

قدرت برنامه‌نویسی با VBA در
Excel 2019

جلد ۲

میخائیل الکساندر
ریچارد دیک کاسلیکا
(بازنویسی کتاب جان واکنباخ)

ترجمه: حسین یعسوبی
انتشارات پندار پارس

سرشناسه	: الکساندر، مایکل، ۱۹۷۲ - م. Alexander, Michael
عنوان و نام پدیدآور	: قدرت برنامه نویسی با VBA در Excel 2019 / میخائیل الکساندر، ریچارد دیک کاسلیک (بازنویسی کتاب جان واکینام) ترجمه حسین یعسوبی.
مشخصات نشر	: تهران: پندار پارس، ۱۳۹۹-
مشخصات ظاهری	: ج.
شابک	: 978-600-8201-90-8
وضعیت فهرست نویسی	: فیبا
یادداشت	: عنوان اصلی: Excel 2019 bible, 2018.
موضوع	: اکسل مایکروسافت (فایل کامپیوتر)
موضوع	: Microsoft Excel (Computer file)
موضوع	: صفحه گسترده الکترونیکی
موضوع	: Electronic spreadsheets
موضوع	: کسب و کار -- برنامه‌های کامپیوتری
موضوع	: Business -- Computer programs
شناسه افزوده	: کاسلیک، دیک
شناسه افزوده	: Kusleika, Dick
شناسه افزوده	: یعسوبی، حسین، ۱۳۵۲ -، مترجم
رده بندی کنگره	: ۵۵۴۸HF
رده بندی دیویی	: ۵۴/۰۰۵
شماره کتابشناسی ملی	: ۶۱۳۲۶۵۲

انتشارات پندار پارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ www.pendarepars.com
 تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۹۲۶۵۷۸ همراه: ۰۹۱۲۲۴۵۲۳۴۸ info@pendarepars.com



نام کتاب	: قدرت برنامه نویسی با VBA در Excel 2019 (جلد ۲)
ناشر	: انتشارات پندار پارس
تالیف	: میخائیل الکساندر، ریچارد دیک کاسلیک
ترجمه	: حسین یعسوبی
چاپ نخست	: خرداد ۹۹
شمارگان	: ۱۰۰۰ نسخه
طرح جلد	: رامین شکرالهی
چاپ، صحافی	: روز
قیمت	: ۹۵۰۰۰ تومان
شابک	: ۹۷۸-۶۰۰-۸۲۰۱-۹۰-۸

هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد

فهرست فصل‌ها

بخش ۲: تکنیک‌های پیشرفته VBA	۳۱۷
فصل ۸: کار با جدول‌های چرخشی (PIVOT TABLES)	۳۱۹
فصل ۹: کار با نمودارها	۳۳۵
فصل ۱۰: تعامل با برنامه‌های دیگر	۳۸۳
فصل ۱۱: کار با داده‌های خارجی و فایل‌ها	۴۰۹
بخش ۳: کار با USERFORM	۴۵۱
فصل ۱۲: شیوه به‌کارگیری پنجره‌های سفارشی	۴۵۳
فصل ۱۳: آشنایی با USERFORM	۴۷۵
فصل ۱۴: نگاهی به مثال‌های USERFORM	۵۱۵
فصل ۱۵: انجام فنون پیشرفته USERFORM	۵۵۳
بخش ۴: توسعه برنامه‌های اکسل	۵۹۹
فصل ۱۶: ایجاد و استفاده از افزونه‌ها	۶۰۱
فصل ۱۷: کار با ریبون	۶۳۱
فصل ۱۸: کار با منوهای میان‌بر	۶۶۹
فصل ۱۹: تهیه HELP برای برنامه	۶۸۹
فصل ۲۰: کار با ماژول‌های کلاس	۷۰۷
فصل ۲۱: آشنایی با مشکلات سازگاری	۷۲۹

فهرست

۳۱۷.....	بخش ۲: تکنیک‌های پیشرفته VBA
۳۱۹.....	فصل ۸: کار با جدول‌های چرخشی (PIVOT TABLES)
۳۱۹.....	مثالی مقدماتی از جدول محوری
۳۲۰.....	ایجاد یک جدول چرخشی
۳۲۲.....	در حاشیه: داده‌های مناسب برای یک جدول چرخشی
۳۲۲.....	آزمایش کدهای ضبط شده برای جدول چرخشی
۳۲۳.....	پاک‌سازی کدهای ضبط شده جدول چرخشی
۳۲۴.....	در حاشیه: سازگاری جدول چرخشی
۳۲۵.....	ایجاد یک جدول چرخشی پیچیده‌تر
۳۲۷.....	کد ایجاد کننده جدول چرخشی
۳۲۸.....	جدول چرخشی پیچیده‌تر چگونه کار می‌کند
۳۲۹.....	ایجاد جدول‌های سفارشی چندگانه
۳۳۲.....	ایجاد یک جدول چرخشی معکوس
۳۳۵.....	فصل ۹: کار با نمودارها
۳۳۵.....	در اعماق نمودارها
۳۳۶.....	مکان نمودارها
۳۳۶.....	ضبط‌کننده ماکرو و نمودارها
۳۳۷.....	در حاشیه: نکته سازگاری
۳۳۷.....	مدل شیء Chart
۳۳۸.....	ایجاد یک نمودار تعبیه‌ای (Embedded Chart)
۳۴۰.....	ایجاد یک نمودار روی یک برگه نموداری
۳۴۰.....	اصلاح نمودارها
۳۴۲.....	استفاده از VBA برای فعال‌سازی یک نمودار
۳۴۲.....	جابه‌جایی یک نمودار
۳۴۳.....	در حاشیه: آشنایی با Chart Names
۳۴۴.....	استفاده از VBA برای غیرفعال کردن یک نمودار
۳۴۴.....	تعیین فعال بودن یا نبودن یک نمودار
۳۴۵.....	حذف از ChartObjects یا Charts Collection
۳۴۶.....	حلقه‌زنی روی همه نمودارها
۳۴۸.....	سایزبندی و ترازبندی ChartObjectها
۳۴۹.....	ایجاد تعدادی نمودار
۳۵۲.....	برون‌ریزی یک نمودار (Export)
۳۵۳.....	برون‌ریزی همه گرافیک‌ها

۲۵۴	تغییر داده‌های استفاده شده در نمودار
۲۵۴	در حاشیه: آشنایی با فرمول SERIES نمودار
۲۵۶	تغییر داده‌های نمودار برپایه سلول فعال
۲۵۷	استفاده از VBA برای تعیین بازه‌های به‌کار رفته در یک نمودار
۲۶۰	نمایش برچسب‌های سفارشی داده‌های نمودار به‌کمک VBA
۲۶۳	نمایش یک نمودار در یک UserForm
۲۶۶	آشنایی با رویدادهای Chart
۲۶۶	مثالی از کاربرد رویدادهای Chart
۲۶۹	توانایی رویدادها برای یک نمودار تعبیه‌ای
۲۶۹	ایجاد ماژول کلاس
۲۶۹	اعلان یک شیء Chart عمومی
۲۶۹	اتصال شیء اعلانی به نمودار
۲۷۰	نوشتن رویه‌های رسیدگی‌کننده رویداد برای کلاس نمودار
۲۷۰	مثال: استفاده از رویدادهای Chart به‌همراه یک نمودار تعبیه‌ای
۲۷۲	کشف ترفندهای نمودارکشی VBA
۲۷۳	چاپ نمودارهای تعبیه‌ای روی تمام صفحه
۲۷۳	ایجاد نمودارهای غیرپیوندی
۲۷۵	نمایش متن با رویداد MouseOver
۲۷۷	جابه‌جا کردن یک نمودار
۲۷۹	کار با Sparkline Charts
۳۸۳	فصل ۱۰؛ تعامل با برنامه‌های دیگر
۳۸۳	آشنایی با اتوماسیون میکروسافت آفیس
۳۸۴	شناخت مفهوم Binding (انقیاد)
۳۸۴	انقیاد زودرس
۳۸۵	انقیاد دیررس
۳۸۶	درحاشیه: GetObject در مقابل CreateObject
۳۸۶	یک مثال خودکارسازی ساده
۳۸۷	خودکارسازی Access از Excel
۳۸۷	اجرای یک کوئری Access از Excel
۳۸۸	اجرای یک ماکروی Access از Excel
۳۸۹	خودکارسازی Word از Excel
۳۸۹	ارسال داده‌های Excel به یک سند Word
۳۹۱	شبیه‌سازی Mail Merge با یک سند Word
۳۹۳	خودکارسازی PowerPoint از Excel
۳۹۳	فرستادن داده‌های اکسل در یک ارائه PowerPoint

۳۹۴	فرستادن همه نمودارهای Excel در یک ارائه PowerPoint
۳۹۶	تبدیل یک کارپوشه به یک ارائه پاورپوینت
۳۹۷	خودکارسازی Outlook از Excel
۳۹۷	ایمیل کردن کارپوشه فعال به شکل یک پیوست
۳۹۸	ایمیل یک بازه مشخص به شکل یک پیوست
۳۹۹	ایمیل کردن یک کاربرگ به شکل یک پیوست
۴۰۰	ایمیل کردن همه آدرس‌های ایمیل درون فهرست تماس‌ها
۴۰۱	آغاز برنامه‌های دیگر از اکسل
۴۰۲	استفاده از تابع Shell در VBA
۴۰۴	در حاشیه: نمایش پنجره یک پوشه
۴۰۴	استفاده از تابع Windows ShellExecute API
۴۰۶	استفاده از AppActivate
۴۰۶	اجرای پنجره‌های Control Panel
۴۰۹	فصل ۱۱: کار با داده‌های خارجی و فایل‌ها
۴۰۹	کار با اتصالات‌های داده‌های خارجی
۴۱۰	مبانی Power Query
۴۱۶	آشنایی با گام‌های کوئری
۴۱۷	تازه‌سازی داده‌های Power Query
۴۱۸	مدیریت کوئری‌های موجود
۴۱۹	به‌کارگیری VBA برای ایجاد اتصالات‌های پویا
۴۲۱	تکرار همه اتصالات‌های درون یک کارپوشه
۴۲۲	استفاده از ADO و VBA برای بیرون کشیدن داده‌های خارجی
۴۲۳	رشته اتصال
۴۲۴	اعلان یک رکورد دست
۴۲۵	ارجاع به کتابخانه شیء ADO
۴۲۶	درج همه آنها با هم در کد
۴۲۸	استفاده از ADO به همراه کارپوشه فعال
۴۲۸	کوئری گرفتن داده‌ها از یک کارپوشه اکسل
۴۲۹	پیوست کردن رکوردها به یک جدول موجود در اکسل
۴۳۰	کار با فایل‌های متنی
۴۳۱	باز کردن یک فایل متنی
۴۳۲	خواندن یک فایل متنی
۴۳۲	نوشتن یک فایل متنی
۴۳۳	دستیابی به یک شماره فایل
۴۳۳	تعیین یا تنظیم موقعیت فایل

۴۳۳	در حاشیه:ویژگی‌های درون‌ریزی و برون‌ریزی فایل متنی اکسل
۴۳۴	گزاره‌هایی برای خواندن و نوشتن
۴۳۴	مثال‌های دست‌کاری فایل متنی
۴۳۵	درون‌ریزی داده‌ها در یک فایل متنی
۴۳۵	برون‌ریزی یک بازه در یک فایل متنی
۴۳۶	درون‌ریزی یک فایل متنی در یک بازه
۴۳۷	ثبت و ضبط میزان استفاده از اکسل
۴۳۸	فیلتربندی یک فایل متنی
۴۳۹	انجام عملیات مرسوم فایل
۴۳۹	به‌کارگیری گزاره‌های فایل-محور VBA
۴۴۰	یک تابع VBA برای تعیین موجود بودن یک فایل
۴۴۱	یک تابع VBA برای تعیین موجود بودن مسیر یک فایل
۴۴۱	رویه VBA برای نمایش فهرستی از فایل‌های درون یک شاخه
۴۴۳	یک رویه VBA برای نمایش فهرستی از فایل‌ها در شاخه‌های تودرتو
۴۴۵	استفاده از شیء FileSystemObject
۴۴۶	استفاده از FileSystemObject برای تعیین موجود بودن فایل
۴۴۶	استفاده از FileSystemObject برای تعیین موجود بودن یک مسیر
۴۴۶	استفاده از FileSystemObject در فهرست‌کردن اطلاعات همه درایوهای دیسک موجود
۴۴۷	فشرده‌سازی و باز کردن فایل‌ها (Ziping)
۴۴۸	زیپ کردن فایل‌ها
۴۴۹	از زیپ خارج ساختن یک فایل
۴۵۱	بخش ۳؛ کار با USERFORM
۴۵۳	فصل ۱۲؛ شیوه به‌کارگیری پنجره‌های سفارشی
۴۵۳	جایگزین‌های UserForm
۴۵۴	استفاده از یک کادر ورودی (Input Box)
۴۵۴	استفاده از تابع InputBox در VBA
۴۵۷	استفاده از متد Application.InputBox
۴۶۰	استفاده از تابع MsgBox در VBA
۴۶۵	استفاده از متد GetOpenFilename در اکسل
۴۶۸	استفاده از متد GetSaveAsFilename در اکسل
۴۶۹	درخواست یک پوشه
۴۶۹	نمایش پنجره‌های محاوره‌ای پیش‌ساخته اکسل
۴۷۱	در حاشیه: اجرای یک دایرکتوری قدیمی Menu Item
۴۷۲	نمایش یک Data Form

۴۷۳	ساخت فرم داده قابل دسترس
۴۷۳	در حاشیه: افزودن Form Command به نوار ابزار Quick Access
۴۷۳	نمایش یک فرم داده با استفاده از VBA
۴۷۵	فصل ۱۳: آشنایی با USERFORM
۴۷۵	چگونگی رفتار اکسل با پنجره‌های سفارشی
۴۷۶	افزودن یک UseForm جدید
۴۷۷	افزودن کنترل‌ها به یک UserForm
۴۷۷	کنترل‌های Toolbox
۴۷۸	CheckBox
۴۷۸	ComboBox
۴۷۹	CommandButton
۴۷۹	Frame
۴۷۹	Image
۴۷۹	Label
۴۷۹	ListBox
۴۸۰	MultiPage
۴۸۰	OptionButton
۴۸۰	RefEdit
۴۸۰	ScrollBar
۴۸۰	SpinButton
۴۸۱	TabStrip
۴۸۱	TextBox
۴۸۱	ToggleButton
۴۸۱	در حاشیه: استفاده از کنترل‌های روی یک کاربرگ
۴۸۳	اصلاح کنترل‌های UserForm
۴۸۴	تنظیم مشخصه‌های یک کنترل
۴۸۵	استفاده از پنجره Properties
۴۸۷	مشخصه‌های مشترک
۴۸۷	در حاشیه
۴۸۸	در حاشیه: استفاده از یک قرارداد نام‌گذاری
۴۸۹	تطبیق صفحه کلید کاربران
۴۸۹	تغییر ترتیب tab کنترل‌ها
۴۹۰	تنظیم hot keys
۴۹۱	در حاشیه: آزمایش یک UserForm
۴۹۱	نمایش یک UserForm

تنظیم مکان نمایش.....	۴۹۲
نمایش یک یوزر فرم modeless.....	۴۹۲
نمایش یک UserForm بر پایه یک متغیر.....	۴۹۳
لود کردن یک UserForm.....	۴۹۳
درباره رویه‌های رسیدگی کننده رویداد.....	۴۹۳
بستن یک UserForm.....	۴۹۴
ایجاد یک یوزر فرم: مثال.....	۴۹۵
ایجاد UserForm.....	۴۹۵
نوشتن کد برای نمایش پنجره محاوره‌ای.....	۴۹۸
آزمایش پنجره.....	۴۹۸
افزودن رویه‌های رسیدگی کننده رویداد.....	۴۹۹
پنجره پایانی.....	۵۰۱
آموزش درباره رویدادها.....	۵۰۱
رویدادهای SpinButton.....	۵۰۳
رویدادهای آغازگر ماوس.....	۵۰۵
رویدادهای آغازگر کیبورد.....	۵۰۵
در مورد رویدادهای آغازگر کد چه؟.....	۵۰۵
جفت کردن یک SpinButton با یک TextBox.....	۵۰۵
در حاشیه: درباره مشخصه Tag.....	۵۰۷
ارجاع به کنترل‌های UserForm.....	۵۰۸
در حاشیه: آشنایی با کلکسیون کنترل‌ها.....	۵۰۹
سفارشی سازی Toolbox.....	۵۱۰
افزودن صفحات جدید به جعبه ابزار.....	۵۱۰
افزودن صفحات جدید به Toolbox.....	۵۱۰
سفارشی سازی یا ترکیب کنترل‌ها.....	۵۱۰
افزودن دیگر کنترل‌های اکتیو ایکس.....	۵۱۲
ایجاد الگوهای UseForm Template.....	۵۱۳
در حاشیه: تقلید پنجره‌های محاوره‌ای اکسل.....	۵۱۳
چک لیست یوزر فرم.....	۵۱۴
فصل ۱۴: نگاهی به مثال‌های USERFORM.....	۵۱۵
ایجاد یک منوی یوزر فرم.....	۵۱۵
استفاده از CommandButtons در یک UserForm.....	۵۱۵
استفاده از یک لیست باکس در یک یوزر فرم.....	۵۱۶
انتخاب بازه‌ها از یک UserForm.....	۵۱۷
ایجاد یک Splash Screen.....	۵۱۹

۵۲۱.....	غیرفعال کردن دکمه Close یوزر فرم
۵۲۲.....	در حاشیه: پیش‌گیری از شیوع ماکرو
۵۲۲.....	تغییر اندازه یک یوزر فرم
۵۲۴.....	زوم کردن و جابه‌جایی برگه‌ای از روی یک یوزر فرم
۵۲۶.....	بررسی فنون ListBox
۵۲۷.....	افزودن آیتم‌ها به یک کنترل ListBox
۵۲۷.....	افزودن آیتم‌ها به یک لیست‌باکس در زمان طراحی
۵۲۸.....	در حاشیه: اطمینان از به‌کار بردن بازه مناسب
۵۲۸.....	افزودن آیتم‌ها به یک لیست‌باکس در زمان اجرا
۵۲۹.....	افزودن تنها عناصر یکتا به یک لیست‌باکس
۵۳۱.....	تعیین عنصر انتخابی در یک لیست‌باکس
۵۳۲.....	تعیین چند انتخاب در یک لیست‌باکس
۵۳۳.....	لیست‌های چندتایی در یک لیست‌باکس تکی
۵۳۴.....	انتقال آیتم لیست‌باکس
۵۳۶.....	جابه‌جایی عناصر درون یک لیست‌باکس
۵۳۷.....	کار با کنترل‌های ListBox چندستونی
۵۳۹.....	استفاده از یک لیست‌باکس برای انتخاب ردیف‌های کاربرگ
۵۴۱.....	استفاده از یک لیست‌باکس برای فعال‌سازی یک برگه
۵۴۴.....	استفاده از یک تکست‌باکس برای فیلتر کردن یک لیست‌باکس
۵۴۶.....	استفاده از کنترل MultiPage در یک یوزر فرم
۵۴۸.....	استفاده از یک کنترل خارجی
۵۵۰.....	متحرک‌سازی یک لیبل
۵۵۳.....	فصل ۱۵: انجام فنون پیشرفته USERFORM
۵۵۳.....	یک پنجره modeless
۵۵۷.....	در حاشیه: یوزر فرم‌های مدلس در اکسل ۲۰۱۹
۵۵۷.....	نمایش یک نماد پیشرفت
۵۵۸.....	در حاشیه: نمایش پیشرفت در نوار وضعیت
۵۵۹.....	ایجاد یک نماد پیشرفت مستقل
۵۶۰.....	ساخت یوزر فرم برای نماد پیشرفت مستقل
۵۶۰.....	ایجاد کد جلوبرنده‌ی میله پیشرفت
۵۶۱.....	فراخوانی نماد پیشرفت مستقل، از درون کد
۵۶۲.....	فایده یک نماد پیشرفت مستقل
۵۶۲.....	نشان دادن نماد پیشرفتی که در یک یوزر فرم یکپارچه شده
۵۶۳.....	اصلاح یوزر فرم برای یک نماد پیشرفت همراه با کنترل MultiPage

افزودن رویه UpdateProgress برای یک نماد پیشرفت همراه با یک کنترل MultiPage.....	۵۶۴
اصلاح رویه برای نماد پیشرفت همراه با یک کنترل MultiPage.....	۵۶۴
چگونه یک نماد پیشرفت، با یک کنترل MultiPage کار می‌کند.....	۵۶۵
نشان دادن یک نماد پیشرفت، بدون استفاده از یک کنترل MultiPage.....	۵۶۵
ایجاد یک نماد پیشرفت غیرگرافیکی.....	۵۶۶
ایجاد یوزرفرم برای نمایش گام‌ها.....	۵۶۷
اصلاح رویه فراخوانی‌کننده برای استفاده از نماد پیشرفت.....	۵۶۷
ایجاد ویزاردها.....	۵۶۸
تنظیم کنترل MultiPage برای ویزارد.....	۵۶۹
افزودن دکمه‌ها به یوزرفرم ویزارد.....	۵۷۰
برنامه‌نویسی دکمه‌های ویزارد.....	۵۷۰
برنامه‌نویسی وابستگی‌ها در یک ویزارد.....	۵۷۲
انجام دادن کار به کمک ویزارد.....	۵۷۴
تقلید از تابع MsfBox.....	۵۷۵
تقلید از MsgBox: کد MyMsgBox.....	۵۷۶
تابع MyMsgBox چگونه کار می‌کند؟.....	۵۷۷
استفاده از تابع MyMsgBox.....	۵۷۹
یوزرفرمی با کنترل‌های متحرک.....	۵۷۹
یوزرفرمی بدون نوار عنوان.....	۵۸۰
شبیه‌سازی یک نوار ابزار به کمک یک یوزرفرم.....	۵۸۲
تقلید یک Task Pane به کمک یک یوزرفرم.....	۵۸۴
یوزرفرمی با قابلیت تغییر سایز.....	۵۸۵
رسیدگی به چند کنترل یوزرفرم با یک رسیدگی‌کننده رویداد.....	۵۸۹
انتخاب یک رنگ در یک یوزرفرم.....	۵۹۲
نمایش یک نمودار در یک یوزرفرم.....	۵۹۴
ذخیره یک نمودار به شکل یک فایل GIF.....	۵۹۵
تغییر مشخصه Picture کنترل Image.....	۵۹۵
ایجاد یک یوزرفرم نیمه شفاف.....	۵۹۵
پازلی روی یوزرفرم.....	۵۹۷
پوکر ویدئویی روی یک یوزرفرم.....	۵۹۸
بخش ۴: توسعه برنامه‌های اکسل.....	۵۹۹
فصل ۱۶: ایجاد و استفاده از افزونه‌ها.....	۶۰۱
افزونه یا Add-In چیست؟.....	۶۰۱
مقایسه یک افزونه با یک کارپوشه استاندارد.....	۶۰۲

۶۰۳	چرا افزونه‌ها را می‌سازیم؟
۶۰۴	در حاشیه: درباره افزونه‌های COM
۶۰۴	آشنایی با مدیر افزونه‌ی اکسل (Add-in Manager)
۶۰۶	ایجاد یک افزونه
۶۰۷	در حاشیه: چند کلام درباره گذرواژه‌ها
۶۰۷	مثالی از یک Add-in
۶۰۹	افزودن اطلاعات توصیفی برای مثال افزونه
۶۰۹	ایجاد یک افزونه
۶۱۰	در حاشیه: درباره مدیر افزونه‌های اکسل
۶۱۱	نصب یک افزونه
۶۱۱	آزمایش افزونه
۶۱۲	توزیع یک افزونه
۶۱۲	اصلاح یک افزونه
۶۱۲	در حاشیه: چک لیست ایجاد یک افزونه
۶۱۳	مقایسه فایل‌های XLAM و XLSM
۶۱۳	فایل XLAM، عضوی از کلکسیون VBA
۶۱۴	مشهود بودن فایل‌های XLAM و XLSM
۶۱۴	کاربرگ‌ها و برگه‌های نموداری در فایل‌های XLAM و XLSM
۶۱۵	دسترسی به رویه‌های VBA در یک افزونه
۶۱۷	در حاشیه: کار روی یک افزونه محافظت شده
۶۱۹	دست‌کاری افزونه‌ها با VBA
۶۱۹	افزودن یک عنصر به کلکسیون AddIns
۶۲۱	حذف یک عنصر از کلکسیون AddIns
۶۲۱	مشخصه‌های شیء AddIn
۶۲۱	مشخصه Name یک شیء AddIn
۶۲۱	مشخصه Path یک شیء AddIn
۶۲۲	مشخصه FullName یک شیء AddIn
۶۲۲	مشخصه Title یک شیء AddIn
۶۲۳	مشخصه Comments یک شیء AddIn
۶۲۳	مشخصه Installed یک شیء AddIn
۶۲۴	دسترسی به یک افزونه به‌عنوان یک کارپوشه
۶۲۵	رویدادهای شیء AddIn
۶۲۶	بهینه‌سازی بازدهی افزونه‌ها
۶۲۷	مشکلات ویژه با افزونه‌ها
۶۲۷	اطمینان از نصب یک افزونه

۶۲۹.....	ارجاع به دیگر فایل‌ها از یک افزونه.....
۶۳۱.....	فصل ۱۷؛ کار با ریبون.....
۶۳۱.....	مبانی ریبون.....
۶۳۳.....	سفارشی‌سازی ریبون.....
۶۳۳.....	افزودن یک دکمه به ریبون.....
۶۳۵.....	در حاشیه: افزودن ماکرو به یک گروه سفارشی روی ریبون.....
۶۳۶.....	افزودن یک دکمه به نوارابزار Quick Access.....
۶۳۸.....	آشنایی با محدودیت‌های سفارشی‌سازی ریبون.....
۶۳۹.....	ایجاد یک ریبون سفارشی.....
۶۳۹.....	افزودن یک دکمه به یک برگه‌ریبونی موجود.....
۶۳۹.....	در حاشیه: مشاهده خطاهای خود.....
۶۴۰.....	در حاشیه: استفاده از کد RibbonX برای تغییر دادن ریبون.....
۶۴۲.....	کد RibbonX.....
۶۴۴.....	رویه‌های پاسخ یا callback.....
۶۴۴.....	بخش CUSTOM UI.....
۶۴۵.....	درحاشیه: استفاده از تصاویر imageMso.....
۶۴۶.....	افزودن یک چک‌باکس به یک برگه‌ریبونی موجود.....
۶۴۶.....	کد RibbonX.....
۶۴۷.....	کد VBA.....
۶۴۸.....	دموی کنترل‌های ریبون.....
۶۴۹.....	ایجاد یک گروه ریبونی.....
۶۵۰.....	ایجاد کنترل‌ها.....
۶۵۷.....	مثالی از کنترل dynamicMenu.....
۶۵۹.....	مطالب بیشتری درباره سفارشی‌سازی ریبون.....
۶۶۱.....	استفاده از VBA با ریبون.....
۶۶۱.....	دسترسی به یک کنترل ریبون.....
۶۶۲.....	کار با ریبون.....
۶۶۴.....	فعال‌سازی یک برگه.....
۶۶۵.....	ایجاد یک نوارابزار سبک قدیمی.....
۶۶۵.....	محدودیت‌های نوارابزارهای سبک قدیم.....
۶۶۶.....	کد ساخت یک نوارابزار.....
۶۶۹.....	فصل ۱۸؛ کار با منوهای میان‌بر.....
۶۶۹.....	بازبینی CommandBar.....
۶۷۰.....	انواع CommandBar.....
۶۷۰.....	منوهای میان‌بر لیست‌بندی.....

۶۷۱	ارجاع به CommandBarها
۶۷۲	ارجاع به کنترل‌های درون یک CommandBar
۶۷۲	در حاشیه: یافتن یک کنترل
۶۷۳	مشخصه‌های کنترل‌های CommandBar
۶۷۴	نمایش همه عناصر منوی میان‌بر
۶۷۵	استفاده از VBA برای سفارشی‌سازی منوهای میان‌بر
۶۷۵	منوی میان‌بر و رابط کاربری تک-سندی
۶۷۶	در حاشیه: سفارشی‌سازی منوهای میان‌بر با کد RibbonX
۶۷۷	بازگرداندن یک منوی میان‌بر
۶۷۹	غیرفعال کردن یک منوی میان‌بر
۶۷۹	غیرفعال کردن عناصر منوی میان‌بر
۶۷۹	افزودن عنصر جدیدی به منوی میان‌بر Cell
۶۸۱	افزودن یک زیرمنو به یک منوی میان‌بر
۶۸۳	محدودسازی یک منوی میان‌بر به یک تک‌کارپوشه
۶۸۴	در حاشیه: یافتن تصاویر FaceID
۶۸۵	منوهای میان‌بر و رویدادها
۶۸۵	افزودن و حذف دستی منوها
۶۸۵	غیرفعال کردن یا پنهان‌سازی عناصر منوی میان‌بر
۶۸۶	ایجاد یک منوی میان‌بر حساس به بزرگی حروف
۶۸۹	فصل ۱۹؛ تهیه HELP برای برنامه‌نامک
۶۸۹	کمک برای برنامه‌های اکسل خود
۶۹۰	در حاشیه: درباره مثال‌های این فصل
۶۹۱	سامانه‌های Help که از اجزای اکسل استفاده می‌کند
۶۹۲	استفاده از کامنت‌های سلولی برای کمک
۶۹۳	استفاده از یک تکست‌باکس برای کمک
۶۹۴	استفاده از یک کاربرگ برای نمایش متن کمک
۶۹۵	نمایش کمک در یک یوزر فرم
۶۹۶	استفاده از کنترل‌های Label برای نمایش متن کمکی
۶۹۷	در حاشیه: استفاده از کنترل tips روی یک یوزر فرم
۶۹۷	استفاده از یک لیبل جابه‌جایی برای نمایش متن کمک
۶۹۸	استفاده از یک کنترل ComboBox برای انتخاب یک عنوان کمکی
۶۹۹	نمایش Help در یک مرورگر وب
۷۰۰	استفاده از فایل‌های HTML
۷۰۰	استفاده از یک فایل MHTML
۷۰۲	استفاده از سامانه HTML Help

۷۰۳.....	استفاده از متد Help برای نمایش HTML Help
۷۰۴.....	وابسته کردن یک فایل کمکی به برنامه
۷۰۵.....	مرتبط ساختن یک عنوان کمکی با یک تابع VBA
۷۰۵.....	در حاشیه: فعالیت
۷۰۷.....	فصل ۲۰: کار با ماژول‌های کلاس
۷۰۷.....	ماژول کلاس چیست؟
۷۰۸.....	ماژول‌های کلاس درون ساخت
۷۰۹.....	ماژول‌های کلاس سفارشی
۷۰۹.....	کلاس‌ها و اشیاء
۷۰۹.....	اشیاء، مشخصه‌ها، و متدها
۷۱۰.....	ایجاد یک کلاس NumLock
۷۱۱.....	افزودن یک ماژول کلاس
۷۱۱.....	افزودن کد VBA به ماژوال کلاس
۷۱۳.....	استفاده از کلاس CNumLock
۷۱۴.....	کدنویسی مشخصه‌ها، متدها و رویدادها
۷۱۴.....	مشخصه‌های برنامه‌نویسی اشیاء
۷۱۶.....	متدهای برنامه‌نویسی اشیاء
۷۱۶.....	رویدادهای ماژول کلاس
۷۱۷.....	افشای رویداد QueryTable
۷۱۸.....	در حاشیه: فعالیت
۷۲۰.....	ایجاد یک کلاس برای نگه داشتن کلاس‌ها
۷۲۱.....	ایجاد کلاس‌های CSalesReps و CSalesRep
۷۲۳.....	ایجاد کلاس‌های CInvoices و CInvoice
۷۲۴.....	پر کردن کلاس‌های والد با اشیاء
۷۲۵.....	محاسبه کمیسیون‌ها
۷۲۹.....	فصل ۲۱: آشنایی با مشکلات سازگاری
۷۲۹.....	سازگاری چیست؟
۷۳۰.....	انواع مشکلات سازگاری
۷۳۱.....	درحاشیه: Microsoft Office Compatibility Pack
۷۳۲.....	پرهیز از به‌کار بردن ویژگی‌های جدید
۷۳۲.....	در حاشیه: تعیین شماره نسخه اکسل
۷۳۴.....	آیا آن روی یک Mac کار می‌کند؟
۷۳۵.....	کار با اکسل ۶۴ بیتی
۷۳۶.....	ایجاد یک برنامه بین‌المللی
۷۳۸.....	برنامه‌های چند زبانی

۷۳۹.....	ملاحظات زبان VBA
۷۳۹.....	استفاده از مشخصه‌های محلی
۷۴۰.....	شناسایی تنظیمات سیستم
۷۴۲.....	تنظیمات تاریخ و زمان

بخش ۲

تکنیک‌های پیشرفته VBA

۳۱۹.....	فصل ۸: کار با جدول‌های چرخشی (PIVOT TABLES)
۳۳۵.....	فصل ۹: کار با نمودارها.....
۳۸۳.....	فصل ۱۰: تعامل با برنامه‌های دیگر.....
۴۰۹.....	فصل ۱۱: کار با داده‌های خارجی و فایل‌ها.....

فصل ۸

کار با جدول‌های چرخشی (Pivot Tables)

در این فصل می‌خوانیم:

- ✓ ایجاد جدول‌های چرخشی با VBA
- ✓ نگاهی به مثال‌هایی از VBA که جدول‌های چرخشی را ایجاد می‌کنند
- ✓ استفاده از VBA در ساخت یک جدول کاربرگی از یک جدول خلاصه‌سازی

مثالی مقدماتی از جدول محوری

ویژگی Pivot Table که آنرا با جدول چرخشی (یا محوری) نیز می‌شناسیم، ابزاری نوآورانه و مفید در اکسل است که در نسخه Excel 5 معرفی شد و به مرور، ارتقاء یافت. در این فصل، فرض را بر این می‌گذاریم که با آن آشنا هستید و نحوه ایجاد و اصلاح دستی آنرا نیز می‌دانید.

همان‌گونه که احتمالاً می‌دانید، ایجاد یک جدول چرخشی از یک دیتابیس یا لیست، امکان خلاصه‌سازی داده‌ها را به روش‌هایی که بدون استفاده از آنها ممکن نیست، فراهم می‌آورد- و به‌طور شگفت‌آوری سریع و بدون نیاز به فرمول است. برای تولید و اصلاح این جدول‌ها، می‌توان از کدهای VBA نیز استفاده کرد.

در این بخش، کار را با یک مثال ساده از کاربرد VBA در ایجاد یک جدول چرخشی آغاز می‌کنیم.

شکل زیر، یک بازه ساده کاربرگی را نشان می‌دهد که دربردارنده چهار فیلد است: SalesRep، Region، Month و Sales. هر رکورد، فروش‌های یک ماه مشخص از یک نماینده فروش مشخص را توصیف می‌کند.

توجه: فایل تمرینی این مثال، simple pivot table.xlsm نام دارد.

	A	B	C	D
1	SalesRep	Region	Month	Sales
2	Amy	North	Jan	33,488
3	Amy	North	Feb	47,008
4	Amy	North	Mar	32,128
5	Bob	North	Jan	34,736
6	Bob	North	Feb	92,872
7	Bob	North	Mar	76,128
8	Chuck	South	Jan	41,536
9	Chuck	South	Feb	23,192
10	Chuck	South	Mar	21,736
11	Doug	South	Jan	44,834
12	Doug	South	Feb	32,002
13	Doug	South	Mar	23,932

ایجاد یک جدول چرخشی

شکل بعدی، یک جدول چرخشی تولید شده از داده‌ها را در کنار ناحیه کاری PivotTable Fields نشان می‌دهد. این جدول چرخشی، بازدهی فروش را برپایه هر نماینده فروش و ماه نشان می‌دهد. این جدول، با فیلدهای زیر تنظیم شده است:

Region: (منطقه) یک فیلد فیلتر گزارش در جدول چرخشی.

SalesRep: یک فیلد ردیفی در جدول چرخشی.

Month: یک فیلد ستونی در جدول چرخشی.

Sales: یک فیلد مقادیر در جدول چرخشی که از تابع SUM استفاده می‌کند.

چنانچه در حال ساختن جدول چرخشی شکل بعدی، ضبط کننده ماکرو نیز روشن باشد و این عملیات را ضبط کند، کدهای زیر تولید خواهد شد:

```
Sub CreatePivotTable()
    Sheets.Add
    ActiveWorkbook.PivotCaches.Create _
        (SourceType:=xlDatabase, _
        SourceData="Sheet1!R1C1:R13C4", _
        Version:=6).CreatePivotTable _
        TableDestination="Sheet2!R3C1", _
        TableName="PivotTable1", _
        DefaultVersion:=6
```

```

Sheets("Sheet2").Select
Cells(3, 1).Select
With ActiveSheet.PivotTables("PivotTable1").PivotFields("Region")
    .Orientation = xlPageField
    .Position = 1
End With
With ActiveSheet.PivotTables("PivotTable1").PivotFields("SalesRep")
    .Orientation = xlRowField
    .Position = 1
End With
With ActiveSheet.PivotTables("PivotTable1").PivotFields("Month")
    .Orientation = xlColumnField
    .Position = 1
End With
ActiveSheet.PivotTables("PivotTable1").AddDataField _
    ActiveSheet.PivotTables("PivotTable1").PivotFields("Sales"), _
    "Sum of Sales", xlSum
End Sub

```

	A	B	C	D	E	F	G	H	I	J
1	Region	(All)								
2										
3	Sum of Sales									
4		Jan	Feb	Mar	Grand Total					
5	Amy	33488	47008	32128	112624					
6	Bob	34736	92872	76128	203736					
7	Chuck	41536	23192	21736	86464					
8	Doug	44834	32002	23932	100768					
9	Grand Total	154594	195074	153924	503592					
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										

Choose fields to add to report:	
Search	
<input checked="" type="checkbox"/> SalesRep	
Drag fields between areas below:	
Filters Region	Columns Month
Rows SalesRep	Σ Values Sum of Sales
<input type="checkbox"/> Defer Layout Update	Update

اگر این ماکرو را اجرا کنید، تقریباً همیشه به خطا منجر خواهد شد. کد را آزمایش کنید و ببینید که ضبط‌کننده ماکرو، یک نام کاربری (Sheet2) برای این جدول چرخشی، هارد-کد کرده است. اگر این برگه از پیش موجود باشد (یا اگر برگه جدیدی که افزوده می‌شود دارای نام دیگری باشد)، ماکرو به خطا منجر می‌شود. اما یک مشکل جدی این ماکرو این است که ضبط‌کننده ماکرو، نام جدول‌های چرخشی را نیز هارد-کد می‌کند. اگر جدول‌های چرخشی دیگری نیز در کارپوشه باشد، نام جدول جدید، PivotTable1 نخواهد بود.

اما حتی اگر ضبط‌کننده ماکرو کار نکند، کاملاً غیرقابل استفاده نیست. این کد، رهنمودی برای کدنویسی به‌منظور ایجاد جدول‌های چرخشی به ما می‌دهد.

در حاشیه: داده‌های مناسب برای یک جدول چرخشی

جدول چرخشی نیاز دارد داده‌ها به شکل یک دیتابیس مستطیلی باشد. این دیتابیس را می‌توان یا در یک بازه کاربرگ (که می‌تواند یک جدول باشد یا تنها یک بازه معمولی) یا در یک فایل دیتابیس خارجی نگهداری کرد. گرچه اکسل می‌تواند یک جدول چرخشی را از هر دیتابیس تولید کند، اما همه دیتابیس‌ها از این داستان سود نمی‌برند.

عموماً، فیلدها در یک جدول دیتابیس، از دو نوع تشکیل می‌شود:

Data: حاوی یک مقدار یا داده‌ی خلاصه‌سازی شده است. برای نمونه، فیلد Sales یک فیلد داده‌ای است.

Category: داده‌ها را توضیح می‌دهد. برای داده‌های فروش، فیلدهای SalesRep، Region و Month، فیلدهای رده‌بندی به‌شمار می‌روند زیرا داده‌های فیلد Sales را تشریح می‌کنند.

هر جدول دیتابیس که برای یک جدول چرخشی مناسب باشد، می‌گوییم نرمال‌سازی شده است. به‌دیگر سخن، هر رکورد (یا ردیف) دارای اطلاعاتی است که داده‌ها را توضیح می‌دهد.

هر جدول چرخشی می‌تواند به‌تنهایی دارای هر تعداد فیلد داده‌ای و فیلد رده‌بندی باشد. هنگام ایجاد یک جدول چرخشی، معمولاً می‌خواهیم یک یا چند فیلد داده‌ای را خلاصه‌سازی کنیم. برعکس، مقادیر درون فیلدهای رده‌بندی، به‌شکل ردیف‌ها، ستون‌ها یا فیلترها در جدول چرخشی ظاهر می‌شوند.

اگر مفهوم آنرا به خوبی درک نکرده‌اید، کارپوشه normalized data.xlsx را بررسی کنید. این کارپوشه حاوی مثالی از بازه‌ای از داده‌ها پیش و پس از نرمال‌سازی شدن برای متناسب شدن برای جدول چرخشی است.

آزمایش کدهای ضبط شده برای جدول چرخشی

کدهای VBA که با جداول چرخشی کار می‌کند می‌تواند گیج‌کننده باشد. برای حس کردن ماکروی ضبط شده، لازم است درباره چند شیء مربوط به این جداول بدانیم که همه آنها در سامانه Help توضیح داده شده است.

PivotCaches: کلکسیون از اشیاء PivotCache در یک شیء Workbook (داده‌های به‌کار رفته با یک جدول چرخشی، در یک مخزن چرخشی ذخیره می‌شود).

PivotTables: کلکسیونی از اشیاء PivotTable در یک شیء Worksheet

PivotFields: کلکسیونی از فیلدها در یک شیء PivotTable

PivotItems: کلکسیونی از آیتم‌های داده‌ای فردی درون یک رده‌بندی فیلد

CreatePivotTable: متدی که یک جدول چرخشی را با استفاده از داده‌های درون یک مخزن چرخشی ایجاد می‌کند

پاک‌سازی کدهای ضبط شده جدول چرخشی

مانند بیشتر ماکروهای ضبط شده، مثال پیشین آنقدر که باید، کارآمد نبود. و همان گونه که اشاره شد، احتمال تولید خطا دارد. می‌توان کد را ساده‌سازی کرد تا بیشتر قابل فهم شود و نیز از خطا جلوگیری شود. کد دست‌سازی که در زیر می‌بینید، جدول چرخشی مشابهی با کد رویه پیشین می‌سازد:

```
Sub CreatePivotTable()
    Dim PTCache As PivotCache
    Dim PT As PivotTable

    ' Create the cache
    Set PTCache = ActiveWorkbook.PivotCaches.Create( _
        SourceType:=xlDatabase, _
        SourceData:=Range("A1").CurrentRegion)

    ' Add a new sheet for the pivot table
    Worksheets.Add

    ' Create the pivot table
    Set PT = ActiveSheet.PivotTables.Add( _
        PivotCache:=PTCache, _
        TableDestination:=Range("A3"))

    ' Specify the fields
    With PT
        .PivotFields("Region").Orientation = xlPageField
        .PivotFields("Month").Orientation = xlColumnField
        .PivotFields("SalesRep").Orientation = xlRowField
        .PivotFields("Sales").Orientation = xlDataField

        'no field captions
        .DisplayFieldCaptions = False
    End With
End Sub
```

رویه CreatePivotTable ساده‌سازی شده است (و شاید فهم آن آسان‌تر شده باشد)؛ زیرا دو متغیر شیء را اعلان کرده است: PTCache و PT. شیء جدید PivotCache با استفاده از متد Create ایجاد شده است. کاربرگی افزوده می‌شود و برگه فعال می‌شود (مقصد جدول چرخشی). سپس با استفاده از متد Add از کلکسیون PivotTables، یک شیء جدید PivotTable ایجاد می‌شود. آخرین بخش کد، چهار فیلد به جدول چرخشی می‌افزاید و مکان آنها درون جدول را با تخصیص مقداری به مشخصه Orientation تعیین می‌کند.

هر دو بازه داده‌ای ماکروی اصلی هارد-کد شده، برای ایجاد شیء PivotCache (یعنی 'Sheet1!R1C1:R13C4')، و محل جدول چرخشی (Sheet2) استفاده می‌شود. در رویه CreatePivotTable، جدول چرخشی برپایه منطقه جاری پوشش داده شده در سلول A1 است. به این معنا که اگر داده‌های بیشتری افزوده شود، ماکرو به درستی به کارش ادامه می‌دهد.

افزودن کاربرگ پیش از ایجاد جدول چرخشی، نیاز به هارد-کد رفرنس برگه را برطرف می‌سازد. تفاوت دیگری که باقی می‌ماند این است که ماکروی دست‌نویس، نامی برای جدول چرخشی تعیین نکرده است. چون متغیر شیء PT ایجاد شده است، کد شما حتی مجبور نیست با نام به جدول چرخشی ارجاع دهد.

نکته: این کد می‌توانست در کل، در استفاده از ایندکس‌ها به‌جای رشته‌های حرفی برای کلکسیون‌های PivotFields عمومی‌تر ساخته شود. با این روش، اگر کاربر سرستون‌ها را تغییر دهد، باز هم کد کار می‌کند. برای نمونه، کد عمومی‌تر می‌توانست به‌جای PivotFields('Region')، از PivotFields(1) استفاده کند.

مانند همیشه، بهترین روش کسب مهارت در این موضوع، ضبط اکشن‌های خود در یک ماکرو است تا اشیاء، متدها و مشخصه‌ها را درک کنیم. سپس عناوین مهم Help را بخوانیم تا کارکرد این اجزا را متوجه شویم. تقریباً در هر موردی، نیاز به اصلاح ماکروهای ضبط شده داریم. یا، پس از فهم شیوه کار با جدول‌های چرخشی، می‌توان کد را از ابتدا نوشت و بی‌خیال ضبط ماکرو شد.

در حاشیه: سازگاری جدول چرخشی

اگر برنامه‌ای برای به‌اشتراک‌گذاری کارپوشه‌ای حاوی یک جدول چرخشی با کاربران نسخه‌های قدیمی اکسل دارید، باید حواستان به مسئله سازگاری باشد. اگر نگاهی به ماکروی ضبط شده بخش "ایجاد یک جدول چرخشی" بیاندازید، گزاره زیر را می‌بینید:

```
DefaultVersion:=6
```

اگر کارپوشه شما در حالت سازگاری باشد، گزاره ضبط شده به این شکل است:

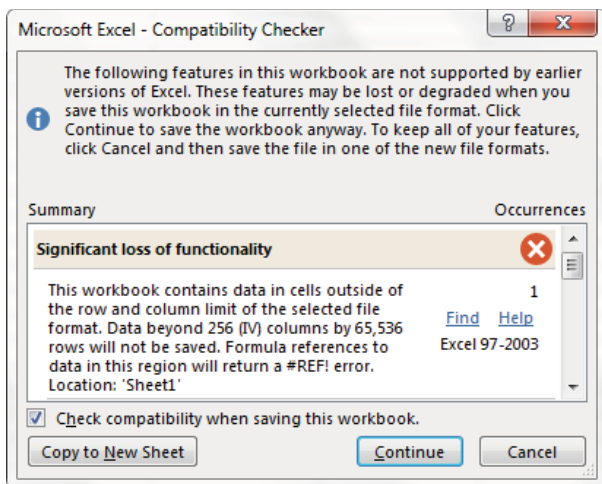
DefaultVersion:=xlPivotTableVersion10

همچنین متوجه می‌شوید که کد ضبط شده، کاملا متفاوت است، زیرا مایکروسافت با آغاز اکسل ۲۰۰۷، تغییرات معناداری در جدول‌های چرخشی ایجاد کرده است.

فرض کنید جدول چرخشی را در اکسل ۲۰۱۹ می‌سازید و کارپوشه را به کارمندی از شرکت که از نسخه ۲۰۰۳ استفاده می‌کند می‌دهید. او جدول چرخشی را می‌بیند اما نمی‌تواند آنرا تازه‌سازی کند. به دیگر سخن، تنها یک جدول مرده از اعداد را می‌بیند.

برای ایجاد یک جدول چرخشی سازگار با گذشته در اکسل ۲۰۱۹، باید فایل را به فرمت XLS ذخیره کنید و دوباره آنرا باز کنید. با این کار، جدول‌های چرخشی که می‌سازید با نسخه‌های پیش از اکسل ۲۰۰۷ کار خواهند کرد. اما، البته که قادر به بهره‌برداری از همه ویژگی‌های جدید جدول چرخشی که

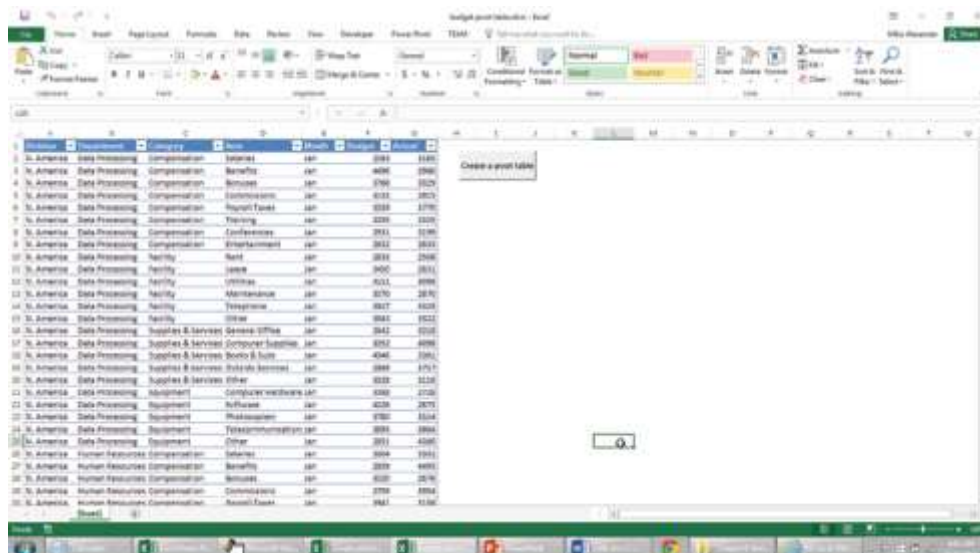
در نسخه‌های بعدی اکسل معرفی شده، نخواهید بود.



خوشبختانه، Excel Compatibility Checker نسبت به این نوع مشکل سازگاری هشدار می‌دهد (شکل آخر را ببینید). گرچه، ماکروهای مربوط به آن جدول‌ها را برای سازگاری بررسی نمی‌کند. ماکروهای این فصل، جدول‌های چرخشی سازگار با گذشته را تولید نمی‌کنند.

ایجاد یک جدول چرخشی پیچیده‌تر

در این بخش، کد VBA مربوط به ایجاد یک جدول چرخشی نسبتا پیچیده را ارائه می‌دهیم. شکل زیر، بخشی از یک جدول کاربرگی بزرگ را نشان می‌دهد. این جدول، دارای ۱۵۸۴۰ ردیف است که داده‌های سلسله‌مراتبی بودجه یک شرکت را دربر می‌گیرد. این شرکت دارای ۵ بخش است و هر بخش دارای ۱۱ اداره است. هر اداره دارای ۴ ردیف (رده‌بندی) بودجه‌ای است و هر ردیف بودجه حاوی چند آیتم بودجه‌ای است. مقادیر واقعی و بودجه‌بندی شده، برای هر ۱۲ ماه موجود است. هدف ما خلاصه‌سازی این اطلاعات با یک جدول چرخشی است.



توجه: فایل تمرینی این مثال، budget pivot table.xlsm نام دارد.

شکل بعدی، یک جدول چرخشی ایجاد شده از این داده‌ها را نشان می‌دهد. دقت کنید که این جدول حاوی یک فیلد محاسباتی به نام Variance است. این فیلد، اختلاف میان مقدار Budget و Actual است.

نکته: گزینه دیگر، افزودن یک ستون جدید به جدول و ایجاد فرمولی برای محاسبه اختلاف میان مقادیر بودجه و هزینه واقعی است. اگر داده‌ها از یک منبع خارجی باشد (به‌جای یک کارپوشه) شاید امکان آن نباشد.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	Division	(All)													
2	Category	(All)													
3															
4			Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Grand Total
5	Accounting														
6	Budget	422,455	433,317	420,522	417,964	411,820	414,012	427,431	418,530	412,134	421,678	426,002	418,445	5,044,910	
7	Actual	422,662	413,163	416,522	420,672	431,309	429,993	425,879	415,253	417,401	417,806	425,271	420,026	5,055,951	
8	Variance	-0,207	-20,154	4,000	-2,708	-19,483	-15,983	1,552	3,277	-5,267	3,872	1,331	-1,593	-11,041	
9	Advertising														
10	Budget	424,590	419,331	417,949	420,324	427,130	424,169	421,183	420,245	429,434	412,078	411,890	423,101	5,051,470	
11	Actual	418,008	420,628	425,437	417,310	419,996	428,330	428,958	420,856	416,067	419,232	411,739	414,482	5,049,253	
12	Variance	8,582	-1,497	-7,488	3,014	7,134	-4,161	-7,775	-0,611	13,367	-7,154	0,157	-1,391	2,217	
13	Data Processing														
14	Budget	422,197	422,057	419,653	417,260	422,848	421,038	421,670	418,093	419,995	418,752	421,100	428,679	5,053,364	
15	Actual	414,743	418,990	430,545	424,214	411,775	421,909	420,210	414,986	419,913	430,262	417,478	408,644	5,031,649	
16	Variance	7,454	-16,933	-10,892	-6,954	11,073	-8,871	1,466	3,127	0,082	-11,510	3,622	20,035	-6,285	
17	Human Resources														
18	Budget	422,053	425,313	418,634	423,038	423,514	419,602	415,197	419,701	422,762	413,741	410,972	412,746	5,037,273	
19	Actual	424,934	429,275	407,053	429,187	430,258	431,870	428,551	422,469	422,252	421,838	415,125	417,222	5,050,034	
20	Variance	-2,881	-3,962	11,581	-6,149	13,256	-2,268	13,354	2,768	0,510	-8,097	-4,153	5,524	-11,761	
21	Operations														
22	Budget	413,530	427,975	419,527	422,299	415,258	414,805	413,149	425,287	412,284	414,342	427,521	410,190	5,026,107	

کد ایجاد کننده جدول چرخشی

کد VBA که این جدول چرخشی را می‌سازد به این شکل است:

```
Sub CreatePivotTable()
    Dim PTcache As PivotCache
    Dim PT As PivotTable

    Application.ScreenUpdating = False
    ' Delete PivotSheet if it exists
    On Error Resume Next
    Application.DisplayAlerts = False
    Sheets("PivotSheet").Delete
    On Error GoTo 0

    ' Create a Pivot Cache
    Set PTcache = ActiveWorkbook.PivotCaches.Create( _
        SourceType:=xlDatabase, _
        SourceData:=Range("A1").CurrentRegion.Address)

    ' Add new worksheet
    Worksheets.Add
    ActiveSheet.Name = "PivotSheet"
    ActiveWindow.DisplayGridlines = False

    ' Create the Pivot Table from the Cache
    Set PT = ActiveSheet.PivotTables.Add( _
        PivotCache:=PTcache, _
        TableDestination:=Range("A1"), _
        TableName:="BudgetPivot")

    With PT
        ' Add fields
        .PivotFields("Category").Orientation = xlPageField
        .PivotFields("Division").Orientation = xlPageField
        .PivotFields("Department").Orientation = xlRowField
        .PivotFields("Month").Orientation = xlColumnField
        .PivotFields("Budget").Orientation = xlDataField
        .PivotFields("Actual").Orientation = xlDataField
        .DataPivotField.Orientation = xlRowField

        ' Add a calculated field to compute variance
        .CalculatedFields.Add "Variance", "=Budget-Actual"
        .PivotFields("Variance").Orientation = xlDataField

        ' Specify a number format
        .DataBodyRange.NumberFormat = "0,000"

        ' Apply a style
        .TableStyle2 = "PivotStyleMedium2"

        ' Hide Field Headers
    End With
End Sub
```

```

.DisplayFieldCaptions = False

' Change the captions
.PivotFields("Sum of Budget").Caption = " Budget"
.PivotFields("Sum of Actual").Caption = " Actual"
.PivotFields("Sum of Variance").Caption = " Variance"
End With
End Sub

```

جدول چرخشی پیچیده‌تر چگونه کار می‌کند

رویه CreatePivotTable با حذف کاربرد PivotSheet در صورت موجود بودن از پیش، کار را آغاز می‌کند. سپس یک شیء PivotCache می‌سازد، کاربرد جدیدی به نام PivotSheet می‌افزاید، و جدول چرخشی را از PivotCache می‌سازد. این کد سپس فیلدهای زیر را به جدول چرخشی می‌افزاید:

Category: یک فیلد (صفحه) فیلتربندی گزارش

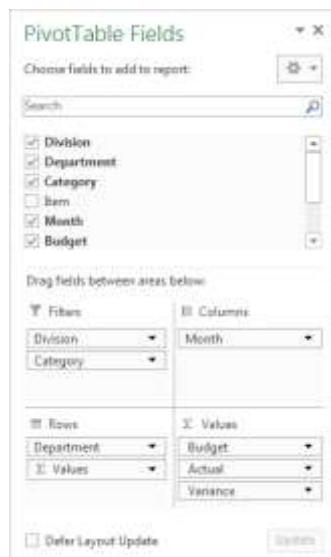
Division: یک فیلد (صفحه) فیلتربندی گزارش

Department: یک فیلد ردیفی

Month: یک فیلد ستونی

Budget: یک فیلد داده‌ای

Actual: یک فیلد داده‌ای



در گزاره زیر دقت کنید که مشخصه Orientation از DataPivotField، روی xlRowField تنظیم است:

```
.DataPivotField.Orientation = xlRowField
```

این گزاره، جهت کلی جدول چرخشی را تعیین می‌کند و بیانگر فیلد Sum Values در ناحیه کاری Pivot Table Fields است (شکل روبه‌رو). این فیلد را به ناحیه Columns منتقل کنید تا ببینید چگونه در طرح‌بندی جدول چرخشی تأثیر می‌گذارد.

سپس رویه مزبور، از متد Add از کلکسیون CalculatedFields برای ایجاد فیلد محاسباتی Variance استفاده می‌کند که مقدار Actual را از مقدار Budget کم کند.

نکته: برای افزودن فیلد محاسباتی به یک جدول چرخشی به شکل دستی، از فرمان PivotTable\Options\Calculations\Fields, Items, & Sets\Calculated Field پنجره Insert Calculated Fields نمایش یابد.

این کد در پایان، چند اصلاح زینتی اعمال می‌کند.

- یک فرمت عددی به DataBodyRange می‌دهد (که به کل داده‌های جدول چرخشی تعمیم یابد).
- یک استایل پیاده‌سازی می‌کند.
- عنوان‌ها را پنهان می‌کند (برابر است با فرمان PivotTable Tools\Analyze\Show\Field (Headers).
- عنوان‌های نمایش یافته در جدول چرخشی را تغییر می‌دهد. برای نمونه، Sum Of Budget با Budget جایگزین می‌شود. دقت کنید که رشته Budget با یک فاصله پیش از آن آمده است. اکسل اجازه نمی‌دهد عنوان متناظر با نام یک فیلد را تغییر دهید، بنابراین با افزودن فاصله، از این محدودیت می‌گذرد.

نکته: فراموش نکنید از ضبط‌کننده ماکرو، بیشترین بهره‌برداری را در یادگیری انواع گوناگون مشخصه‌ها به عمل آورید. بهترین راه برای کشف سینتکس صحیح کدنویسی مورد نیازتان، انجام اکشن‌ها حین ضبط شدن ماکرو است. ترکیب ضبط‌کننده ماکرو، با اطلاعات سامانه Help (و کمی سعی و خطا)، همه اطلاعات لازم در ایجاد کد سفارشی خودتان را در اختیارتان می‌گذارد.

ایجاد جدول‌های سفارشی چندگانه

آخرین مثال، یک سری جدول چرخشی ایجاد می‌کند که داده‌های جمع شده در نظرسنجی یک مشتری را خلاصه‌سازی می‌کند. آن داده‌های نظرسنجی، حاوی ۱۵۰ ردیف است و هر ردیف، حاوی جنسیت پاسخ‌دهنده به علاوه یک امتیازبندی عددی با استفاده از مقیاس ۱ تا ۵ برای هر یک از ۱۴ آیتم نظرسنجی است.

توجه: فایل تمرینی این مثال، survey data pivot tables.xlsx نام دارد.

شکل بعدی، تعدادی از ۲۸ جدول چرخشی تولید شد با ماکرو را نشان می‌دهد. هر آیتم نظرسنجی، در دو جدول چرخشی خلاصه‌سازی شده است (یکی درصدها را نشان می‌دهد و یکی فرکانس‌های حقیقی را).

	A	B	C	D	E	F	G	H	I
3		Female	Male	Grand Total			Female	Male	Grand Total
4	Strongly Disagree	28	40	68	Strongly Disagree	38.4%	50.6%	45.3%	
5	Disagree	20	16	36	Disagree	28.2%	20.3%	24.0%	
6	Undecided	15	9	24	Undecided	21.1%	11.4%	16.0%	
7	Agree	6	14	20	Agree	8.5%	17.7%	13.3%	
8	Strongly Agree	2		2	Strongly Agree	2.8%	0.0%	1.3%	
9	Grand Total	71	79	150					
10									
11	Store hours are convenient				Store hours are convenient				
12	Count of Store hours are convenient				Count of Store hours are convenient				
13		Female	Male	Grand Total		Female	Male	Grand Total	
14	Strongly Disagree	11	13	24	Strongly Disagree	15.5%	16.5%	16.0%	
15	Disagree	7	11	18	Disagree	9.9%	13.9%	12.0%	
16	Undecided	30	26	56	Undecided	42.3%	32.9%	37.3%	
17	Agree	20	22	42	Agree	28.2%	27.8%	28.0%	
18	Strongly Agree	3	7	10	Strongly Agree	4.2%	8.9%	6.7%	
19	Grand Total	71	79	150					
20									
21	Stores are well-maintained				Stores are well-maintained				
22	Count of Stores are well-maintained				Count of Stores are well-maintained				
23		Female	Male	Grand Total		Female	Male	Grand Total	
24	Strongly Disagree	7	14	21	Strongly Disagree	9.9%	17.7%	14.0%	
25	Disagree	7	4	11	Disagree	9.9%	5.1%	7.3%	
26	Undecided	16	14	30	Undecided	22.5%	17.7%	20.0%	
27	Agree	29	29	58	Agree	40.8%	36.7%	38.7%	
28	Strongly Agree	12	18	30	Strongly Agree	16.9%	22.8%	20.0%	
29	Grand Total	71	79	150					

کد VBA تولید کننده این جداول چرخشی به این شکل است:

```

Sub MakePivotTables()
' This procedure creates 28 pivot tables
Dim PTCache As PivotCache
Dim PT As PivotTable
Dim SummarySheet As Worksheet
Dim ItemName As String
Dim Row As Long, Col As Long, i As Long

Application.ScreenUpdating = False

' Delete Summary sheet if it exists
On Error Resume Next
Application.DisplayAlerts = False
Sheets("Summary").Delete
On Error GoTo 0

' Add Summary sheet
Set SummarySheet = Worksheets.Add
ActiveSheet.Name = "Summary"

' Create Pivot Cache
Set PTCache = ActiveWorkbook.PivotCaches.Create( _
    SourceType:=xlDatabase, _
    SourceData:=Sheets("SurveyData").Range("A1"). _
    CurrentRegion)

Row = 1
For i = 1 To 14
    For Col = 1 To 6 Step 5 '2 columns
        ItemName = Sheets("SurveyData").Cells(1, i + 2)
        With Cells(Row, Col)
            .Value = ItemName
            .Font.Size = 16
        End With
    Next Col
Next i

```

```

' Create pivot table
Set PT = ActiveSheet.PivotTables.Add( _
    PivotCache:=PTCache, _
    TableDestination:=SummarySheet.Cells(Row + 1, Col))

' Add the fields
If Col = 1 Then 'Frequency tables
    With PT.PivotFields(ItemName)
        .Orientation = xlDataField
        .Name = "Frequency"
        .Function = xlCount
    End With
Else ' Percent tables
    With PT.PivotFields(ItemName)
        .Orientation = xlDataField
        .Name = "Percent"
        .Function = xlCount
        .Calculation = xlPercentOfColumn
        .NumberFormat = "0.0%"
    End With
End If

PT.PivotFields(ItemName).Orientation = xlRowField
PT.PivotFields("Sex").Orientation = xlColumnField
PT.TableStyle2 = "PivotStyleMedium2"
PT.DisplayFieldCaptions = False
If Col = 6 Then
' add data bars to the last column
    PT.ColumnGrand = False
    PT.DataBodyRange.Columns(3).FormatConditions. _
        AddDatabar
    With pt.DataBodyRange.Columns(3).FormatConditions(1)
        .BarFillType = xlDataBarFillSolid
        .MinPoint.Modify newtype:=xlConditionValueNumber,
newvalue:=0
        .MaxPoint.Modify newtype:=xlConditionValueNumber,
newvalue:=1
    End With
End If
Next Col
Row = Row + 10
Next i

' Replace numbers with descriptive text
With Range("A:A,F:F")
    .Replace "1", "Strongly Disagree"
    .Replace "2", "Disagree"

    .Replace "3", "Undecided"
    .Replace "4", "Agree"
    .Replace "5", "Strongly Agree"
End With
End Sub

```

دقت کنید که همه این جدول‌های چرخشی، از یک شیء PivotCache تکی ایجاد شده‌اند.

این جدول‌های چرخشی، در یک حلقه تودرتو ایجاد می‌شوند. حلقه Col با استفاده از پارامتر Step، از ۱ تا ۶ را می‌شمارد. دستوره‌های ستون دوم جدول‌های چرخشی، کمی متفاوت است. به‌ویژه، این جدول‌ها در ستون دوم خود کارهای زیر را انجام می‌دهند:

- نمایش شمارش به‌شکل درصدی از ستون
- مجموع کل را برای هر یک از ردیف‌ها نمایش نمی‌دهند
- یک فرمت عددی تخصیص می‌دهند
- فرمت‌بندی شرطی روی میله‌های داده‌ها نمایش می‌دهند

متغیر Row، ردیف آغاز کننده هر جدول چرخشی را پیگیری می‌کند. گام آخر، جایگزینی رده‌بندی‌های عددی در ستون‌های A و F با متن است. برای نمونه، ۱ با Strongly Disagree جایگزین می‌شود.

ایجاد یک جدول چرخشی معکوس

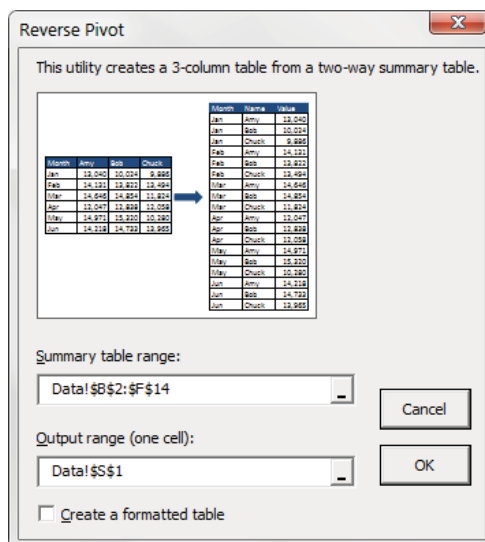
هر جدول چرخشی، خلاصه‌ای از داده‌های درون یک جدول است. اما اگر یک جدول خلاصه شده داشته باشید و بخواهید یک جدول نرمالیزه شده از آن خلاصه ایجاد کنید چه؟ شکل زیر، مثالی را نشان می‌دهد. بازه B2:F14 حاوی یک جدول خلاصه‌سازی شده است - همانند یک جدول چرخشی ساده. ستون‌های I:K حاوی یک جدول ۴۸ ردیفه ایجاد از جدول خلاصه است. در این جدول، هر ردیف، حاوی یک نقطه داده‌ای است و دو ستون نخست، آن نقطه داده‌ای را توصیف می‌کند. به دیگر سخن، داده‌های تغییرشکل یافته، نرمال‌سازی شده است. (عنوان "داده مناسب برای یک جدول چرخشی" در اوایل این فصل را ببینید.)

Month	Amy	Bob	Chuck	Doug
Jan	47,955	34,240	55,560	56,380
Feb	44,715	35,435	61,810	63,325
Mar	41,635	34,005	58,655	60,055
Apr	48,515	32,065	63,530	57,700
May	53,945	39,225	67,860	57,900
Jun	50,990	38,305	64,370	61,760
Jul	49,235	38,675	66,020	65,220
Aug	55,725	34,300	70,160	63,140
Sep	57,710	26,615	68,985	65,740
Oct	54,020	24,220	70,035	63,300
Nov	52,055	19,365	65,240	62,905
Dec	48,690	20,440	64,165	54,915



Column1	Column2	Column3
Jan	Amy	47,955
Jan	Bob	34,240
Jan	Chuck	55,560
Jan	Doug	56,380
Feb	Amy	44,715
Feb	Bob	35,435
Feb	Chuck	61,810
Feb	Doug	63,325
Mar	Amy	41,635
Mar	Bob	34,005
Mar	Chuck	58,655
Mar	Doug	60,055
Apr	Amy	48,515

اکسل روشی برای تبدیل یک جدول خلاصه به یک جدول نرمال‌سازی شده ارائه نمی‌دهد، پس برای یک ماکروی VBA کار خوبی است. برای نمونه، UserForm شکل زیر، بازه‌های ورودی و خروجی را می‌گیرد و نیز دارای گزینه‌ای برای تبدیل بازه خروجی به یک جدول است.



توجه: فایل تمرینی این مثال، reverse pivot table.xlsm نام دارد.

هنگامی که کاربر روی دکمه OK در این UserForm کلیک می‌کند، کد VBA بازه‌ها را اعتبارسنجی می‌کند و سپس رویه ReversePivot را با این گزینه فراخوانی می‌کند:

```
Call ReversePivot(SummaryTable, OutputRange, cbCreateTable)
```

گزاره بالا، سه آرگومان پاس می‌دهد:

SummaryTable: یک شیء Range که گویای جدول خلاصه است.

OutputRange: یک شیء Range که گویای بالاترین سلول سمت چپ بازه خروجی است.

cbCreateTable: شیء Checkbox روی UserForm

این رویه برای هر اندازه جدول خلاصه‌ای کار می‌کند. تعداد ردیف‌های داده در جدول خروجی، برابر با $(r-1)*(c-1)$ است که در آن r و c نشانگر تعداد ردیف‌ها و ستون‌ها در SummaryTable است.

کد رویه ReversePivot به این صورت است:

```

Sub ReversePivot(SummaryTable As Range, _
  OutputRange As Range, CreateTable As Boolean)
  Dim r As Long, c As Long
  Dim OutRow As Long, OutCol As Long

  ' Convert the range
  OutRow = 2
  Application.ScreenUpdating = False
  OutputRange.Range("A1:C3") = Array("Column1", "Column2",
  "Column3")
  For r = 2 To SummaryTable.Rows.Count
    For c = 2 To SummaryTable.Columns.Count
      OutputRange.Cells(OutRow, 1) = SummaryTable.Cells(r, 1)
      OutputRange.Cells(OutRow, 2) = SummaryTable.Cells(1, c)
      OutputRange.Cells(OutRow, 3) = SummaryTable.Cells(r, c)
      OutputRange.Cells(OutRow, 3).NumberFormat = _
        SummaryTable.Cells(r, c).NumberFormat
      OutRow = OutRow + 1
    Next c
  Next r

  ' Make it a table?
  On Error Resume Next
  If CreateTable Then _
    ActiveSheet.ListObjects.Add xlSrcRange, _
      OutputRange.CurrentRegion, , xlYes
  On Error Goto 0
End Sub

```

رویه کاملاً ساده‌ای است. کد روی ردیف‌ها و ستون‌های بازه ورودی حلقه می‌زند و سپس داده را در بازه خروجی می‌نویسد. بازه خروجی همواره سه ستون دارد. متغیر `OutRow`، ردیف جاری در بازه خروجی را پیگیری می‌کند. در پایان، اگر کاربر چک باکس را تیک بزند، بازه خروجی با استفاده از متد `Add` کلکسیون `ListObjects` تبدیل به یک جدول می‌شود.

فصل ۹

کار با نمودارها

در این فصل می‌خوانیم:

- ✓ کشف اطلاعات ضروری پس‌زمینه نمودارهای اکسل
- ✓ فهم تفاوت میان نمودارهای تعبیه‌ای (گنجانده شده)^۱ و برگه‌های نمودار
- ✓ آشنایی با مدل شیء Chart
- ✓ استفاده از متدهای دیگری به‌جز ضبط‌کننده ماکرو برای کمک به یادگیری اشیاء Chart
- ✓ بررسی مثال‌هایی از مرسوم‌ترین کارهای نمودارکشی که از VBA استفاده می‌کند
- ✓ واکاوی ماکروهای پیچیده‌تر نمودارکشی
- ✓ یافتن ترفندهای جالب و مفید نمودارکشی
- ✓ کار با نمودارهای Sparkline

در اعماق نمودارها

ویژگی نمودارکشی اکسل، امکان ایجاد دامنه گسترده‌ای از نمودارها با استفاده از داده‌هایی که در یک کاربرگ ذخیره شده است را فراهم می‌آورد. در اینجا با کنترل کامل روی هر زاویه‌ای از هر نمودار، سروکار داریم.

هر نمودار اکسل، به‌سادگی با اشیاء بسته‌بندی می‌شود، که هر یک از آنها دارای مشخصه‌ها و متدهای خودش است. به این دلیل، دست‌کاری نمودارها با VBA می‌تواند اندکی چالش‌برانگیز باشد. در این فصل، روی مفاهیم کلیدی که برای درک کدنویسی VBA لازم داریم بحث می‌کنیم تا نمودارها را تولید یا دست‌کاری کنیم. راز، همان‌گونه که خواهید دید، شناخت دقیق سلسله‌مراتب شیء برای نمودارهاست.

^۱ در کتاب مرجع کامل Excel 2019 به‌دلیل پارسی‌نویسی، به جای واژه تعبیه‌ای از واژه گنجانده شده استفاده کرده‌ام. اما به نظر، نمودار تعبیه‌ای شاید بهتر مفهوم embedde chart را برساند.

مکان نمودارها

نمودارها در اکسل می‌تواند در یکی از دو محل تعیین شده در یک کارپوشه باشد:

- به شکل یک شیء تعبیه‌ای روی یک کاربرگ: هر کاربرگ می‌تواند حاوی هر تعداد نمودار تعبیه‌ایی باشد.
- در یک برگه نمودار مجزا: برگه نمودار، معمولا یک نمودار واحد را نگه می‌دارد.

بیشتر کاربران، نمودارها را به شکل دستی با استفاده از فرمان‌های گروه‌ریبونی Insert\Chart می‌سازند. اما می‌توان با استفاده از VBA نیز نمودارها را ترسیم کرد. و البته، می‌توان از VBA برای اصلاح نمودارهای موجود نیز استفاده کرد.

نکته: سریع‌ترین روش ایجاد دستی یک نمودار، انتخاب داده‌ها و فشار Alt+F1 است. اکسل یک نمودار تعبیه‌ای را می‌سازد و از نوع نمودار پیش‌گزینه استفاده می‌کند. برای ایجاد یک نمودار پیش‌گزینه جدید روی یک برگه نمودار، داده‌ها را انتخاب و F11 را بفشارید.

یک مفهوم کلیدی در کار با نمودارها، active chart است؛ یعنی نموداری که فعلا انتخاب و فعال شده است. هنگامی که کاربر روی یک نمودار تعبیه‌ای کلیک می‌کند یا برگه نموداری را فعال می‌کند، یک شیء Chart فعال می‌شود. مشخصه ActiveChart در VBA، شیء Chart فعال‌شده (اگر باشد) را برمی‌گرداند. می‌توان برای کار با این شیء، کدنویسی کرد، درست مانند کاری که می‌توانید برای کار با شیء Workbook بازگشتی توسط مشخصه ActiveWorkbook کدنویسی کنید.

با یک مثال آغاز می‌کنیم: اگر نموداری فعال شود، گزاره زیر، مشخصه Name را برای شیء Chart نمایش خواهد داد:

```
MsgBox ActiveChart.Name
```

چنانچه نموداری فعال نباشد، گزاره بالا تولید خطا می‌کند.

نکته: همان‌گونه که خواهید دید، برای دست‌کاری کدهای VBA یک نمودار، نیازی به فعال کردن آن نیست.

ضبط‌کننده ماکرو و نمودارها

همان‌گونه که در فصل‌های پیش خواندید، می‌دانید که ما اغلب، استفاده از ضبط‌کننده ماکرو را برای آموختن اشیاء، مشخصه‌ها، و متدها پیشنهاد می‌دهیم. مانند همیشه، ضبط‌کننده ماکرو، بهترین ابزار آموزشی را به نمایش می‌گذارد. کد ضبط شده، تقریبا همیشه شما را به اشیاء، مشخصه‌ها، و متدهای مربوطه هدایت می‌کند.

در حاشیه: نکته سازگاری

کد VBA این فصل، از مشخصه‌ها و متدهای نمودار-محور استفاده می‌کند که در اکسل ۲۰۱۳ ارائه شده است. برای نمونه، اکسل ۲۰۱۳، متد AddChart2 را معرفی کرد. هنوز متد AddChart کار می‌کند اما روی بیشترین تغییرات اخیر متمرکزیم که اغلب استفاده آسان‌تری دارد. در نتیجه، برخی کدهایی که اینجا ارائه می‌دهیم با نسخه‌های پیش از اکسل ۲۰۱۳ کار نمی‌کند.

مدل شیء Chart

هنگامی که شروع به بررسی مدل شیء برای یک شیء Chart می‌کنید، احتمالاً کمی گیج شوید - که تعجبی ندارد، زیرا مدل شیء، کلا گیج‌کننده است. این مدل، ژرف و عمیق است.

برای نمونه، فرض کنید می‌خواهید عنوان نمایش یافته در یک نمودار تعبیه‌ای را تغییر دهید. البته که شیء سطح بالا، Application است (اکسل) که حاوی یک شیء Workbook است و خود Workbook نیز حاوی یک شیء Worksheet است. شیء Worksheet نیز دربردارنده یک شیء ChartObject است که خودش، یک شیء Chart را شامل می‌شود. شیء Chart دارای یک شیء ChartTitle است که خودش دارای یک مشخصه Text است که متن نمایش یافته به شکل عنوان نمودار را ذخیره می‌کند.

روش دیگر نگرستن به این سلسله مراتب یک شیء تعبیه‌ای، به این صورت است:

```
Application
  Workbook
    Worksheet
      ChartObject
        Chart
          ChartTitle
```

البته کد VBA باید از این مدل شیء به دقت پیروی کند. برای نمونه، برای تنظیم یک عنوان نمودار برای YTD Sales می‌توان یک دستور VBA همانند این نوشت:

```
Worksheets(1).ChartObjects(1).Chart.ChartTitle.Text = "YTD Sales"
```

این گزاره فرض می‌کند کارپوشه فعال، شیء Workbook است. گزاره با نخستین شیء درون کلکسیون ChartObjects روی نخستین کاربرد کار می‌کند. مشخصه Chart، شیء Chart واقعی را برمی‌گرداند و مشخصه ChartTitle نیز شیء ChartTitle را برمی‌گرداند. در پایان، به مشخصه Text می‌رسیم.

دقت کنید که گزاره بالا در صورتی که نمودار، عنوانی نداشته باشد شکست می‌خورد. برای افزودن یک عنوان پیش‌گزیده (که متن Chart Title را نمایش دهد) به نمودار، از این گزاره استفاده کنید:

```
Worksheets("Sheet1").ChartObjects(1).Chart.HasTitle = True
```

برای یک برگه نمودار، سلسله مراتب شیء، کمی متفاوت است، زیرا شیء Worksheet یا شیء ChartObject را درگیر نمی‌کند. برای نمونه، در اینجا سلسله مراتب شیء ChartTitle برای نموداری در یک برگه نمودار ارائه می‌شود:

```
Application
  Workbook
    Chart
      ChartTitle
```

از این گزاره VBA می‌توان برای تنظیم عنوان در یک برگه نمودار برای YTD Sales استفاده کرد:

```
Sheets("Chart1").ChartTitle.Text = "YTD Sales"
```

هر برگه نموداری الزاماً یک شیء Chart است و آن شامل هیچ شیء ChartObject نمی‌شود. به روش دیگر، شیء والد یک نمودار تعبیه‌ای، یک شیء ChartObject است و شیء والد یک نمودار روی یک برگه نمودار مجزا، یک شیء Workbook است.

هر دو گزاره زیر، پنجره‌ای با واژه Chart را نمایش می‌دهند:

```
MsgBox TypeName(Sheets("Sheet1").ChartObjects(1).Chart)
Msgbox TypeName(Sheets("Chart1"))
```

نکته: هنگام ایجاد یک نمودار تعبیه‌ای، درحال افزودن به کلکسیون ChartObjects و کلکسیون Shapes مندرج در یک کاربرگ مشخص هستید. (هیچ کلکسیون Charts برای یک کاربرگ وجود ندارد.) هنگامی که یک برگه نمودار جدید می‌سازید، درحال افزودن به کلکسیون Charts و کلکسیون Sheets یک کارپوشه مشخص هستید.

ایجاد یک نمودار تعبیه‌ای (Embedded Chart)

ChartObject، یک نوع ویژه از شیء Shape است. بنابراین، عضوی از کلکسیون Shapes است. برای ایجاد یک نمودار جدید، از متد AddCharts2 از کلکسیون Shapes استفاده کنید. گزاره زیر، یک نمودار تعبیه‌ای خالی با همه تنظیمات پیش‌گزیده را ایجاد می‌کند:

```
ActiveSheet.Shapes.AddChart2
```

متد AddChart2 می‌تواند هفت آرگومان بگیرد (همگی اختیاری):

Style: یک کد عددی که سبک استایل (یا منظر کلی) نمودار را تعیین می‌کند.

xlChartType: نوع نمودار. اگر وارد نشود، از نوع نمودار پیش‌گزیده استفاده می‌کند. ثابت‌ها برای همه انواع نمودارها فراهم شده است (مانند xlArea و xlColumnClustered).

Left: موقعیت سمت چپ نمودار، به نقطه. اگر نادیده گرفته شود، اکسل نمودار را به‌شکل افقی وسط‌چین می‌کند.

Top: موقعیت سمت بالای نمودار، به نقطه. اگر نادیده گرفته شود، اکسل نمودار را به شکل عمودی وسط چین می‌کند.

Width: پهنای نمودار، به نقطه. اگر نادیده گرفته شود، اکسل از ۳۵۴ استفاده می‌کند.

Height: ارتفاع نمودار، به نقطه. اگر نادیده گرفته شود، اکسل از ۲۱۰ استفاده می‌کند.

NewLayout: یک کد عددی که طرح‌بندی نمودار را تعیین می‌کند.

در زیر، گزاره‌ای است که یک نمودار ستونی خوشه‌ای را با استفاده از Style 201 و Layout 5 که ۵۰ پیکسل فاصله از چپ و ۶۰ پیکسل فاصله از بالا و ۳۰۰ پیکسل پهنا و ۲۰۰ پیکسل ارتفاع دارد ایجاد می‌کند:

```
ActiveSheet.Shapes.AddChart2 201, xlColumnClustered, 50, 60,
300, 200, 5
```

در بسیاری مواقع شاید به این نتیجه برسید که ایجاد یک متغیر شیء وقتی نمودار ایجاد می‌شود، کارآمد است. رویه زیر، یک نمودار خطی ایجاد می‌کند که می‌توانید با استفاده از متغیر شیء MyChart به آن در کد ارجاع دهید. دقت کنید که متد AddChart2 تنها دو آرگومان نخست را مشخص می‌کند. پنج آرگومان دیگر، از مقادیر پیش‌گزیده استفاده می‌کنند.

```
Sub CreateChart ()
    Dim MyChart As Chart
    Set MyChart = ActiveSheet.Shapes.AddChart2(212, xlLineMarkers)
    .Chart
End Sub
```

نمودار بدون داده، مفید نیست. به دو روش می‌توان داده‌هایی را برای یک نمودار تعیین کرد:

- انتخاب سلول‌ها پیش از اینکه کد، نمودار را بسازد
- استفاده از متد SetSourceData شیء Chart پس از ایجاد نمودار

در زیر، رویه ساده‌ای است که بازه‌ای از داده‌ها را انتخاب و سپس نموداری ایجاد می‌کند:

```
Sub CreateChart2 ()
    Range("A1:B6").Select
    ActiveSheet.Shapes.AddChart2 201, xlColumnClustered
End Sub
```

رویه زیر، متد SetSourceData را به تصویر می‌کشد. این رویه، از دو متغیر شیء استفاده می‌کند: DataRange (برای شیء Range که داده‌ها را نگه می‌دارد) و MyChart (برای شیء Chart). متغیر شیء MyChart هم‌زمان با ایجاد نمودار تنظیم می‌گردد.

```
Sub CreateChart3 ()
    Dim MyChart As Chart
    Dim DataRange As Range
    Set DataRange = ActiveSheet.Range("A1:B6")
```

```
Set MyChart = ActiveSheet.Shapes.AddChart2.Chart
MyChart.SetSourceData Source:=DataRange
End Sub
```

دقت کنید که متد AddChart2 هیچ آرگومانی ندارد، پس نموداری پیش‌گزیده ایجاد می‌شود.

ایجاد یک نمودار روی یک برگه نموداری

بخش پیش، رویه‌های پایه‌ی ایجاد یک نمودار تعبیه‌ای را توصیف کرد. برای ایجاد یک نمودار به شکل مستقیم روی یک برگه نمودار، از متد Add2 کلکسیون Charts استفاده کنید. متد Add2 کلکسیون Charts از چند آرگومان انتخابی استفاده می‌کند، اما این آرگومان‌ها موقعیت برگه نمودار را مشخص می‌کنند نه اطلاعات مرتبط با نمودار را.

مثال زیر، نموداری را روی یک برگه نمودار ایجاد می‌کند و بازه داده‌ای و نوع نمودار را تعیین می‌کند:

```
Sub CreateChartSheet ()
Dim MyChart As Chart
Dim DataRange As Range
Set DataRange = ActiveSheet.Range("A1:C7")
Set MyChart = Charts.Add2
MyChart.SetSourceData Source:=DataRange
ActiveChart.ChartType = xlColumnClustered
End Sub
```

اصلاح نمودارها

بهینه‌سازی‌ها با اکسل ۲۰۱۳ ارائه شد و کار ایجاد و اصلاح نمودارها را حتی برای کاربران نهایی، ساده‌تر ساخت. برای نمونه، هنگامی که نموداری فعال‌سازی می‌شود، اکسل سه آیکن در سمت راست نمودار نمایش می‌دهد: Chart Elements (برای افزودن یا حذف عناصر نمودار استفاده می‌شود)، Style & Color (برای انتخاب یک سبک نمودار یا تغییر پالت رنگ استفاده می‌شود)، و Chart Filters (برای پنهان‌سازی سری‌ها یا نقاط داده‌ای استفاده می‌شود).

کد VBA می‌تواند همه کارهایی را که از طریق کنترل‌های نمودار جدید در دسترس است انجام دهد. برای نمونه، اگر درحالی که عناصری را به یک نمودار می‌افزایید یا از آن حذف می‌کنید ضبط‌کننده ماکرو را روشن کنید، خواهید دید که متد مرتبط با آن، SetElement (متدی از شیء Chart). این

متد، یک آرگومان می‌گیرد و ثابت‌های از پیش تعریف شده‌ای برایش موجود است. برای نمونه، برای افزودن خطوط مشبک افقی اصلی به نمودار فعال، از این گزاره استفاده کنید:

```
ActiveChart.SetElement msoElementPrimaryValueGridLinesMajor
```

برای برداشتن این خطوط از نمودار فعال، از این گزاره استفاده کنید:

```
ActiveChart.SetElement msoElementPrimaryValueGridLinesNone
```

همه ثابت‌ها در سامانه Help فهرست شده‌اند یا می‌توانید از ضبط‌کننده ماکرو برای یافتن آنها استفاده کنید.

برای تغییر نمودار به یک سبک از پیش تعریف شده، از مشخصه ChartStyle استفاده کنید. سبک‌ها، اعداد هستند و هیچ ثابت توصیفی وجود ندارد. برای نمونه، گزاره زیر، سبک نمودار فعال را به 215 تغییر می‌دهد:

```
ActiveChart.ChartStyle = 215
```

مقادیر معتبر برای مشخصه ChartStyle، مقادیر بین فواصل ۱-۴۸ و ۲۴۸-۲۰۱ است. گروه دوم حاوی سبک‌هایی است که در اکسل ۲۰۱۳ معرفی شد. همچنین فراموش نکنید که شکل ظاهری واقعی سبک‌ها، میان نسخه‌های مختلف اکسل یکسان نیست. برای نمونه، پیاده‌سازی سبک ۴۸، ظاهری متفاوت در اکسل ۲۰۱۰ خواهد داشت.

برای تغییر رنگ شمای استفاده شده در نمودار، مشخصه ChartColor را روی مقداری بین ۱ و ۲۶ تنظیم کنید. به مثالی توجه کنید:

```
ActiveChart.ChartColor = 12
```

هنگامی که مقدار ChartStyle را با ۲۶ گزینه ChartColor ترکیب کنیم، ۲۴۹۶ ترکیب مختلف خواهیم داشت که برای راضی کردن هر فردی کافیهست. همچنین، اگر انتخاب‌های پیش‌ساخته آنها کافی نباشد، روی هر عنصری در یک نمودار کنترل دارید. برای نمونه، کد زیر پرکردن رنگ یک نقطه در سری‌های یک نمودار (محور افقی) را تغییر می‌دهد:

```
With ActiveChart.FullSeriesCollection(1).Points(2).Format.Fill
    .Visible = msoTrue
    .ForeColor.ObjectThemeColor = msoThemeColorAccent2
    .ForeColor.TintAndShade = 0.4
    .ForeColor.Brightness = -0.25
    .Solid
End With
```

دوباره در اینجا هم می‌توانیم با ضبط کردن کارهایی که حین اعمال تغییرات در یک نمودار انجام می‌دهیم، اطلاعات مورد نیاز برای کدنویسی را درباره مدل شیء کسب کنیم.

استفاده از VBA برای فعال‌سازی یک نمودار

وقتی کاربری روی ناحیه‌ای از یک نمودار تعبیه‌ای کلیک می‌کند، نمودار فعال می‌شود. کد VBA می‌تواند نمودار تعبیه‌ای را به همراه متد Activate فعال سازد. گزاره VBA زیر، برابر با کلیک روی یک نمودار تعبیه‌ای برای فعال کردن آن است:

```
ActiveSheet.ChartObjects("Chart 1").Activate
```

اگر نمودار روی یک برگه نمودار باشد، از گزاره‌ای همانند زیر استفاده کنید:

```
Sheets("Chart1").Activate
```

به‌شکل جایگزین، نمودار را می‌توان با انتخاب شکلی (shape) که نمودار در آن جای گرفته، فعال کرد:

```
ActiveSheet.Shapes("Chart 1").Select
```

هنگامی که نموداری فعال می‌شود می‌توان در کد، با استفاده از مشخصه ActiveChart به آن ارجاع داد (که یک شیء Chart را برمی‌گرداند). برای نمونه، دستور زیر نام نمودار فعال را نمایش می‌دهد. اگر نمودار فعالی موجود نباشد تولید خطا می‌شود.

```
MsgBox ActiveChart.Name
```

برای اصلاح یک نمودار با کد VBA، نیازی به فعال کردن آن نیست. دو رویه زیر، اثر یکسانی دارند. هر دو، نمودار تعبیه‌ای که Chart 1 نام گذاری شده را به یک نمودار فضایی تغییر می‌دهند. نخستین رویه، نمودار را پیش از انجام دستکاری‌ها، فعال می‌کند؛ رویه دوم این کار را نمی‌کند.

```
Sub ModifyChart1 ()
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveChart.ChartType = xlArea
End Sub
```

```
Sub ModifyChart2 ()
    ActiveSheet.ChartObjects("Chart 1").Chart.ChartType = xlArea
End Sub
```

جابه‌جایی یک نمودار

هر نمودار تعبیه‌ای روی یک کاربرگ را می‌توان به یک برگه نموداری تبدیل کرد. برای انجام دستی این کار، کافیسست نمودار تعبیه‌ای را فعال و مسیر Chart Tools\Design\Location\Move Chart را دنبال کرد. در پنجره Move Chart نیز گزینه New Sheet را انتخاب کرد و یک نام به آن داد.