Range.FormulaR1C1 و سبک R1C1 مهم و مفید است	
۳۱۲	۸-۳. سبکR1C1 : نحوه ایجاد آدرس های سلولی۸	
۳۱۳	۸–۳–۱. آدرس های نسبی در سبکR1C1	
۳۱۷	۸-۳-۲. آدرس های مطلق در سبک R1C1	
۳۱۷	۸-۳-۳. آدرس های ترکیبی در سبک R1C1	
۳۱۸	۸-۳-۴. ارجاع به یک سطر یا ستون کامل با استفاده از سبکR1C1	
۳۱۹	۴-۸. ویژگی Range ، FormulaR1C1	
۳۱۹	۸-۴-۱. بررسی اجمالی	
۳۲۰	۸-۴-۲. ساختار ویژگی	
۳۲۰	۸-۴-۳. خواندن ویژگی	
۳۲۰	۸-۴-۴. کار با محدوده ها	
۳۲۱	۸-۴-۸. تنظیم تاریخ	
۳۲۱	۵-۸. ویژگی Range ، FormulaR1C1 Local و ملاحظات زبان اکسل	
۳۲۱	۸-۵-۱. فرمول های اکسل و ملاحظات زبان اکسل	
۳۲۲	۲-۵-۸. ویژگی Range ، FormulaR1C1Local	
ل ۳۲۳	۸-۶. مثال برای ویژگیFormulaR1C1 : تنظیم فرمول یک محدوده سلولی برای ایجاد جدوا	
۳۲۸	۸-۶-۸. ویژگی Range . Formula : تنظیم فرمول یک محدوده سلولی برای ایجاد جدول	
۳۳۰	۸-۷. یک نکته در ماکروهایی که از آدرس نسبی استفاده می کند	

۳۳۳	فصل نهم
۳۳۳	BA کار با سلول ها و محدوده ها در
٣٣٤	مقدمه
۳۳٤	۹-۱. بررسی اجمالی
۳۳٦	۲-۹. چند ویژگی مفید شیء Range
۳۳۷	۲-۹-۱. ویژگی Value
۳۳۸	۲-۲-۹. ویژگی Text
۳۳۸	۲-۲-۹. ویژگی Count.
۳۳۸ Η	۲-۹-۴۰ ویژگی Column و Row
۳۳۹	۵-۲-۹. ویژگی Address
۳۳۹	۲-۹-۶ ویژگی HasFormula
٣٤٠	۲-۹-۷ ویژگی Font.
۳٤٢	۸-۲-۹. ویژگی Interior.
۳٤٢	۹-۲-۹. ویژگی Formula
۳٤٣Νι	۱۰-۲-۹. ویژگی umberFormat
۳٤٣ Ra	۳-۹. چند شیوهعمل مفید شیء nge
٣٤٣	۹-۳-۹. شيوه عمل Select
۳٥٠ Paste	۲-۳-۹. شیوه های عمل Copy و
۳٥٥	۳-۳-۹. شيوه عمل Clear
۳٥٥	۴-۳-۹. شيوه عمل Delete
یک محدوده)	۹-۳-۹. شيوه عمل Cut (جابجای
٣٥٦	۴-۹. نکاتی برای کار با محدوده
۳٥٧	فصل دهم
ف اکسل	دسترسی به ماکرو از فایل های مختل
۳٥٨	(Personal Macro Workbook)

۳٥Λ	مقدمه
Personal N چیست؟	lacro Workbook .١-١٠
ی ماکروهای شخصی ۳۵۹	۲-۱۰. مسیر دسترسی به فایا
نظر در فایل ماکروهای شخصی۳٦۱	۱۰–۳. کپی ماکرو های مورد
رو در فایل ماکروهای شخصی ۳٦۱	۱-۳-۱۰. مراحل کپی ماک
ره شده در فایل ماکروهای شخصی۳٦٣	۴-۱۰. اجرای ماکروهای ذخیر
رو به ريبون۳٦٣	۱۰-۴-۱۰. اضافه کردن ماک
٣٦٤	فصل یازدهم
س یک	تنظيمات ويرايشكر ويژوال بي
ฑาอ	مقدمه
گر ویژوال بیسیککر ویژوال بیسیک	۱۹–۱۰. سفارشی کردن ویرایش
דאEc	۲-۱۱. گزینه های زبانه litor
רדיז	de Settings . 1-7-11
۳۷۵Wind	ow Settings .۲-۲-۱۱
۳۷۹	۳-۱۱ زبانه ditor Format
۳۸۱	Code Colors .1-۳-11
۳۸۳	Font .۲-۳-۱۱
۳۸٥	Size .٣-٣-١١
۳۸۵ Margin Ir	ndicator Bar .۴–۳–۱۱
۳۸٥	۴-۱۱. زبانه General
ፕለጓ Form G	rid Settings .1-4-11
۳۸٦Sho	w ToolTips .۲-۴-۱۱
۳۸۷Collapse Proj. Hide	es Windows .۳-۴-۱۱
۳۸۸Edit and Continue & Notify Before	e State Loss .۴-۴-۱۱
۳۸۸Εrr	or Trapping .۵-۴-۱۱
۳۸۹	Compile .۶-۴-۱۱

۳۹٠	۵-۱۱. زبانه Docking
٣٩٢	منابع

فهرست جلد دوم

مقدمه۱
مطالبی در رابطه با مطالعه کتاب ۲
محتویات کتاب ٦
فصل اول
دستور IF…Then در VBA دستور IF…Then دستور
مقدمه
۱–۱. ترکیب IFThenElse
۲-۱. نحوه استفاده از دستورIF در VBA (مثال)۲-۱
۱–۳. دستور IF تو در تو۲–۳. دستور IF
IFThen .۱-۳-۱ با شروط چندگانه
۲-۳-۱. دستور IFThenElse
۲-۳-۱. ساختار IF…Then… Elseif…Else
۰۹ استفاده از AND و OR و OR در دستور IF…Then…Else
۵-۱. استفاده از عملگر نامساوی (<>) در دستور IFThen۰۰۰ استفاده از عملگر نامساوی (<>)
۶-۱. استفاده از Not با Not استفاده از Not ا
ف صل دوم
دسـتور Select Case در VBA در VBA در دسـتور که در ک
مقدمه
۲۲ Select Case

۲۳	۲-۲. نمونه هایی از دستور Select Case
۲۳	۲-۲-۱. بررسی اعداد وارد شده
۲٥	۲-۲-۲. استفاده از "شرط IS " در ساختار Select Case
۲٥	۲-۲-۲. استفاده از Case Else برای بررسی همه مقادیر
۲٥	۲-۲-۴. بررسی مجموعه اعداد
۲٦	۲-۲-۵. رتبه بندی بر اساس شماره های وارد شده
۲۸	۲-۲-۶. بررسی اعداد زوج و فرد با استفاده از Select Case
۲۸	۲-۲-۷. بررسی روزهای هفته (شروط چندگانه)
۲۹	۸-۲-۲ دستور Select Case تو در تو ۸-۲-۲
۳۱	۹-۲-۲. بررسی رشته متن با Select Case
۳۲	فصل سوم
۳۲	حلقه ها در VBA
۳۳	مقدمه
٣٤	۲-۱. حلقه VBA اکسل چیست؟
۳٥	۲-۳. چرا باید حلقه های VBAرا یاد بگیرید
۳٦	۳-۳. حلقه ها چقدر سریع است؟
۳۹	۴-۳. حلقه For…Next
۳۹	۲-۴-۳. ترکیب حلقه ForNext
٤٣	۲-۴-۳. نحوه عملکرد حلقه ForNext
٤٥	۳-۴-۳. مثال ها
٤٨	۵-۳. حلقه For EachNext د حلقه
٥٠	۱-۵-۳. ترکیب حلقه For EachNext
٥٢	۲-۵-۳. نحوه عملکرد حلقه For EachNext
٥٤	۳-۵-۳. مثال های حلقه For Each…Next
~7	
01	۶-۳. حلقه های Do While وDo Until

٦٠	۲-۶-۳. عملکرد و روند حلقه های Do While و Do Until
้าา	۳-۶-۳. مثال های حلقه Do While
٦٨	۴-۶-۳. مثال های حلقهDo Until
٧٠	۳-۷. حلقهDo و اجتناب از حلقه های بی پایان
٧٠	۲-۷-۳. ترکیب حلقهDo ترکیب حلقه
٧٢	۸-۳. حلقه WhileWend
٧٣	۱-۸-۳. ترکیب حلقه WhileWend
٧٤	۲-۸-۳. عملکرد و روند حلقه WhileWend
Vo	۳-۹. حلقه های VBA و دستور Exit
VV	۳–۱۰. حلقه های تو در تو۳
٧٩	۳-۱۰-۱. مثال هایی از حلقه های تو در تو
Λ٣	۳-۱۱- دستور GoTo
Λ٤	۱-۱۱-۳. ایجاد حلقه با IF و GoTo
Λο	فصل چهارم
Λο	فصل چهارم توابع برگه و توابع داخلی VBA در برنامه نویسی
Λο Λο Λ٦	فصل چهارم . توابع برگه و توابع داخلی VBA در برنامه نویسی م _{قدمه} .
Λο Λο Λ٦ Λ٦	فصل چهارم . توابع برگه و توابع داخلی VBA در برنامه نویسی ^{مقدمه} . ۱-۴. تابع چیست ؟
Λο Λο Λι Λι Λν	فصل چهارم توابع بر گه و توابع داخلی VBA در برنامه نویسی مقدمه. ۴-۱. تابع چیست ؟ ۴-۲. توابع استاندارد بر گه اکسل
Λο Λο Λ٦ ΛΥ ΛΛ	فصل چهارم . توابع بر که و توابع داخلی VBA در برنامه نویسی مقدمه . ۴-۱. تابع چیست ؟ . ۴-۲. توابع استاندارد بر که اکسل ۴-۲-۱. نحوه استفاده از توابع بر که اکسل در VBA
Λο Λο Λτ Λν Λλ ۹)	فصل چهارم . توابع بر که و توابع داخلی VBA در برنامه نویسی . مقدمه. ۴-۱. تابع چیست ؟ . ۴-۲-۲. توابع استاندارد بر که اکسل . ۲-۲-۴. نحوه استفاده از توابع بر که اکسل در VBA
Λο Λο Λ٦ ΛΥ ΛΝ ۹١ ۹٢	فصل چهارم . توابع بر که و توابع داخلی VBA در برنامه نویسی مقدمه. ۴-۱. تابع چیست ؟ ۴-۲-۴. نحوه استفاده از توابع بر که اکسل در VBA ۴-۲-۴. ساده سازی آدرس ارجاع به شیء ۴-۲-۳. چگونه می توان تابع بر که اکسل را در VBA وارد کرد
Λο Λο ΛΊ ΛΊ ΛΥ ΛΝ ۹١ ۹٢	فصل چهار م توابع بر که و توابع داخلی VBA در برنامه نویسی مقدمه. ۴-۱. تابع چیست ؟ ۴-۲-۲. توابع استاندارد بر که اکسل . ۴-۲-۲. نحوه استفاده از توابع بر که اکسل در VBA ۴-۲-۲. ساده سازی آدرس ارجاع به شیء ۴-۲-۳. چگونه می توان تابع بر که اکسل را در VBA وارد کرد ۴-۲-۴. کدام توابع بر که اکسل در VBA قابل استفاده نیست
Λο Λο Λ٦ Λ٦ ΛΥ ΛΝ ٩١ ٩٢ ١٠٦ ١.٦	فصل چهارم . توابع بر که و توابع داخلی VBA در برنامه نویسی . مقدمه. ۱۹-۲. تابع چیست ؟ ۲-۴. تابع چیست ؟ ۲-۴-۲. نحوه استفاده از توابع بر که اکسل در VBA ۲-۴-۲. ساده سازی آدرس ارجاع به شیء. ۲-۴-۲. چگونه می توان تابع بر که اکسل را در VBA وارد کرد ۴-۲-۴. کدام توابع بر که اکسل در VBA قابل استفاده نیست ۴-۲-۴. مثال هایی از توابع بر که اکسل در VBA قابل استفاده نیست
∧○ ∧○ ∧□、 ∧□、 ∧∇ ∧∧ へ∧ へ∧ へ∧ へへ へく へく へく へく へく へく く く く へく へく へく へく く く く く く へく へく へく へ、	فصل چهار م توابع بر که و توابع داخلی VBA در برنامه نویسی مقدمه ۴-۱. تابع چیست ؟ ۴-۲. توابع استاندارد بر که اکسل ۴-۲-۱. نحوه استفاده از توابع بر که اکسل در VBA ۴-۲-۲. ساده سازی آدرس ارجاع به شیء ۴-۲-۳. چگونه می توان تابع بر که اکسل را در VBA وارد کرد ۴-۲-۴. کدام توابع بر که اکسل در VBA قابل استفاده نیست ۴-۲-۹. مثال هایی از توابع بر که اکسل در NBA
۸٥ ۸٦ ٨٦ ٨٦ ٨٧ ٨٨ ٩١ ٩٢ ٩٢ ١٠٦ ١·٦ ١·٩	فصل چهارم . توابع بر که و توابع داخلی NBA در برنامه نویسی . مقدمه . ۹-۱. تابع چیست ؟ . ۴-۲. توابع استاندارد بر که اکسل . ۴-۲-۱. نحوه استفاده از توابع بر که اکسل در NBA . ۴-۲-۲. یحوه استفاده از توابع بر که اکسل در در NBA . ۴-۲-۳. چگونه می توان تابع بر که اکسل را در NBA وارد کرد ۴-۲-۹. کدام توابع بر که اکسل در NBA قابل استفاده نیست ۴-۳-۹. مثال هایی از توابع بر که اکسل در NBA . ۴-۳-۹. توابع داخلی NBA .

۱۱Σ	۴-۴. توابع سفارشی (فرآیند function)
וות	فصل پنجم
าาา	چند تابع مهم VBA
וות	مقدمه
١٢٠	۵-۱. توابع LBOUND وBOUND
١٢٠	۵-۱-۱. ترکیب توابع
١٢٠	۵-۱-۲ مثال های نمونه
١٢١	۵–۲. تابع Split
177	۵-۲-۵. ترکیب تابع
١٢٣	۵-۲-۲. مثال های نمونه
170	۵–۳. تابع ARRAY
۱۲٥	۵-۳-۱. ترکیب تابع
۱۲٥	۵-۳-۲. مثال های نمونه
١٢٧	۵–۴. تابع DIR
١٢٨	۵-۴-۱. ترکیب تابع
179	۵-۴-۲ استفاده از نماد های جایگزین پذیر با تابع DIR
179	۵-۴-۳. مثال های نمونه
١٣٣	۵-۵. تابع InStr
١٣٣	۵-۵-۱. ترکیب تابع
١٣٤	۵-۵-۲. نکات تکمیلی تابع InStr
١٣٤	۵-۵-۳. مثال های نمونه
١٣٦	۵-۶ تابع TRIM
۱۳۷	۵-۶-۱. ترکیب تابع
۱۳۷	۵-۶-۲. مثالهای نمونه
۱۳۹	۲-۵. تابع SGN
۱۳۹	۵-۷-۱. ترکیب تابع

١٤٠	۵–۸. تابع INT ۸–۵
١٤٠	۵–۸–۱. ترکیب تابع
۱٤)	۵–۹. تابع FIX
۱٤۱	۵-۹-۱. ترکیب تابع
١٤٢	۵-۱۰-۵ تابع JOIN
١٤٢	۵-۱۰-۱۰. ترکیب تابع
١٤٢	۵-۱۰-۲. مثال های نمونه
۱٤٣	فصل ششم
١٤٤	کار با برگه های اکسل در VBA
١٤٤	مقدمه
١٤٤	۶-۱. تفاوت بین Worksheets و Sheets در VBA
۱٤٦	۶-۲. ارجاع به برگه های اکسل
۱٤٦	۶-۲-۲. ارجاع به کلیه برگه های فایل اکسل
١٤٧	۶-۲-۲. ارجاع به برگه فعال
١٤٧	۶-۲-۳. ارجاع به برگه با استفاده از نام برگه
۱٤۸	۶-۲-۶. ارجاع به برگه با استفاده از شماره شاخص
10•	۶-۲-۶. ارجاع با استفاده از نام کد برگه
١٥٣	۶-۲-۶ ارجاع به چندین برگه
١٥٤	۳-۲-۳. ارجاع به برگه ای در یک فایل اکسل دیگر
100	۶-۳. افزودن برگه جدید به فایل اکسل
۲٥٦	۶-۴. حذف یک برگه از فایل اکسل
ر حلقه For EachNext ر	۶-۵. بررسی همه برگه های فایل اکسل با استفاده از دستو
ر حلقه ForNext ر حلقه	۶-۶. بررسی همه برگه های فایل اکسل با استفاده از دستو
كوسكوس	۶-۷. حلقه کردن تمام برگه های فایل اکسل به ترتیب مع
109	۶-۸. تغییر نام برگه های فایل اکسل
١٦٠	۶-۹. اختصاص شیء Worksheet به متغیر

ודו	۶-۱۰۰ مخفی کردن برگه ها (Hidden + Very Hidden)
ודו	۶-۱۰-۱. روش عادی پنهان کردن یک برگه در اکسل
אדר	۶-۱۰-۲. مخفی کردن برگه ، بطوری که نتوان آن را به راحتی نمایان کرد
ه است	۶-۱۰-۶. نمایان کردن برگه ای که با ویژگی Very Hidden مخفی شد
١٦٥	۶-۱۰-۴. مخفی/ نمایان کردن برگه ها با استفاده از VBA
١٦٥	۶-۱۱. پنهان کردن برگه ها بر اساس عنوان برگه
דרו	۶-۱۲ مرتب کردن برگه ها به ترتیب حروف الفبا
ארו V	۶–۱۳. حفاظت (Protect) تمام برگه های یک فایل
אר ו	۶–۱۴ حفاظت تمام برگه ها با استفاده از رمز ورود
١٦٨	۶-۱۵. ایجاد فهرست برگه ها (با لینک)
١٦٩	۶–۱۶. حذف برگه های خالی یک فایل
١٧٠	فصل هفتم
\\/ .	VDA A (Workhook) 1 ala hibita (
TV *	الاربا فايل هاي النسل (workbook) در DA
۱۷۰	مقدمه
۱۷۰ ۱۷۱	مقدمه مقدمه ۲-۱. ارجاع به یک فایل با استفاده از VBA
1V• 1V1 1V1	مقدمه مقدمه ۲–۱. ارجاع به یک فایل با استفاده از VBA ۲–۱. ارجاع با استفاده از نام فایل
 1 V · 1 V · 1 V 1 1 V 1 1 V 1 1 V τ 	کر با گایل های ۱ دسل (WOIKBOOK) کر ۷۵۸ مقدمه. ۱–۱ ارجاع به یک فایل با استفاده از VBA ۱–۱–۲. ارجاع با استفاده از نام فایل
 1 ∨ · 1 ∨ · 1 ∨ 1 1 ∨ 1 1 ∨ 1 1 ∨ τ 1 ∨ τ 1 ∨ τ 	کار با کایل های ۱ دسل (WOIKDOOK) کر ۷۵۸مقدمه ۲–۱. ارجاع به یک فایل با استفاده از VBA ۲–۱–۱. ارجاع با استفاده از نام فایل ۲–۱–۲. ارجاع با استفاده از شماره شاخص
 1∨• 1∨1 1∨1 1∨1 1∨T 1∨T 1∨T 1∨T 1∨T 	کر با کیل های ۲ دسل (WOIKDOOK) کر ۷۵۸ مقدمه. ۲–۱۰ ارجاع به یک فایل با استفاده از VBA ۲–۱–۱. ارجاع با استفاده از نام فایل ۲–۱–۲. ارجاع با استفاده از شماره شاخص ۲–۱–۲. ارجاع با استفاده از فایل فعال
1V• 1V1 1V1 1V1 1VT 1VT 1VT 1VT 1VT	کر با کیل های ۲ دسل (WOIKDOOK) کر ۷۵۸ مقدمه ۲–۱۰ ارجاع به یک فایل با استفاده از VBA ۲–۱–۱. ارجاع با استفاده از نام فایل ۲–۱–۲. ارجاع با استفاده از فایل فعال ۲–۱–۴. ارجاع با استفاده از فایل فعال ۲–۱–۲. ارجاع با استفاده از مایل معال
 NV• NV1 NV1 NVT NVT NVT NVT NVT NVT NVT NVT 	کر با کیل های ۲ دسل (WOIKDOOK) کر ۷۵۸ مقدمه. ۲–۱۰ ارجاع به یک فایل با استفاده از VBA ۲–۱–۲. ارجاع با استفاده از نام فایل ۲–۱–۳. ارجاع با استفاده از فایل فعال ۲–۱–۴. ارجاع با استفاده از فایل فعال ۲–۲–۲. ایجاد یک فایل جدید ۲–۳. باز کردن یک فایل با استفاده از VBA
1 V · 1 V · 1 V 1 1 V 1 1 V Γ 1 V Γ 1 V Γ 1 V Γ 1 V Γ 1 V Σ 1 V Δ	کر با کیل های ۲ دسل (WOIKDOOK) کر VBAمقدمه. مقدمه. ۲–۱۰ ارجاع به یک فایل با استفاده از VBA ۲–۱۰۰ ارجاع با استفاده از نام فایل ۲–۱۰۰ ارجاع با استفاده از فایل فعال ۲–۱۰۰ ارجاع با استفاده از فایل فعال ۲–۱۰ ایجاد یک فایل جدید ۲–۲ ایجاد یک فایل با استفاده از VBA
NV• NV1 NV1 NVT NVT	کر با کیل های اکسل (WOrkbook) کر VBAمقدمه. مقدمه. ۲–۱. ارجاع به یک فایل با استفاده از VBA ۲–۱–۲. ارجاع با استفاده از نام فایل ۲–۱–۳. ارجاع با استفاده از شماره شاخص ۲–۱–۳. ارجاع با استفاده از فایل فعال ۲–۲. ایجاد یک فایل جدید ۲–۳. باز کردن یک فایل با استفاده از VBA ۲–۹. ذخیره فایل اکسل
NV· NVI NVI NVT NVT NVT NVT NVT NVT NVE NVE NVO NVT	مقدمه مقدمه ۲–۱. ارجاع به یک فایل با استفاده از VBA ۲–۱–۱. ارجاع با استفاده از نام فایل ۲–۱–۲. ارجاع با استفاده از شماره شاخص ۲–۱–۳. ارجاع با استفاده از فایل فعال ۲–۱–۴. ارجاع با استفاده از فایل فعال ۲–۲. ایجاد یک فایل جدید ۲–۳. باز کردن یک فایل با استفاده از VBA ۲–۹. ذخیره فایل اکسل ۲–۹. ذخیره تمام فایل های باز
NV· NVI NVI NVT NVE NVO NVN NVN	کر با کایل های ۱ کسل (WORKDOOK) کر VBA مقدمه ۲–۱-۱. ارجاع به یک فایل با استفاده از VBA ۲–۱-۲. ارجاع با استفاده از نام فایل ۲–۲-۳. ارجاع با استفاده از فایل فعال ۲–۲۰. ارجاع با استفاده از فایل فعال ۲–۲۰. ایجاد یک فایل جدید ۲–۲. ایجاد یک فایل جدید ۲–۲. دخیره فایل اکسل ۲–۹. ذخیره تمام فایل های باز ۲–۹. ذخیره کپی از فایل فعال (با تایمر)

١٧٩	۹-۲. اختصاص شیء Workbook به متغیر۹
١٧٩	۲-۱۰. ایجاد حلقه از طریق فایل های باز۷
۱۸۰ (F	۱۱-۲. خطا هنگام کار با شیء Workbook (′ 9 ′)
۱۸۱	۷–۱۲. رمزگذاری برگه های فایل های باز۷
۱۸۱	۲–۱۳. تهیه فهرست فایل های باز
١٨٢	فصل هشتم
١٨٣	ایجاد و اجرای توابع سفارشی (UDF) در VBA
١٨٣	مقدمه
١٨٣	۸–۱. تفاوت فرآیند Function و فرآیند sub در VBA
١٨٥	۲-۸. چرا فرآیند های Function را ایجاد و استفاده می کنیم؟
ነለኘ	۸-۳. ترکیب اصلی یک فرآیند Function
١٨٧	۸-۳-۱. دستور اعلان یک فرآیند Function
١٨٧	۸-۳-۲ نامگذاری فرآیند های Function
۱۸۸	۸-۳-۳. تعیین اینکه فرآیند Function چه مقداری را باید برگرداند
19•	۸-۳-۴. ترکیب جامع فرآیند Function
197	۴-۸. ایجاد یک تابع سفارشی ساده در VBA
۱۹۳	۸-۴-۱. آناتومی تابع ایجاد شده
۱۹۸	۵-۸. مولفه های فرآیند Function
۲۰۰	۸-۵-۱. ایجاد تابع سفارشی بدون مولفه
۲۰٤	۸-۵-۲. ایجاد تابع سفارشی با یک مولفه ورودی
۲۰۷	۸-۵-۳. ایجاد تابع سفارشی با چند مولفه ورودی
۲۱۱	۸-۵-۴. استفاده از آرایه به عنوان مولفه
۲۱۳	۸-۵-۵ ایجاد تابع سفارشی با مولفه های اختیاری
۲۱۵	۸-۵-۶ ایجاد تابع سفارشی با مولفه های اجباری و اختیاری
719	۸-۵-۷. ایجاد تابع سفارشی با تعداد مولفه ورودی نامحدود
۲۲۳	

۲۲٤	۸–۶–۱. بازگرداندن اسامی ماه ها
۲۲۷	۸-۶-۲. بازگرداندن لیست مرتب شده
۲۲۸	۸-۷. سایر موارد اختیاری در ایجاد فرآیند Function
۲۲۸	۸-۷-۸. فرآیند Function را کجا ذخیره کنیم
۲۳۰	۲-۷-۸. نوع داده های VBA و فرآیند های ۲-۷-۸.
۲۳۰	۸-۷-۲. نحوه اجرای یک فرآیندFunction
۲۳۸	۸-۷-۴. اضافه کردن توضیحات برای فرآیندFunction
٢٤٨	۸-۶-۵ تغییر دسته بندی فرآیندFunction
۲٥٠	۸-۷-۶ ایجاد دسته بندی جدید برای فرآیندFunction
707	۸-۸. استفاده از دستور Exit Function در VBA
۲٥٢	۸-۹. خطایابی تابع تعریف شده توسط کاربر
۲٥٣	۸-۹-۱. خطایابی تابع سفارشی با استفاده از پنجره پیام
۲٥٣	۸-۹-۲. خطا یابی تابع سفارشی با تنظیم نقطه توقف (Breakpoint)
۲٥٤	۸-۹-۳. خطا یابی تابع سفارشی با استفاده از دستور Debug.Print
٢٥٥	۸-۱۰. توابع استاندارد برگه اکسل در مقایسه با توابع سفارشی VBA
٢٥٥	۸–۱۱. توابع پوششی (Wrapper Functions)
۲۵٦	۸-۱۱-۱. توابع پوششی برگه اکسل
۲۷۵	۸–۱۱–۲. توابع سفارشی مفید برای استفاده در کد شما
۲۸۰	۸–۱۲ فراخوانی توابع API ویندوز
۲۸۱	۸–۱۲–۱. اکسل ۶۴ بیتی و توابع API
۲۸۲	۸–۱۲–۲. مثال هایی از نحوه استفاده توابع API ویندوز
797	فصل نهم
۲۹۳	تکنیک های مدیریت خطا د ر VBA
۲۹۳	مقدمه
٢٩٤	۹-۱. مثالی از بررسی خطاها همگام با کد نویسی
۲۹۸	۲-۹. انواع خطاهای VBA اکسل

۲۹۸	۰-۲-۹. خطاهای ترکیب (Syntax errors)
۳۰۰	۲-۲-۹. خطای همگردانی (Compilation errors)
۳۰۱	۲-۹-۳-۳. خطاهای زمان اجرا (Runtime errors)
۳۰۳	۴-۲-۹. خطاهای منطقی (Logical Errors)
۳۰۳	۳-۹. پیکربندی تنظیمات خطا در Error Trapping)VBA)
٣٠٤	۴-۹. مدیریت خطا با استفاده از دستور On Error
۳۰٥	On Error Resume Next .١-۴-٩
۳۰۷	On Error GoTo 0 .۲-۴-۹
۳۰۸	On Error Goto [Label] .٣-۴-٩
۳۱٤	۵-۹. برای مقابله با خطاهای VBA چه چیز دیگری باید بدانیم؟
۳۱٤	۱-۵-۹. شیء Err
۳۱۵	۱–۵–۹. ویژگی های شی Err
۳۱۸	۲–۵–۹. شيوه های عمل شیء Err
۳۲۳	۶-۹. نكات استفاده از دستورات مديريت خطا به صورت خلاصه
۳۲۳	۹–۷. تکنیک های رفع خطا
٣٢٤	۹-۲-۹. بررسی کد
٣٢٤	۲-۷-۹. استفاده از تابع MsgBox
۳۲٥	۳-۲-۹. درج دستور Debug.Print
۳۲٦	۹-۷-۹. استفاده از ابزارهای رفع خطای VBA
۳۳۲	۸-۹. نکات کاهش خطا
۳۳٤	فصل دهم
٣٣٤	رخدادها و فرآیند های خودکار
۳۳٥	مقدمه
۳۳٥	۱-۱۰. رخداد VBA اکسل چیست
۳۳٦	۲-۱۰. آیا رخداد های VBA مفید است
۳۳۷	۰۱-۳-۱۰ انواع مختلف رخدادهای VBA

۳۳۸	۱۰-۱۰. نحوه ایجاد فرآیند های حامل رخداد
۳٤٠	۱۰–۴-۱۰ انتخاب ماژول مناسب
۳٥٣	۲-۴-۱۰. کد نویسی فرآیند حامل رخداد
۳٥٧	۵-۱۰. نقش مولفه ها در رخدادهای VBA
۳٥٩	۰۱-۶. لیست رخداددهای VBA اکسل: سازماندهی و سطوحی که میتوانید آنها را کنترل کنید
<i>״</i> ¬״	۱۰-۶-۱۰ رخدادهای مربوط به ایجاد یا حذف شیء
ฑาง	۱۰-۶-۲. رخدادهای مربوط به باز شدن ، بسته شدن ، چاپ و ذخیره
۳۸۰	۱۰–۶–۳. رخدادهای مربوط به فعال سازی شی ، انتخاب ، تغییر اندازه و کلیک ماوس
٤•٦	۲۰–۶–۴. رخدادهای مربوط به تغییرات و محاسبات
٤١٦	۵-۶-۱۰. رخدادهای مربوط به پنجرهProtected View
٤٢٠	۲۰-۶-۶ رخدادهای مربوط به PivotTable (جدول محوری)
٤٢٩	۲-۶-۱۰. رخدادهای مربوط به افزونه ها (Add-Ins)
٤٣٠	۸-۶-۱۰. رخدادهای مربوط به مدل داده های اکسل (Data Model)
٤٣١	۹-۶-۱۰. رخدادهای مربوط به داده XML Data) XML)
٤٣٤	۱۰-۶-۱۰. سایر رخدادها
٤٣٥	۱۰-۶-۱۱. رخدادهای غیر شیء
٤٣٦	OnTime .١-١١-۶-١٠
٤٣٩	۲-۱۱-۶-۱۰. رخداد OnKey
٤٤٤	۲-۱۰. رایج ترین رخدادهای سطح فایل اکسل و سطح بر گه اکسل
٤٤٦	۸-۱۰. غیر فعال کردن رخدادها در VBA
٤٤٧	۹-۱۰. تأثیر رخدادها در پشته دستور Undo
٤٤٨	فصل یازدهم
٤٤٩	برقراری ارتباط با کاربر ، پنجره های محاورهای
٤٤٩	مقدمه
٤٤٩	۱-۱۱. پنجره های محاوره ای ساده جایگزین UserForm
٤٥٠	۱-۱-۱۱ تابع MsgBox

٤٧١	۲-۱-۱۱. تابع InputBox (پنجره ورودی)
٤٨٢	۲-۱-۱۱. فرم داده (Data Form)
٤٨٤	۴-۱-۱۱. شيوه عمل GetOpenFilename
٤٨٩	۵-۱-۱۱. شيوه عمل GetSaveAsFilename شيوه عمل
٤٩٠	۶-۱-۱۱. ویژگی FileDialog
٤٩٢	۱۱–۱-۷. نمایش پنجره های محاوره ای داخلی اکسل
٤٩٥	۲-۱۱. مبانی UserForm
٤٩٥	۱-۲-۱۱. چه موقع از UserForm استفاده کنیم
٤٩٧	UserForm (یک مرور کلی)
٤٩٨	UserForm کار با UserForm ها
٥•٨	۳-۱۱. ایجاد یک نمونه UserForm
٥•٩	۱-۳-۱۱ ایجاد UserForm
٥•٩	۲-۳-۱۱. افزودن CommandButton
٥١١	۳-۳-۱۱. افزودن OptionButton
٥١٣	۴-۳-۱۱. اضافه کردن فرآیندهای حامل رخداد
010	۱۱-۳-۵. ایجاد ماکرویی برای نمایش پنجره محاوره ای
017	۹۱–۳–۶. در دسترس بودن ماکرو
٥١٧	۲-۳-۱۱. تست ماکرو
٥١٨	۴-۱۱. کنترل های UserForm
٥١٨	۱-۴-۱۱. شروع به کار با کنترل های UserForm
٥٢٣	۲-۴-۱۱. کنترل های UserForm: جزئیات
٥٣٥	۲-۱۹-۳. کار با کنترل های پنجره محاوره ای سفارشی
٥٤٠	۴-۴-۱۱. تکنیک ها و ترفندهای UserForm
٥٤٨	۵-۱۱. مطالبی در مورد رخداد های کنترل
00+	۱-۵-۱۱. رخدادهای کنترل SpinButton
٥٥٤	۰۱-۶. ارجاع به کنترل های UserForm

۲-۱۱. سفارشی کردن جعبه ابزار
۱۱-۷-۱۱ اضافه کردن زبانه های جدید به جعبه ابزار
۲-۷-۱۱. شخصی سازی یا ترکیب کنترل ها
۵۵۸ افزودن دیگر کنترل های ActiveX
UserForm الگو (UserForm)۵۵۹
۵٦٠ UserForm چک لیست ایجاد ۹-۱۱
۱۰-۱۱. نگاهی به مثال های UserForm
UserForm استفاده از UserForm به عنوان "منو"
۲-۱۰-۱۱. انتخاب یک محدوده از طریق UserForm
۵٦٧-۱۰-۳. ایجاد صفحه معرفی برنامه۵۷
۴-۱۰-۱۱. استفاده از چند مجموعه OptionButton
۵–۱۰–۵۰ استفاده از SpinButton و SpinButton می ۲۰۰۱۰
۵۷۲-۱۰-۹. غیرفعال کردن دکمه Close در UserForm
۷-۱۰-۱۱. تغییر اندازه UserForm
۷-۱۰-۱۱. تغییر اندازه UserForm
۵۷۳ UserForm UserForm ۵۷۲ ۵۷۲
۵۷۳UserFormUserForm۵۷۲۵۷۲۵۷۲۵۷۲۵۷۲۵۷۲۵۷۲۵۷۲۵۷۲
۵۷۳UserFormUserForm۵۷۷ ۵۷۵۵۷۵ UserForm۵۷۵۵۷۵ ۵۷۶۵۷۹۵۷۹۵۷۹ ۱۰-۱۰-۱۹. کاوشی در تکنیک های ListBox۵۷۹ ۲۰۱۱-۱۰-۱۱ استفاده از کنترل MultiPage در UserForm۵۷۶ ۱۰-۱۰-۱۱ استفاده از کنترل خارجی۵۷۶
۵۷۳UserFormUserForm۵۷۷ ۵۷۵۱۰-۱۰-۸ بزرگنمایی و پیمایش برگه از یک UserForm۵۷۷ ۵۷۵۱۰-۱۱-۹. کاوشی در تکنیک های ListBox۵۷۲ ۱۱-۱۰-۱۱ استفاده از کنترل Bage در UserForm۵۷۶ ۱۱-۱۰-۱۱ استفاده از کنترل خارجی۵۷۶ ۱۲-۱۰-۱۱ متحرک کردن Label۵۷۶
۵۷۳ UserForm
۵۷۳ UserForm UserForm ۵۷۵ UserForm UserForm ۵۷۲ در تکنیک های ListBox ۱۰-۱۰-۱۱ لیتفاده از کنترل Box و MultiPage ۱۰-۱۰-۱۱ استفاده از کنترل خارجی ۱۰-۱۱ دمتحرک کردن Label ۱۰-۱۱ دمتحرک کردن اعادا ۱۰-۱۱ دمتحرک کردن اعادا
۵۷۳
۵۷۳
UserForm الا-۱۰-۸ بزرگنمایی و پیمایش برگه از یک UserForm UserForm
UserForm بزرگنمایی و پیمایش برگه از یک UserForm

۲–۱۱–۱۱. شبیه سازی نوار ابزار با UserForm	
Task Pane با Task Pane با ۲۵۲۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰	
UserForm .۹-۱۱-۱۱ و قابلیت تغییر اندازه با کلیک کاربر	
۱۰-۱۱-۱۱. مدیریت چند کنترل UserForm با یک فرآیند حامل رخداد	
۱۱-۱۱-۱۱ انتخاب رنگ در UserForm	
۱۲-۱۱-۱۱. نمایش نمودار در UserForm	
UserForm ایجاد UserForm نیمه شفاف	
۱۱–۱۱–۱۴ یک پازل در UserForm	
۱۵–۱۱–۱۵. پوکر ویدیویی در یک UserForm	
صل دوازدهم	ۏ
مایر مباحث	J,
قدمه	م
۲۱–۱۲. کار با شیء Range	
١-١-١-٢. تغيير اندازه محدوده	
۲۱–۱۲ نوشتن مقدار در سلول۲۲۳	
۲۵-۱-۳ وارد کردن مقدار به سلول خالی بعدی	
۲۱-۱-۴. متوقف کردن ماکرو برای دریافت محدوده انتخاب شده توسط کاربر۲	
۵-۱-۱۲ شمارش سلول های منتخب	
٦٢-١-٩. تعيين نوع محدوده انتخاب شده	
۲۱-۱۲. پردازش کارآمد در یک محدوده۲۷۲	
۸-۱-۱۲. حذف تمام سطرهای خالی	
۲۵–۱۱-۹. تعداد تکرار سطرها در لحظه۲۷۷	
۱۰-۱-۱۲. تعیین اینکه یک محدوده داخل محدوده دیگری است یا خیر	
٦٢-١-١٢. تعيين نوع داده سلول٩٧٩	
۱۲-۱-۱۲. خواندن و نوشتن در محدوده	

۱۴-۱-۱۲. کپی محدوده های غیر مجاور
۱۵-۱-۱۲. مخفی کردن تمام سطر ها و ستون ها به جز موارد انتخاب شده
۱۶-۱-۱۲. همگام سازی برگه ها
Find شيوه عمل Find شيء Find شيء ۲۹۱
٦٩٢-١-١٧-١. تركيب شيوه عمل
۲-۱۲. تغییر تنظیمات اکسل
۲-۱۲-۱۲. تغییر تنظیمات منطقی
۲-۲-۱۲ تغییر تنظیمات غیر منطقی
۲-۱۲. افزایش سرعت اجرای کد VBA
۱۲-۳-۱۲. خاموش کردن به روزرسانی صفحه نمایش
۲۰–۳۰-۲ خاموش کردن محاسبه خودکار
۲۰-۳-۳ غیر فعال کردن پیام های هشدار مزاحم
۲-۳-۱۲. ساده سازی آدرس ارجاع به شیء
۱۲-۳-۵. اعلان نوع داده متغیر
۷۰۲-۳-۱۶ استفاده از دستور WithEnd With
۲-۱۲. نحوه ایجاد و استفاده از Add-in (افزونه) اکسل۲۰۰۰. نحوه ایجاد و استفاده از Add-in
۱۲-۴-۱۲. افزونه چیست؟
۲-۴-۱۲. مقایسه افزونه با فایل استاندارد
۲۰۴-۱۲. چرا افزونه ایجاد می کنیم؟
Add-In ها Add-In ها
۲۰–۴–۱۲. ایجاد افزونه
۲۹-۴-۱۲ Add-In
۲۰۴-۱۲. مقایسه یک افزونه با یک فایل استاندارد (مقایسه فایلهای XLAM و XLSM)
۸–۴–۱۲. دستکاری افزونه ها با VBA
۹–۴–۱۲. بهینه سازی عملکرد افزونه ها ۷۳۰
۱۰–۴–۱۲. مشکلات ویژه با افزونه ها ۷۳۱

مقدمه

کتاب حاضر به گونهای جامع به بحث مبانی برنامه نویسی با اکسل (ماکروها) پرداخته است و به همین دلیل برای طیف وسیعی از علاقهمندان از مبتدی تا حرفهای مناسب خواهد بود.

به مجموعهای از کارها که به صورت خودکار و پشت سرهم انجام شود ، ماکرو گفته می شود. تمام محدودیتهایی که فکر می کنید نرم افزار اکسل داشته باشد ، به کمک ماکروها می توان پوشش داد.

در اکسل به دو صورت می توان ماکرو ایجاد کرد. ایجاد ماکرو به کمک اکسل و دیگری ، نوشتن ماکرو توسط کاربر.

میتوانید یک سری دستور که میخواهید به صورت خودکار انجام شود را یک بار انجام دهید و از اکسل بخواهید تا آنها را ثبت کند و سپس در موقع لزوم ، دستورهای ثبت شده را هر بار که شما بخواهید خودکار اجرا کند. بسیاری از کارهایی که نیاز داریم به صورت خودکار انجام شود ، ممکن است به چند راه منتهی شود. بنابراین برای انجام ، حتما باید در مواردی تصمیم گیری و عملیات منطقی هم به صورت خودکار انجام شود که تحقق این امر با ماکرویی که توسط خود اکسل ایجاد میشود غیر ممکن است. در چنین مواردی ایجاد ماکرو به کمک مجموعهای از کدهای برنامه نویسی میتواند کارساز باشد.

یادگیری مبحث اول آسان و عملی ساختن آن بسیار راحت است ولی متاسفانه کاربرد کمتری نسبت به مبحث دوم دارد. مبحث دوم ، نوشتن ماکرو ، حداقل نیاز به اندکی آشنایی با برنامه نویسی دارد. هدف کتاب حاضر نیز در این راستا میباشد. بنابراین ...

آیا میخواهید یک برنامه نویس شوید

خبر خوب این است که برنامه نویسی با اکسل به همان اندازه که دشوار به نظر میرسد ، چندان هم دشوار نیست. اگر موافق نیستید ، صبر کنید ...

چرا زحمت؟

اکثر کاربران اکسل هیچ وقت زحمت یادگیری برنامه نویسی با VBA را به خود نمیدهند. علاقه شما به این موضوع قطعا شما را در بین گروهی نخبه قرار میدهد. اگر هنوز هم قانع نشدهاید که برنامه نویسی با اکسل ایده خوبی است ، چند دلیل برای شما آوردهام که شاید بخواهید برای یادگیری آن وقت بگذارید:

 ✓ این یادگیری شما را به بازار کار عرضه میکند. دوست داشته باشید یا نه ، نرمافزارهای مایکروسافت بسیار محبوب است. شاید میدانید همه برنامههای مجموعه Microsoft Office از VBA پشتیبانی میکند. هرچه بیشتر درباره VBA آگاهی داشته باشید ، شانس پیشرفت شما در شغلتان بیشتر میشود.

فرض کنید یک مدیر فروش هستید و باید گزارش فروش ماهانه را تهیه کنید. میتوانید یک برنامه VBA بنویسید تا این کار را برای شما انجام دهد. رئیس شما تحت تأثیر کیفیت بالای گزارشات شما قرار خواهد گرفت و شما به پست جدیدی ارتقا خواهید یافت که برای آن بسیار واجد شرایط هستید.

- ✓ این امکان را به شما میدهد تا از نرمافزار خود بیشترین بهره را ببرید. استفاده از اکسل بدون استفاده از VBA به نوعی مانند خرید تلویزیون و تماشای تنها تعدادی از کانالهای آن است.
- ✓ (در نهایت) باعث بهبود بهرهوری شما خواهد شد. تسلط بر VBA قطعا مدتی طول خواهد کشید اما شما بیشتر از آنچه فکر می کنید صرفه جویی می کنید ، مانند آنچه درباره رفتن به دانشگاه به شما گفتهاند.
- ✓ سرگرم کننده است (خوب ، گاهی اوقات). بعضی افراد واقعا از اکسل لذت میبرند و کارهایی انجام میدهند که در غیر این صورت غیر ممکن است. با تمام شدن مطالعه این کتاب ، ممکن است شما نیز یکی از این افراد باشید.

حالا متقاعد شدهاید؟

مطالبی در رابطه با مطالعه کتاب

الف- دستورات اكسل

دکمههای قسمت فوقانی ریبون اکسل (مانند Insert و View) به عنوان زبانه شناخته می شود. اگر روی یک زبانه کلیک کنید نماد (آیکن) دستورات مربوط به آن زبانه نمایش داده می شود. هر نماد دارای عنوانی است که (معمولا) در کنار یا زیر آن درج شده است. نمادها به صورت گروهی تنظیم شده است و نام گروه در زیر آنها ظاهر می شود.

قرارداد مورد استفاده در این کتاب ، درج نام **زبانه** به دنبال نام **گروه** و به دنبال آن نام **نماد** است. به عنوان مثال ، مراحل انتخاب دستور **تنظیم طول متن داخل یک سلول** به شرح زیر است:

Home ➡ Alignment ➡ Wrap Text

با کلیک اولین زبانه ریبون با برچسب File نمای پشت صفحه اکسل (پنجره Backstage) نمایش داده می شود. دستوراتی در امتداد لبه سمت چپ این پنجره وجود دارد. برای نشان دادن دستورات این پنجره از File و به دنبال آن از عنوان دستور استفاده می کنیم. به عنوان مثال ، دستور زیر پنجره محاورهای Excel Options را نشان می دهد:

File ⇔ Options

ب- دستورات ویرایشگر ویژوال بیسیک

ویرایشگر ویژوال بیسیک پنجرهای است که در آن با کدهای VBA کار خواهید کرد. این پنجره از رابط قدیمی نوار منو و نوار ابزار بهره میبرد. دستور زیر به معنای کلیک منوی Tools و انتخاب References است:

Tools ⇔ References

ج-کد VBA

این کتاب حاوی چند خط کد VBA است. برخی از این خطوط کد بین حاشیههای کتاب جای نمی گیرد. در چنین مواردی از نویسه استاندارد خطدنباله کد VBA (_) زیرخط به دنبال یک فضای خالی ، استفاده می-کنیم. به عنوان مثال:

Selection.PasteSpecial Paste:=xlValues, _

Operation:=xlNone. SkipBlanks:=False,Transpose:=False

هنگامی که از این کد استفاده میکنید ، میتوانید آن را به صورت فوق تایپ کنید یا با حذف فضای خالی و نماد زیر خط ، در یک خط قرار دهید.

د – دسترسی به فایل کدهای ارائه شده در کتاب

نمونههای ارائه شده در پوشه پیوست کتاب در دسترس است. با داشتن فایلهای آماده ، در زمان لازم برای تایپ مجدد کدها صرفهجویی خواهد شد. با این وجود بهتر است با آنها بازی کنید و با تغییرات مختلف ماکروها را آزمایش کنید. آزمایش بهترین روش برای تسلط بر VBA است.

قانون طلایی یادگیری VBA

اگر میخواهید کدنویسی با VBA را یاد بگیرید ، یک چیز به شما کمک خواهد کرد. اگرچه کپی و چسباندن کد متداول است ، اما به شما کمک نمی کند VBA را سریع یاد بگیرید. قانونی وجود دارد که از شما میخواهم به آن عمل کنید:

... خودتان کد را تایپ کنید.

چرا از شما میخواهم این کار را انجام دهید؟

زیرا به شما کمک میکند تا زبان VBA را بسیار سریعتر یاد بگیرید. اگر ترجیح میدهید نمونهها را خودتان وارد کنیدAlt + F11 را فشار دهید تا محیط ویرایشگر ویژوال بیسیک فعال شود. سپس یک ماژول استاندارد VBA ایجاد کرده و کد را داخل آن تایپ کنید. کنترل کنید که فایل به درستی تنظیم شده است. به عنوان مثال ، اگر در مثال از دو برگه به نامهایSheet1 و Sheet2 استفاده می شود اطمینان حاصل کنید که فایل شما نیز حاوی برگههایی با این عناوین است.

محتويات كتاب

مجموعه کتابهای برنامهنویسی با اکسل به گونهای فصل بندی شده است که مطالعه مطالب آن برای خواننده راحت و قابل درک باشد. هر فصل پیشنیاز فصل بعدی است ، بر این اساس جلد اول کتاب در یازده فصل تنظیم شده است.

فصل اول: ضبط ماكرو با اكسل

پس از مطالعه این فصل ، میتوانید اولین ماکروی خود را ایجاد کرده و با کدهای VBA مربوطه آشنا شوید.

ایجاد ماکرو با استفاده از ابزار **ضبط ماکرو** نسبتا ساده است و در هفت مرحله آسان انجام میشود. اگر هدف اصلی شما این است که ماکروهایتان را تنها با ابزار ضبط ماکروی اکسل ایجاد کنید ، با مطالعه این فصل برای این کار آماده خواهید بود.

اما اگر هدف شما دستیابی به سطح یک کاربر حرفهای اکسل است ، مطالب این فصل ایده خوبی از مقدمات اولیه برای برنامه نویسی با VBA ارائه میدهد. در واقع این فصل ، تمام آنچه که برای شروع ، ضبط و استفاده از ماکروها در اکسل باید بدانید را پوشش میدهد.

فصل دوم: امنيت ماكروها

VBA یک زبان قدر تمند است. هر کدام از گزینههای خوب و بد که در ادامه لیست می شود ، ناشی از این قدرت است:

- ✓ از یک سو ، ماکروها برای استفاده به جای کارهای تکراری و بهرهوری در استفاده از اکسل بسیار مفید است.
 - ✓ از سوی دیگر ، هنگامی که به طور نامناسب استفاده شود ، میتواند خسارت زیادی ایجاد کند.

آیا احتمال ریسک بالقوه ماکروها عاملی برای عدم استفاده از آنها است؟

بهطور کلی ، نه. همانطور که کریس هافمن بیان میکند:

ماکروها بالقوه خطرناک است اما ، مانند شیر در باغ وحش. باید چگونه رفتار کردن با آن را بدانید.

در این فصل مهمترین مطالبی که در مورد فعال کردن ماکروها باید بدانید توضیح داده شده است. بنابراین درک خوبی از فعالیتها و تنظیمات امنیتی ماکرو برای وضعیت خاص خود خواهید داشت.

فصل سوم: ۱۰ اصطلاح مهم برنامه نویسی با اکسل

در این فصل حداقل با۱۰ اصطلاح ضروری که برای یادگیری برنامه نویسی با VBA نیاز است آشنا خواهید شد. همچنین قادر به درک نحوه استفاده از این اصطلاحات و برخی از بحثهای مربوط به آنها خواهید بود. اگر هدف شما رسیدن به سطح یک کاربر حرفهای اکسل است ، درک مفهوم این اصطلاحات و تسلط بر نحوه کار با آنها اختیاری نیست. علاوه بر این ، خواهید دیدید که چگونه این مفاهیم برای تشکیل یک ماکرو گرد هم میآیند.

مطالعه این فصل ثابت خواهد کرد که برنامه نویسی با VBA چندان هم دشوار نیست.

فصل چهارم: نوعدادههای VBA

هدف اصلی VBA دستکاری دادهها است. بنابراین ، برای حرفهای شدن در VBA و ماکروها باید درک خوبی از نوعدادههای مختلف آن داشته باشید. پس از مطالعه این فصل ، درک خوبی از نوعدادههای مختلف VBA خواهید داشت و قادر خواهید بود نوعداده مناسب برای هر موقعیتی را تعیین کنید. انتخاب نوعداده ممکن است در ابتدا کمی دشوار باشد اما نگران نباشید ...

فصل پنجم: متغيرها در VBA

متغیرها عناصر بسیار انعطاف پذیری است که دائما هنگام کار با VBA با آنها روبرو خواهید شد. به عنوان یک نتیجه از این انعطاف پذیری و حضور همه جا ، مهم است که درک خوبی از نحوه کار با آنها داشته باشید. در غیر این صورت در نهایت با مشکلات زیادی در خطایابی کدهای VBA مواجه خواهید شد و یا حتی بدتر ماکروهای شما ممکن است همان طور که انتظاردارید (بدون اینکه متوجه شوید) کار نکند و نتایج اشتباهی برگرداند. این فصل مهمترین موضوعات مربوط به اعلان متغیرهای VBA را پوشش داده است و چند پیشنهاد و بهترین شیوه در رابطه با ، کار با متغیرهای VBA بیان شده است.

- ✓ متغیر VBA چیست؟
- ✓ چرا بهتر است که متغیرهای VBA را به صراحت اعلان کنید.
 - ✓ چطور میتوان متغیرها را به صورت صریح اعلان کرد.
- ✓ چگونه می توان متغیرهای VBA را با توجه به دامنه و طول عمر آنها اعلان کرد.

- نحوه نام گذاری متغیرها
- چگونه مقادیر یا ابزارهای بیان را به متغیرهای VBA اختصاص دهید.

فصل ششم: آرایهها در VBA

مطالبی که با مطالعه این فصل آشنا خواهید شد ، به این شرح است: آرایههای یک بعدی و چند بعدی VBA شباهتها و تفاوت آنها
آرایههای ثابت و پویا ، شباهتها و تفاوتهای آنها
آرایههای ثابت و پویا ، شباهتها و تفاوتهای آنها
رابطه بین اندازه آرایه ، نوعداده و الزامات حافظه
نحوه اعلان یک آرایه (یک بعدی یا چند بعدی) بسته به اینکه ثابت است یا پویا
تعیین کران پایین آرایهها اگر به صورت صریح اعلان نشده باشد
دستور Base کرده و الزامات حافظه

✓ نحوه استفاده از دستور Erase هنگام کار با آرایهها.

فصل هفتم: روشهای رایج ارجاع به شیء Range در VBA در

شیء Range از مهمترین و پرتکرارترین اشیاء VBA اکسل است. متاسفانه ، موضوع برخی از کاربردهای ان ممکن است دشوار باشد. یکی از دلایل اصلی آن ، این است که روشهای مختلفی برای ارجاع به شیء Range وجود دارد.

این فصل پس از ارائه یک مقدمه کوتاه در مورد موضوع شیء Range به توضیح برخی از رایجترین روشهای ارجاع مناسب به این شیء VBA میپردازد. پس از مطالعه این فصل درک خوبی از شیء Range خواهید داشت و با گزینههای مختلفی که میتوانید برای ارجاع به آن استفاده کنید آشنا خواهید شد.

فصل هشتم: سبک Range.FormulaR1C1 و ویژگی Range.FormulaR1C1 در

با مطالعه این فصل دانش کافی برای شروع به استفاده از سبک R1C1 و ویژگیRange.FormulaR1C1 برای ایجاد ماکروهای کارآمد ، انعطاف پذیر و قدرتمند اکسل خواهید داشت.

برخی از مطالب ارائه شده به این شرح است :

- ✓ سبک A1 وR1C1 چیست؟
- ✓ چرا سبکRange.FormulaR1C1 و ویژگی Range.FormulaR1C1 مفید و مهم است.
- ✓ اقداماتی که باید برای جلوگیری از اشتباهات بالقوه هنگام کار با سبک R1C1 و همچنین آدرسدهی نسبی در VBA در نظر بگیرید.
 - ✓ خصوصیات اصلی ویژگی FormulaR1C1 چیست؟
- ✓ ویژگی Range.FormulaR1C1Local چیست و چگونه میتوان از آن برای مقابله با سازگاری زبان اکسل وقتی در محیط چند زبانه کار میکنید ، استفاده کرد.

فصل نهم: کار با سلولها و محدودهها در VBA

بیشتر زمانی که با اکسل صرف میکنیم ، با سلولها سر و کار داریم. بنابراین منطقی است که برای دستیابی به سطح یک کاربر حرفهای ، یکی از موضوعاتی که باید یاد بگیریم این است که چگونه با سلولها در محیط VBA کار کنیم. این فصل با ارائه مثالهایی به نحوه کار با سلولها و محدودهها در VBA پرداخته است.

فصل دهم: دسترسی به ماکرو از فایلهای مختلف اکسل

هنگامی که ماکرویی را در یک فایل اکسل ایجاد/ضبط میکنید ، میتوان آن را تنها در آن فایل مورد استفاده قرار داد. اما اگر فهرستی از ماکروهای مفید اکسل داشته باشید که میخواهید در تمام فایلهایتان در دسترس بوده و به راحتی استفاده کنید چه؟ در چنین مواردی ، بهتر است این ماکروها را در Personal Macro Workbook (فایل ماکروهای شخصی) ذخیره کنید. انجام این کار به شما اجازه میدهد تا از طریق هر فایلی در سیستم خود به کد این ماکروها دسترسی پیدا کنید.

با مطالعه این فصل نحوه ایجاد و کار با Personal Macro Workbook را یاد خواهید گرفت.

فصل یازدهم: تنظیمات ویرایشگر ویژوال بیسیک

اگر هدف شما دستیابی به سطح یک کاربر حرفهای در ماکروها و VBA است ، باید با تنظیمات ویرایشگر ویژوال بیسیک در ابتدا ویژوال بیسیک آشنا شوید تا بتوانید با آن به خوبی کار کنید. با وجود اینکه ویرایشگر ویژوال بیسیک در ابتدا ممکن است پیچیده به نظر برسد ، با مطالعه این فصل به اندازه کافی در مورد آن خواهید دانست و میتوانید آن را متناسب با نیازهای خود سفارشی و تنظیم کنید ، که شروعی مناسب برای ایجاد ماکروها خواهد بود.


حتی اگر در دنیای VBA تازه کار هستید ، میتوانید به راحتی یک ماکرو ضبط و بعضی از کارهایتان را خود کار کنید. این فصل ، تمام آنچه که برای شروع ، ضبط و استفاده از ماکروها در اکسل باید بدانید را پوشش می دهد.

آمادهاید اولین ماکروی خود را ایجاد کنید؟

پس بیایید مقدمات کار را آماده کنیم:

مقدمه

۱-۱. اضافه کردن زبانه Developer به ریبون اکسل

اولین گام برای ضبط یک ماکرو ، اضافه کردن زبانه Developer به ریبون اکسل است. اگر این زبانه در ریبون موجود است به بند بعد بروید ، در غیر این صورت مراحل زیر را دنبال کنید: روی یکی از زبانههای موجود در ریبون ، راست کلیک کنید و در پنجره باز شده Customize the Ribbon را انتخاب کنید تا پنجره Excel Options باز شود. زبانه Customize Ribbon واقع در نوار لبه سمت چپ این پنجره را انتخاب کنید. سپس از لیست کشویی Customize the Ribbon در سمت راست این پنجره گزینه Main Tabs را انتخاب کنید. از کادر پایین نیز Developer را انتخاب و روی دکمه ok کلیک کنید.

Formulas Proofing	 Choose commands from: (3)		Customize the Bithere	
Proofing	Choose commands from: (2)		Constanting the Dille and (1)	
Causa -			Customize the Ri <u>b</u> bon:	
	Popular Commands	•	Main Tabs	
Language	Add or Remove Filters	•	Main Tabs	
Advanced	Borders	•		
Customize Ribbon	Calculate Now	E	 	
Quick Access Toolbar	(Copy		🛨 📝 Data	
Add-Ins	Custom Sort	<u>A</u> dd >:	> 🖸 🗹 Review	
Total	Cut	<< <u>R</u> em	ove 🕑 🔽 View	-
	A Decrease Font Size		Developer	
	Delete Sheet Columns		🗄 🔽 Add-Ins	
	Delete Sheet Rows		🗄 🗹 Background Removal	
	Email		New Tab <u>N</u> ew Group Rename	
5 M	Font	I.	Customizations: Reset - 0	
	A Font Color	•		
	Font Size	<u>I</u> • •	Import/Export	

•••

با انجام مراحل فوق زبانه Developer به ريبون اضافه خواهد شد.

isual Macros	E Record Macro Use Relative Refe Macro Security Code	erences Add- ins	Excel CC Add-Ins Add Add-Ins	DM Insert	Design Mode I Run D	rties Code ialog	E Map Properties	Import Export
							XML	
						ا اکسل	ضبط ماکرو ہ	۲۰. مراحل
						تيب است:	ماکرو به این تر	حل ضبط
					کنید.	De کلیک	زبانه veloper؛	۱. روی
حالت ضب	مئن شويد كه	Use R مطر	elative F	Reference	نرل گزینه s	Cod با کنن سرو	زارهای گروه e 	۲. در اب ۱
1 6 . d		_		4		ل باشد.	درس نسبی فعا	با اد
File Ho	me Insert	• Page Layout	Formulas	Data Rev	view View	Developer	♀ Tell me what you	u want to do
<u>۹</u>	🔡 Record Macro 🙀 Use Relative Refe	erences Add-	Excel C	OM Insert	Design	code Source	Map Properties	Import

اگر این حالت فعال نیست ، روی Use Relative References کلیک کنید تا رنگ دکمه تغییر و فعال شود.

H 5	- ¢- (ì 🖬 🕯		V)							
File	Home	Insert	Page Layout	Formulas	Data	Rev	view	View	Developer	♀ Tell me what you	ı want to do
Visual M Basic	Macros	cord Macro e Relative f acro Securi	References Add- ins	Excel Add-Ins	COM Add-Ins	Insert	Design Mode	E Prope	code Dialog	Map Properties Expansion Packs	ि Import ि Export

هنگامی که حالت ضبط با آدرس نسبی غیرفعال است (این حالت به صورت پیشفرض غیرفعال میباشد) کدهای VBA همیشه به همان سلولهایی که استفاده میکنید مراجعه میکند ، در حالی که وقتی این حالت روشن است هر عمل ضبط شده توسط اکسل ، هنگام اجرا با سلول فعال مقایسه میشود. به عنوان مثال ، فرض کنید ماکرویی به صورت زیر ضبط کردهاید:

در سلول فعال عبارت This is the best Excel tutorial را تایپ و کلید Enter را فشار میدهید.

اگر در زمان ضبط ماکرو A2 سلول فعال باشد و حالت ضبط با آدرس نسبی را فعال نکنید ، با اجرای ماکرو مهم نیست که کدام سلول فعال است ، ابتدا سلول A2 انتخاب شده و عبارت This is the best Excel tutorial در آن درج و سپس سلول A3 فعال می شود.

اگر از حالت ضبط با آدرس نسبی استفاده کنید ، کد VBA نمی تواند آدرس سلول ها را به صورت مطلق بکار برد. به عنوان مثال ، فرض کنید که سلول A1 سلول فعال باشد و شروع به ضبط ماکرو در حالت ضبط با آدرس نسبی می کنید. حال سلول پایینی ، سلول A2 را انتخاب کنید ، عبارت Inis is the best Excel کنید ، عبارت A2 را انتخاب کنید ، عبارت Inti is the best Excel کنید ، عبارت A2 را انتخاب کنید ، عبارت Intorial را وارد و کلید This is the best Excel را انتخاب کنید ، می کنید ، دیگر سلول A2 انتخاب کنید ، عبارت Intorial می کنید. حال سلول پایینی ، سلول A2 را انتخاب کنید ، عبارت Into is the best Excel کنید ، عبارت A2 را انتخاب کنید ، عبارت A2 را انتخاب کنید ، عبارت A2 را انتخاب می کنید ، دیگر سلول A2 انتخاب نمی شود بلکه نسبت به سلول فعال فعلی ، سلولی انتخاب خواهد شد. به عنوان مثال اگر زمان اجرای ماکرو سلول K3 سلول فعال باشد به سلول is is the best Excel tutorial را درج و سلول K3 را انتخاب می کند.

۳. از ابزارهای گروه Code روی دکمه Record Macro کلیک کنید.

<mark>ਜ਼</mark> 5ਾ ਟੋਾ	à 🖻 🖻	Ŧ					
File Home	Insert	Page Layout	Formulas	Data Revie	v View	Developer	${\mathbb Q}$ Tell me what you want to do
Visual Macros Basic	Record Macro Use Relative R Macro Securit	eferences Ado y	d- Excel CO s Add-Ins Add Add-Ins	M Insert D	E Prop a View sign ode I Run Controls	code Dialog	Import Import Import Import
	بک کنید.	مىشود كل	ضعيت ظاهر	چپ نوار وه	در سمت	Reco را که	یا دکمه ord Macro
	94						2
	95	9	07				- F
	96						- 5
	97		Sheet1	Sheet2	((f)	
	Rea				1 0	/	2
	ىىدھد:	شما امکان م	ن پنجره به ه	اهد شد. ایر	Re باز خو	cord Mac	۴. پنجره محاورهای ro
	1				اب کنید.	ل خود انتخا	🗸 عنوانی برای ماکروی
	Macro3	، Macro2	Macro1	ص میدهد:	رو اختصا	رض به ماک	اکسل یک نام پیش
) بیان میکند	وزش اكسل	گان حوزہ آم	لرين نويسند	(از برجسته	والكنباخ	ور که جان	با این حال ، همانط
			ـت.	رو مفید اس	عنوان ماک	یفی برای :	انتخاب یک نام توص

مواردی وجود دارد که هنگام نامگذاری یک ماکرو باید رعایت کنید. به عنوان مثال ، نام ماکرو باید با یک حرف (نه یک عدد) یا نویسه زیرخط (_) شروع شود و نمیتواند نویسه فاصله یا نویسههای خاصی به غیر از زیرخط داشته باشد. همچنین نباید مشابه عناوین ماکروهای موجود باشد.

به عنوان مثال Best Excel tutorial یک نام قابل قبول نیست ، اما Best_Excel_Tutorial مناسب است.

موضوع نام گذاری ماکروها (هم برای فرآیندهای Sub و هم فرآیندهای Function) در بند ۳–۵–۵ به تفضیل پوشش داده شده است.

Record Macro	? ×
Macro name:	$\left(q_{\lambda} \right)$
Best_Excel_Tutorial	
Shortcut <u>k</u> ey: Ctrl+	907
Store macro <u>i</u> n:	
This Workbook	▼
Description:	
92	
67	OK Cancel

🗸 یک میانبر صفحه کلید به ماکرو اختصاص دهید.

این مرحله اختیاری است. می توانید ماکرو را بدون میانبر صفحه کلید نیز ضبط کنید ، اما اختصاص کلید میانبر امکان میدهد که آن را به سادگی با فشردن چند کلید اجرا کنید.

میانبرهای صفحه کلید به فرم"کلید ترکیبی + Ctrl" است. کلید ترکیبی ، میتواند یک حرف انگلیسی و یا ترکیبی از یک حرف به علاوه کلید Shift باشد.

هنگام ایجاد میانبر صفحه کلید برای ماکروها ، در انتخاب ترکیب کلیدها دقت کنید. اگر یکی از میانبرهای داخلی اکسل را دوباره انتخاب کنید ، این انتخاب کلید میانبر پیشین را غیرفعال میکند. از آنجایی که اکسل چندین کلید میانبر داخلی به فرم" حرف انگلیسی+ Ctrl" دارد احتمال غیرفعال کردن این میانبرها زیاد است. به عنوان مثال ، کلید میانبر Ctrl+B که یک میانبر داخلی برای دستور Bold است را در نظر بگیرید. اگر این کلید را به یک ماکرو اختصاص دهید ، کلید میانبر داخلی اکسل برای دستور Bold غیر فعال میشود. اگر Ctrl+B را فشار دهید ماکرو اجرا میشود ، اما فونت متن انتخاب شده به صورت ضخیم نوشته نخواهد شد.

برای رفع این مشکل بهتر است که میانبرهای صفحه کلید را به فرم" حرف انگلیسی+ Ctrl+ Shift" اختصاص دهید تا احتمال غیر فعال کردن کلید میانبری که از قبل وجود دارد کمتر شود. به این معنا که ، به جای انتخاب Ctrl+Shift+B میتوانیم ترکیب Ctrl+Shift+B را به عنوان کلید میانبر به ماکرو اختصاص دهیم.

Record Macro
Macro name:
Best_Excel_Tutorial
Shortcut key: Ctrl+Shift+ B
Store macro in:
Description:
OK Cancel

در هر صورت ، در مورد ترکیب کلیدی که انتخاب می کنید احتیاط کنید.

🗸 دامنه و حوزه عمل ماکرو را مشخص کنید.

می توانید ماکرو را در فایلی که روی آن کار می کنید (This Workbook) یا یک فایل اکسل جدید (New Workbook) و یا فایل ماکروهای شخصی (Personal Macro Workbook) ذخیره کنید. انتخاب پیش فرض این است که ماکرو در فایلی که روی آن کار می کنید ذخیره شود. در این صورت تنها زمانی که این فایل باز است قادر به استفاده از این ماکرو خواهید بود.

این مرحله تضمین می کند که ماکرو بخشی از فایل اکسل است.

Reco	rd Macro		? <mark>x</mark>	
<u>M</u> ac	ro name:			
Shou	steut kan			
31101	Ctrl+Shift+ B			
Stor	e macro <u>i</u> n:			(QA)
	This Workbook		~	
<u>D</u> eso	cription:			
			37	
		OK	Cancel	

اگر New Workbook را انتخاب کنید ، اکسل یک فایل جدید ایجاد می کند و شما قادر به ضبط و ذخیره ماکرو در این فایل خواهید بود ، اما همانطور که در رابطه با انتخاب This Workbook مشاهده کردید در این مورد نیز ، ماکرو تنها در فایل ایجاد شده کار می کند. گزینه ذخیره پیشرفته تر ، انتخاب Personal Macro Workbook (فایل ماکروهای شخصی) است. بیل ژیلن در کتاب Excel 2013 In Depth فایل ماکروهای شخصی را به این صورت تعریف می کند:

(...) یک فایل اختصاصی برای ذخیره ماکروهای عمومی ، که امکان استفاده از آنها در هر فایلی وجود دارد.

مزیت اصلی ذخیره ماکرو در Personal Macro Workbook این است که این ماکروها میتواند در فایلهای اکسلی که در آینده ایجاد خواهد شد نیز مورد استفاده قرار گیرد. زیرا هنگام استفاده از اکسل ، همه این ماکروها در همان سیستمی که ذخیره شدهاند در دسترس است ، صرف نظر از اینکه در حال کار روی یک فایل اکسل جدید یا فایلی متفاوت از آنچه که ماکرو را در آن ایجاد کردهاید باشید (نحوه ایجاد و کار با Personal Macro Workbook در فصل دهم پوشش داده شده است).

🗸 توضیحی برای ماکرو درج کنید.

این مرحله اختیاری است. با این حال ، بهتر است عادت کنید هر زمان که یک ماکروی جدید ایجاد می کنید در قسمت Description توضیحاتی را ثبت کنید تا همواره خودتان و یا همکارانتان زمان اجرای ماکرو ، از آنچه که از ماکرو انتظار می ود اطلاع داشته باشید.

		باشد.
	8 X	
		(96)
	- 4	
	$a\Lambda^{\prime}$	
ОК	Cancel	
OK	Cancel	
	OK	2 X Y OK Cancel

۵. پس از تعیین عنوان ماکرو ، میانبر صفحه کلید ، مشخص کردن مکان ذخیره ماکرو و درج توضیحات
 ماکرو روی دکمه OK کلیک کنید تا پنجره محاورهای Record Macro بسته شود.

aX

Macro name:			
Best_Excel_Tu	utorial		
Shortcut <u>k</u> ey:			
Ctrl+Shift+ E	3		
Store macro <u>i</u> n:			
This Workbo	ok		
Description:			
		01	Con est

۶. در این مرحله اقداماتی که میخواهید ضبط و ذخیره شود را انجام دهید. دکمه Stop recording ((توقف ضبط) را میتوانید در زبانه Developer مشاهده کنید که نشان میدهد ضبط ماکرو در حال انجام است.

)• ♂• [à 🖻 🖸) -							
File	Home	Insert	Page Layout	Formulas	Data	Review	View	Devel	oper	${f Q}$ Tell me what you want to do
Visual Basic	Macros	op Recordin se Relative R lacro Securit	g eferences Add y	- Excel Add-Ins	COM Add-Ins	Insert Desig	Prop G View n e 🗐 Run	erties Code Dialog	Source	Map Properties Import
	Co	de		Add-Ins		C	ontrols			XML

۷. در پایان روی دکمه Stop recording و یا دکمه توقف ضبط که در سمت چپ نوار وضعیت ظاهر



۱-۳. مثالی از نحوه ضبط ماکرو با اکسل

اگر مراحل توضیح داده شده در بخش قبل را دنبال کنید ، می توانید شروع به ایجاد ماکروهای خود کنید. در این بند ، ماکرویی ضبط خواهد شد که موارد زیر را انجام می دهد:

جارت This is the best Excel tutorial را به <u>سلول فعال وار</u>د می کند.
 عرض ستون سلول فعال را به صورت خودکار تنظیم می کند.
 رنگ زمینه سلول فعال را به قرمز تغییر می دهد.
 رنگ فونت سلول فعال را به آبی تغییر می دهد.

در این مثال ، هنگام کار با پنجره محاورهای Record Macro از مولفههای توضیح داده شده در مراحل هفت گانه بند ۱-۲ استفاده می کنیم. به عبارت دیگر: عنوان ماکرو Best_Excel_Tutorial میانبرصفحه کلید Ctrl+Shift+B و محل ذخیره ماکرو This Workbook خواهد بود.

- سلول A1 را انتخاب كنيد (سلول فعال).
 - ۲. روی زبانه Developer کلیک کنید.
- ۳. دکمه Use Relative References را کلیک کنید تا فعال شود.
 - ۴. سپس روی دکمه Record Macro کلیک کنید.

۵. در پنجره محاورهای قسمتهای مورد نیاز را پر کنید.
 ۶. دکمه OK را کلیک کنید.
 ۷. مجددا سلول A1 (سلول فعال) را انتخاب کنید.
 ۸. متن را داخل سلول <u>تایپ کنید</u> و...

ادامه مراحل ضبط ماکرو را از تصویر متحرک موجود در پوشه فصل ۱ پیوست کتاب دنبال کنید.

تبریک! شما اولین ماکروی اکسل خود را ایجاد کردهاید! اکنون میتوانید این ماکرو را اجرا کنید.

سلولی به غیر از سلول A1 انتخاب کنید. با استفاده از کلیدهای ترکیبی Ctrl+Shift +B ماکرو را اجرا کنید (کلید Ctrl و Shift را نگه دارید سپس کلید B را فشار دهید). این میانبری بود که هنگام ضبط ماکرو به آن اختصاص دادیم (در بند ۳–۵–۷ خواهید دید که برای اجرای ماکرو چند روش دیگر نیز وجود دارد).

چه اتفاقی میافتد؟ آیا نشانگر ماوس به سلول Al بازگشت.

چنین نمی شود، زیرا ماکرو را در حالت آدرس نسبی ضبط کرده اید. بنابراین مکان نما نسبت به سلول فعال فعلی (سلولی که انتخاب می کنید) عمل می کند.

در واقع چون در مرحله ۷ ضبط این ماکرو سلول فعال را انتخاب کردید ، هر سلولی را انتخاب کنید ماکرو در آن سلول اجرا خواهد شد.

با اجرای این ماکرو تمام مراحلی که هنگام ضبط ماکرو انجام دادید ، در کمتر از یک ثانیه انجام می شود.

تمرين:

۱. مراحل فوق را با انتخاب سلول دیگری به غیر از A1 در مرحله ۷ تکرار کنید چه اتفاقی میافتد؟ ۲. مراحل فوق را در حالت ضبط ماکرو با آدرس مطلق نیز تکرار کنید.

امیدوارم بتوانید نخستین ماکروی خود را ایجاد کنید ، حداقل متوجه شوید که ایجاد ماکرو با اکسل ، چنانچه در ابتدا به نظر میرسد ، چندان پیچیده نیست. ماکرویی که ایجاد کردید یک مثال بسیار ساده بود. در فصلهای آتی درباره VBA و مباحث پیچیدهتری که به شما اجازه میدهد ماکروهای قدرتمندی ایجاد کنید ، بحث خواهد شد.

با این حال ، اطلاعاتی که در بندهای پیشین ارائه شد جهت ایجاد طیف گستردهای از ماکروها ، برای مبتدیها کافی خواهد بود. جان والکنباخ در کتاب Excel 2013 Bible توضیح میدهد که:

در بیشتر موارد ، می توانید اقدامات خود را به عنوان یک ماکرو ضبط و سپس به سادگی ماکرو را اجرا کنید؛ نیازی به درک کدی که به طور خودکار ایجاد شده است ندارید.

بنابراین ، بار دیگر برای ایجاد نخستین ماکروی خود ، تبریک می گویم!

۱-۴. ورود به دنیای VBA

مرحله بعدی در ایجاد ماکروهای اکسل ، ورود به دنیای VBA است.

به نظر جان والکنباخ که در بند قبل ذکر شد برگردیم ؛ این نظر بدان معنا نیست که نیازی به یادگیری برنامه نویسی نیست.

اگر متعهد به استفاده از قدرت ماکروی اکسل هستید ، باید VBA را یاد بگیرید.

کدهای نوشته شده با استفاده از VBA به دلایل مختلف قدرتمندتر از ماکروی ایجاد شده با ابزار ضبط ماکرو است ، یکی از مهمترین دلایل این است که با استفاده از کد VBA میتوانید کارهایی انجام دهید که با استفاده از ابزار ضبط ماکرو نمیتواند ثبت شود.

۱-۴-۱. مواردی که ابزار ضبط ماکرو قادر به انجام آن نیست

در کتاب Excel 2013 Bible جان والکنباخ فهرستی از اقداماتی که نمی توان آنها را ضبط کرد ، مانند نمایش پنجرههای محاورهای ، پردازش دادههای چند فایل اکسل ، ایجاد add-in و… لیست کرده است.

دركتاب Excel 2013 VBA and Macros بيل ژيلن و تريسي سيرستاد بيان ميكنند:

مهم است بدانید که ابزارضبط ماکرو هرگز قصد دکمه AutoSum را به طور صحیح ثبت نمیکند.

در ادامه لیستی از کارهایی که نمی توان با ابزار ضبط ماکرو انجام داد ارائه شده است:

- ب با ابزار ضبط ماکرو نمی توانید یک تابع سفارشی ایجاد کنید؛ با استفاده از VBA می توانید توابع سفارشی ایجاد کنید که می توان از آنها مانند سایر توابع برگه اکسل استفاده کرد (نحوه ایجاد توابع سفارشی در جلد دوم کتاب ارائه شده است).
- ✓ نمی توانید کدها را بر اساس یک رخداد اجرا کنید ؛ در VBA می توانید از بسیاری از رخدادها مانند باز شدن فایل اکسل ، اضافه کردن یک برگه ، دوبار کلیک روی سلول و ... برای اجرای کد مرتبط با آن رخداد استفاده کنید (رخدادهای VBA در جلد دوم کتاب پوشش داده شده است).
- ✓ نمی توانید با ابزار ضبط ماکرو حلقه تکرار ایجاد کنید ؛ هنگامی که به صورت دستی کد نویسی می کنید
 می توانید از حلقه های تکرار (از جمله Do until ، Do While ، For Next) در VBA استفاده کنید.
 اما با ضبط کننده ماکرو نمی توانید این کار را انجام دهید.
- ✓ نمی توانید شرایط را تجزیه و تحلیل کنید ؛ اگر یک کد VBA را به صورت دستی بنویسید ، می توانید
 با استفاده از دستور IF...Then...Else برای تجزیه و تحلیل یک شرط استفاده کنید و اگر شرط
 برقرار بود کد اجرا شود (یا اگر برقرار نبود کد دیگری اجرا شود).
- ✓ هنگامی که یک ماکرو ضبط می کنید ، هیچ مولفه ای نخواهید داشت ؛ یک فرآیند می تواند از مولفه های ورودی استفاده کند که برای انجام کاری در ماکرو دریافت می شود. در حالی که با ضبط کننده ماکرو این کار را نمی توان انجام داد ، زیرا ماکروهای ضبط شده مستقل است و به هیچ یک از ماکروهای موجود دیگر وابسته نیست.

۱-۴-۱. دسترسی به محیط ویرایشگر ویژوال بیسیک

برای شروع یادگیری برنامه نویسی با اکسل ، بهتر است دستورات یا کدهایی که هنگام ضبط ماکرو تولید کردهاید را بررسی کنیم. برای انجام این کار ، باید ویرایشگر ویژوال بیسیک (VBE) را باز کنید. راههای مختلفی برای نمایش ویرایشگر ویژوال بیسیک در اکسل وجود دارد:

با استفاده از میانبر صفحه کلید (سادهترین و سریعترین راه)

سادهترین راه برای باز کردن ویرایشگر ویژوال بیسیک این است که از میانبر صفحه کلید ALT+F11 استفاده کنید (کلید ALT را نگه دارید و F11 را فشار دهید). این میانبر به صورت یک حلقه عمل می کند. بنابراین هنگامی که مجددا از آن استفاده کنید به برنامه اکسل باز خواهید گشت (بدون بستن ویرایشگر ویژوال بیسیک).

✓ با استفاده از زبانه Developer

زبانه Developer را انتخاب و در گروه ابزارهای Code روی دکمه Visual Basic کلیک کنید.

🗄 ጛィ 👌 🖻 🖻 ፣							
File Home Insert Page La	yout Formulas	Data	Review	View	Developer	${f Q}$ Tell me what you want to do	
Visual Basic	Add- Excel ins Add-Ins	COM Ir Add-Ins	nsert Design Mode	E Prope	erties Code Dialog	Map Properties Import Expansion Packs Refresh Data	
Code	Add-Ins		Cont	trols		XML	

محیط ویرایشگر ویژوال بیسیک ظاهر می شود که تقریبا به شکل تصویر زیر دیده می شود:

Microsoft Visual Basic for Applications - Book1		
<u> </u>	N L /	
IN 🖬 - 📮 X IS 18. AS 147 (* 1 - 11 - 12 - 12 - 12 - 12 - 12 - 12 -	A KA	
BS VEADwater (Rook1)		
American Microsoft Every Objects		
Sheet1 (Sheet1)		
ThisWorkbook		
🗎 Properties - Sheet1 👝 🖸 💌		
Sheet1 Worksheet		
Alphabetic extractional		
Aphrabetic Categorized		
(Name) Sheet1		
DisplayPageBreaks False		
DisplaykightToLeft Paise		
EnableCalcitation Take		
EnableEconation		
EnableOutlining False		
EnablePivotTable False		
EnableSelection 0 - xlNoRestrictions		
Name Sheet1		
ScrollArea		
StandardWidth 8.43		
Visible -1 - xlSheetVisible		

محیط ویرایشگر ویژوال بیسیک قابل تنظیم است ، بنابراین ممکن است پنجرهای که در رایانه شما نمایش داده می شود کمی متفاوت تر از تصویر فوق باشد.

اولین بار که این پنجره را مشاهده کردم ، سوالم این بود که کدهای ایجاد شده ماکرو کجا قرار گرفتهاند؟

ممكن است شما هم اين سؤال را داشته باشيد ، پس اجازه دهيد پاسخ دهيم:

۱-۴-۳. آناتومی محیط ویرایشگر ویژوال بیسیک

در محیط ویرایشگر ویژوال بیسیک چه میبینید. گزینهها و بخشهای مختلفی وجود دارد که در ابتدا ممکن است کاملا جدید باشد. میتوانید این محیط را به ۶ بخش اصلی تقسیم کنید:

Microsoft Visual Basic for Applications - Re	cord Macro .xlsm		×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>I</u> nsert F <u>o</u> rmat <u>D</u> eb	ug <u>R</u> un <u>T</u> ools <u>A</u> dd-Ins <u>W</u> indow <u>H</u> elp (1)		
S 🔄 - 🛃 X 🗈 🛍 🗚 🔊 (°	🕨 💷 🕍 😻 🚰 🤔 🖄 🛛 🖓 🛛 Ln 1, Col 1	(2) =	
Project - VBAProject 🗖 🗉 🖾	🔀 Record Macro .xlsm - ThisWorkbook (Code)		x
II II 🔁 🔁 📮	(General)	✓ (Declarations)	7
Worksong Microsoft Excel Objects Microsoft Excel Objects Sheet1 (Sheet1) Sheet1 (Sheet1) Microsoft Excel Objects Modules Modules Module1	5	200 100	
Properties - ThisWo Properties - ThisWo ThisWorkbook Aphabetic Categorized Name ThisWorkbook Categorized Name ThisWorkbook Categorized Categorized			E
ChangeHistoryOuratio 0 ChardbataPointTrack True CheckCompatibility False ConflictResolution 1 - xlUserResolution Jate 1904 False DisplayOrawingObject -4104 - xlDisplaySha DisplayOrawingObject -4104 - xlDisplaySha	<u>=</u>]= <	907	► •
EnableAutoRecover True	Immediate		×
EncryptionProvider EnvelopeVisible False Final False ForceFullCalculation False HighlightChangesOnS(False	6		E

۱. نوار منو: این قسمت ، شامل چند منو است. هر یک از این منوها نیز حاوی دستوراتی است که میتوانید آنها را برای تعامل و کار با اجزای مختلف ویژوال بیسیک بکار ببرید. یکی از نکاتی که هنگام کلیک روی هر منو متوجه خواهید شد ، این است که برخی از دستورها دارای میانبر صفحه کلید است که در آن نمایش داده میشود. برای مثال ، به منوی Debug نگاهی بیندازید:

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>I</u> nsert F <u>o</u> rmat	<u>D</u> eb	ug	<u>R</u> un	<u>T</u> o	ools	<u>A</u> d	d-Ins	<u>W</u> indow	<u>H</u> elp
905		Cor	npi <u>l</u> e	VBAI	Proje	ct			
	Θī	Step	o <u>I</u> nto					F8	1
	(ji	Step	o <u>O</u> ve	r				Shift+F8	
	Ē	Step	o O <u>u</u> t				Ctrl+S	Shift+F8	
	÷≣	<u>R</u> un	To C	urso	r			Ctrl+F8	
		<u>A</u> do	Wate	ch					
		<u>E</u> dit	Wato	:h				Ctrl+W	
	63	<u>Q</u> ui	ck Wa	atch.			1	Shift+F9	
R "		Tog	igle Bi	reakp	point			F9	1
		<u>C</u> lea	ar All I	Breal	kpoir	nts	Ctrl+S	Shift+F9	
	⇒	Set	Next S	State	men	t		Ctrl+F9	1
	\$	Sho	w Ne	<u>x</u> t Sta	atem	ent			
	_								_
		٢٢							

۲. **نوار ابزار**: مانند نوار ابزار دسترسی سریع اکسل است که شامل برخی از گزینههای مفید بوده و میتوانید گزینههای بیشتری نیز به آن اضافه کنید. مزیت آن این است که تنها با یک کلیک میتوانید از ابزارهای موجود استفاده کنید. نوار ابزار نمایش داده شده در تصویر فوق ، نوار ابزار استاندارد و پیشفرض ویرایشگر ویژوال بیسیک است. با این وجود شامل ۳ **نوار ابزار** دیگر نیز میباشد:

- ✓ نوار ابزار Debug
- ✓ نوار ابزار Edit
- √ نوار ابزار UserForm

علاوه بر موارد فوق ، ویرایشگر ویژوال بیسیک امکان میدهد که نوار ابزار را ، به چند روش سفارشی کنید. تمام این تنظیمات را با رفتن به منوی View و انتخاب گزینه Toolbars میتوانید تغییر دهید. ویرایشگر ویژوال بیسیک یک لیست با ۴ نوار ابزار مختلف و گزینهای برای دسترسی به پنجره محاورهای Customize نمایش میدهد.



نوار ابزاری که با نماد تیک در سمت چپ آن مشخص شده است در حال حاضر توسط اکسل نمایش داده می شود. می توانید با کلیک روی عنوان نوار ابزار مورد نظر ، آن را اضافه یا حذف کنید. همچنین می توانید چند نوار ابزار را به ویرایشگر ویژوال بیسیک اضافه کنید. اگر روی Customize کلیک کنید ویرایشگر ویژوال بیسیک پنجره محاورهای Customize را نشان می دهد که به شکل زیر است:

Customize		X
Toolbars Commands	Options	
Toolb <u>a</u> rs:		
Debug		^ <u>N</u> ew
Menu Bar		Rename
Standard		Delete
UserForm		9
		<u><u>R</u>eset</u>
	O A	2
		-
	<u></u>	
		Close

با استفاده از این پنجره ، میتوانید جنبههای اضافی مربوط به نوار ابزار که توسط ویرایشگر ویژوال بیسیک نمایش داده میشود را کنترل کنید. برای مثال این پنجره شامل ، امکان کنترل نمایش نوار ابزار ، منوی میانبر یا اضافه کردن نوار ابزار جدید است.

همچنین می توانید با کلیک روی فلش کوچک انتهای نوار ابزار نیز ، با حذف یا اضافه کردن گزینه ها آن را سفارشی کنید. با کلیک روی سه نقطه خاکستری (ابتدای نوار ابزار) و کشیدن آن می توانید نوار ابزار را حرکت دهید.

 Image: Image

۳. پنجره Project Explorer این قسمت محلی است که می توانید لیستی از تمام فایل های اکسل باز و بر گههای آنها و افزونههایی که بار گذاری شدهاند را در آن مشاهده کنید. به عنوان مثال اگر یک فایل با ۳ برگه داشته باشید ، در این پنجره نشان داده می شود. هر فایل اکسل و یا افزونهای که در حال حاضر باز است در Project Explorer به عنوان یک Project (پروژه) جداگانه ظاهر می شود. ماژولها (modules) فرمهای کاربر (userforms) و ماژولهای کلاس (class modules) موجود در فایلهای اکسل نیز در این پنجره لیست می شود. یک پروژه ، همان طور که در وب سایت مایکروسافت بیان شده است ، مجموعهای از ماژولها است. جان والکنباخ در کتاب Excel VBA Programming for Dummies بیان می کند که یک پروژه می تواند به عنوان مجموعهای از اشیاء که به صورت یک طرح کلی مرتب شده است شده است شود. هر پروژه ممکن است شامل پوشه های زیر باشد:

- ✓ یک پوشه به نام Microsoft Excel Objects (اشیاء مایکروسافت اکسل) که همیشه در هر پروژه
 ظاهر می شود. این پوشه معمولا شامل ۲ نوع شیء است (با اشیاء در فصل سوم آشنا خواهید شد):
- ۱. برگههای فایل اکسل مربوطه. به عبارت دیگر هر یک از برگهها ، یک شیء جداگانه محسوب می شود. می شود.
 - ۲. خود فایل اکسل که ThisWorkbook نامیده می شود.
 - پوشه ماژول هنگامی ایجاد می شود که پروژه ، حاوی ماژول های VBA باشد.
- √ اگر پروژه حاوی شیء UserForm که برای ایجاد پنجره محاورهای سفارشی استفاده می شود باشد Project Explorer یک پوشه به نام Forms نشان می دهد.
- ✓ یک پروژه همچنین میتواند شامل ماژولهای کلاس باشد ، در این حالت پنجره پروژه یک پوشه به نام Class Modules را نیز نشان خواهد داد.
- √ در نهایت ، اگر پروژه دارای Reference است یک پوشه هم به نام References نشان داده می شود.



همان طور که در تصویر فوق می بینید ، تنها پروژهای که ظاهر شده است پروژه Record Macro.xlsm است. در پوشه Microsoft Excel Objects می توانید ببینید که فایل اکسل دارای یک برگه است. در نهایت این پروژه شامل یک ماژول VBA نیز است ، بنابراین پوشه ماژول قابل مشاهده است. هیچ شیء UserForm دراود بنابراین ، پوشههای مربوط به آنها ظاهر نمی شود.

ویرایشگر ویژوال بیسیک این امکان را میدهد که قسمتهای مختلف لیست را با کلیک روی "+" یا "-" گسترش دهید یا جمع کنید. هنگامی که یک پوشه گسترش مییابد میتوانید اجزای داخل آن را ببینید.

اگر Project Explorer را نمی بینید ، می توانید آن را با استفاده از روش های زیر نمایش دهید:

✓ از کلید میانبر Ctrl+R استفاده کنید.

Ŧ

✓ روی آیکن Project Explorer در نوارابزار کلیک کنید.

ें 🔣 🔤 - 🛃 | 🐰 🖦 🛍 🥙 (M | 🕨 💷 🕍 💽 🚰 😚 🕨 🕜 | Ln 1, Col 1

e <u>E</u> dit	<u>V</u> iew	Insert	F <u>o</u> rmat	Debug	<u>R</u> un	<u>T</u> ools	<u>A</u> dd-Ins	<u>W</u> indow	<u>H</u> elp
	E	<u>C</u> ode		/ F7	1				
	-8	0 <u>b</u> ject	62	Shift+F7					
	1	<u>D</u> efinition		Shift+F2					
		Last Pøsit	on Ctrl+	Shift+F2					
	· ۴	<u>O</u> bject Bro	owser	F2					
	<u>e</u>	[mmediat	e Window	Ctrl+G					
		Local <u>s</u> Wi	ndow						
	J	Watc <u>h</u> Wi	ndow						
0	44	Call Stac <u>k</u>		Ctrl+L					
	S	<u>P</u> roject Ex	plorer	Ctrl+R					
\sim		Properties	: <u>W</u> indow	F4					
	衆	Toolbo <u>x</u>							
		T <u>a</u> b Order	r						
		<u>T</u> oolbars			•				
		Microsoft	Excel	Alt+F11					

✓ به منوی View بروید و روی گزینه Project Explorer کلیک کنید.

می توانید پنجره Project Explorer را با استفاده از یکی از روشهای زیر مخفی کنید:

- با کلیک دکمه ضربدر
 با راست کلیک در هر نقطهای از پنجره Project Explorer و انتخاب گزینه Hide
- ۴. پنجره Properties این پنجره ویژگی اشیائی که در حال حاضر در Project Explorer انتخاب شده است را نمایش میدهد و این امکان را میدهد که آنها را ویرایش کنید. به این پنجره به ویژه هنگام ایجاد UserForm ها بیشتر نیاز خواهید داشت. اغلب از این پنجره برای نام و تغییر ویژگی Visible برگه اکسل (کار با برگههای اکسل در جلد دوم کتاب ارائه شده است) استفاده می شود.

امکان پنهان کردن یا نمایش پنجره Properties نیز وجود دارد. اگر ویرایشگر ویژوال بیسیک در حال حاضر پنجره Properties را نمایش نمی دهد ، از کلید میانبر F4 استفاده کنید و یا روی نماد پنجره Properties در نوار ابزار کلیک کنید.

े 🔣 🔤 - 📕 🗼 🖻 🖺 🖽 🔊 (୯୦୦) 🖬 🗉 🔟 😫 🚰 🚰 🖓 🖓 Ln 1, Col 1

برای نمایش آن ، همچنین می توانید از منوی view گزینه Properties Window را انتخاب کنید.

<u>File Edit View Insert</u> F<u>o</u>rmat Debug Run Tools Add-Ins Window Help Code ○ **F** Object Shift+F7 Definition Shift+F2 Last Position Strl+Shift+F2 🚰 🛛 Object Browser F2 Immediate Window Ctrl+G 953 Locals Window E. 1 Watch Window Call Stack Ctrl+L Project Explorer Ctrl+R Properties Window F4 Toolbox Tab Order Toolbars ۶ Microsoft Excel Alt+F11

این پنجره را نیز میتوانید به کمک یکی از روشهای زیر مخفی کنید:

Ŧ

√ با کلیک دکمه ضربدر

با راست کلیک در هر نقطهای از پنجره Properties و انتخاب گزینه Hide

۵. پنجره کد: در این پنجره ، کد VBA ضبط شده با اکسل یا کد نوشته شده توسط کاربر نمایش داده می شود. در ادامه خواهید دید که ماکروی ضبط شده شما نیز ، در پنجره کد یک ماژول ذخیره شده است. علاوه بر نمایش کد ، در این پنجره می توانید کدهای VBA را بنویسید یا ویرایش کنید.

برای هر شیء موجود در قسمت Project Explorer مانند فایلها ، برگهها ، ماژولها و ... یک پنجره کد وجود دارد. با انجام هر کدام از موارد زیر میتوانید به پنجرهی کد شیء مورد نظر موجود در Project Explorer دسترسی داشته باشید:

✓ دوبار کلیک روی شیء

استثناء این قانون UserForm است. اگر روی UserForm دوبار کلیک کنید ، ویرایشگر ویژوال بیسیک آن را در نمای طراحی نمایش میدهد ، موضوعی که در جلد دوم کتاب به آن خواهیم پرداخت.



✓ انتخاب شیء و سپس کلیک روی گزینه Code در منوی View

✓ انتخاب شیء و کلیک روی آیکن View Code که در بالای پنجره Project ظاهر می شود. 😹 Project - VBAProject 📃 🔲 🔜 ■ States Sta - Microsoft Excel Objects --- 🎟 Sheet 1 (Sheet 1) 🕷 ThisWorkbook - Modules Module 1 ✓ راست کلیک روی شیء و انتخاب View Code از لیست باز شده View Code -8 View Object VBAProject Properties... I<u>n</u>sert Import File... Export File... Remove Module1... 🚔 <u>P</u>rint... Dockable <u>H</u>ide

- ۶. پنجره Immediate هدف اصلی این پنجره این است که به شما کمک کند تا متوجه خطاهای موجود در کدها شوید و برای کنترل و یا خطایابی کد VBA از آن استفاده کنید. یکی از راههای استفاده از این پنجره ، استفاده از دستور Print.Debug در داخل کد و سپس اجرای ماکرو است.
 اگر نتیجه Print.Debug را در پنجره Immediate دریافت کنیم ، میدانیم که حداقل کد تا آن خط مشکلی ندارد.
 این پنجره به طور پیشفرض قابل مشاهده نیست و میتوانید آن را با انجام هر یک از موارد زیر نمایش دهید:
 با استفاده از میانبر صفحه کلید Gtrl + G
 - ✓ با کلیک روی منوی view و انتخاب گزینه Immediate Window



در تصویر فوق VBAProject دو پوشه را نشان میدهد: Microsoft Excel Objects و ماژولها. میتوانید عناصر داخل پوشه اول (اشیاء مایکروسافت اکسل) را ببینید ، اما اجزای داخل پوشه دوم (ماژولها) قابل مشاهده نیست.

برای گسترش پوشه ماژول ها و دیدن اجزای آن ، روی "+" کلیک کنید. پنجره Project Explorer تقریبا به شکل زیر نمایش داده می شود:



مواردی که در پوشه اشیاء مایکروسافت اکسل ظاهر می شود آشنا است. اما...

ماژول چیست؟

ماژول جایی است که کد VBA در آن ذخیره میشود.

هنگام ضبط ماکروی Best_Excel_Tutorial موارد زیر در ویرایشگر ویژوال بیسیک رخ میدهد:

- ✓ یک ماژول جدید به صورت خودکار ایجاد میشود.
- √ یک ماکرو با نامی که مشخص کردید ثبت میشود (Best_Excel_Tutorial).
- ✔ کدها در پنجره کد ماژول (دقیقتر در پنجره کد Module1) درج و ذخیره می شود.

برای نمایش کد VBA روی Module1 دو بار کلیک کنید. ویرایشگر ویژوال بیسیک کدهای VBA را در **پنجره کد** ماژول نمایش میدهد.

اگر طبق مراحل بند مثال ضبط ماکرو با اکسل (بند۱-۳) ماکروی Best_Excel_Tutorial را ایجاد کرده باشید ، کدهای شما تقریبا به صورت تصویر صفحه بعد خواهد بود.



آن با زبان محاورهای انگلیسی) چگونه میتواند یک گام مفید در انتخاب دستورالعملهایی باشد که ماکرو باید در VBA دنبال کند. بنابراین ، احتمالا قادر به درک برخی از کلمات و شاید برخی از دستورالعملهای ماکروی فوق باشید. به عنوان مثال ، ممکن است خطوط کد مشخص شده در تصویر زیر را در این ماکرو تشخیص دهید و مفهوم آنها را متوجه شوید:

🦓 Record Macro .xlsm - Module1 (Code)	
(General) -	Best_Excel_Tutorial
Sub Best_Excel_Tutorial () ' Best_Excel_Tutorial Macro ' Type "This is the best Excel tuto ' Auto-fit the column width of the ' Color the active cell red. ' Change the font color of the active ' Keyboard Shortcut: Ctrl+Shift+B ' ActiveCell.Select ActiveCell.FormulaR1C1 = "This Selection.Columns.AutoFit With Selection.Interior .Pattern = xlSolid .PatternColorIndex = xlAutor .Color = 255 .TintAndShade = 0 .PatternTintAndShade = 0 End With With Selection.Font .Color = -4165632	rial" into the active cell. active cell. ve cell to blue. is the best Excel tutorial" matic
End With End Sub	
	▼

ActiveCell.Select ✓

Active cell سلولی است که در حال حاضر در برگه اکسل انتخاب شده است (سلول فعال). حتی اگر با اکسل و VBA آشنا نباشید ، حتما میدانید که معنی select چیست. درست است ، احتمالا همان طور که متوجه شدهاید ، این قطعه کد **سلول فعال فعلی را انتخاب میکند**.

Selection.Columns.Autofit \checkmark

این خط کد نیز با "انتخاب" شروع شده و پس از آن اشاره به "ستونها" و "تنظیم خودکار" می شود. اگر به خاطر داشته باشید ، دومین موردی که در ضبط ماکروی Best_Excel_Tutorial باید انجام می شد تنظیم خودکار عرض ستون سلول فعال بود. با توجه به موارد فوق ، حتما تصور می کنید که هدف از این قطعه کد VBA این است که به طور خودکار عرض ستونی که سلول فعال در آن قرار گرفته است را متناسب با طول متنی که ماکرو وارد می کند تنظیم کند.

ممکن است بخواهید بدانید که هر یک از خطوط کد ماکروی فوق به چه معنی است ، اجازه دهید ابتدا بخشی از مبانی کد VBA را توضیح دهیم.

۱-۵-۱. مبانی کدهای VBA ماکروی ایجاد شده

برای درک هر یک از دستورالعملهای ماکرویی که ضبط کردهاید ، اجازه دهید خطوط کد را که برحسب مورد به صورت تکی و مجموعه اجرا میشود ، بررسی کنیم.

اگر مفهوم هر کدام از خطوط کد را در ک نکردید نگران نباشید. هدف این بند ، این است که ایده اولیهای درباره نحوه عملکرد VBA و مهمتر از همه ، دستورالعملهایی که اکسل برای تایپ This is the best Excel tutorial تنظیم خودکار عرض ستون سلول فعال ، تغییر رنگ زمینه سلول فعال به قرمز و تغییر رنگ فونت سلول فعال به آبی اجرا میکند را به شما نشان دهد.

متوجه خواهید شد که (نه تنها در این مثال ، بلکه معمولا هنگام ضبط ماکرو) کد VBA ممکن است شامل برخی از اقداماتی باشد که شما انجام ندادهاید. جان والکنباخ در کتاب Excel 2013 Bible بیان میکند:

این تنها نتیجه فرعی روشی است که اکسل برای ترجمه عملیات کد استفاده میکند.

به عبارت دیگر ، در حال حاضر جای نگرانی در مورد خطوط کدی که شما ایجاد نکردهاید نیست.

پنجره کد ماکرویی که ایجاد کرده اید ، شامل خطوط کد زیر است:

Sub Best_Excel_Tutorial () −۱ آیتم

عبارت Sub اساس یک فرآیند Sub است. این فرآیند یکی از دو نوع فرآیندی است که میتوانید در اکسل ایجاد کنید. نوع دیگر ، فرآیند Function است (توضیحات تکمیلی این نوع فرآیند در جلد دوم کتاب ارائه شده است).

خوب ... این خط کد چه عملی انجام میدهد؟

به سادگی به اکسل می گوید که در حال نوشتن یک فرآیند Sub هستید.

فرآیندهای Sub بایستی همیشه با موارد زیر شروع شود: ۷ عبارت Sub

Best_Excel_Tutorial نام فرآیند ؛ در این مورد

√ يرانتز

علاوه بر این ، فرآیندهای Sub همیشه باید با عبارت End Sub پایان یابد ، همانطور که میتوانید در آخرین خطکد نشان داده شده در تصویر زیر (با شماره ۸ مشخص شده است) ببینید.

Record Macro .xlsm - Module1 (Code) (General) Best_Excel_Tutorial (1)Sub Best_Excel_Tutorial() (2)Best_Excel_Tutorial Macro Type "This is the best Excel tutorial" into the active cell. Auto-fit the column width of the active cell. Color the active cell red. Change the font color of the active cell to blue. Keyboard Shortcut: Ctrl+Shift+B (3) ActiveCell.Select ActiveCell.FormulaR1C1 = "This is the best Excel tutorial" (4) 5 Selection.Columns.AutoFit With Selection.Interior (6) .Pattern = xlSolid .PatternColorIndex = xlAutomatic .Color = 255 .TintAndShade = 0 .PatternTintAndShade = 0 End With With Selection.Font $\overline{7}$.Color = -4165632.TintAndShade = 0 End With End Sub (8 ∃ ∢

آیتم ۲ - خطوط کد با فونت سبز رنگ که با (') شروع شدهاند.

این خطوط تنها برای ارائه اطلاعاتی قرار داده شده است و دارای خصوصیات زیر میباشد: ✓ با یک آپستروف (`) شروع میشوند.

 ✓ VBA اساسا خطوط کدی را که با آپستروف شروع می شود ، نادیده می گیرد. بنابراین ، اجرای ماکرو به این توضیحات وابسته نیست. اگر این خطوط کد را حذف کنید ، ماکرو همچنان به صورت صحیح اجرا می شود. هدف اصلی توضیحات ، نمایش اطلاعات مربوط به ماکرو است و به درک آن کمک میکند. توضیحات ممکن است شامل مواردی مانند ، هدف ماکرو یا آخرین اصلاحات انجام شده در این ماکرو باشد.

♦ ActiveCell.Select – ۳

همان طور که در بند ۱-۴ توضیح داده شد ، این خط کد به اکسل می گوید که سلول فعال فعلی را انتخاب کند. اگر دقیق تر بیان کنیم:

- ✓ ActiveCell اشاره به سلول فعال در برگه فعال اکسل دارد.
- ✓ Select یک شیء را در برگه فعال اکسل انتخاب می کند. شیء در این خط کد ، سلول فعال است
 که ActiveCell به آن ارجاع می دهد.
 - ActiveCell.FormulaR1C1 = "This is the best Excel tutorial" -۴ آيتم ۴-

این خط به اکسل فرمان میدهد که متن This is the best Excel tutorial در سلول فعال وارد شود. اجازه دهید هر یک از قسمتهای مختلف این دستور را بررسی کنیم:

- ✓ در آیتم قبل یاد گرفتید که هدف ActiveCell چیست.
- ✓ FormulaR1C1 به اکسل می گوید فرمول شیء را تعیین کند ، در این مورد شیء ، سلول فعال است
 که ActiveCell به آن اشاره دارد.

R1C1 اشاره به سبک آدرس دهی R1C1 دارد (توضیحات کامل در خصوص FormulaR1C1 در فصل هشتم ارائه شده است). اگر به خاطر داشته باشید ، در ابتدای فصل (بند۱-۲) بیان شد که چرا باید ضبط در حالت آدرس نسبی را فعال کنید و نحوه انجام این کار را یاد گرفتید.

- ✓ "This is the best Excel tutorial " بیان می کند که فرمول چیست که باید در شئ سلول فعال قرار
 گیرد (در این ماکرو فرمول یک رشته متن است).
 - Selection.Columns.AutoFit ۵
 آیتم ۵

همان طور که در بند ۱-۵ بیان شد ، این دستور باعث می شود که اکسل ستون سلول فعال را متناسب با متنی که ماکرو درج کرده است به صورت خودکار تنظیم کند. بخش های مختلف این دستور به شرح زیر است: ✓ Selection : نشان دهنده انتخاب فعلی (در این مورد سلول فعال) است. ✓ Columns: ستون های شیء Selection را انتخاب می کند که در اینجا ستون سلول فعال است. ✓ AutoFit: عرض ستون(های) انتخاب شده (در این مثال) و یا ارتفاع سطر(های) انتخاب شده را به بهترین شکل ممکن ، متناسب با طول متن وارد شده تنظیم می کند.

ایتم ۶- دستور With...End With اول

به بلوک کد زیر توجه کنید که به عنوان یک دستور With...End With شناخته می شود.

```
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .Color = 255
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
```

ماکروی شما در حال حاضر دو مورد از چهار مورد درخواستی را اجرا کرده است:

✓ متن This is the best Excel tutorial را در سلول فعال درج کرده است.
 ✓ عرض ستون سلول فعال را متناسب با طول متن وارد شده تنظیم کرده است.

همان طور که انتظار دارید ، اقدام بعدی که اکسل انجام خواهد داد تغییر رنگ زمینه سلول فعال به قرمز است.

شاید انتظار داشته باشید که تغییر رنگ زمینه سلول فعال یک گام ساده باشد. ولی به نظر میرسد که اکسل نیاز به انجام چند مرحله برای این اقدام دارد.

هدف اصلی دستور With...End With ساده سازی ترکیب اجرای چند دستورالعمل است که همه آنها به یک شیء واحد اشاره میکند بدون اینکه هر بار به آن شیء مراجعه شود. در مورد این مثال این شیء سلول فعال است.

همانطور که در تصویر صفحه بعد مشاهده می *ک*نید ماکروی ضبط شده حاوی دو دستور With...End With میباشد.





حال که یک درک اولیه از آنچه که دستور With...End With انجام میدهد دارید ، اجازه دهید نگاهی به خطوط کد این دستور داشته باشیم:

With Selection.Interior .

این خط به اکسل می گوید که هنگام اجرای دستورالعملها که بخشی از دستور With...End With است باید همیشه به داخل سلول فعال ارجاع شود. نحوه انجام کار چگونه است؟

- ✓ With شروع دستور With...End With است و اکسل را مطلع می کند که دستورالعملهای
 بعدی روی شییء که در این خط ذکر شده است اجرا خواهد شد.
- ✓ Selection.Interior ابزار بیان شیء است که در بند ساختار دستور With...End With توضیح
 داده شده است.

Selection نشان دهنده انتخاب فعلی است که در این مثال سلول فعال است. Interior به درون یک شی اشاره می کند ، که در این مورد داخل سلول فعال است. Pattern = xlSolid.**۲.** این کد اولین خط از دستور With...End With است که اشاره به داخل سلول فعال دارد و به اکسل می گوید رنگ الگوی درون سلول فعال یک رنگ توپر و یکپارچه است. این کار به شرح زیر انجام می شود:

- ✓ Pattern الگوی داخلی را تنظیم می کند.
- ✓ xlSolid مشخص مى كند كه الكو بايد يك رنگ يكپارچه باشد.

.PatternColorIndex = xlAutomatic.r

این خط کد نحوه تنظیم درون سلول فعال را مشخص می کند:

- ✓ PatternColorIndex رنگ داخلی را تنظیم می کند.
- ✓ xlAutomatic مشخص می کند که باید خودکار رنگ شود.
 - .Color = 255.۴

این خط دستوری است که در واقع به اکسل اعلام میکند چه رنگی باید برای پرکردن داخل سلول فعال استفاده کند. Color رنگ سلول را با توجه به شمارهای (در این مورد ۲۵۵ است) مشخص میکند که در ماژول Best_Excel_Tutorial قرمز است.

.TintAndShade = 0.

این خط کد تنظیم وضوح رنگی را که برای پر کردن سلول فعال انتخاب شده است به اکسل دستور میدهد. TintAndShade وضوح یا تیرگی یک رنگ را تنظیم میکند. هنگامی که TintAndShade برابر با 0 تعیین میشود (همانطور که در این مورد است) این ویژگی رنگ تغییری نمیکند.

.PatternTintAndShade = 0.9

همان طور که ممکن است تصور کنید این خط به اکسل انتقال میدهد که الگو باید بدون طرح سایه داخل سلول فعال قرار گیرد. PatternTintAndShade الگوی رنگ و سایه را برای داخل یک شیء ، در اینجا سلول فعال ، تنظیم می کند.

End With.Y

این خط به اکسل اعلام می کند که پایان دستور With...End With است. بنابراین ، خطوط بعدی به یک شیء متفاوت از آنچه که در دستور With...End With به آن اشاره می شد ارجاع می شود. در مورد این مثال ، پایان دستور With...End With اول نشان میدهد که دستورات بعد دیگر به الگوی زمینه سلول فعال ارجاع نمیدهد.

ایتم ۷- دستور With...End With دوم ♦

در آیتم قبل دستور With...End With را یاد گرفتید و با ساختار کلی آن آشنا شدید. بنابراین ، مستقیما به توضیح خط به خط دستور With...End With دوم پرداخته شده است.

همانطور که احتمالا انتظار دارید ، آخرین دستورالعملهایی که هنگام ایجاد ماکرو انجام دادید ، باید انجام شود: تغییر رنگ فونت سلول فعال به رنگ آبی.

With Selection.Font.

همانطور که بیان شد ، این خط شروع دستور With...End With است و به اکسل می گوید که دستورات زیر با شیئ که در اینجا اعلام می شود ، کار می کند. در اینجا این شیء Selection.Font است.

Selection در ماکروی Best_Excel_Tutorial سلول فعال است. به عبارت دیگر Selection.Font به معنای فونت متن در سلول فعال است. بنابراین Selection.Font اساسا اکسل را مطلع میکند که تمام خطوط کد که بخشی از دستور With...End With است ، اشاره به فونت سلول فعال دارد.

این خط مشابه چهارمین خط کد دستور With...End With اول است و به اکسل می گوید چه رنگی باید برای فونت سلول فعال استفاده شود.

Color رنگ را تعیین می کند ، عدد (در این مورد 4165632-) کد رنگی است که به رنگ آبی اشاره دارد.

.TintAndShade = 0.

این خط دقیقا مشابه یکی از خطوط دستور With...End With اول است. از آنجا که TintAndShade وضوح یک رنگ را تعیین می کند ، هنگامی که TintAndShade برابر با 0 (همان طور که در این مورد است) تعیین شود ، این ویژگی فونت سلول فعال تغییری نمی کند.

End With.

این خط پایان دستور With...End With است. بنابراین خطوط کد بعدی به فونت سلول فعال ارجاع نمیدهد.

[.]Color = -4165632.Y

🛠 آيتم ۸ – End Sub

دستور End خاتمه دادن به اجرای چیزی است و در این مورد پایان فرآیند Sub است. این بدان معنی است هنگامی که اکسل این خط کد را اجرا میکند ، ماکرو متوقف خواهد شد. به عبارت دیگر ، این پایان کد نخستین ماکرویی است که ایجاد کردید.

- ۱-۶. چند نکته در خصوص نحوه یادگیری برنامه نویسی با اکسل
- ✓ قسمتهایی از کد VBA را تغییر دهید تا چیزهای جدیدی تجربه کنید.

در خط کد "ActiveCell.FormulaR1C1 = "This is the best Excel tutorial از عبارت This is is the best Excel tutorial استفاده شده است ، میتوانید هر متن دیگری (مثلا نام خودتان) را نیز انتخاب کنید. همچنین میتوانید با تغییر اعداد ، رنگهای دیگری برای زمینه و فونت سلول انتخاب کنید.

به صفحه اصلی اکسل برگردید و دوباره ماکرو را (با استفاده از میانبر صفحه کلید Ctrl+Shift+B که به آن اختصاص دادهاید) اجرا و بررسی کنید چه اتفاقی میافتد.

√ برخی از دستورات را از کد حذف کنید تا ببینید که چگونه بر ماکرو تاثیر میگذارد.

به عنوان مثال ، اگر Selection.Columns.AutoFit را حذف کنید چه اتفاقی رخ میدهد؟ آن را امتحان کنید. به صفحه اصلی اکسل برگردید و دوباره ماکرو را با این تغییر اجرا کنید. چه اتفاقی میافتد؟ آیا این تغییر چیزی بود که انتظار داشتید؟ در مراحل بعدی:

- جیزهای جدیدی امتحان کنید و ماکروهای متفاوتی را ضبط کنید.
- ۲. ویرایشگر ویژوال بیسیک را باز کنید و کد VBA را خط به خط بررسی کنید.

با بررسی کدهای نوشته شده توسط ابزار ضبط ماکرو ، میتوانید ایدههایی برای شروع ماکرونویسی بدست آورید. به عنوان مثال ، اگر از شما خواسته شود یک ستون را با استفاده از کدنویسی فیلتر کنید ، اگر در ابتدا هیچ نظری نداشته باشید که به چه صورت است ، میتوانید به سرعت یک ماکرو ضبط کنید بعد به سراغ ماکروی ایجاد شده رفته تا ببینید اکسل چه کدهایی ثبت کرده است. با بررسی و ویرایش این کدها میتوانید ماکروی خود را بنویسید. ۷-۱ کدهای آماده VBA را کجا قرار دهیم
 دو مکان است که میتوانید کد VBA را به اکسل اضافه کنید:
 ✓ پنجره کد یک ماژول
 ✓ پنجره کد شیء ؛ این شیء میتواند یک فایل اکسل ، یک برگه اکسل ، User Form و ... باشد.

۱-۷-۱. تفاوت پنجره کد ماژول و پنجره کد شیء هنگامی که کدی را به پنجره کد هر یک از اشیاء اضافه میکنید ، اجرای آن وابسته به برخی رخدادهای آن شیء است. به عنوان مثال ، اگر میخواهید به محض باز شدن فایل اکسل پیامی نمایش دهید ، این کد در پنجره کد شیء ThisWorkbook قرار میگیرد (که نشانگر فایل اکسل است). در این مورد رخداد شیء ، باز شدن فایل اکسل است. این رخدادها به عنوان محرکهایی است که با روی دادن آنها ماکرو اجرا خواهد شد (رخدادهای اشیاء به طور کامل در جلد دوم کتاب پوشش داده شده است).

به همین ترتیب ، اگر میخواهید از یک برگه اکسل محافظت کنید ، کد را بایستی در پنجره کد Worksheet وارد کنید. وارد کنید. برعکس ، کدهای ذخیره شده در ماژول باید به صورت دستی (و یا از طریق کدهای دیگر) اجرا شود. هنگامی که ماکرویی ضبط میکنید ، اکسل به طور خودکار یک ماژول ایجاد میکند و کدهای ضبط شده را در آن وارد میکند. حال اگر بخواهید این کد را اجرا کنید ، باید به صورت دستی این کار را انجام دهید.

۱-۷-۲. ایجاد ماژول و اضافه کردن کد به پنجره کد آن

به طور پیشفرض ، ماژول بخشی از فایل اکسل نیست. هنگامی که یک ماکرو ضبط میکنید ، اکسل به طور خودکار در فایل اکسل یک ماژول قبل از شروع به ضبط ایجاد میکند. مواقعی ممکن است بخواهید بدون ضبط ، ماکرویی ایجاد کنید. به کمک یکی از روشهای زیر میتوانید این کار را انجام دهید. روش۱- چگونه یک ماژول VBA ایجاد کنیم

- در ابتدا وارد محیط ویرایشگر ویژوال بیسیک اکسل شوید.
- ۲. در پنجره Project Explorer پروژهای را انتخاب کنید که میخواهید یک ماژول به آن اضافه کنید (اگر پنجره Project Explorer را نمی بینید از میانبر صفحه کلید Ctrl+R استفاده نمایید). به عنوان مثال در تصویر زیر ، یک ماژول به (VBAProject (Book1.xlsm افزوده می شود که تنها پروژه باز و

موجود در این پنجره است. روی یکی از اشیاء فایل اکسل که در پنجره Project Explorer مشاهده میکنید راست کلیک کنید. ۳. در لیست باز شده مکاننما را روی گزینه Insert قرار دهید. ۴. در لیست باز شده روی Module کلیک کنید. 😹 Project - VBAProject 🛛 🗖 💌 🖂 🖃 🚞 WBAProject (Book1.xlsx) 🚊 🖷 Microsoft Excel Objects Bheet1 (Sheet1) 🕷 ThisWorkb 📧 🛛 View Code View Object VBAProject Properties... I<u>n</u>sert UserForm Import File... <u>M</u>odule ~~ Export File... Class Module Remove Sheet 🚔 Print... Dockable <u>H</u>ide روش۲- چگونه یک ماژول VBA ایجاد کنیم در پنجره Project Explorer پروژهای را انتخاب کنید که میخواهید ماژول به آن اضافه شود. سپس از منوی Insert گزینه Module , اانتخاب کنید. File Edit View Insert Format <u>D</u>ebug Run Tools Add-Ins Window <u>H</u>elp Procedure... UserForm Module ~~ C) Class Module File... با اجرای این مراحل ، بلافاصله پوشه Modules ایجاد و یک شیء با عنوان Module1 به آن اضافه می شود.
اگر در سمت چپ محیط ویرایشگر ویژوال بیسیک قسمت Project Explorer را مشاهده کنید خواهید دید که ماژولی با نام Module1 نمایش داده میشود. اگر قبلا یک ماژول وارد کرده باشید ، مراحل فوق یک ماژول جدید ایجاد میکند. هنگامی که ماژول ایجاد شد میتوانید روی آن دوبار کلیک کرده و پنجره کد آن را باز کنید. حال میتوانید کد را داخل پنجره کد کپی کنید و یا خودتان آن را بنویسید و سپس ذخیره کنید.

۱-۲-۲-۱. حذف یک ماژول

درست همانطور که می توانید ماژول های جدید VBA را به یک پروژه اضافه کنید ، می توانید آنها را با استفاده از یکی از دو روش زیر حذف کنید. توجه داشته باشید که ، همانطور که جان والکنباخ در کتاب Excel VBA Programming for Dummies بیان می کند ، تنها ماژول های VBA را می توانید حذف کنید:

راهی برای حذف سایر ماژولهای VBA وجود ندارد ، آنهایی که برای اشیاء Sheet یا ThisWorkbook استفاده می شود.

روش۱- نحوه حذف ماژول VBA

با استفاده از این روش ، میتوانید ماژول VBA را با دنبال کردن دو مرحله ساده زیر حذف کنید.

۱. به پنجره پروژه بروید و ماژولی که میخواهید حذف شود را انتخاب کنید.



۲. به منوی File بروید و Remove module_name را انتخاب کنید.

module_name نام ماژولی است که میخواهید حذف شود. برای مثال ، هنگام حذف Tibe منوی File تقریبا به شکل زیر نمایش داده می شود:

1	<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	Insert	F <u>o</u> rmat	<u>D</u> ebug	<u>R</u> un	<u>T</u> ools	<u>A</u> dd-Ins	<u>W</u> indow	<u>H</u> elp	
	H	<u>S</u> ave B	ook1.xls	ix i		Ctrl+	+S					
		Import	File			Ctrl+	м					
		<u>E</u> xport	File			Ctrl+	۰E				-	
		<u>R</u> emov	/e Modu	ule2								
	h	Print				Ctrl+	.p				a	
	x	<u>C</u> lose a	and Retu	urn to Mi	icrosoft Exc	el Alt+	Q					
	_									C C C C C C C C C C C C C C C C C C C		
									VB	ماژول A	ہ حذف	روش۲- نحو
			كنىد.	ن کلیک	شەد ، است	د حذف	خماهد	. که مہ	وي مادول	ژه د وید و	نحاه د ه	. به ن
			••	*					رری رز , م	J	<i>)</i>];].	* •
						ید.	ک کنے	Ren کلی	nove mod	ule_name	گزینه ۲	۲. روی
									907			
						View C <u>o</u>	de					
					-8	View Ob	ject	<u> </u>	2			
						VBAProj	ect Pro	p <u>e</u> rties				
						I <u>n</u> sert	5		•			
						Import F	File	/				
						<u>E</u> xport F	ile					
						Remove	Modu	le2	1			
						Print			- 1			
						Doc <u>k</u> abl	e					
						<u>H</u> ide						
						-	_					
					0.77							

صرف نظر از اینکه کدام یک از دو روش فوق را برای حذف ماژول VBA انتخاب می کنید ویرایشگر ویژوال بیسیک یک پنجره محاورهای نشان می دهد و از شما می خواهد قبل از حذف ماژول آن را منتقل (export) کنید.

Micro	osoft Visual Basic for Applications
	Do you want to export Module2 before removing it?
	Yes No Cancel Help
S	٤٦

در اغلب موارد ، دلیل حذف یک ماژول VBA این است که کدی داخل آن وجود ندارد. در این صورت ، روی No کلیک کنید.

اگر به هر دلیلی میخواهید ماژول قبل از حذف ذخیره شود ، روی Yes کلیک کنید. این فایل با فرمت bas. ذخیره میشود.

برای انتقال یک ماژول بدون حذف آن ، روی آن راست کلیک و در لیست باز شده گزینه...Export File را انتخاب کنید. از میانبر صفحه کلید Ctrl + E نیز میتوانید استفاده کنید. فایل ذخیره شده را میتوانید با کلیک راست روی پروژه دیگر و انتخاب گزینه ...ImportFile به آن واردکنید.

E	View C <u>o</u> de	
-8	View O <u>b</u> ject	
	VBAProject Prop <u>e</u> rties)/
	Insert	ľ
	Import File	
	Export File	
	<u>R</u> emove Module2	
	Print	
	Doc <u>k</u> able	
	<u>H</u> ide	
_		

۱–۷–۳. اضافه کردن کد به پنجره کد شیء

برای باز کردن پنجره کد یک شیء روی آن دوبار کلیک کنید. وقتی که پنجره کد باز شد ، میتوانید داخل آن کدی بنویسید و یا کدهای آماده را کپی کنید.

۱-۸. ذخیره فایل اکسل حاوی ماکرو

قبل از اکسل ۲۰۰۷ فرمت فایلها xls. بود. اما از نسخه ۲۰۰۷ به بعد xlsx. به عنوان فرمت فایل استاندارد اکسل معرفی شد. فایلهایی که به عنوان xlsx. ذخیره میشود نمیتواند ماکرو را در خود ذخیره کند.

بنابراین اگر فایلی با فرمت xlsx. داشته باشید و ماکرویی در آن ایجاد کنید ، اگر بخواهید آن را ذخیره کنید پنجره محاورهای صفحه بعد نمایش داده خواهد شد.

اگر Yes را انتخاب کنید اکسل به طور خودکار ، تمام کد را از فایل حذف خواهد کرد و آن را با فرمت xlsx. ذخیره می کند.



اما اگر No را کلیک کنید ، پنجره محاورهای زیر ظاهر خواهد شد و بایستی آن را در فرمت xlsm. ذخیره کنید (گزینه Excel Macro - Enabled Workbook را انتخاب کنید) تا این ماکرو را حفظ کنید.

پس اگر فایل اکسلی حاوی ماکرو را ، برای بار نخست بخواهید ذخیره کنید باید آن را در فرمت xlsm. ذخیره کرد. در غیر اینصورت کد ماکرو را از دست خواهید داد.

Organize New folder Image: Search Desktop Organize New folder Image: System Folder
Organize New folder Image: System Folder Image: System Folder Image: System Folder <t< th=""></t<>
Microsoft Excel Favorites Recent Places Desktop Downloads Downloads Documents Documents Pile name: BookLaxism Save as type: Excel Macro-Enabled Workbook (*.xlsm)
Authors: mehraj.mohammadi.ha Tags: Add a tag
A Hide Folders Tools



اگر در حال مطالعه این فصل هستید فرض می کنیم تنظیمات اکسل شما ، همان تنظیمات پیش فرض است. تنظیم پیش فرض این است که تمام ماکروها را آگاهانه غیرفعال کند. این موضوع یک اقدام پیشگیرانه است زیرا ممکن است که ماکرو به نوعی یک ویروس باشد. به همین دلیل است که اگر فایل اکسل حاوی ماکرو را باز کنید ، ممکن است پیام هشدار **ماکروها غیرفعال شدهاند** را که به صورت نوار زرد رنگی در قسمت بالای

صفحه نمایش اکسل ظاهر میشود ، مشاهده کنید.

ļ	SECUR	UTY W	/ARM	NING	Macro	os have	e been (disabled.	Enable Content	Ϊ }
L97			Ŧ	:	×	~	f_{x}		<i>v</i> _b	ž
	А		в		С		D	E	Se G	н

به طور کلی میتوانید تنظیمات امنیتی ماکروهای اکسل را تغییر دهید تا مشخص کنید وقتی فایل اکسل را باز می *ک*نید چه اتفاقی افتد:

- √ کدام ماکرو میتواند اجرا شود.
- √ ماكروها تحت چه شرايطي ميتوانند اجرا شوند.

در هر صورت ، توجه داشته باشید:

قدمه

- ✓ اگر مدیر شبکه شما به منظور جلوگیری از ویرایش فایل ، تنظیمات پیش فرض را تغییر داده باشد قادر به تغییر تنظیمات امنیتی ماکرو نخواهید بود. اگر چنین است ، موضوع را با مدیر شبکه در میان بگذارید.
- ✓ هر تغییری که در تنظیمات امنیتی ماکروی اکسل انجام میدهید ، تنها در اکسل اعمال می شود و به برنامه های دیگر Microsoft Office تاثیری ندارد.

اولین سوال شما شاید این باشد که ...

۲-۱. چرا ماکروهای اکسل به صورت پیش فرض فعال نیست؟
 شاید این مساله برای شما موضوع سادهای باشد ...

فصلی سوم مرابع میں بر نامہ نویسی با اکسل میں بر نامہ نویسی با اکسل

با این حال ، انتظار میرود که VBA از زبانهای محاورهای معمول متفاوت باشد. تصور کنید که در حال یادگیری یک زبان جدید (مانند فرانسوی ، ایتالیایی یا اسپانیایی) هستید. یکی از موضوعات مورد نیاز برای مطالعه ، ساختار آن زبان است و اصطلاحاتی مانند اسمها ، ضمایر ، فعلها و اگر مفهوم دقیق این اصطلاحات را فراموش کردهاید ، یادگیری زبان جدید میتواند کاملا چالش برانگیز باشد. خوب...

زبان برنامهنویسی نیز به طور قابل توجهی متفاوت نیست ، بنابراین برای یادگیری یک زبان مانند VBA باید ساختار آن را درک کنید. از آنجا که این ساختار دقیقا همانند زبانهای محاورهای نیست ، با برخی از اصطلاحات خاص (و متفاوت) که بایستی یاد بگیرید ، روبرو خواهید شد.

اینجاست که مطالب این فصل به کمک شما خواهد آمد. در ادامه برخی از مهمترین اصطلاحاتی که برای یادگیری برنامه نویسی با VBA باید بدانید توضیح داده شده است. در پایان فصل ، میتوانید با ساختار اصلی دستورات VBA آشنا شوید که به شما امکان میدهد مفهوم کدهای این زبان برنامهنویسی را درک کنید و سریعتر یاد بگیرید.

در این فصل ، از شباهت میان اجزای VBA و بخشهایی از زبان محاورهای به میزان مشخص استفاده شده است ، در صورت نیاز از تصاویر نیز استفاده کردهایم.

۲-۳. ماکرو و تفاوت آن با VBA

ماکرو و VBA دقیقا یک چیز نیست ، اگر چه خیلی به هم مرتبط است و گاهی اوقات کاربران اکسل از آنها به یک معنی استفاده میکنند. همان طور که ذکر شد VBA یک زبان برنامه نویسی است که میتواند در چندین نرم افزار مانند Access ، Power Point ، Excel و Access که جزء نرم افزارهای مایکروسافت آفیس است استفاده شود.

ماکرو یک زبان برنامه نویسی نیست. همان طور که جان والکنباخ (یکی از برجستهترین نویسندگان حوزه اکسل) بیان میکند:

ماکرو دنبالهای از دستورالعملها است که برخی از جنبههای اکسل را خودکار میکند.

به عبارت دیگر:

*ماکرو توالی دستورالعملهایی است که میخواهید اکسل دنبال کند تا به هدف خاصی دست پیدا کنید.

* VBA یک زبان برنامه نویسی است که میتوانید از آن برای ایجاد ماکرو استفاده کنید.

۳-۵-۶-۳. فرآیندهای Sub خصوصی یکی از روشهای اجرای یک فرآیند (بند ۳-۵-۷) فراخوانی آن با استفاده از یک فرآیند دیگر است. در حقیقت ، در بعضی از موارد ممکن است فرآیند خاصی داشته باشید که مستقل نیست و طراحی شده است که توسط یکی دیگر از فرآیندهای Sub فراخوانی شود. اگر از اینگونه فرآیندها که یکی از رایجترین روشهای اجرای فرآیندهای Sub است در یک فایل اکسل دارید توصیه می شود آنها را خصوصی کنید. با انجام این کار این فرآیندها در پنجره محاورهای Macro لیست نمی شود.

Sub .اجرا و فراخوانی فرآیندهای Sub

همانطور که یاد گرفتید یک فرآیند مجموعهای از دستورالعملها است. به عنوان یک قاعده دستورالعملهای اجرایی یک فرآیند باعث اجرای آن میشود:

- به عنوان یک واحد؛ و
- √ به ترتيبي که نوشته شدهاند.

به عبارت دیگر ، دستورالعملهای یک فرآیند به شرح زیر اجرا میشود:

- √ دستور اول در ابتدا اجرا می شود ؛
- ✓ سپس دستور دوم اجرا می شود ؛ و
 - ✓ به همین ترتیب ، تا انتها ؛
- أخرين دستور اجرا شده و فرآيند پايان مىيابد.

استثناهایی در این قاعده وجود دارد (مانند استفاده از دستور IF...Then...Else که در جلد دوم کتاب یاد خواهید گرفت).

می توانید یک فرآیند Sub را با روش های مختلفی اجرا و فراخوانی کنید. تا کنون ، تنها یک راه برای اجرای فرآیند فرآیند Sub در اکسل بیان کردیم که استفاده از میانبر صفحه کلید بود. برای اجرای ماکرو با استفاده از این روش ، کلیدهای ترکیبی مربوطه را فشار می دهید. در ادامه توضیحات تکمیلی این روش نیز ارائه خواهد شد. در این بخش ۹ روش که می توانید یک فرآیند Sub را اجرا کنید ، ارائه خواهد شد. مزیت این روش ها این است که حتی اگر فایل اکسل را با کسی که با Sub آشنایی ندارد به اشتراک بگذارید ، مثلا می تواند روی یک دکمه کلیک و ماکرو را اجرا کند (بدون دانستن اینکه چه اتفاقی می افتای می در این بخش ۹ روش که می توانید یک فرآیند Sub آشنایی ندارد به اشتراک بگذارید ، مثلا می تواند روی یک دکمه کلیک و ماکرو را اجرا کند (بدون دانستن اینکه چه اتفاقی می افتد).

Object.Property

Object.Method

شاید سوالاتی در مورد این ساختارها داشته باشید ، بنابراین بیایید به مفهوم یکی از کلمات کلیدی آن نگاهی داشته باشیم

۷-۳. شیء (Object)

همانطور که مشاهده کردید یک فرآیند (procedure) اقداماتی را انجام میدهد. شاید سوال کنید که هدف این اقدامات چیست؟

پاسخ **اشیاء** است. در VBA از اشیاء برای ارجاع به چیزی استفاده میکنیم ، چیزی که عملی روی آن انجام می شود. در محیط اطرافمان هم می توانید اشیاء را در هرجایی پیدا کنید ، مانند لپ تاپی که برای کار با اکسل استفاده می کنید ، اسب ها مثال دیگری از اشیاء است (از کودکی به اسب ها علاقه زیادی داشتم ، لذا در این فصل هم از آنها برای بیان مطالب استفاده کردهام).



در VBA نیز اشیاء چیز خیلی متفاوتی نیست ، به این دلیل که VBA دستکاری اشیاء است. به نظر والکنباخ ، بیش از ۱۰۰ نوع از اشیاء وجود دارد که میتواند در ویژوال بیسیک مورد استفاده قرار گیرد. برخی از نمونههای رایج اشیاء VBA عبارتند از:

- ✓ فایل اکسل (Workbook)
- ✓ برگه اکسل (Worksheet)



جان والکنباخ درباره هدف اصلی VBA در کتاب Excel VBA Programming for Dummies بیان میکند: هدف اصلی VBA دستکاری دادهها است.

عبارت کلیدی در جمله فوق ، داده است. اگر هدف اصلی VBA دستکاری دادهها است ، درک درستی از نوعدادههای مختلف VBA برای مدیریت آن ضروری است.

۱-۴. نوع داده های VBA

مقدمه

نوعدادههای VBA نحوه ذخیره دادهها در حافظه رایانه را تعیین میکند. **نوعداده**ای که به یک متغیر اختصاص داده می شود وابسته به دادهای است که می خواهید متغیر نگه دارد (متغیرها در فصل پنچم به تفضیل بیان شده است).



هنگامی که در کد خود **نوعداده** را برای متغیر مشخص می کنید به VBA می گویید که این متغیر را چگونه ذخیره کند و چه مقدار فضا برای آن اختصاص دهد. به عنوان مثال ، اگر نیاز به استفاده از یک متغیر برای نگه داشتن شماره ماه دارید ، می توان از **نوعداده** BYTE که می تواند مقادیر ۰ تا ۲۵۵ را نگه دارد ، استفاده کنید. از آنجایی که تعداد ماهها بیشتر از ۱۲ نبوده و همیشه به صورت عدد صحیح می باشد ، این انتخاب درست بوده و فضای کمتری از حافظه را نیز اشغال می کند.

برعکس ، اگر نیاز به یک متغیر برای ذخیره تعداد سطرهای اکسل دارید ، باید از یک **نوعداده** استفاده کنید که می تواند یک عدد تا ۱۰۴۸۷۵۶ را در خود جای دهد. بنابراین بهتر است از **نوعداده** Long استفاده کنید.

در جداول صفحه بعد تمام **نوعداده**هایی که میتوانید در VBA استفاده کنید نشان داده شده است.



این مورد مربوط به موضوع **نوع داده متغیر و اعلان متغیر** است. ۲. چگونه می توان تعیین کرد که کدام برنامه یا بخشی از برنامه می تواند از متغیر استفاده کند. این بحث به موضوع **دامنه متغیر و طول عمر** آن اشاره دارد. ۳. چگونه متغیرهای VBA را نام گذاری کنید.

متغیرهای VBA را با تعریف و اعلان آنها میتوان ایجاد کرد. با اعلان متغیر ، نام و ویژگیهای آن را مشخص می کنید. یکی از ویژگیهایی که میتوانید تعیین کنید ، **نوعداده** متغیر است.

به عنوان مثال ، اگر میخواهید عدد صحیح را در یک متغیر ذخیره کنید ، نوعداده شما Integer خواهد بود و اگر میخواهید متنی را ذخیره کنید نوعداده String میباشد. متغیر را قفسهای از یخچال فرض و نام آن را سبز میگذاریم و اعلان میکنیم که در قفسهای با نام سبز تنها سبزیجات قرار گیرد. اعلان متغیر عدد صحیح با Integer نیز مشابه این مثال است. مهارت در تعیین خصوصیات متغیرهایی که در VBA ایجاد میکنید شاید دلیل اصلی اهمیت درک نوعداده VBA است. بیایید ببینیم چرا این گونه است:

۵-۲. چرا باید متغیر ها را به صراحت اعلان و تعریف کرد؟

اعلان متغیر به رایانه می گوید که چه مقدار از فضای حافظه برای استفاده بعدی ذخیره شود.

در حال حاضر ، از توصیههای رایج در خصوص اعلان متغیر VBA استنباط می شود که اعلان متغیر روشی خوب در کد نویسی است.

چند دلیل برای آن وجود دارد و مهمترین آن با نوعدادههای VBA مرتبط است. **نوعداده** VBA نحوه ذخیره داده در حافظه رایانه را مشخص می کند. می توانید نوعداده متغیر را هنگام اعلان آن تعیین کنید ، همچنین می توانید بدون مشخص کردن نوعداده آن را اعلان کنید. اگر نوعداده را برای متغیر مشخص نکنید VBA از نوعداده پیش فرض (Variant) استفاده می کند.

اگر به خاطر داشته باشید تکیه بر Variant علی رغم انعطاف پذیری بیشتر نسبت به سایر نوع داده ها مشکلاتی نیز داشت. بعضی از این مشکلات بالقوه شامل استفاده ناکار آمد از فضای حافظه و اجرای کند برنامه های VBA بود.

وقتی متغیری را تعریف می کنید ، در واقع نوعداده آن را به VBA می گویید و دیگر متکی به Variant نیستید. انجام این کار اطمینان می دهد که VBA تنها متغیر اختصاصی را اعمال می کند و این باعث می شود که از سرعت اجرای کد شما کاسته نشود.



در فصل پنجم در رابطه با موضوع متغیر ، یکی از عناصر ذخیره و بازیابی دادهها در حافظه اصلی رایانه بحث کردیم. آرایهها نیز (در نهایت) نوعی متغیر است.

متغیر معمولی میتواند یک مقدار را ذخیره کند ، اما یک متغیر آرایهای مقادیر متعدد را ذخیره میکند. بنابراین ، کار با آرایههای VBA به میزان مشخصی بسیار شبیه به کار با متغیرها است. با این حال ، آرایه ویژگیهای خاصی دارد که از متغیر متفاوت است. آرایه:

- ✓ امکان میدهد اطلاعات مربوط به گروهها را دستهبندی کنید و به طور کلی ، دستکاری دادهها را راحت میکند. این قابلیت کمک میکند که خطوط کد را سادهتر و کوتاهتر کنیم.
 - کمک می کند روند دریافت اطلاعات را کاهش دهید.
 - میتوانید خوانایی کد را بهبود بخشید.
 - ✓ امکان افزایش سرعت اجرای برنامههای VBA را فراهم می کند.

نشانه قدرت آرایههای VBA توسط ریچارد منسفیلد در Mastering VBA for Microsoft Office 20۱۶ به این شرح ارائه شده است:

آرایهها **نوع فوق متغیر** یا ستاره **متغیر [ها]** است.

پس:

مقدمه

هر چند توضيحات پايه از آرايه در مقدمه فوق ارائه شد ، بياييد آن را عميق تر بررسي كنيم ...

۶-۸. آرایه VBAچیست؟

مرکز توسعه مایکروسافت ، آرایه را: مجموعهای از عناصر ترتیبی یکپارچه با نوعداده مشابه توصیف میکند. به عبارت دیگر آرایه برای ذخیره مجموعهای از عناصر استفاده میشود که نوعداده یکسانی دارند. هر عنصر یک آرایه دارای شماره شاخص منحصر به فرد است. تغییرات اعمال شده به یک عنصر آرایه بر عناصر دیگر تاثیر نمی گذارد.

در کتاب Excel VBA Programming For Dummies والکنباخ تعریف گسترده ای از آرایه ارائه می دهد:

آرایه گروهی از متغیرهایی است که یک نام را به اشتراک میگذارند.

| Element |
|---------|---------|---------|---------|---------|---------|---------|---------|
| #1 | #2 | #3 | #4 | #5 | #6 | #7 | (#8) |

یک شبکه در نظر بگیرید که عناصر در سطرها و ستونهای آن مرتب شده است.

تصویر زیر یک آرایه دو بعدی با ۱۶ عنصر که در ۴ سطر و ۴ ستون تشکیل شده است را نشان میدهد:

Element	Element	Element	Element
#(1, 1)	#(1, 2)	#(1, 3)	#(1, 4)
Element	Element	Element	Element
#(2, 1)	#(2, 2)	#(2, 3)	#(2, 4)
Element	Element	Element	Element
#(3, 1)	#(3, 2)	#(3, 3)	#(3, 4)
Element	Element	Element	Element
#(4, 1)	#(4, 2)	#(4, 3)	#(4, 4)

توجه کنید که چگونه با استفاده از دو عدد ، به هر عنصر آرایه اشاره می کنیم. شماره اول اشاره به مکان بّعد اول که عنصر در آن واقع شده است (در این تصویر ، سطر) و شماره دوم به محل بّعد دوم (در این مورد ، ستون) اشاره دارد. نحوه ارجاع به عناصر آرایه در ادامه بیان خواهد شد.

اگر بٌعد دیگری اضافه کنید ، یک آرایه سه بعدی دریافت خواهید کرد. میتوانید این آرایه را به عنوان یک مکعب تصور کنید.





فصل هفتم

در این فصل ، تمام جزئیات مورد نیاز برای کار با شیء Range را توضیح میدهیم. Range شییء است که برای ارجاع و کار با سلولها در VBA استفاده می شود. با این حال ، اهمیت شیء Range با مطلب فوق به پایان نمی رسد.

جان والكنباخ در كتاب Excel VBA Programming for Dummies بيان مىكند:

اکثر کارهای برنامهنویسی در اکسل روی شیء Range تمرکز دارد.

بیلژیلن (آقای اکسل) و تریسی سیرستاد درکتاب Excel 2013 VBA and Macros اظهار میکنند که شیء Range بیشترین شیء مورد استفاده در VBA اکسل است.

با وجود اهمیت شیء Range آدرس دهی و ارجاع به اشیاء به طور کلی یکی از پیچیده ترین مباحث برای کاربرانی است که در ماکرو نویسی و VBA تازه کار هستند. جان والکنباخ بیان می کند برخی از سردر گمی ها در مورد این موضوع تا حدودی به علت این واقعیت است که:

VBA اجازه میدهد تا با استفاده از روشهای مختلفی به شیء Range ارجاع دهید.

صرف نظر از اینکه موضوع شیء Range چقدر گیج کننده است ، برای دست یابی به سطح یک کاربر حرفهای و متخصص VBA باید به آن مسلط شوید. پس بیایید شروع کنیم:

۱-۷. شیء Range چیست

مقدمه

از شیء Range می توانید برای نشان دادن محدوده ای در یک برگه اکسل استفاده کنید. این بدان معنی است که با استفاده از شیء Range می توانید به موارد زیر ارجاع دهید:

یک سلول واحد
 سطر یا ستونی از سلولها
 محدودهای از سلولها ، صرف نظر از اینکه در مجاورت هم باشد یا خیر
 محدوده سه بعدی

همان طور که از موارد فوق می توان نتیجه گرفت ، اندازه شیء Range می تواند متفاوت باشد. در سطح پایه می توانید به یک سلول واحد ارجاع دهید. از سوی دیگر ، به تمام سلول های یک بر گه اکسل نیز می توانید ارجاع دهید.



اگر از ابزار ضبط ماکرو استفاده کرده باشید ، حتما با سبک R1C1 و **ویژگی**FormulaR1C1 آشنا هستید. به این دلیل که این ابزار همیشه از FormulaR1C1 استفاده میکند.

مقدمه

برای مثال ، حتی اگر از ویژگی Range.Value به منظور مقداردهی یک سلول استفاده کنید ، ابزار ضبط ماکرو از ویژگی Range.FormulaR1C1 برای این هدف استفاده خواهد کرد. برای اینکه موضوع روشن تر شود از ابزار ضبط ماکرو برای ایجاد ماکروی Enter_Value_FormulaR1C1 استفاده کردهایم.

در این ماکرو (i) اعداد ۱ تا ۵ در سلول B5 تا B9 وارد شده و در پایان (ii) سلول B10 انتخاب می شود.



توجه کنید که چگونه ضبط کننده ماکرو هر بار که عددی وارد می کنید ، از ویژگی FormulaR1C1 استفاده می کند.

همانطور که در کتاب Excel 2016 Power Programming with VBA بیان شده است ، به طور معمول اگر از Range Value برای وارد کردن مقدار در یک سلول استفاده کنید ، در عمل استفاده از FormulaR1C1 نیز "همان نتیجه را تولید میکند".

با این حال ، هدف این فصل مقایسه ویژگی Range.FormulaR1C1 با ویژگی Range.Value نیست. بلکه خود ویژگی Range.FormulaR1C1 و **سبک** R1C1 است.

۸-۵-۱. فرمول های اکسل و ملاحظات زبان اکسل

به عنوان موضوع کلی ، توابع اکسل محلی است. به این معنی که نام یک تابع مانند IFERROR با توجه به زبان اکسل تغییر می کند. اجازه دهید موضوع را روشن کنیم:

توابع <mark>متن</mark> مانند FIND ، LEN ، MID ، RIGHT ، LEFT و SEARCH را در نظر بگیرید. این اسامی عناوینی است که در نسخه انگلیسی اکسل استفاده می شود. نام توابع با توجه به زبانی که در

اکسل استفاده می کنید تغییر می کند. جدول زیر نام این توابع را در ۱۰ زبان مختلف نشان می دهد:

SEARCH	FIND	LEN	MID	RIGHT	LEFT	انگلیسی	1
HALLAR	ENCONTRAR	LARGO	EXTRAE	DERECHA	IZQUIERDA	اسپانیایی	2
SUCHEN	FINDEN	LÄNGE	TEIL	RECHTS	LINKS	آلمانی	3
CHERCHE	TROUVE	NBCAR	STXT	DROITE	GAUCHE	فرانسه	4
VIND.SPEC	VIND.ALLES	LENGTE		RECHTS	LINKS	هلندى	5
RICERCA	TROVA	LUNGHEZZA	STRINGA.ESTRAI	DESTRA	SINISTRA	ايتالي <mark>ا</mark>	6
LOCALIZAR	PROCURAR	NÚM.CARACT	EXT.TEXTO	DIREITA	ESQUERDA	برزيل	7
SZUKAJ.TEKST	ZNAJDŹ	DŁ	FRAGMENT. TEKSTU	PRAWY	LEWY	لهستانی	8
SØG	FIND	LÆNGDE	Ø MIDT	HØJRE	VENSTRE	دانمارکی	9
MBUL	BUL	UZUNLUK	PARÇAAL	SAĞDAN	SOLDAN	تركيه	10

اگر فایل اکسل خود را به کسی که از زبان دیگری غیر از انگلیسی برای اکسل استفاده می کند ارسال کنید هنگام باز کردن فایل ، توابع و فرمولهای مورد استفاده به صورت خودکار توسط اکسل ترجمه می شود. با این حال اگر به طور مستقیم فرمولها را با زبان دیگری غیر از انگلیسی به برگه اکسل وارد کنید ، ترجمه خودکار معمولا انجام نمی شود. به عبارت دیگر هنگام کار با ویژگی FormulaR1C1: اگر با استفاده FormulaR1C1 بخواهید فرمولی را بخوانید یا تنظیم کنید ، توابع معمولا به طور خودکار ترجمه نمی شود. ساده ترین راه برای مقابله با مشکالت سازگاری زبان اکسل هنگام کار با ویژگی

FormulaR1C1 استفاده از ویژگی FormulaR1C1Local به جای آن است.

Range.FormulaR1C1Local ويژگى ۸–۵–۲

برای درک اینکه چرا Range.FormulaR1C1Local ممکن است مفید باشد ، بیایید به آنچه که در مورد زبان استفاده شده توسط Range.FormulaR1C1 بیان شد برگردیم. بدین معنا که ویژگی FormulaR1C1 از زبان ماکرو استفاده میکند. از سوی دیگر ، ویژگی Range.FormulaR1C1Local از زبان اکسل استفاده میکند.



فصل نهم

اولین چیزی که موقع فکر کردن به اکسل به ذهنتان میرسد چیست؟

در مورد من ، سلولها است. خوب بالاخره بیشتر زمانی که با اکسل صرف میکنیم ، با سلولها سر و کار داریم. بنابراین ، منطقی است که برای رسیدن به سطح یک کاربر حرفهای اکسل ، یکی از موضوعاتی که باید یاد بگیریم ، این است که چگونه با سلولها در محیط VBA کار کنیم.

در فصل هفتم روشهای مختلف ارجاع به شیء Range را یاد گرفتید. در این فصل ، دقیق تر به اشیاء Range می پردازیم. اکسل در مورد سلولها است و شیء Range ظرفی برای سلولها است.

۹–۱. بررسی اجمالی

مقدمه

یک شیء Range محدوده موجود در یک شیء Worksheet (برگه اکسل) را نشان میدهد. اشیاء Range نیز مانند سایر اشیاء ، دارای ویژگیها (که میتوانید آنها را بررسی کرده و بعضا تغییر دهید) و شیوههای عمل (که اقداماتی را روی شیء انجام میدهد) است.

یک شیء Range می تواند به اندازه یک سلول واحد (به عنوان مثالB4) یا به اندازه ۱۷۱۷۹۸۶۹۱۸۴ سلول (A1:XFD1048576) یک برگه باشد.

وقتی به یک شیء Range ارجاع میدهید ، آدرس همیشه با نمادهای نقل قول احاطه می شود ، مانند این: Range ("A1:C5 ")

اگر Range از یک سلول تشکیل شده باشد ، شما همچنان به نقل قولها نیاز دارید:

Range("K9") اگر محدودهای استفاده می شود که نامگذاری شده است ، می توانید از دستوری مانند این استفاده کنید: Range("PriceList")

(با استفاده از دستور Formulas⇔Defined Names⇔Define Name میتوانید یک محدوده نامگذاری شده ایجاد کنید.)

یک شیء Range می تواند از یک یا چند سطر یا ستون کامل تشکیل شود. با استفاده از ترکیبی مانند زیر می توانید به یک سطر کامل (در این مورد ، سطر ۳) ارجاع دهید:

Range("3:3")

همچنین میتوانید به یک ستون کامل (ستون چهارم در این مثال) ارجاع دهید:

Range("D:D")

دسترسی به ماکرو از فایل های مختلف اکسل

فصل دهم

(Personal Macro Workbook)

هنگامی که ماکرویی را در یک فایل اکسل ایجاد/ ضبط میکنید ، تنها در آن فایل در دسترس بوده و میتوان استفاده کرد. اما اگر فهرستی از ماکروهای مفید اکسل دارید که میخواهید در تمام فایلهایتان در دسترس بوده و به راحتی استفاده کنید ، راه حل چیست؟

در چنین مواردی ، ایده خوبی است که این ماکروها را در Personal Macro Workbook ذخیره کنید.

Personal Macro Workbook .۱-۱۰ چیست؟

مقدمه

Personal Macro Workbook (فایل ماکروهای شخصی) یک فایل ویژه پنهان در سیستم شما است که هر بار اکسل را باز کنید باز خواهد شد.

این فایل اختصاصی مکانی است که میتوانید در آن کد ماکروهایی که از هر فایل اکسلی میتوان به آنها دسترسی داشت را ذخیره کنید. این فایل مکان بسیار خوبی برای ذخیره آن دسته از ماکروهایی است که اغلب استفاده میکنید. به عبارت دیگر ، ماکروهای ذخیره شده در فایل ماکروهای شخصی ، هنگام استفاده از اکسل در همان رایانه در دسترس خواهد بود ، صرف نظر از این که شما در یک فایل اکسل دیگر یا همان فایلی که ماکرو را در آن ایجاد کردهاید کار کنید. این کار باعث صرفهجویی در زمان خواهد شد ، زیرا مجبور نیستید هر بار همان ماکرو را برای هر فایل بازسازی کنید. در عوض ، میتوانید آن را برای یک بار ایجاد و در فایل ماکروهای شخصی ذخیره کنید و از هر فایلی به آن دسترسی داشته باشید.

برای مثال ، فرض کنید به طور منظم از همکاران خود اطلاعاتی دریافت می کنید و هر بار باید دادهها را ویرایش و قالببندی کنید. از آنجا که نیاز دارید هر دفعه همان مراحل را دنبال کنید ، می توانید یک ماکرو برای انجام این کار ایجاد و آن را در فایل ماکروهای شخصی ذخیره کنید. حالا ، هر بار که فایل اکسل را دریافت می کنید تنها باید ماکرو را اجرا کنید (که در فایل ماکروهای شخصی ذخیره شده است) تا ویرایش فایلتان انجام شود.

۲-۱۰. مسیر دسترسی به فایل ماکروهای شخصی

به طور پیشفرض ، در اکسل **فایل ماکروهای شخصی** وجود ندارد. ابتدا باید آن را ایجاد کنید.

برای ایجاد فایل ماکروهای شخصی در اکسل مراحل زیر را دنبال کنید:

- ✓ یک فایل جدید یا هر فایل اکسل موجود را باز کنید.
 - ✓ به زبانه Developer در ریبون بروید.



فصل یازدهم

با این که تنظیمات پیشفرض ویرایشگر ویژوال بیسیک برای اکثر کاربران مناسب است ، این امکان را میدهد تا رابط کاربری و چند ویژگی آن را سفارشی کنید. ۱۱–۱۱. سفارشی کردن ویرایشگر ویژوال بیسیک

مقدمه

اگر میخواهید ویرایشگر ویژوال بیسیک را سفارشی کنید ، اولین چیزی که باید انجام دهید این است که پنجره محاورهای Options را باز کنید. برای انجام این کار ، به منوی Tools بروید و گزینه Options را کلیک کنید.



پنجره محاورهای Options دارای چهار زبانه (در تصویر زیر نشان داده شده است) میباشد که شامل گزینههای سفارشی مختلفی برای ویرایشگر ویژوال بیسیک است.

Options	Editor Format General Docking	
	e Settings Auto Syntax Check Require Variable Declaration	Auto Indent
য য য	Auto List Members Auto Quick Info Auto Data Tips	Tab Width: 4
Wind	low Settings Drag-and-Drop Text Editing	
<u>य</u>	Default to Full Module View Procedure Separator	
	OK	Cancel Help
		,











Copyrighted Material

Keith Darlington



Michael Alexander **Richard Kusleika**

WILEY

AS CREATE COMPELLING ANALYZE AND PRESENT TS CHARTS AND DIACRAMS DATA WITH PRVDT TARKES



