

قدرت برنامه‌نویسی VBA در
Excel 2019

جلد ۱

میخائیل الکساندر
ریچارد دیک کاسلیکا
(بازنویسی کتاب جان واکنباخ)

ترجمه: حسین یعسوبی
انتشارات پندار پارس

فیپا



انتشارات پندار

دفتر فروش: انقلاب

تلفن: ۶۶۵۷۲۳۳۵



نام کتاب : قدرت برنامه‌نویسی VBA در Excel 2019 (جلد ۱)

ناشر : انتشارات پندار پارس

تالیف : میخائیل الکساندر، ریچارد دیک کاسلیکا

ترجمه : حسین یعسوبی

چاپ نخست : اردیبهشت ۹۹

شمارگان : ۱۰۰۰ نسخه

طرح جلد : رامین شکرالهی

چاپ، صحافی : روز

قیمت : ۸۵۰۰۰ تومان

شابک : ۹۷۸-۶۰۰-۸۲۰۱-۸۵-۴



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

پیش‌گفتار مترجم

مدت‌ها بود به دلیل پیشنهاد و درخواست بسیاری از علاقه‌مندان و کمبود کتابی مرجع درباره برنامه‌نویسی VBA در اکسل، به‌دنبال فرصتی برای آماده‌سازی چنین کتابی می‌گشتم. از آنجا که کتاب مرجع کامل Access به‌همراه برنامه‌نویسی با VBA که سال‌ها از ترجمه آن می‌گذرد و چندباری آنرا به‌روزرسانی کرده و بارها به چاپ رسانده‌ایم با استقبال باورنکردنی روبه‌رو شده، و نیز دیگر کتاب‌های مرجع کامل Excel و حتی کتاب مرجع توابع و فرمول‌های Excel این حقیر، به یاری پروردگار، مورد توجه علاقه‌مندان قرار گرفته‌اند و ده‌ها هزار نسخه از اینها تاکنون چاپ و به‌فروش رسانده شده است، آرزو داشتم پس از چاپ کتاب Power BI، کار را تمام کنم و با ترجمه VBA در اکسل، تیر آخر را از کمان رها کنم، اما نمی‌شد که نمی‌شد...

تا اینکه ویروس عرفانی کوبیت ۲۰۱۹ آمد و جهان را به چالش کشید، بسیاری را به کام مرگ کشاند و خیلی‌ها را خانه‌نشین کرد و بذر ترس را در دل مردمان جهان کاشت.

نخست مانند بسیاری، ترسی بر دلم افتاد و پایان جهان را متصور شدم. سال ۱۳۹۹ خورشیدی سر رسید. پس از گذشت ۳ روز، به یک باره واژگانی از میان افکارم به سرعت برق و باد عبور کرد. بله خودش بود! چرا که نه؟ همان که مدت‌ها به‌دنبال زمانی برای انجامش بودی. از فرصت استفاده کن و در این خود قرنطینگی، به‌جای دلسرد شدن و پارو زدن در امواج منفی ذهن، به خواسته‌ی قدیمی خود جامه عمل بپوشان و کتاب VBA در اکسل را ترجمه کن.

این بود که با انرژی وصف ناشدنی، کار ترجمه کتاب Excel 2019, Power Programming with VBA آقای میخائیل الکساندر را آغاز کردم و برای اینکه زودتر به دست مخاطبان برسد، تصمیم بر این شد که آنرا در دو جلد روانه بازار کنیم. جلد نخست آن هر آنچه لازم است در باب VBA بدانید تا یک برنامه‌کاربری با آن بسازید را شامل می‌شود. جلد دوم نیز مباحث پیشرفته را دربر می‌گیرد که امیدوارم بتوانم آنرا نیز به‌زودی ترجمه و روانه بازار کنیم. هرچند، تصمیم داریم هر دو جلد را در یک جلد یکپارچه با جلد سخت نیز به دست علاقه‌مندان برسانیم.

اگر اکسل کار هستید، تازمانی‌که با VBA کار نکنید هیچ لذتی از اکسل نخواهید برد. دنیایی با افقی پهناور که با دوربین VBA می‌توان دوردست‌هایش را کنکاش کرد و به ماورایش رفت و در سایه ماژول‌های غول پیکرش، آبجکت‌هایش را نوازش کرد و توابعش را در آغوش کشید و فرمول‌های ظریفی تولید کرد!

به نظر حقیر، فردی که VBA بداند و با ماکرونویسی آشنا شود و وارد Power BI شود و با کوثری‌نویسی، داشبوردهای مدیریتی هدفمندی تولید کند، قابلیت استخدام در هر شرکت و بنگاه تولیدی یا اقتصادی را دارد. چرا که اینک، تحلیل داده‌های تجاری، حرف نخست را در دنیای پس از کویت ۲۰۱۹ می‌زند و رقابت بر سر تحلیل اطلاعات جدیدی است که از جوامع مختلف پس از کرونا کسب می‌شود. دست بجنبید و فرصت یادگیری را از دست ندهید. با حرکت، ساختارها آغاز می‌شود. در اینجا لازم می‌دانم به برادر عزیزم که از کمک‌های شایانش مرا بی‌نصیب نگذاشت و انگیزه‌ام را دو چندان کرد درود بفرستم و برایش بهترین‌ها را آرزو کنم.

حسین یعسوبی - ۲۹ فروردین ۱۳۹۹

فهرست فصل‌های جلد ۱

بخش یک: مقدمه‌ای بر EXCEL VBA	۷
فصل ۱: الزامات توسعه برنامه کاربردی صفحه گسترده	۹
فصل ۲: معرفی زبان VBA	۳۱
فصل ۳: بنیان برنامه‌نویسی VBA	۸۱
فصل ۴: کار با رویه‌های ساب VBA	۱۲۷
فصل ۵: ایجاد رویه‌های FUNCTION	۱۶۷
فصل ۶: آشنایی با رویدادهای اکسل (EXCEL EVENTS)	۲۰۹
فصل ۷: مثال‌ها و تکنیک‌های برنامه‌نویسی VBA	۲۴۵

فهرست فصل‌های جلد ۲

بخش دوم: تکنیک‌های پیشرفته VBA	
فصل ۸: کار با جدول‌های چرخشی (PIVOT TABLES)	
فصل ۹: کار با نمودارها	
فصل ۱۰: تعامل با دیگر برنامه‌های کاربردی	
فصل ۱۱: کار با داده‌ها و فایل‌های خارجی	
بخش سوم: کار با USERFORMها	
فصل ۱۲: کار با پنجره‌های محاوره‌ای سفارشی	
فصل ۱۳: معرفی USERFORMها	
فصل ۱۴: نگاهی به مثال‌های USERFORM	
فصل ۱۵: پیاده‌سازی تکنیک‌های پیشرفته USERFORM	
بخش چهارم: توسعه برنامه‌های کاربردی اکسل	
فصل ۱۶: ایجاد و استفاده از افزونه‌ها (ADD-INS)	
فصل ۱۷: کار با ریبون	
فصل ۱۸: کار با منوهای میان‌بر	
فصل ۱۹: تهیه HELP برای برنامه‌کاربردی خود	
فصل ۲۰: کار با ماژول‌های CLASS	
فصل ۲۱: شناخت مشکلات سازگاری	
بخش پنجم: پیوست: مرجع گزاره‌ها و توابع VBA	

فهرست

۷	بخش یک؛ مقدمه‌ای بر EXCEL VBA
۹	فصل ۱؛ الزامات توسعه برنامه کاربردی صفحه گسترده
۹	برنامه کاربردی صفحه گسترده چیست؟
۱۰	گام‌هایی برای توسعه برنامه
۱۱	تعیین نیازهای کاربر
۱۲	طراحی برنامه‌ای که نیازهای کاربر را رفع می‌کند
۱۵	تعیین مناسب‌ترین رابط کاربری
۱۵	سفارشی‌سازی ریبون
۱۶	سفارشی‌سازی منوهای میان‌بر
۱۶	ایجاد کلیدهای میان‌بر
۱۷	ایجاد پنجره‌های محاوره‌ای سفارشی
۱۷	افزودن کنترل‌های ActiveX به کاربرگ
۱۹	تلاش برای توسعه
۲۰	درباره خودتان همراه با کاربر نهایی
۲۰	آزمایش برنامه
۲۱	در حاشیه؛ آزمایش بتا چیست؟
۲۲	ضدضربه کردن برنامه
۲۳	در حاشیه؛ گذرواژه‌های اکسل چندان مطمئن نیستند
۲۴	افزایش جذابیت ظاهری برنامه
۲۵	ایجاد یک سامانه Help
۲۵	مستندسازی تلاش توسعه دهنده
۲۶	توزیع برنامه برای کاربر
۲۶	به‌روز رسانی برنامه هنگام نیاز
۲۷	مشکلات دیگر توسعه
۲۷	نسخه نصب شده اکسل کاربر
۲۷	مشکلات زبان
۲۸	سرعت سیستم
۲۸	حالت‌های ویدیویی
۳۱	فصل ۲؛ معرفی زبان VBA
۳۱	درباره ضبط کننده ماکروی اکسل (Macro Recorder)
۳۲	ایجاد نخستین ماکرو
۳۴	بررسی ماکرو
۳۵	آزمایش ماکرو
۳۵	ویرایش ماکرو
۳۶	مقایسه ضبط ماکروی مستقل و وابسته
۳۶	ضبط ماکروها با رفرنس‌های مستقل
۳۹	ضبط ماکروها با رفرنس‌های وابسته

۴۱	مفاهیم دیگر ضبط ماکرو
۴۱	امنیت ماکرو در اکسل
۴۲	محل‌های ایمن شده
۴۳	نگهداری ماکروها در کارپوشه ماکروهای شخصی
۴۴	تخصیص یک ماکرو به یک دکمه و دیگر کنترل‌های فرم
۴۵	درج یک ماکرو در نوارابزار Quick Access
۴۶	کار با محیط Visual Basic Editor
۴۶	آشنایی با اجزای VBE
۴۷	Menu bar
۴۷	Toolbar
۴۸	پنجره Project
۴۸	پنجره Code
۴۸	پنجره Immediate
۴۸	کار با پنجره Project
۴۹	افزودن یک ماژول VBA جدید
۵۰	حذف یک ماژول VBA
۵۱	کار با یک پنجره Code
۵۱	بزرگ و کوچک کردن پنجره‌ها
۵۲	رسیدن به کد VBA در درون یک ماژول
۵۵	سفارشی‌سازی محیط VBA
۵۵	برگه Editor
۵۷	برگه Editor Format
۵۸	برگه General
۵۸	برگه Docking
۵۹	اصول VBA
۵۹	شناخت اشیاء (آبجکت‌ها)
۶۰	آشنایی با کلکسیون‌ها
۶۱	آشنایی با مشخصه‌ها (Properties)
۶۱	تعیین مشخصه‌های شیء فعال
۶۴	آشنایی با متدها (Methods)
۶۴	در حاشیه: مطالب بیشتری درباره آرگومان‌ها
۶۵	شیرجه عمیق: کار با اشیاء Range
۶۶	یافتن مشخصه‌های شیء Range
۶۶	مشخصه Range
۶۷	مشخصه Cells
۶۹	در حاشیه: گرفتن اطلاعات از یک سلول
۷۰	مشخصه Offset
۷۱	مفاهیم ضروری که باید به‌یاد داشت

۷۲	در حاشیه: درباره کد مثال‌ها
۷۳	وحشت نکنید، تنها نیستید
۷۳	آخر کتاب را بخوانید
۷۴	اجازه دهید اکسل به ماکرونویسی شما کمک کند
۷۴	استفاده از سامانه Help
۷۴	اهمیت مکان هنگام درخواست کمک
۷۵	نیاز به اتصال به اینترنت دارید
۷۵	استفاده از Object Browser
۷۷	کش رفتن کدها از اینترنت
۷۷	قدرت انجمن‌های کاربری
۷۸	بررسی بلاگ‌های حرفه‌ای
۷۸	YouTube، معدن ویدئوهای آموزشی
۷۹	یادگیری از Microsoft Office Dev Center
۷۹	تشریح دیگر فایل‌های اکسل در سازمان خود
۸۱	فصل ۳؛ بنیان برنامه‌نویسی VBA
۸۱	نگاهی به عناصر زبان VBA
۸۲	در حاشیه: وارد کردن کد VBA
۸۳	کامنت‌ها
۸۵	متغیرها، نوع داده‌ها، و ثابت‌ها
۸۶	تعریف نوع داده (Data Type)
۸۸	اعلان متغیرها
۸۹	تعیین یک نوع داده
۹۰	مجبور کردن خود به اعلان همه متغیرها
۹۰	قلمروبندی متغیرها
۹۱	در حاشیه: نکته‌ای درباره مثال‌های این فصل
۹۱	متغیرهای محلی
۹۲	متغیرهای چند ماژولی (module-wide)
۹۳	متغیرهای عمومی (Public)
۹۳	متغیرهای ایستا (Static)
۹۳	کار با ثابت‌ها (Constants)
۹۴	اعلان ثابت‌ها
۹۴	استفاده از ثابت‌های از پیش تعریف شده
۹۵	کار با رشته‌ها (Strings)
۹۶	کار با تاریخ‌ها
۹۶	در حاشیه: درباره باگ تاریخ در اکسل
۹۷	گزاره‌های تخصیص (Assignment Statements)
۹۹	آرایه‌ها
۱۰۰	اعلان آرایه‌ها

۱۰۰.....	اعلان آرایه‌های چند بعدی
۱۰۱.....	اعلان آرایه‌های پویا (Dinamic Arrays)
۱۰۱.....	متغیرهای شیء (Object Variables)
۱۰۲.....	نوع داده‌های User-Defined
۱۰۳.....	توابع پیش‌ساخته
۱۰۵.....	در حاشیه: تابع MsgBox
۱۰۶.....	دست‌کاری اشیاء و کلکسیون‌ها
۱۰۶.....	ساختارهای With-End With
۱۰۷.....	ساختارهای For Each-Next
۱۰۹.....	کنترل اجرای کد
۱۱۰.....	گزاره‌های GoTo
۱۱۱.....	ساختارهای If-Then
۱۱۴.....	در حاشیه: تابع IIF در VBA
۱۱۵.....	سازنده‌های Select Case
۱۱۸.....	بلوک‌های حلقه‌زنی دستورها
۱۱۹.....	حلقه‌های For-Next
۱۱۹.....	در حاشیه: برنامه‌نویسی ساخت‌یافته چیست؟
۱۲۳.....	حلقه‌های Do While
۱۲۴.....	حلقه‌های Do Until
۱۲۷.....	فصل ۴؛ کار با رویه‌های ساب VBA
۱۲۷.....	درباره رویه‌ها
۱۲۸.....	اعلان یک رویه Sub
۱۲۹.....	در حاشیه: نامگذاری رویه‌ها
۱۲۹.....	قلمروی یک رویه
۱۲۹.....	رویه‌های عمومی
۱۳۰.....	رویه‌های خصوصی
۱۳۰.....	اجرای رویه‌های Sub
۱۳۱.....	اجرای یک رویه با فرمان Run Sub\UserForm
۱۳۱.....	اجرای رویه از پنجره Macro
۱۳۲.....	اجرای یک رویه با کلیدهای میان‌بر ترکیبی و Ctrl
۱۳۴.....	اجرای یک رویه از روی ریبون
۱۳۴.....	اجرای یک رویه از یک منوی میان‌بر سفارشی‌سازی شده
۱۳۴.....	اجرای یک رویه از رویه دیگر
۱۳۶.....	فراخوانی یک رویه در ماژولی متفاوت
۱۳۶.....	فراخوانی یک رویه در کارپوشه‌ای دیگر
۱۳۹.....	در حاشیه: چرا رویه‌های دیگر را فرامی‌خوانیم؟
۱۳۹.....	اجرای یک رویه با کلیک روی یک شیء
۱۴۱.....	اجرای یک رویه هنگام رخ دادن یک رویداد

۱۴۲.....	اجرای یک رویه از پنجره Immediate
۱۴۲.....	پاس دادن آرگومان‌ها به رویه‌ها
۱۴۵.....	درحاشیه: استفاده از متغیرهای عمومی به جای پاس دادن آرگومان‌ها به یک رویه
۱۴۶.....	فنون رسیدگی به خطا
۱۴۶.....	به دام انداختن خطاها
۱۴۸.....	مثال‌های رسیدگی‌کننده خطا
۱۵۰.....	نمونه‌ای واقعی از کاربرد رویه‌های Sub
۱۵۱.....	هدف
۱۵۱.....	ملزومات پروژه
۱۵۱.....	چه چیزی می‌دانید
۱۵۲.....	روش کار
۱۵۳.....	برخی از ضبط کردن‌های مقدماتی
۱۵۴.....	گام ابتدایی
۱۵۵.....	کدنویسی
۱۵۶.....	نوشتن رویه Sort
۱۶۰.....	آزمایش‌های بیشتر
۱۶۱.....	برطرف کردن مشکلات
۱۶۴.....	در دسترس قرار دادن ابزار
۱۶۵.....	ارزیابی پروژه
۱۶۷.....	فصل ۵؛ ایجاد رویه‌های FUNCTION
۱۶۷.....	رویه‌های SUB در برابر رویه‌های Function
۱۶۸.....	دلیل ایجاد توابع سفارشی
۱۶۹.....	مثال ابتدایی تابع
۱۶۹.....	استفاده از یک تابع در یک کاربرگ
۱۷۰.....	استفاده از تابع در یک رویه VBA
۱۷۱.....	تحلیل تابع سفارشی این مثال
۱۷۳.....	در حاشیه: توابع کاربرگی سفارشی چه کاری نمی‌توانند بکنند؟
۱۷۳.....	رویه‌های Function
۱۷۵.....	قلمروی یک تابع
۱۷۵.....	اجرای رویه‌های فانکشن
۱۷۶.....	اجرای تابع از یک رویه
۱۷۶.....	اجرای تابع در فرمول کاربرگی
۱۷۷.....	در یک فرمول فرمت‌بندی شرطی
۱۷۸.....	اجرای تابع سفارشی از پنجره VBE Immediate
۱۷۹.....	آرگومان‌های تابع
۱۷۹.....	مثال‌های تابع
۱۷۹.....	توابعی بدون هیچ آرگومان
۱۸۱.....	در حاشیه: کنترل محاسبه دوباره تابع

۱۸۲	تابعی با یک آرگومان
۱۸۴	در حاشیه: استفاده از آرگومان‌ها، نه رفرنس‌های سلولی
۱۸۵	تابعی با دو آرگومان
۱۸۵	تابعی با یک آرگومان آرایه‌ای
۱۸۶	تابعی با آرگومان‌های اختیاری
۱۸۸	تابعی که یک آرایه VBA برمی‌گرداند
۱۹۰	تابعی که یک مقدار خطا برمی‌گرداند
۱۹۲	تابعی با تعداد نامحدودی آرگومان
۱۹۳	شبیه‌سازی از تابع SUM اکسل
۱۹۶	توابع تاریخی بسط یافته
۱۹۸	توابع دیباگ
۲۰۰	واکاوی پنجره Insert Function
۲۰۰	استفاده از متد MacroOptions
۲۰۲	تعیین یک رده‌بندی تابع
۲۰۳	افزودن دستی یک توصیف تابع
۲۰۴	استفاده از Add-Ins برای ذخیره توابع سفارشی
۲۰۵	استفاده از Windows API
۲۰۵	در حاشیه: اکسل ۶۴ بیتی و توابع API
۲۰۶	مثال‌های Windows API
۲۰۶	تعیین دایرکتوری ویندوز
۲۰۷	کشف کلید Shift
۲۰۸	درباره توابع API بیشتر بدانیم
۲۰۹	فصل ۶؛ آشنایی با رویدادهای اکسل (EXCEL EVENTS)
۲۰۹	درباره رویدادها چه چیزی باید دانست
۲۱۱	آشنایی با ترتیب رویدادها
۲۱۱	محل درج رویه‌های event-handler
۲۱۲	غیرفعال‌سازی رویدادها
۲۱۴	وارد کردن کد رسیدگی‌کننده رویداد
۲۱۵	رویه‌های رسیدگی‌کننده رویداد آرگومان‌دار
۲۱۶	آشنایی با رویدادهای سطح-کارپوشه
۲۱۸	رویداد Open
۲۱۹	رویداد Activate
۲۱۹	رویداد SheetActivate
۲۱۹	رویداد NewSheet
۲۲۰	رویداد BeforeSave
۲۲۰	رویداد Deactivate
۲۲۱	رویداد BeforePrint
۲۲۱	به‌روزرسانی یک هدر و فوتر

۲۲۲.....	پنهان‌سازی ستون‌ها پیش از چاپ.....
۲۲۲.....	BeforeClose رویداد.....
۲۲۴.....	بررسی رویدادهای کاربرگ.....
۲۲۶.....	Change رویداد.....
۲۲۷.....	مانیتورینگ یک بازه مشخص برای تغییرات.....
۲۲۸.....	مانیتورینگ یک بازه برای ضخیم کردن (bold) فرمول‌ها.....
۲۲۹.....	مانیتورینگ یک بازه برای اعتبارسنجی داده‌های ورودی.....
۲۳۲.....	SelectionChange رویداد.....
۲۳۳.....	BeforeDoubleClick رویداد.....
۲۳۳.....	BeforeRightClick رویداد.....
۲۳۴.....	در حاشیه: Object Browser برای مکان‌یابی رویدادها.....
۲۳۴.....	مانیتورینگ با رویدادهای Application.....
۲۳۵.....	فعال‌سازی رویدادهای سطح-Application.....
۲۳۷.....	تعیین زمان بازشدن یک کارپوشه.....
۲۳۸.....	مانیتورینگ رویدادهای سطح-Application.....
۲۳۹.....	دسترسی به رویدادهای نامرتبط به یک شیء.....
۲۳۹.....	OnTime رویداد.....
۲۴۰.....	OnKey رویداد.....
۲۴۱.....	مثالی از رویداد OnKey.....
۲۴۲.....	کدهای کلید.....
۲۴۳.....	غیرفعال‌سازی منوهای میان‌بر.....
۲۴۵.....	فصل ۷؛ مثال‌ها و تکنیک‌های برنامه‌نویسی VBA
۲۴۵.....	آموزش با مثال.....
۲۴۶.....	کار با بازه‌ها.....
۲۴۶.....	کپی یک بازه.....
۲۴۷.....	انتقال یک بازه.....
۲۴۸.....	کپی یک بازه با سایز متغیر.....
۲۴۹.....	در حاشیه: نکاتی برای کار با بازه‌ها.....
۲۴۹.....	انتخاب یا شناسایی انواع گوناگون بازه‌ها.....
۲۵۱.....	در حاشیه: روش دیگر ارجاع به یک بازه.....
۲۵۱.....	تغییر اندازه یک بازه.....
۲۵۲.....	درخواست سریع برای مقدار یک سلول.....
۲۵۳.....	واردکردن مقداری در سلول خالی بعدی.....
۲۵۴.....	متوقف کردن یک ماکرو برای رسیدن به بازه انتخابی کاربر.....
۲۵۶.....	شمارش سلول‌های انتخابی.....
۲۵۷.....	تعیین نوع بازه انتخابی.....
۲۵۹.....	حلقه‌زنی کارآمدتر روی یک بازه انتخابی.....
۲۶۲.....	حذف همه ردیف‌های خالی.....

۲۶۳.....	تکراری کردن ردیف‌ها به تعداد یک متغیر
۲۶۴.....	تعیین اینکه بازه‌ای در بازه دیگر است یا نه
۲۶۵.....	تعیین نوع داده یک سلول
۲۶۶.....	خواندن و نوشتن بازه‌ها
۲۶۷.....	روشی بهتر برای نوشتن در یک بازه
۲۶۹.....	انتقال آرایه‌های تک بعدی
۲۷۰.....	انتقال یک بازه به یک آرایه متغیر
۲۷۱.....	انتخاب سلول‌ها برپایه مقدار
۲۷۲.....	کپی یک بازه غیرمتوالی
۲۷۴.....	کار با کارپوشه‌ها و برگه‌ها
۲۷۴.....	ذخیره همه کارپوشه‌ها
۲۷۵.....	ذخیره‌سازی و بستن همه کارپوشه‌ها
۲۷۵.....	مخفی‌سازی همه به‌جز انتخاب شده‌ها
۲۷۷.....	ایجاد یک جدول فرایبندی از محتویات
۲۷۸.....	همگام‌سازی کاربرگ‌ها
۲۷۹.....	تکنیک‌های VBA
۲۷۹.....	ضامنی کردن یک مشخصه Boolean
۲۸۰.....	نمایش تاریخ و زمان
۲۸۱.....	نمایش دلخواه زمان
۲۸۲.....	دریافت فهرستی از فونت‌ها
۲۸۴.....	مرتب‌سازی یک آرایه
۲۸۵.....	پردازش یک سری فایل
۲۸۷.....	برخی توابع مفید برای استفاده در کدها
۲۸۷.....	تابع FileExists
۲۸۷.....	تابع FileNameOnly
۲۸۸.....	تابع PathExists
۲۸۸.....	تابع RangeNameExists
۲۸۹.....	تابع SheetExists
۲۸۹.....	تابع WorlbookIsOpen
۲۸۹.....	در حاشیه: آزمایش عضویت در یک کلکسیون
۲۹۰.....	بازیابی یک مقدار، از یک کارپوشه بسته شده
۲۹۲.....	برخی توابع کاربرگی مفید
۲۹۲.....	بازگرداندن اطلاعات فرمت‌بندی
۲۹۴.....	کاربرگ سخن‌گو
۲۹۴.....	نمایش داده‌ها هنگام ذخیره یا چاپ شدن یک فایل
۲۹۵.....	شناخت والدین شیء
۲۹۶.....	شمارش سلول‌های میان دو مقدار
۲۹۷.....	تعیین آخرین سلول غیرخالی در یک ستون یا ردیف

۲۹۸.....	آیا رشته از یک الگو پیروی می‌کند؟.....
۲۹۹.....	استخراج n امین عنصر از یک رشته.....
۳۰۰.....	هجی کردن یک عدد.....
۳۰۱.....	یک تابع چندعملکردی.....
۳۰۲.....	تابع SHEETOFFSET.....
۳۰۳.....	بازیابی بیشینه مقدار میان همه کاربرگ‌ها.....
۳۰۴.....	بازگرداندن آرایه‌ای از اعداد صحیح تصادفی غیرتکراری.....
۳۰۵.....	تصادفی کردن یک بازه.....
۳۰۶.....	مرتب‌سازی یک بازه.....
۳۰۸.....	فراخوانی‌های API ویندوز.....
۳۰۸.....	آشنایی با اعلان‌های API.....
۳۰۹.....	اعلان‌های ۳۲ بیتی درمقابل ۶۴ بیتی.....
۳۰۹.....	تعیین مشارکت‌های فایل.....
۳۱۰.....	تعیین اطلاعات چاپگر پیش‌فرض.....
۳۱۱.....	تعیین اطلاعات نمایش ویدیو.....
۳۱۲.....	خواندن از و نوشتن در رجیستری.....
۳۱۳.....	خواندن از رجیستری.....
۳۱۴.....	نوشتن در رجیستری.....
۳۱۵.....	در حاشیه: روش ساده‌تر دسترسی به رجیستری.....

مقدمه

برای بیشتر ما، مسیر برنامه‌نویسی با VBA در اکسل، با نیاز به انجام برخی کارهایی که نمی‌توانیم با ابزارهای استاندارد درون اکسل انجام دهیم آغاز می‌شود. این کارها برای هر یک از ما متفاوت است. کارهایی همچون ایجاد کارپوشه‌های جداگانه به شکل خودکار برای هر یک از ردیف‌های درون یک مجموعه داده، یا خودکارسازی ارسال ایمیل از ده‌ها گزارش. به احتمال زیاد کاربران دیگری هم این مشکلات را دارند و به دنبال حل آن به کمک VBA اکسل هستند.

مسئله جالب درباره VBA اکسل این است که مجبور نیستید برای آغاز حل مشکلات خود در این موارد، یک کاربر خبره و باتجربه در اکسل باشید. می‌توانید تنها برای حل یک مشکل خاص، تنها به آموختن راه‌حل همان مشکل بسنده کنید یا جلوتر روید و روش‌های رسیدگی به انواع سناریوهای خودکارسازی را نیز بیاموزید.

هر هدفی که داشته باشید، این کتاب در آموزش قدرت زبان VBA در خودکارسازی کارها، هوشمندتر کردن کار و پربارتر شدن، به شما کمک خواهد کرد.

عناوین پوشش داده شده

تمرکز این کتاب، روی زبان برنامه‌نویسی Visual Basic for Application (VBA) است که زبان برنامه‌نویسی درون اکسل است (و دیگر برنامه‌های کاربردی آفیس). به شکلی ویژه‌تر، نشان می‌دهد چگونه برنامه‌هایی بنویسید که کارهای متنوعی را در اکسل خودکارسازی کند. در این کتاب، هر آن چیزی که فکرش را کنید یا نکنید پوشش داده شده است؛ از ضبط ماکروهای ساده تا ساخت برنامه‌های کاربردی و ابزارهای جنبی کاربر-محور هوشمند و خبره.

از هر کجای این کتاب می‌توانید شروع به خواندنش کنید. می‌توانید از فصل نخست آن شروع کنید و تا آخر بروید یا برحسب سرتیترهای فهرست، مورد به مورد وارد شوید. زبان VBA اغلب یک تلاش کار-محور است. بنابراین اگر به یک کار محاسباتی برخورد کنید، شاید لازم شود ابتدا به فهرست یا ایندکس کتاب رجوع کنید.

ابزار VSTO¹ در این کتاب پوشش داده نشده است. تکنولوژی که از VB.NET و Visual C# استفاده می‌کند و نیز برای کنترل اکسل و دیگر برنامه‌های آفیس استفاده می‌شود.

¹ Microsoft Visual Studio Tools for Office

همان‌گونه که می‌دانید، اکسل ۲۰۱۹ در دسترس دیگر پلتفرم‌هاست. برای نمونه، می‌توان از Excel Web App در مرورگر خود و حتی در آی-پد و تبلت‌ها استفاده کرد. این نسخه‌ها از VBA پشتیبانی نمی‌کنند. به بیانی دیگر، این کتاب برای نسخه دسکتاپ (رومیزی) اکسل ۲۰۱۹ تحت ویندوز است.

آنچه نیاز است بدانید

این کتاب برای کاربران مبتدی اکسل نیست. اگر تجربه‌ای در اکسل ندارید بهتر است نخست، کتاب مرجع کامل اکسل ۲۰۱۹ همین نویسنده (و ترجمه یعسوبی-انتشارات پندار پارس) را بخوانید که پوشش کاملی بر همه ابزارهای اکسل دارد و مناسب کاربران در همه سطوح است.

این کتاب در اصل، مناسب کاربرانی است که در سطح متوسط باشند و کارهای زیر را بتوانند انجام دهند:

- ایجاد کارپوشه‌ها، افزودن کاربرگ‌ها، ذخیره فایل و ...
- حرکت و مسیریابی درون کارپوشه
- استفاده از رابط کاربری ریبون
- فرمول‌نویسی
- استفاده از توابع کاربرگی اکسل
- نام‌گذاری سلول‌ها و بازه‌ها
- استفاده از ابزارهای پایه ویندوز، همچون تکنیک‌های مدیریت فایل و حافظه‌موقت (کلیپ‌بورد)

آنچه نیاز است داشته باشید

اکسل در چند نسخه در دسترس است؛ شامل نسخه وب و نسخه تبلت و موبایل. این کتاب تنها برای نسخه رومیزی اکسل ۲۰۱۹ تحت ویندوز نوشته شده است. اگر طرحی برای توسعه برنامه‌های کاربردی که در نسخه‌های قدیمی‌تر اکسل استفاده خواهد شد دارید، پیشنهاد اکید می‌شود که از همان نسخه قدیمی‌تر اکسل استفاده کنید. در این چند سال اخیر، مایکروسافت یک چرخه انتشار چاپک را برای نسخه وب اکسل به کمک Office 365 اتخاذ کرده است، که انتشار به‌روزرسانی‌ها را به‌ویژه در پایه ماهانه تولید می‌کند.

مهم است که یک نصب کامل از اکسل داشته باشید و اگر می‌خواهید فصل‌های پیشرفته‌تر مربوط به ارتباط میان اکسل و دیگر برنامه‌های آفیس را امتحان کنید، به نصب کامل آفیس نیاز دارید.

نسخه ویندوزی که استفاده می‌کنید مهم نیست. هر سیستم کامپیوتری که بتواند ویندوز را اجرا کند کافی است، اما بهتر است سرعت بالا و حافظه مناسبی داشته باشد. اکسل برنامه بزرگی است و استفاده از آن روی یک سیستم ضعیف با حافظه اندک، می‌تواند کمی آزاردهنده باشد.

فراموش نکنید که این کتاب برای اکسل مکتبتاش قابل استفاده نیست.

قراردادهای به‌کار رفته در کتاب

چند دقیقه برای قراردادهای نوشتاری که در این کتاب به کار رفته اختصاص دهید تا فهم آن ساده‌تر شود.

فرمان‌های اکسل

اکسل از یک سامانه منوی ریبونی مفهومی استفاده می‌کند. کلماتی که در نوار بالایی آمده است، به عنوان برگه‌ریبونی یا tab شناخته می‌شود (مانند Insert و View). کلیک روی یک tab، ریبون حاوی آیکن‌ها را باز می‌کند که فرمان‌های پر کاربرد را نمایش دهد. به جای tab از واژه برگه یا برگه ریبونی استفاده می‌کنیم. هر آیکن، دارای نامی است که معمولاً کنار یا زیر آن نمایش داده شده است. آیکن‌ها در گروه‌هایی چیده شده‌اند و نام هر گروه، زیر آیکن‌های آن درج شده است. به هر گروه، گروه‌ریبونی می‌گوییم.

در این کتاب به جای اینکه بگوییم از گروه ریبونی Allignment در برگه ریبونی Home، روی فرمان Wrap Text کلیک کنید، می‌گوییم مسیر ریبونی Home\Allignment\Wrap Text را دنبال کنید.

کلیک روی نخستین برگه‌ریبونی که File نام دارد، پنجره Backstage را احضار می‌کند که فرمان‌هایی در سمت چپ آن دیده می‌شود. برای اشاره به این فرمان‌ها، مثلاً فرمان Options، می‌گوییم مسیر File\Options را دنبال کنید.

فرمان‌های Visual Basic Editor

محیط VB Editor پنجره‌ای است که در آن می‌توان با کدهای VBA خود، کار کرد و دارای رابط کاربری منو و نوارابزار دیگری است. مثلاً فرمان Tools\References می‌گوید روی منوی Tools کلیک و آیتم منویی References را انتخاب کنید.

قرار دادهای صفحه‌کلید

برای وارد کردن داده‌ها لازم است از صفحه‌کلید استفاده کنید. همچنین، می‌توان مستقیماً به وسیله صفحه‌کلید با منوها و پنجره‌های محاوره‌ای کار کرد- روشی که شاید برایتان ساده‌تر باشد اگر انگشتانتان قبلاً روی کلیدها قرار داشته باشد.

کدهای VBA

کدهای VBA که در این کتاب به کار رفته را مستقیماً از ماژول VBA کپی و در کتاب paste کرده‌ایم. فونت کدها متفاوت است و برای خوانایی و کمک به فهم آنها از یک یا چند تب (تورفتگی) استفاده کرده‌ایم. اگر یک خط کد در یک خط از کتاب جا نشده باشد، از استاندارد توالی ادامه‌دادن خط VBA استفاده کرده‌ایم؛ یعنی در انتهای خط، یک فاصله و سپس یک کارکتر آندرلاین (زیرخط یا _) گذاشته‌ایم و ادامه کد را در خط بعدی آورده‌ایم. برای نمونه، دو خط زیر، یک گزاره کد واحد است:

```
columnCount = Application.WorksheetFunction. _
CountA(Range("A:A")) + 1
```

یعنی می‌توان این کد را در دو خط به همین شکل وارد کرد یا پس از برداشتن فاصله و آندرلاین، آنرا در یک تک خط وارد محیط برنامه کرد.

سازماندهی کتاب

این کتاب از ۵ بخش اصلی تشکیل شده است که احتمالاً ترجمه آنرا در دو جلد ارائه دهیم.

بخش ۱؛ مقدمه‌ای بر Excel VBA (جلد ۱ کتاب)

در این بخش، با ارائه اساس کار برنامه‌نویسی که برای ساخت و مدیریت توابع و زیررویه‌های اکسل نیاز خواهید داشت، شما را با VBA آشنا می‌کنیم. فصل ۱ با نگاهی مفهومی به توسعه برنامه کاربردی اکسل، زمینه را برایتان فراهم می‌کند. فصل‌های ۲ تا ۶، هر چیزی که برای آغاز کدنویسی در VBA لازم است را بیان می‌دارد. در فصل ۷ نیز مثال‌هایی مفید و کاربردی ارائه می‌شود.

بخش ۲: تکنیک‌های پیشرفته VBA (جلد ۲ کتاب)

در فصل ۸ و ۹ روی شیوه استفاده از VBA برای کار با جدول و نمودارهای چرخشی (pivot) بحث می‌شود (و نیز گراف‌های Sparkline). فصل ۱۰ روی تکنیک‌های مختلفی که برای تعامل با دیگر برنامه‌های کاربردی می‌توان استفاده کرد بحث می‌کند (مانند Word و Outlook). فصل ۱۱ روی شیوه کار با فایل‌ها و منابع داده‌ای خارجی بحث می‌کند، از جمله نحوه کنترل Power Query از VBA.

بخش ۳: کار با UserForms

چهار فصل این بخش، پنجره‌های سفارشی را پوشش می‌دهد (UserForms). فصل ۱۲، برخی جایگزین‌های پیش‌ساخته برای ایجاد UserForm های سفارشی را ارائه می‌دهد. فصل ۱۳، مقدمه‌ای بر UserForm ها و انواع کنترل‌هایی که می‌توان استفاده کرد را بیان می‌دارد. در فصل ۱۴ و ۱۵ نیز مثال‌هایی از پنجره‌های سفارشی، از پایه تا پیشرفته آورده شده است.

بخش ۴: توسعه برنامه‌های کاربردی اکسل

فصل‌های این بخش، با عناصر مهم ایجاد برنامه‌های کاربردی کاربر-محور (User-oriented) سروکار دارد. فصل ۱۶، ایجاد افزونه‌ها را مطرح می‌کند. فصل ۱۷ و ۱۸ شیوه اصلاح و تغییر ریبون اکسل و منوهای میان‌بر را بیان می‌دارد. در فصل ۱۹ چند روش برای ارائه کمک برخط برای برنامه‌های کاربردی که نوشته‌اید آمده است. فصل ۲۰، یک راهنما برای توسعه برنامه‌های کاربردی کاربر-محور ارائه می‌شود. در فصل ۲۱ با برخی اطلاعات مربوط به سازگاری، برنامه‌نویسی VBA را مرور می‌کنیم.

بخش ۵: پیوست

راهنمای ارجاع به همه گزاره‌ها و توابع VBA به وسیله کلیدواژه‌های تخصیص یافته به آنها.

فایل‌های تمرینی کتاب

فایل‌های تمرینی کتاب در سایت کتاب اصلی، و نیز در صفحه این کتاب در سایت انتشارات پندارپارس قرار دارد:

www.wiley.com/go/excel2019powerprogramming

www.pendarepars.com

توجه: پس از جست‌وجوی کتاب در سایت پندارپارس، وارد صفحه کتاب شوید و روی سورس کد و ضمائم کلیک کنید. لازم است در سایت ثبت نام کنید.

بخش یک

مقدمه‌ای بر Excel VBA

در این بخش، داریم:

فصل ۱: الزامات توسعه برنامه کاربردی صفحه گسترده.....	۹
فصل ۲: معرفی زبان VBA.....	۳۱
فصل ۳: بنیان برنامه‌نویسی VBA.....	۸۱
فصل ۴: کار با رویه‌های ساب VBA.....	۱۲۷
فصل ۵: ایجاد رویه‌های FUNCTION.....	۱۶۷
فصل ۶: آشنایی با رویدادهای اکسل (EXCEL EVENTS).....	۲۰۹
فصل ۷: مثال‌ها و تکنیک‌های برنامه‌نویسی VBA.....	۲۴۵

فصل ۱

الزامات توسعه برنامه کاربردی صفحه گسترده

در این فصل می‌خوانیم:

- ✓ آشنایی با گام‌های اساسی مربوط به توسعه برنامه کاربردی صفحه گسترده
- ✓ تعیین نیازهای کاربران نهایی
- ✓ طراحی برنامه‌های کاربردی برای رفع نیاز کاربران
- ✓ توسعه و آزمایش برنامه‌های کاربردی خود
- ✓ مستندسازی تلاش‌های توسعه‌ای خود و نوشتن سند کاربری

برنامه کاربردی صفحه گسترده چیست؟

برای اهداف این کتاب، برنامه کاربردی صفحه گسترده، یک فایل صفحه گسترده است (یا گروهی از فایل‌های مرتبط) که به‌گونه‌ای طراحی شده تا هر فردی به‌جز توسعه‌دهنده آن بتواند کارهای مشخصی را بدون آموزش زیادی انجام دهد. نسبت به این تعریف، بیشتر فایل‌های صفحه گسترده که توسعه می‌دهید ممکن است کاملاً برنامه کاربردی صفحه گسترده به شمار نیاید. شاید ده‌ها یا صدها فایل صفحه گسترده در درایو سخت خود داشته باشید، اما این یک شرط مطمئن است که بیشتر آنها برای استفاده دیگران طراحی نشده‌اند.

هر برنامه کاربردی صفحه گسترده خوب، کارهای زیر را انجام می‌دهد:

- کاربر نهایی را قادر می‌سازد کار مشخصی را انجام دهد که او شاید امکان انجامش به روش دیگر را نداشته است.
- راه‌حل مناسبی برای انجام کار، ارائه می‌دهد. (محیط صفحه گسترده، همیشه رویکرد بهینه‌ای ندارد).
- انجام کاری که قرار است انجام شود. این شرط لازم شاید بدیهی باشد، اما نه در همه کارهای غیرمرسوم برنامه‌های کاربردی که از این آزمون رد می‌شوند.

- تولید نتایج دقیق و بدون باگ.
- استفاده از راهکارهای مناسب و کافی و الگوریتم‌هایی برای انجام کار آنها.
- گیر انداختن خطاها پیش از اینکه کاربر با آنها برخورد کند.
- اجازه حذف تصادفی (یا عمدی) یا تغییر اجزای مهم به کاربر داده نمی‌شود.
- داشتن یک محیط کاربری شفاف و ثابت که کاربر همواره بداند چگونه پیش رود.
- داشتن فرمول‌ها، ماکروها، و عناصر محیط کاربری که به‌خوبی مستندسازی شده‌اند و در صورت لزوم، امکان تغییرات بعدی را داشته باشند.
- طراحی به‌گونه‌ای است که بتوان به روش‌های ساده و بدون ایجاد تغییرات عمده، آنرا اصلاح کرد. واقعیت مهم این است که کاربران همواره نیاز به ایجاد تغییرات دارند.
- دارای سامانه کمکی قابل دسترس باشد تا اطلاعات مهم دست‌کم در رویه‌های عمده، ارائه شود.
- طراحی به‌گونه‌ای باشد که برنامه، قابل حمل باشد و روی هر سامانه‌ای که دارای نرم‌افزار مناسبی باشد اجرا شود (در این مورد، یک کپی از یک نسخه پشتیبانی شده از اکسل).

جای شگفتی ندارد که ممکن است برنامه‌های کاربردی صفحه گسترده، برای بسیاری از سطوح کاربردی متفاوت تولید شود، دامنه‌ای از یک الگوی ساده جای-خالی-را-پر-کن تا یک برنامه کاربردی کاملاً پیچیده که از یک محیط کاربری سفارشی استفاده می‌کند و شاید حتی شبیه صفحه گسترده هم نباشد.

توجه: زین پس در برگردان **Application**، به جای واژه "برنامه کاربردی"، از واژه "برنامک" استفاده می‌کنیم.

گام‌هایی برای توسعه برنامک

هیچ دستور ساده و مطمئنی برای توسعه یک برنامک صفحه گسترده کارآمد وجود ندارد. هر فردی دارای سبک خودش برای ایجاد چنین برنامکی است. افزون بر این، هر پروژه‌ای متفاوت است و نیاز به رویه خودش دارد. در نتیجه، تقاضاها و تجارب افرادی که با آنها یا برایشان کار می‌کنید نیز، در شیوه توسعه فرایند محصولات، نقش بازی می‌کند.

توسعه دهندگان صفحه گسترده معمولاً فعالیت‌های زیر را انجام می‌دهند:

- تعیین نیازهای کاربر (کاربران)
- طراحی یک برنامک که این نیازها را مرتفع سازد.
- تعیین مناسب‌ترین رابط کاربری (user interface)

- ایجاد صفحه گسترده، فرمول‌ها، ماکروها، و رابط کاربری
- آزمایش و دیباگ برنامه
- کوشش برای ضدضربه کردن برنامه
- ایجاد ظرافت و جذابیت‌های ظاهری برنامه
- مستندسازی تلاش صورت گرفته در این توسعه
- توسعه سند کاربری و سامانه‌های کمکی Help
- توزیع برنامه برای کاربری مشخص
- به‌روزرسانی برنامه در صورت نیاز

همه موارد بالا برای هر برنامه‌ای لازم نیست و ترتیب انجام این موارد، از پروژه‌ای به پروژه دیگر، متفاوت است. در ادامه کتاب، هر یک از این موارد را شرح می‌دهیم. بیشتر اینها نیز شامل جزئیات فنی می‌شود که در فصل‌های مربوطه آمده است.

تعیین نیازهای کاربر

هنگامی که یک پروژه اکسل را برمی‌دارید، یکی از نخستین گام‌های شما، تشخیص دقیق کاری است که کاربران نهایی لازم دارند. ناتوانی در ارزیابی کامل نیازهای کاربران نهایی، اغلب به‌زودی منجر به کار اضافه می‌شود، و مجبورید برنامه را برای رسیدن به هدف اصلی خود اصلاح کنید.

گاهی با کاربران نهایی آشنایی کامل دارید (یا اینکه خودتان کاربر نهایی هستید). در موارد دیگر، (مثلا اگر شما مشاور توسعه‌ای یک پروژه برای یک مشتری جدید هستید)، شاید چیز زیادی درباره کاربران یا وضعیت‌شان ندانید (یا اصلا ندانید).

نیاز کاربران را چگونه تعیین می‌کنید؟ اگر خواسته باشید یک برنامه صفحه گسترده را توسعه دهید، ایده خوب این است که با کاربران نهایی ملاقات کنید و سؤال‌های خاصی را از آنها بپرسید. بهتر است هر چیزی را بنویسید، دیاگرام‌های جریان را ایجاد کنید، به جزئیات ریز توجه کنید، و برای اطمینان از اینکه محصولی که تحویل می‌دهید، همان است که نیاز داشته‌اند، هر کاری که لازم است را انجام دهید.

به برخی راهنمایی‌های زیر که به انجام ساده‌تر این گام کمک می‌کند توجه کنید:

- فرض نکنید که آنچه کاربر نیاز دارد را می‌دانید. حدس بعدی در این مرحله همیشه موجب مشکلات بعدی می‌شود.
- در صورت امکان، مستقیماً با کاربران نهایی برنامه صحبت کنید، نه فقط با سرپرست یا مدیر.

- برای رسیدن به نیازهای کاربران، کاری که فعلا انجام می‌شود را یاد بگیرید (در صورت وجود). شاید بتوانید با وفق دادن ساده یک برنامه موجود، از انجام تکراری برخی کارها جلوگیری کنید. در کمترین حدش، نگاهی به راه‌حل‌های جاری بیاندازید تا با عملیات کاری آشنا شوید.
 - منابع موجود در سایت کاربر را شناسایی کنید. برای نمونه، سعی کنید تعیین کنید که آیا باید پیرامون هر محدودیت سخت‌افزاری یا نرم‌افزاری کار کنید یا خیر.
 - در صورت امکان، سامانه‌های سخت‌افزاری مشخصی که استفاده خواهد شد را تعیین کنید. اگر برنامه شما در سامانه‌های کند قابل اجراست، لازم است آنرا به حساب آورید.
 - تعیین اینکه کدام نسخه‌های اکسل استفاده می‌شود. به یاد داشته باشید که کاربران می‌توانند نسخه‌های اکسل را در macOS، پلتفرم‌های موبایل، و ویندوز اجرا کنند. این باید هنگام طراحی یک راه‌حل اکسل خودکارسازی شده، مدنظر قرار گیرد. گرچه مایکروسافت، هرچه در توانش است را برای برانگیختن کاربران در به‌روزرسانی به آخرین نسخه نرم‌افزار انجام می‌دهد، اما بیشتر کاربران اکسل این کار را نمی‌کنند.
 - شناخت سطح مهارت کاربران نهایی. این اطلاعات در طراحی برنامه به شکل اختصاصی، به شما کمک می‌کند.
 - تعیین اینکه برنامه چه مدت استفاده خواهد شد و آیا تغییری در طول عمر پروژه پیش‌بینی می‌شود. دانستن این اطلاعات می‌تواند در مقدار تلاشی که در پروژه می‌کنید تأثیر داشته باشد و در طراحی تغییرات کمک کند.
- در آخر، اگر ویژگی‌های پروژه، پیش از تکمیل برنامه تغییر یابد، شگفت‌زده نشوید. این رخداد مرسوم است و موقعیت بهتر این است که انتظار تغییرات را داشته باشید تا اینکه با آنها شگفت‌زده شوید. کافی است مطمئن شوید که در قراردادتان (اگر دارید) به مسئله تغییر ویژگی‌ها اشاره شده باشد.

طراحی برنامه‌ای که نیازهای کاربر را رفع می‌کند

پس از تعیین نیازهای کاربران نهایی، پریدن مستقیم روی اکسل و کلنجر رفتن با آن، هوس انگیز است. اما به پیش نیازهای دیگری نیاز دارید.

معماران، خانه‌ای بدون داشتن مجموعه نقشه‌های فنی نمی‌سازند، و شما هم نباید برنامه صفحه گسترده را بدون برخی از انواع طرح‌ها ایجاد کنید. رسمیت طرح‌تان بستگی به قلمروی پروژه و سبک

عمومی کار کردن دارد، اما باید دست‌کم، مقدای زمان صرف فکر کردن درباره اینکه چه کاری قرار است انجام دهید کنید و با یک طرح از عملیاتی که قرار است انجام پذیرد، وارد عمل شوید.

پیش از اینکه آستین‌های خود را بالا بزنید و دکمه‌های کیبورد را بفشارید، خوب است مقداری وقت صرف بررسی روش‌های گوناگونی کنید که می‌توان با آنها مشکل کاربر را حل کرد. این دوره زمانی طراحی و برنامه‌ریزی، جایی است که دانش اکسل به کار نمی‌آید. پرهیز از ساخت کوچه‌های بن‌بست به‌جای گرفتار شدن در آنها، همیشه ایده خوبی است.

اگر از ده‌ها خبره اکسل بخواهید یک برنامه‌ک را بر مبنای مشخصات دقیقی طراحی کنند، این شانس را دارید که ده‌ها پیاده‌سازی متفاوت از آن پروژه را دریافت کنید که همگی، آن مشخصات را داشته باشند. در این راه‌حل‌ها، برخی بهتر از بقیه هستند، زیرا اکسل اغلب چند گزینه برای تکمیل یک کار ارائه می‌دهد. چنانچه با زیر و بم اکسل آشنا باشید، ایده خوبی از متدهای نهفته‌ای که در اختیار دارید خواهید داشت و می‌توانید یکی از مناسب‌ترین آنها را برای پروژه خود به دست بگیرید. اغلب، ذره‌ای از تفکر خلاق، منجر به یک رویه غیرمعمول می‌شود که از دیگر روش‌ها برتری خواهد داشت.

در ابتدای مرحله این دوره طراحی و برنامه‌ریزی، برخی گزینه‌های عمومی را در نظر بگیرید؛ همچون: **ساختار فایل:** درباره اینکه آیا می‌خواهید از یک کارپوشه با چند کاربرگ استفاده کنید یا از چند کارپوشه تک کاربرگی، یا از یک فایل الگو، بیاندیشید.

ساختار داده: همواره این را مدنظر بگیرید که چگونه داده‌ها ساختار بندی خواهد شد و تعیین اینکه آیا از فایل‌های دیتابیس خارجی، منابع داده‌ای ذخیره شده در فضای ابری، یا نگهداری همه چیز در کاربرگ‌ها استفاده خواهید کرد.

افزونه یا فایل کارپوشه: در برخی موارد، شاید یک افزونه (add-in) بهترین انتخاب برای محصول نهایی‌تان باشد، یا شاید بتوانید از یک افزونه به‌همراه یک کارپوشه استاندارد استفاده کنید.

نسخه اکسل: آیا برنامه اکسل تنها با اکسل ۲۰۱۹ قابل استفاده خواهد بود یا نیاز به اجرا در نسخه‌های قدیمی‌تر اکسل خواهد داشت؟ درباره نسخه‌های اکسل اجرایی روی دیگر پلتفرم‌ها همچون macOS یا دستگاه‌های موبایل چطور؟ این توجه‌ها مهم هستند زیرا هر نسخه جدید اکسل، ابزارهایی را اضافه می‌کند که در نسخه‌های پیشین موجود نیست.

رسیدگی به خطا: رسیدگی به خطا، یک مسئله اصلی در برنامه‌هاست. نیاز دارید تعیین کنید که برنامه‌تان خطاها را شناسایی و به آنها رسیدگی کند. برای نمونه، اگر برنامه شما عملیات جدول چرخشی را روی برگه فعال انجام می‌دهد، لازم است امکان رسیدگی به مواردی که در آنها جدول چرخشی در برگه فعال وجود ندارد رسیدگی شود.

استفاده از ابزارهای ویژه: اگر برنامه شما نیاز به خلاصه‌سازی^۱ مقداری از داده‌ها داشته باشد شاید بخواهید استفاده از ابزارهای جدول چرخشی را مدنظر قرار دهید، یا شاید بخواهید از ابزار اعتبارسنجی داده‌های اکسل به‌عنوان کنترلی روی ورود داده‌های معتبر استفاده کنید.

مسائل بازدهی: زمان شروع فکر کردن به افزایش سرعت و کارآمدی برنامه در حالت توسعه، زمانی نیست که برنامه تکمیل شده است و کاربران مشغول نالیدن از آن باشند.

سطح امنیت: همان‌گونه که شاید بدانید، اکسل چندین گزینه محافظتی برای محدودیت دسترسی به عناصر مشخصی از کارپوشه را دارد. برای نمونه، می‌توان سلول‌ها را قفل کرد تا فرمول‌ها نتواند تغییر یابد، و می‌توان یک گذرواژه تخصیص داد تا مانع مشاهده یا دسترسی کاربران غیرمجاز به فایل‌های خاصی شد. تعیین دقیق آنچه نیاز به محافظت دارد از همان ابتدای کار (و به چه سطحی از محافظت نیاز است)، کارتان را آسان‌تر خواهد کرد.

نکته: توجه داشته باشید که ابزارهای محافظتی اکسل، صد درصد مؤثر نیستند. اگر اصرار کامل و قطعی به این کار دارید، شاید اکسل بهترین بستر برای کارتان نباشد.

در این فاز، شاید مجبور باشید با ملاحظات چند پروژه مشخص دیگر سروکار داشته باشید. همه گزینه‌ها را درنظر بگیرید و روی نخستین راه‌حلی که به ذهنتان خطور کرد بنمایید.

ملاحظه دیگر طراحی، یادآوری برنامه‌ریزی برای تغییر است. چنانچه برنامه را تا حد امکان به‌شکل عمومی بسازید، به نفع خودتان کار کرده‌اید. برای نمونه، رویه‌ای ننویسید که تنها یک بازه مشخص از سلول‌ها را بپذیرد، در عوض به‌گونه‌ای طراحی کنید که پذیرای هر بازه‌ای به‌عنوان آرگومانش باشد. هنگامی که تغییرات گریزناپذیر درخواست شود، چنین طرحی ارائه نسخه‌های جدیدتر را برایتان آسان‌تر خواهد کرد. همچنین ممکن است متوجه شوید کاری که روی یک پروژه انجام می‌دهید مشابه کاری است که در پروژه‌ای دیگر انجام می‌دهید. هنگام طراحی و برنامه‌ریزی یک پروژه، قابل استفاده مجدد بودن را نیز برای خود لحاظ کنید.

از اینکه به کاربر نهایی اجازه دهید خط مشی خود در حل یک مشکل را به شما دیکته کند پرهیز کنید. در واقع نباید گذاشت کاربران به شما خط دهند. برای نمونه، فرض کنید با یک مدیر ملاقات می‌کنید و می‌گویید ادارات نیاز به برنامه‌ای دارند که فایل‌های متنی که در برنامه دیگر درون‌ریزی^۲ خواهد شد را بنویسد. نیاز کاربران به راه‌حلی مشخص، نباید سردرگمی ایجاد کند. نیاز واقعی کاربران، به اشتراک‌گذاری داده‌هاست. استفاده از یک فایل متنی واسط برای این کار، تنها یکی از راه‌حل‌های

¹ Summarize: Sum, Average, Max, Min, ...

² Import

ممکن است؛ شاید روش بهتری برای رفع این مشکل موجود باشد. به بیانی دیگر، به کاربران اجازه ندهید مشکل خود را با پافشاری روی یک راه‌حل تعریف کنند. تعیین بهترین دیدگاه، کار شماسست.

تعیین مناسب‌ترین رابط کاربری

هنگام توسعه صفحه‌گسترده‌هایی که دیگران قرار است استفاده کنند، نیاز به توجه ویژه به رابط کاربری دارید (یا واسط کاربری؛ User Interface). با رابط کاربری، روشی را تعریف می‌کنیم که با آن مشخص می‌شود کدام کاربر با برنامه‌ک تعامل داشته باشد و ماکروهای VBA را اجرا کند.

از زمان تولید اکسل ۲۰۰۷ که منوها و نوارابزارها جای خود را به ریبون دادند، برخی از این تصمیم‌های رابط کاربری نامربوط شده‌اند. منوها و نوارابزارهای سفارشی، با همه اهداف و سلیقه‌ها، منسوخ شده‌اند. در نتیجه، توسعه‌دهندگان می‌بایست نحوه کار با ریبون را بیاموزند.

اکسل چندین ویژگی ارائه می‌دهد که برای طراحی رابط کاربری کفایت می‌کند:

- سفارشی‌سازی ریبون
 - سفارشی‌سازی منوی میان‌بر
 - کلیدهای میان‌بر
 - پنجره‌های محاوره‌ای سفارشی (UserForms)
 - کادرهای پیام (پنجره‌ها) و کادرهای ورودی
 - کنترل‌هایی که مستقیماً روی کاربرگ جای می‌گیرند (مانند ListBox یا CimmandButton)
- بحث درباره این ویژگی‌ها در صفحات بعدی آمده است و در فصل‌های بعدی کاملاً تشریح می‌شود.

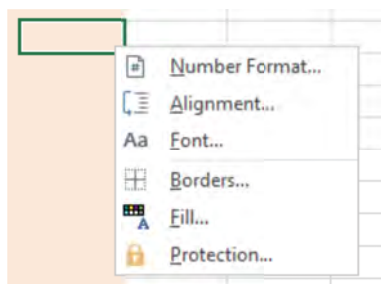
سفارشی‌سازی ریبون

به عنوان یک توسعه‌دهنده، کنترل کافی روی محتویات ریبون که شامل برگه‌ها و فرمان‌های درون آنها پس از باز شدن برنامه‌ک اکسل شماسست، دارید. هرچند، اکسل به کاربران نهایی اجازه می‌دهد ریبون را تغییر دهند، اما تغییرات ایجاد در UI به‌وسیله کدنویسی، کار ساده‌ای نیست.

یادآوری: در فصل ۱۷، "کار با ریبون"، اطلاعات کافی درباره کار روی ریبون آمده است.

سفارشی‌سازی منوهای میان‌بر

اکسل به توسعه‌دهنده VBA اجازه سفارشی‌سازی منوهای میان‌بر کلیک راست را می‌دهد. این منوها می‌توانند روش سریع‌تر اجرای یک دستور را، بدون رفتن به مسیری دورتر در لابه‌لای ریبون، نسبت به بازه‌ای که در حال کار در آن هستند، پیشنهاد می‌دهند. شکل زیر، یک منوی میان‌بر سفارشی-سازی شده را پس از کلیک راست روی یک سلول نشان می‌دهد.



یادآوری: فصل ۱۸ درباره نحوه کار با منوهای میان‌بر با استفاده از VBA بحث می‌کند که شامل برخی محدودیت‌های ناشی از رابط کاربری تک‌سندی است که در اکسل ۲۰۱۳ معرفی شده است.

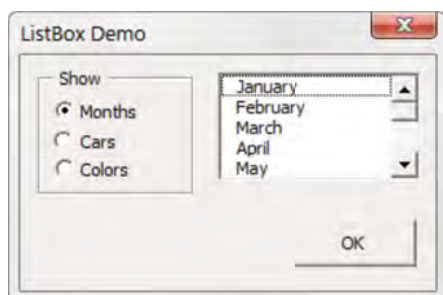
ایجاد کلیدهای میان‌بر

گزینه دیگر رابط کاربری که در اختیارتان است، یک کلید میان‌بر سفارشی است. اکسل اجازه می‌دهد یک کلید Ctrl (یا Shift+Ctrl) ترکیبی را برای یک ماکرو تعریف کنید. با فشار این کلیدهای ترکیبی توسط کاربر، ماکرو اجرا می‌شود.

هرچند، دو اخطار وجود دارد. نخست اینکه کاربر را به‌خوبی توجیه کنید که کدام کلیدها فعال است و چه کاری انجام می‌دهند. دوم اینکه، یک ترکیب کلیدی که پیش‌تر برای کار دیگری استفاده شده است را تخصیص ندهید. هر ترکیب کلیدی که به یک ماکرو تخصیص دهید، اولویتی بالاتر از کلیدهای میان‌بر درون‌ساخت می‌گیرد. برای نمونه، Ctrl+S یک کلید میان‌بر درون‌ساخت اکسل برای ذخیره فایل جاری است. اگر این ترکیب کلید را به یک ماکرو تخصیص دهید، قابلیت ذخیره کردن فایل را پس از فشار این کلیدها از دست خواهید داد. فراموش نکنید که کلیدهای میان‌بر، نسبت به بزرگی و کوچکی واژگان حساس هستند، پس می‌توان از یک ترکیبی همچون Ctrl+Shift+S استفاده کرد (Shift+S در تایپ S بزرگ به‌کار می‌رود).

ایجاد پنجره‌های محاوره‌ای سفارشی

هر فردی که سال‌ها از یک کامپیوتر شخصی استفاده کرده باشد با واژه پنجره محاوره‌ای آشناست (کادر محاوره‌ای یا همان dialog box که به اختصار، پنجره می‌گوییم). در نتیجه، پنجره‌های سفارشی اکسل می‌تواند نقش اصلی را در رابط کاربری که برای برنامه‌ خود طراحی می‌کنید بازی کند. در شکل زیر، نمونه‌ای از آن را می‌بینید.



پنجره سفارشی را با نام UserForm می‌شناسیم. این پنجره می‌تواند ورودی کاربر را بگیرد، گزینه‌ها یا ترجیحات کاربر را دریافت، و مستقیماً کل برنامه شما را دنبال کند. عناصری که یک UserForm را ایجاد می‌کند (دکمه‌ها، لیست‌های پایین‌رونده، چک‌باکس‌ها و غیره)، control نامیده می‌شود – به‌شکلی ویژه‌تر، کنترل‌های ActiveX. اکسل، ترتیب استاندارد از کنترل‌های اکتیوایکس را ارائه می‌دهد و می‌توان کنترل‌های شخص‌سومی^۱ را نیز داخل کرد.

پس از افزودن یک کنترل به یک پنجره، می‌توان آنرا به یک سلول کاربرگ پیوند زد تا نیازی به هیچ ماکرویی نشود (به‌جز ماکروی ساده‌ای که پنجره را باز می‌کند). پیوند دادن یک کنترل به یک سلول، کار ساده‌ای است، اما همیشه بهترین راه برای رسیدن به ورودی کاربر، از یک پنجره نیست. بیشتر وقت‌ها می‌توان ماکروهای VBA را توسعه داد که با پنجره‌های سفارشی ما کار کنند.

یادآوری: جزئیات UserForm‌ها در بخش ۳ کتاب آمده است.

افزودن کنترل‌های ActiveX به کاربرگ

اکسل همچنین اجازه افزودن کنترل‌های UserForm ActiveX را به لایه ترسیم کاربرگ می‌دهد (لایه‌ای نامرئی در بالای یک برگه که تصاویر، نمودارها و اشیاء دیگر را نگه می‌دارد). شکل زیر، یک مدل کاربرگ ساده‌ای را نشان می‌دهد با چند کنترل UserForm که مستقیماً به آن افزوده شده است.

^۱ Third-party: اشخاص یا شرکت‌های واسطه ارائه دهنده یک سرویس

این برگه، حاوی این کنترل‌های اکتیوایکس است: یک ChechBox، یک ScrollBar و دو سری OptionButtons. این کارپوشه فاقد ماکرو است. در عوض، کنترل‌هایش به سلول‌های کاربرگ پیوند خورده‌اند.

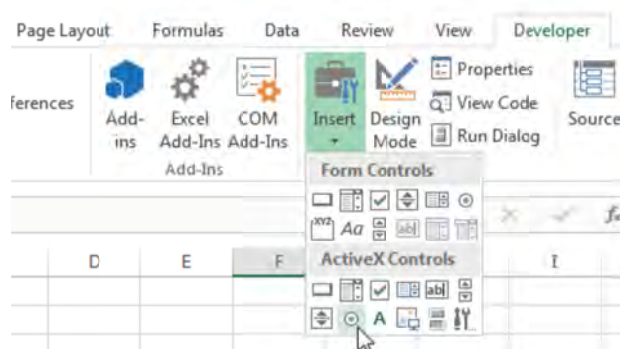
Mortgage Loan Parameters		Linked Cells	
Purchase Price:	\$345,000	565 Interest from scroller	
<input type="checkbox"/> Finance the \$5,000 loan		5.65 Percent	
Pct. Down Payment:	Loan Term:	FALSE	Loan Fee
<input type="radio"/> 10%	<input type="radio"/> 30-year fixed		\$276,000 Loan
<input type="radio"/> 15%	<input checked="" type="radio"/> 15-year fixed	FALSE	30-year
<input checked="" type="radio"/> 20%		TRUE	15-year
Loan Amount:	\$276,000		15 Year term
Interest:	5.65%	FALSE	10% down
Monthly Payment:	\$2,277.18	FALSE	15% down
		TRUE	20% down
			20% Down payment

توجه: فایل تمرینی این کاربرگ، Controls.xlsx نام دارد.

شاید مرسوم‌ترین کنترل، CommandButton باشد. این کنترل به‌خودی‌خودش کاری نمی‌کند و نیاز دارید به هر دکمه کامندباتونی، یک ماکرو را پیوست کنید. (گاهی به کنترل دکمه می‌گوییم).

استفاده مستقیم از کنترل‌های پنجره در یک کاربرگ، اغلب نیاز به پنجره‌های سفارشی را رفع می‌کند. اغلب می‌توان به شکل ماهرانه‌ای، عملیات یک صفحه گسترده را با افزودن چند کنترل اکتیوایکس (یا کنترل فرم) به کاربرگ، ساده‌سازی کرد. این کنترل‌های اکتیوایکس، به کاربر اجازه می‌دهند به جای وارد کردن در سلول‌ها، با عملیات کنترل‌های مشابه، انتخاب خود را انجام دهند.

دسترسی به این کنترل‌ها از مسیر ریبونی Developer\Controls\Insert است (شکل زیر). اگر برگه ریبونی Developer را ندارید، آنرا از برگه Customize Ribbon در پنجره Excel Options به ریبون بیافزایید.



دو نوع کنترل موجود است: کنترل‌های فرم و کنترل‌های اکتیوایکس. هر دو دسته، دارای مزایا و معایب خود است. عموماً، استفاده از کنترل‌های فرم ساده‌تر است، اما دیگری کمی پیچیده‌تر است. خلاصه‌ای از کنترل‌های هر دو دسته را در جدول ۱-۱ ببینید.

جدول ۱-۱. کنترل‌های ActiveX در مقابل کنترل‌های Form

	ActiveX Controls	Form Controls
Excel versions	97, 2000, 2002, 2003, 2007, 2010, 2013, 2016, 2019	5, 95, 97, 2000, 2002, 2003, 2007, 2010, 2013, 2016, 2019
Controls available	CheckBox, TextBox, Command Button, OptionButton, ListBox, ComboBox, ToggleButton, Spin Button, ScrollBar, Label, Image (and others can be added)	GroupBox, Button, Check Box, OptionButton, List-Box, DropDown (ComboBox), ScrollBar, Spinner
Macro code storage	In the code module for the sheet	In any standard VBA module
Macro name	Corresponds to the control name (for example, CommandButton1_Click)	Any name you specify
Correspond to	UserForm controls	Pre-Excel 97 dialog sheet controls
Customization	Extensive, using the Properties box	Minimal
Respond to events	Yes	Click or Change events only

تلاش برای توسعه

پس از شناسایی نیازهای کاربر، رویه‌ای را برای رفع این نیازها تعیین می‌کنید و در مورد اجزایی که برای رابط کاربری استفاده خواهید کرد تصمیم می‌گیرید. سپس، زمان ایجاد برنامه‌ک فرا می‌رسد. البته این گام، شامل مقدار بسیار زیادی از کل زمانی است که روی یک پروژه خاص صرف می‌کنید.

شیوه‌ای که درباره توسعه برنامه در پیش می‌گیرید بستگی به سبک شخصی شما و طبیعت برنامه دارد. به جز برای کارپوشه‌های الگوی ساده fill-in-the-blank، احتمالاً برنامه شما از ماکروها استفاده خواهد کرد. ایجاد ماکروها در اکسل آسان است اما ایجاد ماکروهای خوب، دشوار است.

درباره خودتان همراه با کاربر نهایی

در این بخش، روی مشکلات مهم توسعه بحث می‌کنیم؛ زمانی که ظاهر برنامه شما تقریباً نهایی می‌شود و زمان بسته‌بندی و توزیع کارتان نزدیک می‌شود.

آزمایش برنامه

چند بار از یک برنامه نرم‌افزاری تجاری استفاده کرده‌اید و تنها در یک لحظه وخیم، متوجه وجود باگ و خراب کارکردن آن شده‌اید؟

م احتمالاً این است که دلیل مشکل آن است که آزمایش ناکافی موجب شده که همه باگ‌ها را نتوانید بگیرید. همه نرم‌افزارها غیرمستقیم دارای باگ هستند اما در بهترین نرم‌افزار، باگ‌ها کاملاً مبهم هستند. همان‌گونه که خواهید دید، گاهی در اکسل باید روی باگ‌ها کار کرد تا برنامه به جایی برسد تا به خوبی کار کند.

پس از ساخت برنامه خود، باید آنرا آزمایش کنید. آزمایش، یکی از بحرانی‌ترین گام‌هاست؛ غیرمرسوم نیست که مقدار زمانی که صرف تست و دیباگ یک برنامه می‌کنید، با زمانی که صرف ساخت آن کرده‌اید برابر شود. در واقع، باید وقت زیادی برای آزمایش در حین فاز توسعه صرف کنید. پس از نوشتن یک روتین VBA یا ایجاد فرمول‌هایی در یک کاربرگ و همه این کارها، می‌خواهید مطمئن شوید برنامه شما طبق انتظارتان کار می‌کند.

همانند برنامه‌های کامپایل شده استاندارد، برنامه‌های صفحه گسترده‌ای که توسعه می‌دهید، مستعد باگ خواهند بود. هر باگ می‌تواند این‌گونه تعریف شود: (۱) چیزی که حین اجرای یک برنامه یا برنامه اتفاق می‌افتد اما نباید اتفاق بیافتد، یا (۲) چیزی که اتفاق نمی‌افتد درحالی که باید اتفاق بیافتد. هر دو نوع باگ کاملاً ناپسند است و باید برنامه‌ریزی کنید که یک بخش مناسب از زمان توسعه‌ای خود را به آزمایش برنامه اختصاص دهید تا همه شرایط معقول لحاظ شود و مشکلات یافت شده، رفع شود.

مهم است که هر برنامه‌گسترده‌ای که برای دیگران توسعه می‌دهید را تا آخر آزمایش کنید. و بسته به بینندگان احتمالی آن، شاید بخواهید برنامه‌کاربرد را ضد ضربه کنید. به دیگر سخن، تلاش کنید به همه خطاها رسیدگی کنید و آنقدر آنرا واکاوی کنید که خطاهای احتمالی هم بیرون زند و تلاش‌های موزونی برای دوری از خطاها کنید- یا دست کم به شکل مناسبی به آنها رسیدگی کنید.

این دوراندیشی نه تنها به کاربران نهایی کمک می‌کند بلکه کار شما را نیز ساده‌تر می‌کند و از شهرت و اعتبار شما نیز محافظت می‌کند. همچنین، استفاده از آزمون بتا را نیز در نظر داشته باشید- کاربران نهایی شما به مانند کاندیدهایی عمل می‌کنند، چراکه همین افراد قرار است در آخر، از محصول شما استفاده کنند.

هرچند شاید آزمایش همه احتمالات، امکان‌پذیر نباشد، اما ماکروها باید قادر به رسیدگی به انواع خطاهای مرسوم باشند. برای نمونه، آیا اگر کاربر یک رشته متنی را به جای مقدار عددی وارد کند چه می‌شود؟ یا اگر کاربر تلاش کند ماکروی شما را هنگامی که کارپوشه باز نیست اجرا کند چه؟ یا اگر کاربر، پنجره‌ای را بدون انجام هیچ انتخابی ببندد چه؟ یا اگر کلیدهای Ctrl+F6 را بفشارد و به پنجره بعدی بپرد چه؟ پس از کسب تجربه، با این گونه مسائل به‌خوبی آشنا خواهید شد و بدون حتی فکر کردن در این باره، اینها را لحاظ می‌کنید.

در حاشیه؛ آزمایش بتا چیست؟

تولیدکنندگان نرم‌افزار، نوعا دارای یک چرخه شدید آزمایشی برای تولیدات جدید خود هستند. پس از آزمایش‌های داخلی گسترده، محصول پیش-انتشار معمولا برای گروهی از کاربران علاقه‌مند فرستاده می‌شود که تست بتا را انجام دهند. این فاز اغلب، مشکلات دیگری را آشکار می‌کند که معمولا پیش از انتشار نهایی محصول، اصلاح می‌گردد.

اگر یک توسعه دهنده برنامه‌اکسل هستید که بیش از چند نفر از آن استفاده خواهند کرد، شاید بخواهید یک آزمایش بتا برای آن لحاظ کنید. این آزمایش، کاربران موردنظر را قادر می‌سازد که از برنامه‌کاربرد شما روی سخت‌افزارهای مختلف، با تنظیمات پیشنهادی برنامه‌کاربرد، استفاده کنند.

دوره بتا باید پس از تکمیل همه آزمایش‌های شخصی شما و پس از اینکه احساس کردید آماده توزیع نهایی است آغاز شود. لازم است گروهی از کاربران را برای کمک شناسایی کنید. چنانچه هر چیزی که نهایتا در برنامه‌کاربرد قرار می‌گیرد را توزیع می‌کنید، این فرایند بهتر کار می‌کند: مستندسازی کاربر، نصب برنامه، سامانه help، و غیره. آزمایش بتا را می‌توانید به راه‌های گوناگونی انجام دهید، همچون بحث‌های رو در رو، ایمیل، پرسش‌نامه‌ای، و تماس‌های تلفنی یا پیامکی.

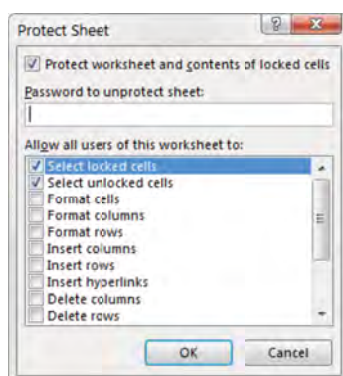
تقریباً همیشه از مشکلاتی که نیاز دارید برطرف شود یا بهبودهایی که نیاز دارید پیش از توزیع گسترده برنامه‌ها به‌عهد بگیرد آگاه می‌شوید. هرچند، آزمایش بتا نیاز به زمان بیشتری دارد و همه پروژه‌ها نمی‌توانند این‌گونه عمل کنند.

ضدضربه کردن برنامه

خراب کردن یک صفحه گسترده، بسیار آسان است، اگر کمی به آن فکر کنید. پاک کردن یک فرمول حیاتی یا یک مقدار، می‌تواند سبب بروز خطاهایی در کل کاربرگ شود- و شاید حتی در دیگر کاربرگ‌های وابسته به آن. حتی به شکلی دیگر، اگر کاپوشه خراب نخیره شود، جایگزین کپی سالم آن روی دیسک می‌شود. اگر یک رویه پشتیبان‌گیری در جایی نباشد، کاربر برنامه ممکن است به دردمر بیافتد و احتمالاً شما را مقصر بداند.

واضح است که می‌توان به آسانی دید که چرا هنگامی که کاربران- به‌ویژه تازه‌کارها- قرار است از کاربرگ‌هایتان استفاده کنند لازم است برخی موارد حفاظتی را به آن بیافزایید. اکسل چند تکنیک را برای محافظت از کاربرگ‌ها و بخش‌هایی از آنها ارائه می‌دهد:

قفل کردن سلول‌های مشخص: سلول‌های مشخصی را می‌توان قفل کرد (با استفاده از برگه Protection در پنجره Format Cells) تا کاربران نتوانند آنها را تغییر دهند. این کار تنها زمانی کارساز است که سند به‌وسیله فرمان Review\Changes\Protect Sheet محافظت شده باشد. پنجره Protect Sheet گزینه‌هایی دارد که اجازه می‌دهد تعیین کنید کاربران چه کارهایی بتوانند در برگه محافظت شده انجام دهند. (شکل بعدی)



پنهان‌سازی فرمول‌های درون سلول‌های مشخص: به‌کمک برگه Protection در پنجره Format Cells می‌توان فرمول‌های درون سلول‌های مشخصی را پنهان ساخت تا دیگران نتوانند آنها را ببینند.

این کار هم زمانی اثربخش است که سند با فرمان `Review\Changes\Protect Sheet` محافظت شده باشد.

محافظت از کل یک کارپوشه: از کل یک کارپوشه می‌توان محافظت کرد- ساختار کارپوشه، موقعیت و اندازه پنجره، یا هر دو. برای این کار، از فرمان `Review\Protect\Protect Workbook` استفاده کنید.

قفل کردن اشیاء روی کاربرگ: از ناحیه `Properties` در نوار وظیفه، برای قفل کردن اشیاء استفاده کنید (مانند شکل‌ها) و مانع جابه‌جا شدن یا تغییر آنها شوید. برای دسترسی به این ناحیه از `task pane` روی آن شیء کلیک راست و `Size and Properties` را انتخاب کنید. قفل کردن اشیاء تنها هنگامی تأثیر دارد که سند با فرمان `Review\Protect\Protect Sheet` محافظت شده باشد. به حالت پیش‌گزیده، همه اشیاء قفل می‌شود.

پنهان‌سازی ردیف‌ها، ستون‌ها، برگه‌ها، و اسناد: می‌توان ردیف‌ها، ستون‌ها، برگه‌ها، و کل کارپوشه را مخفی ساخت. این کار مانع این می‌شود که در کاربرگ، به هم ریختگی به‌نظر رسد و نیز برخی محافظت‌های معتدلی را در برابر چشمان کنجکاو کاربران ارائه می‌دهد.

نامزد کردن کارپوشه اکسل به‌عنوان پیشنهاد فقط-خواندنی: برای اطمینان از اینکه فایل نمی‌تواند با هیچ تغییری بازنویسی شود می‌توان به کارپوشه اکسل پیشنهاد فقط-خواندنی بودن (و استفاده از یک گذرواژه) داد. این کاندید شدن را می‌توان در پنجره `General Options` تنظیم کرد. از مسیر `File\Save As`، این پنجره را باز کنید و پس از انتخاب یک مسیر، در پنجره `Save As` روی دکمه `Tools` کلیک کنید. از پنجره `General Options` نیز برای تعیین گذرواژه مناسب استفاده کنید.

تخصیص گذرواژه: برای جلوگیری از باز کردن فایل توسط کاربران غیرمجاز، می‌توان یک گذرواژه به آن تخصیص داد. مسیر `File\Info\Protect Workbook\Encrypt with Password` را دنبال کنید.

استفاده از یک افزونه محافظت گذرواژه‌ای: این گونه افزونه‌ها مانع این می‌شود که کاربران تغییری در کاربرگ‌ها دهند.

در حاشیه؛ گذرواژه‌های اکسل چندان مطمئن نیستند

آگاه باشید که گذرواژه‌های اکسل اغلب می‌تواند به‌سادگی توسط برنامه‌های شکستن گذرواژه تجاری موجود که هزاران واژه را آزمایش می‌کنند شکسته شود. پس به محافظت گذرواژه‌ای اکسل

با دید ضد ضربه بودن ننگرید. هرچند، برای کاربران عادی می‌تواند خوب باشد. اما اگر فردی واقعا درصدد شکستن رمز برآید، احتمالا موفق خواهد شد.

افزایش جذابیت ظاهری برنامه

اگر از بسته‌های نرم‌افزاری گوناگون استفاده کرده باشید، بدون تردید نمونه‌هایی از رابط‌های کاربری با طراحی بسیار ضعیف، برنامه‌های دشوار از لحاظ کارکردن، و صفحات یکنواخت و زشت را دیده‌اید. اگر صفحه گسترده را برای دیگران توسعه می‌دهید، باید توجه ویژه‌ای به ظاهر برنامه معطوف دارید.

چگونه به نظر رسیدن یک برنامه کامپیوتری، می‌تواند سلیقه‌ای باشد و از دید برخی در یک کشور، زیبا و از دید برخی در کشورهای دیگر، زشت و بدقیافه باشد و این درباره برنامه‌هایی که شما با اکسل توسعه می‌دهید هم صدق می‌کند. زیرا زیبایی، یک معیار نسبی و رفتار چشمی است. اگر مهارت‌های شما بیشتر در جهت تحلیلی باشد، از کمک فردی با دانش زیبایی‌شناسی و طراحی بهره جویید.

کاربران نهایی، از یک رابط کاربری خوش‌منظر قدردانی می‌کنند و اگر زمان بیشتری به طراحی و معیارهای زیبایی‌شناسی تخصیص دهید، برنامه شما ظاهری حرفه‌ای و سیقل داده شده به خود می‌گیرد. برنامه‌ای که ظاهر خوبی داشته باشد، این را القا می‌کند که توسعه‌دهنده‌اش به خوبی مراقب محصول خود بوده و زمان و تلاش کافی صرف کارش کرده است. پیشنهاد‌های زیر را به‌خاطر بسپارید:

کوشش برای ثبات: هنگام طراحی مثلا پنجره‌ها، تلاش کنید تا حد امکان، ظاهری همانند اکسل داشته باشند. فرمت‌بندی، فونت و اندازه متن و رنگ‌ها هم همین‌طور.

ساده بودن: اشتباه مرسوم می‌کند این است که تلاش می‌کنند اطلاعات بسیار زیادی را در یک صفحه یا پنجره واحد بگنجانند. یک قانون خوب هست که می‌گوید تنها یک یا دو تکه اطلاعات در یک لحظه ارائه شود.

شکستن صفحات ورودی: اگر از یک صفحه ورودی برای درخواست اطلاعات از کاربر استفاده می‌کنید، بهتر است آنرا به چند صفحه تقسیم کنید تا از شلوغی صفحه جلوگیری شود. چنانچه از یک تک پنجره استفاده می‌کنید، می‌توانید به کمک کنترل MultiPage، آن را بشکنید و یک پنجره با چند برگه یا تب ایجاد کنید.

رنگ‌های کمتر: هر چه از رنگ‌های کمتری استفاده کنید، شکل‌تر دیده می‌شود.

تصاویر و حروف بصری: به فرمت‌های عددی توجه داشته باشید و از طرح حروف، اندازه فونت و بردارهای یکسانی در همه جا استفاده کنید.

ارزیابی کیفیت‌های زیبایی‌شناسی، امری ذهنی است. اگر دچار تردید شدید، برای ساده‌سازی و شفاف بودن کار، بکوشید.

ایجاد یک سامانه Help

با توجه به مستندسازی کاربر، بهترین تمرین این است که مستندات کاغذ-محور یا الکترونیکی را به کاربران ارائه دهید (یا هر دو). ارائه سامانه کمکی الکترونیکی، یک قانون استاندارد در برنامه‌های ویندوزی است. خوشبختانه برنامه‌های اکسل نیز می‌توانند سامانه help ارائه دهند- حتی کمک حساس به محتوا. توسعه متن کمکی، نیاز به تلاش اضافه‌ای دارد، اما برای یک پروژه بزرگ، می‌تواند با ارزش باشد.

نکته دیگری که باید توجه داشت، پشتیبانی از برنامه خود است. به بیانی، اگر کاربر با مشکل مواجه شود، چه کسی تلفن را پاسخ می‌دهد؟ اگر آمادگی رسیدگی به پرسش‌های روتین را ندارید، نیاز به فردی دارید که این کار را برایتان انجام دهد. گاهی می‌خواهید آنرا تفکیک کنید تا تنها مسائل و مشکلات فنی مهم یا باگ‌های احتمالی را خودتان پاسخ دهید.

یادآوری: در فصل ۱۹ به سامانه کمکی Help می‌پردازیم.

مستندسازی تلاش توسعه دهنده

قرارگرفتن شما و کاربران یک برنامه صفحه‌گسترده در کنار هم، یک چیز است، قابل فهم کردن برنامه برای آنها، چیزی دیگر است. همانند برنامه‌های تجاری، مهم است که کارتان را به‌طور کامل مستندسازی کنید. چنین مستندسازی، هنگام بازگشت به عقب برای بررسی یک موضوع، به شما و نیز به فرد دیگری که برنامه را به او داده باشید کمک می‌کند.

چگونه یک برنامه کارپوشه را مستندسازی می‌کنید؟ می‌توانید یا اطلاعات را در یک کاربرگ ذخیره کنید یا از فایل دیگری استفاده کنید. حتی اگر بخواهید می‌توانید از یک سند کاغذی استفاده کنید. شاید ساده‌ترین راه، استفاده از یک کاربرگ مجزا برای ذخیره کامنت‌ها و اطلاعات کلیدی پروژه باشد. برای کدهای VBA، آزادانه از کامنت‌ها استفاده کنید. (اگر متن VBA با یک آپوستروف آغاز شود، به‌عنوان کد تشخیص داده نمی‌شود و برنامه به چشم کامنت به آن می‌نگرد.) هرچند یک قطعه کد VBA زیبا ممکن است امروز برایتان کاملاً واضح و معنادار باشد، اما چند ماه بعد پس از مراجعه

به آن، شاید کاملاً آنرا فراموش کرده باشید و منظورش را متوجه نشوید، مگر اینکه از ویژگی کامنت‌گذاری VBA استفاده کرده باشید.

توزیع برنامه برای کاربر

پروژه را تکمیل کرده‌اید و آماده انتشار آن برای کاربران نهایی هستید. چگونه می‌خواهید آنرا توزیع کنید؟ به چند روش می‌توان برنامه را توزیع کرد و روش انتخابی، بستگی به چند عامل دارد.

می‌توانید تنها یک لینک دانلود یا درایو موقتی به آنها بدهید، چند شیوه‌نامه تحویلشان دهید و به راه خود ادامه دهید. یا شاید بخواهید برنامه را خودتان نصب کنید- اما این دیدگاه همیشه شدنی نیست. گزینه دیگر، توسعه یک برنامه setup (تنظیم و نصب) رسمی است که این کار را به شکل خودکار انجام دهد. چنین برنامه‌ای را می‌توان با یک زبان برنامه‌نویسی تجاری، خرید یک برنامه setup عمومی، یا نوشتن setup شخصی خود با VBA انجام داد.

اکسل یک تکنولوژی به توسعه دهندگان ارائه می‌دهد که بتوانند برنامه‌های خود را به شکل دیجیتالی امضا کنند. این فرایند به کاربران نهایی در شناسایی نویسنده یک برنامه کمک می‌کند تا مطمئن شوند این پروژه، تغییر نکرده است، و برای کمک به جلوگیری از گسترش ویروس‌های ماکروبی یا دیگر کدهای بالقوه مخرب است. برای امضای دیجیتالی یک پروژه، نخست باید درخواست یک گواهی‌نامه دیجیتال از یک مرجع رسمی گواهی‌نامه کنید (یا می‌توانید پروژه خود را با ایجاد گواهی‌نامه دیجیتال شخصی خود، خویش-امضایی کنید). اطلاعات بیشتر در سایت میکروسافت آمده است.

به‌روز رسانی برنامه هنگام نیاز

پس از توزیع برنامه، آیا فکر می‌کنید کارتان به پایان رسیده است؟ می‌توانید پشت آن بنشینید و از کارتان لذت ببرید، و سعی کنید به مشکلاتی که در حین توسعه با آن مواجه شده بودید (و حل شد) فکر کنید. در موارد نادری شاید کارتان به پایان رسد. هرچند بیشتر مواقع کاربران برنامه به شکل کامل راضی نخواهند بود. یقیناً، برنامه شما به همه ویژگی‌های اصلی پایبند است، اما چیزهایی تغییر می‌کند. مشاهده کار کردن یک برنامه، اغلب سبب می‌شود کاربر به موارد دیگری فکر کند که برنامه نمی‌تواند آنها را انجام دهد.

هنگامی که نیاز به به‌روز رسانی یا تجدیدنظر در برنامه باشد، از خود خرسند خواهید شد که در آغاز، زیربنای آنرا به گونه‌ای مناسب طراحی کرده‌اید و کارهایتان را مستندسازی کرده‌اید و در ادامه، به مشکل برنخواهید خورد.