

Excel VBA voor Professionals Derde Editie

Wim de Groot, 2020

Ieder hoofdstuk begint met een inleiding, hier vindt u alle inleidingen bij elkaar.

Inleiding

We beginnen met enkele algemene opmerkingen, zodat u prettig met dit boek werkt. U krijgt een globaal overzicht van de inhoud. Ik deel een aantal afspraken met u over de opmaak en het gebruik van lettertypes, zowel voor de VBA-code als voor kaders. Ook is het belangrijk om te weten hoe coderegels worden afgebroken, want dat moet in dit boek hier en daar, vanwege de breedte van de pagina's. Voor de liefhebbers van sneltoetsen vertel ik hoe deze in de tekst te vinden zijn. En u leest waar u de gratis voorbeeldbestanden kunt ophalen.

1. Macro's opnemen en uitvoeren

Met een macro kunt u vaak voorkomende handelingen automatiseren. De eenvoudigste manier om het gemak van VBA te leren kennen, is door een macro op te nemen. Tijdens de opname van de macro voert u die handelingen één keer uit. Daarna kunt u die macro zo vaak uitvoeren als u maar wilt en worden die handelingen supersnel voor u gedaan. U drukt daarvoor op een zelfgekozen sneltoets. Die sneltoets moet u onthouden, maar u kunt de macro ook met een knop bedienen; dat is prettig als iemand anders met uw macro gaat werken. U plaatst de knop in het werkblad. Hiervoor voegt u een vorm in, die u van een opschrift voorziet. U kunt uw macro's ook bedienen met keuzerondjes om zo een adreslijst op verschillende manieren te sorteren. Werkt uw macro nog niet meteen, dan moet u mogelijk de beveiliging bijstellen.

2. Werken met de Visual Basic Editor

Als u de opgenomen macro wilt bijstellen, kunt u die helemaal opnieuw opnemen, maar u kunt sneller een kleine aanpassing in de code zelf aanbrengen. We nemen een kijkje achter de schermen en maken kennis met de Visual Basic Editor, de omgeving waarin we de code kunnen bewerken en waar we kunnen programmeren. De Visual Basic Editor wordt ook wel de IDE genoemd: de integrated development environment, de 'geïntegreerde ontwikkelomgeving'. U ziet uit welke onderdelen deze omgeving bestaat: herkenbare Windows-elementen als vensters en werkbalken, al is het lint van Office hierin niet doorgevoerd. Als u de code opent, ziet u hoe Visual Basic de opdrachten die u tijdens de opname van de macro gaf, heeft omgezet in code en uit welke onderdelen die code is opgebouwd. U kunt de code bijstellen die de opname in het eerste hoofdstuk hebt geproduceerd. Daarna is de volgende stap: zelf code gaan schrijven.

3. Code schrijven

U kunt inmiddels de code van een bestaande macro wijzigen en uitbreiden. Weet u de code van een opdracht niet, dan neemt u die apart op. Om verder te profiteren van de ontelbare mogelijkheden van VBA gaat u zelf uw procedures schrijven. De eerste stappen leert u aan de hand van het berichtvenster. Bij het opstellen van code schiet de Visual Basic Editor u van alle kanten te hulp. De AutoLijstweergave neemt u veel tikwerk uit handen, de Syntaxisinfo toont in welke volgorde u een instructie moet typen en er is een zeer uitvoerige Help-functie beschikbaar. U hoeft niet alles zelf te bedenken: op internet zijn veel voorbeelden te vinden. U leest hoe u code van het web haalt en in uw eigen code opneemt. Zoals voor elke taal gelden

er ook voor de programmeertaal VBA een aantal regels. Ook die komt u in dit hoofdstuk tegen.

4. Werken met objecten van Excel

VBA is een zogeheten objectgeoriënteerde programmeertaal, het werkt met objecten. Welke objecten heeft Excel, wat kunt u ermee en hoe werkt dat? Hoe laat u een procedure heen en weer gaan door het werkblad? Hoe maakt u selecties en hoe bewerkt u die? Daarbij speelt het verschil tussen absoluut en relatief een rol. U stuurt objecten aan met methoden, eigenschappen en argumenten. Zoals elke taal heeft de programmeertaal VBA regels voor de zinsopbouw: de grammatica of syntaxis. Het is essentieel om daarmee te kunnen werken. Een groep van objecten vormt een verzameling. Daardoor kunt u al die objecten tegelijk bewerken. Als u meer eigenschappen op hetzelfde object wilt toepassen, maakt u met het koppel With en End With uw code korter en eenvoudiger te lezen. Verschillende methoden hebben meer informatie nodig, zoals bij kopiëren: waar moet de kopie naartoe? Of bij afdrukken: hoeveel exemplaren wilt u? Die nadere informatie heet een argument. U kunt argumenten met naam en toenaam noemen. Maar u kunt ook volstaan met de waarde ervan, als u precies weet welke plaats zo'n argument in de reeks argumenten heeft.

5. Variabelen en constanten gebruiken

In een verhuisdoos kunt u boeken vervoeren. Er passen ook schoenen in, of keukenservies. Wat u er ook in doet, de doos blijft even groot. Met een doos vergeleken, is een variabele een stukje geheugenruimte waar u allerlei gegevens in kunt plaatsen. Variabelen worden in VBA veel gebruikt. Ze vormen een van de belangrijkste bouwstenen en het is erg handig om ermee te werken. U kunt er tekst in opslaan, een getal, een datum enzovoort. Bevat een variabele eenmaal een waarde, dan kunt u in de rest van de code die variabele gebruiken. Daarbij wil VBA ook weten welk type gegevens u in de variabele gaat bewaren. Lucifer bewaart u in een luciferdoosje, wijn gaat in een fles. Wijn doet u niet in een luciferdoosje, lucifers niet in een wijnfles. Afhankelijk van het type artikel dat u wilt bewaren, kiest u de verpakking. Als u van tevoren opgeeft welk type gegevens u in de variabele gaat bewaren, werkt de procedure efficiënter. Andere belangrijke bouwstenen zijn constanten; hun waarde ligt vast en is niet variabel.

6. Code verkorten en versnellen

In dit hoofdstuk vindt u tips om sneller te werken met VBA. Aan de ene kant kunt u tijd besparen als u codes sneller kunt opstellen. Codewoorden uit een menu aanklikken gaat sneller dan ze zelf typen (en foutloos). Codes die u kreeg door een macro op te nemen, kunt u vaak korter maken. Als u weet hoe dat gaat, kunt u ze zelf ook meteen kort typen. Als u eenmaal een macro hebt en u wilt die door een andere procedure laten gebruiken, doet u aan hergebruik. Aan de andere kant kunt u de uitvoering van de procedure (iets) sneller laten verlopen. In de eerste plaats door overbodige opdrachten weg te halen, die de zaak vertragen. Verder door eigenschappen die u tijdens een opname niet hebt gekozen, te verwijderen. Of door eigenschappen die bij hetzelfde object horen, in een blok With...End With te plaatsen. Ook zaken als schermverversing en automatisch het werkblad berekenen vragen onnodig tijd. U leest hoe u duizenden keren een dobbelsteen gooit; daarbij kom ik nog even terug op het werken met variabelen.

7. Beslissingen nemen

VBA denkt niet zelf na. Hoeveel werk een macro u ook uit handen neemt, deze doet alleen wat u hem opdraagt. Niet meer en niet minder. De 'beslissingen' die een code neemt, moet u vooraf programmeren en daarvoor moet u zelf vooraf het denkwerk verrichten. Mensen vertrouwen vaak blindelings op de computer en een macro lijkt slim. Maar die doet alleen wat u zelf instelt. U kunt aan een procedure duidelijk maken wat deze moet doen in een bepaald geval. U geeft een voorwaarde op en als aan die voorwaarde wordt voldaan, wordt een opdracht uitgevoerd. Daarvoor is er het koppel If...Then. Als er meer mogelijkheden zijn, komen Else en ElseIf erbij. Hebt u een hele serie gevallen waaruit de code moet kiezen, dan kan Select Case ze afhandelen. U kunt ook een venster laten verschijnen met een vraag en daar een If...Then-blok aan koppelen. Afhankelijk van het antwoord wordt dan het ene of het andere deel van de procedure uitgevoerd. Een constructie met If...Then wordt een structuur genoemd; in het volgende hoofdstuk vindt u meer structuren. In dit hoofdstuk doet u de inzichten hiervoor op. Eerst zetten we de verschillende operatoren op een rij.

8. Handelingen herhalen

In het vorige hoofdstuk maakt u kennis met If...Then. Een ander sterk koppel is For...Next. Dat vormt een lus, waarmee u een deel van de code herhaalt, zo vaak u wilt. U kunt er bijvoorbeeld een reeks cellen mee vullen en een inhoudsopgave maken van alle werkbladnamen in een werkmap. Ik laat ook zien hoe u een jaaroverzicht met verkoopgegevens die in een werkblad staan, kunt verdelen over werkbladen met maanden. Met de variant For Each...Next kunt u zaken herhalen als u het aantal objecten niet weet. Alle tabbladen een andere naam geven gaat ook snel met deze lus. Mocht er iets fout gaan, dan is het statement On Error Resume Next erg handig. Het duo Do en Loop weet van geen ophouden. Maar met de lus Do While...Loop bouwt u een voorwaarde in, waardoor de herhaling stopt. En met Do While...Until brengt u ook een grens aan, deze werkt andersom. Met deze lus kunt u willekeurige getallen onder elkaar in het werkblad plaatsen, bijvoorbeeld voor de lotto. De opdracht GoTo maakt een sprong en gaat in de procedure terug naar waar u wilt. Bijvoorbeeld omdat een invoervenster niet correct is ingevuld of om in een lijst met datums naar de cel met vandaag te gaan.

9. Algemeen bruikbare macro's maken

De meeste macro's doen iets wat alleen in een bepaald werkblad moet gebeuren. In de meeste gevallen is het dan ook verstandig de macro op te slaan in de werkmap waarin deze moet functioneren, want dezelfde macro kan in een ander werkblad ongewenste zaken doen. Er zijn echter macro's die handig zijn om in alle werkmappen te gebruiken. Bijvoorbeeld een macro die het hele pad van het bestand laat zien, een macro die alle hyperlinks uit een werkblad verwijdert, of die de vrij kleine letters van alle opmerkingen in het werkblad vergroot. Werkt u met geblokkeerde cellen, dan is aan het werkblad niet te zien welke cellen geblokkeerd zijn en welke niet. Met twee macro's maakt u dat snel zichtbaar. Om uw algemene macro's in ieder werkblad te kunnen oproepen, plaatst u daarvoor een extra knop in de werkbalk Snelle toegang. Of u voegt ze toe aan een van de tabs in het lint. Of voeg uw compleet eigen tabblad aan het lint toe. Daar kunt u ook meteen vaak gebruikte knoppen op plaatsen. U leest hoe u een bestaande macro vanuit één werkmap algemeen bruikbaar maakt, en omgekeerd hoe u een algemene macro alleen in een bepaalde werkmap laat werken.

10. Reageren op gebeurtenissen

In hoofdstuk 7 leest u over procedures met If...Then; als een bepaalde voorwaarde opgaat, doen ze iets. Dit hoofdstuk gaat over procedures die automatisch worden uitgevoerd zodra er iets gebeurt. Deze zelfstartende macro's worden aangestuurd door een gebeurtenis, een 'event'; vandaar de naam eventmacro. Zo'n gebeurtenis is bijvoorbeeld dat gegevens worden ingevoerd, dat een werkblad wordt geactiveerd of dat een werkmap wordt geopend. Zodra dat gebeurt, wordt een procedure in werking gezet. Een eventmacro kan ook reageren als u op een cel klikt. De hele rij krijgt dan een achtergrondkleur, of de hele kolom of allebei. U kunt instellen dat u bij het openen van een werkblad altijd in dezelfde cel wilt beginnen of ervoor zorgen dat u bij het vullen van cellen binnen een bepaalde tabel blijft. U kunt in een werkmap automatisch een vast werkblad tonen, draaitabellen automatisch laten bijwerken, zorgen dat een maandkeuzelijst altijd de huidige maand toont enzovoort. Wilt u een werkmap beveiligen, dan moet u alle werkbladen een voor een beveiligen. Een eventmacro knapt dit automatisch voor u op vlak voor het sluiten. Kunt u met macro's al veel automatiseren, nu gaan die zelfs automatisch van start.

11. Openen, opslaan en sluiten

Dit hoofdstuk gaat over het beheren van uw werkmappen. U kunt werkmappen door middel van code opslaan en afsluiten. Er is verschil tussen opslaan en afsluiten en dat onderscheid houden we hier ook aan. Wilt u een nieuw bestand opslaan? Of een bestaande werkmap? Wilt u er meteen ook een naam aan geven? Het kan met VBA. Ook voor de manier waarop u een werkmap wilt afsluiten hebt u de beschikking over een groot aantal opties. U kunt bijvoorbeeld automatisch een bestand opslaan met als bestandsnaam de naam van de gebruiker en de datum van vandaag. Ook de map waarin het bestand terecht komt, kunt u instellen. U kunt een werkmap opslaan als pdf-bestand. Dat is van belang voor bestanden die niet gewijzigd mogen worden, zoals facturen. Over facturen gesproken: dit hoofdstuk eindigt met een model voor uw facturen. Na het invullen van de gegevens krijgt de factuur automatisch een volgend factuurnummer en wordt deze opgeslagen als pdf, met het factuurnummer in de bestandsnaam.

12. Berichtvensters maken

Vanaf dit hoofdstuk brengt u meer interactie in uw werkbladen, want de komende hoofdstukken gaan over berichten en vensters. U laat VBA meer communiceren met de gebruiker. Helder communiceren is erg belangrijk, vooral als andere mensen uw programma's gaan gebruiken. Een venster dat een boodschap in beeld brengt, is een message box. Dat komt in eerdere hoofdstuk al even voorbij, hier leert u onder meer hoe u daar diverse knoppen en pictogrammen op plaatst. Afhankelijk van de knop die men aanklikt moet er iets gebeuren: cellen wel of niet wissen bijvoorbeeld. Gewoonlijk wordt de boodschap in het berichtvenster weergegeven als één lange zin. Die kunt u ook verdelen over meer regels. Als u een tabel hebt waarin u per dag uw activiteiten bijhoudt, kan Excel de gewerkte tijd per activiteit optellen. U gaat deze tijden wegschrijven in een tabel voor het hele jaar, en daarbij controleren of de medewerker zijn/haar naam heeft ingevuld, of er een datum is ingevuld, of het aantal tijdstippen klopt met het aantal activiteiten en of er niet eerder uren bij dezelfde datum zijn genoteerd. Bij ontbrekende gegevens verschijnt er een berichtvenster. Een bijzonder soort berichtvenster is de statusbalk, onder in beeld. Daarin kunt u de voortgang van een proces weergeven in procenten, als dat iets langer duurt dan de fractie van een seconde die de meest macro's erover doen. Of u laat in de statusbalk een mededeling weergeven terwijl de procedure wordt uitgevoerd.

13. Invoervensters maken

Als u de gebruiker vraagt gegevens in te voeren, gebruikt u het invoervenster, de InputBox. Met de ingevulde waarde kan de macro verder werken. U kunt bijvoorbeeld een getal invullen, waarna u meteen het kwadraat krijgt. Dat getal kunt u afronden. Als u via dat venster een datum invoert of procenten, moet u op enkele zaken letten. Wilt u alle werkbladen beveiligen met een wachtwoord, dan moet u dat voor ieder werkblad apart doen en dat steeds twee keer. U vindt hier een code om alle werkbladen tegelijk te beveiligen met een wachtwoord dat u in een invoervenster opgeeft. U kunt in het invoervenster ook een celbereik opgeven om met dat gebied iets te doen. Voor het declareren van reiskosten kunt u een keten van vensters opzetten. U kunt niet altijd voorspellen wat gebruikers doen als u vraagt om gegevens in te voeren. Maar het is goed om op mogelijke fouten te anticiperen. Met If...Then-constructies kunt u verwachte fouten afvangen door te controleren of men een bedrag invoert als dat moet. Ook leest u hoe u het invoervenster laat reageren als de gebruiker op Annuleren klikt of op de Esc-toets drukt, of niets invult en op OK klikt. Zo maakt u uw procedure 'monkey proof'. Ten slotte geef ik een procedure waarin de gebruiker een pincode moet typen om de beveiliging van het werkblad te halen. Na drie onjuiste pogingen wordt het bestand gesloten.

14. Dialoogvensters opbouwen

Het berichtvenster en het invoervenster uit het vorige hoofdstuk zijn eenvoudige vormen van een dialoogvenster. Daarmee communiceert de gebruiker met de computer. Een geavanceerder soort dialoogvenster is het UserForm. Dat kunt u naar wens voorzien van knoppen, invoervakken, pijlknoppen, keuzelijsten enzovoort. In dit en in het volgende hoofdstuk leest u hoe u eigen dialoogvensters opbouwt. U leert welke codes u nodig hebt om het venster werkend te maken. Het eerste dialoogvenster is er een met tekstvakken. De gegevens die u invult komen automatisch op de goede plaats in het werkblad terecht. Ook kunt u in een dialoogvenster keuzelijsten inbouwen. U leest hoe u de opties in het menu zet, hoe u zorgt dat de gemaakte keuze verwerkt wordt en hoe u ruimte maakt om daarin ook zelf nog iets in te vullen. Moeten enkele vakken of keuzelijsten verplicht worden ingevuld, dan kunt u daarvoor een controle inbouwen. Wilt u gebruikers in het dialoogvenster laten kiezen tussen meer opties, dan bouwt u daarvoor keuzerondjes in. Als voorbeeld neem ik het omzetten van bedragen naar btw: moet de btw eraf of erbij? Moet dat tegen een hoog of een laag tarief? Een ander voorbeeld is het aanvullen van een gegevenslijst: de naam, de aanduiding Man, Vrouw of Onbekend en de geboortedatum komen automatisch op de volgende lege rij onder aan de lijst. U leest eerst aanwijzingen die voor alle dialoogvensters van belang zijn.

15. Meer elementen in dialoogvensters plaatsen

In het vorige hoofdstuk leest u hoe u een dialoogvenster maakt met tekstvakken, keuzelijsten en optierondjes. In dit hoofdstuk komen aan de orde: selectievakje, keuzelijst, wisselknop, schuifbalk, kringveld, tabbladen, celverwijzing en datum- en tijdkiezer. U vindt van al deze elementen een voorbeeld en u leest hoe u ze in een dialoogvenster inbouwt en gebruikt. Aan het eind leest u hoe u een dialoogvenster gebruiksvriendelijker maakt voor mensen die graag met toetsen werken. U leest hoe u bij invoervakken een tip laat verschijnen, het dialoogvenster in uw eigen kleur steekt of van een afbeelding voorziet.

16. Werken met functies voor getallen

U kunt met VBA rekenen. Hiervoor heeft het programma een aantal rekenfuncties. U kunt de uitkomsten op allerlei manieren opmaken. Het aantal rekenfuncties van VBA is beperkt, maar voor ingewikkelde berekeningen kunt u altijd terugvallen op de functies van Excel. Die kunt u binnen de VBA-code plaatsen. Een Excel-functie moet u in het Engels invoeren. De vertaling van de besproken Excel-functies vindt u in appendix A en u kunt een volledige lijst met vertalingen gratis downloaden. Gebruikt u Excel-functies in VBA, dan loopt u tegen een paar ongerijmdheden aan. Als VBA de rekenfunctie zelf heeft, is die van Excel niet beschikbaar. Er zijn functies die in VBA wel dezelfde naam hebben als de Excel-functie en toch iets anders doen. Enkele functies doen hetzelfde als in Excel, maar hebben in VBA een andere naam. Dat kan verwarrend zijn. Dit hoofdstuk geeft helderheid over de rekenfuncties.

17. Werken met andere functies

In het vorige hoofdstuk maakt u kennis met rekenfuncties van VBA. In dit hoofdstuk komen functies voor tekst en informatie aan bod. U kunt tekenreeksen (strings) bewerken. U wilt een getal als een tekst zien, of omgekeerd. Met de functie Replace vervangt u letters in een tekst. Als er bijvoorbeeld een reeks namen in een kolom staat met de voor- en achternaam in dezelfde cel, en u wilt de lijst sorteren op de achternaam, dan moet u de achternaam in een eigen kolom zetten. Daarvoor gebruiken we enkele tekstfuncties. Als u die in een macro gebruikt, plaatst u achternamen en plaatsnamen in aparte kolommen. Met een andere groep functies converteert u een waarde van het ene gegevenstype naar een ander type. Er zijn functies die informatie geven over het bestand en functies die controleren of een variabele van een bepaald gegevenstype is. Daarmee kunt u bijvoorbeeld controleren wat in een invoervenster is ingevuld.

18. Eigen functies maken

Excel heeft ongeveer 480 functies in huis. Mocht u daaraan niet genoeg hebben, dan kunt u zelf functies maken. Die werken ook met VBA-code. Als u werkt met de maten van ronde vormen, zoals een cirkel of een cilinder, en u wilt daarvan de oppervlakte of de inhoud berekenen, hebt u dan de formules daarvoor paraat? Het is iets met pi en de straal. Als u hiervoor zelf een functie maakt, hoeft u alleen de straal maar op te geven en wordt de rest automatisch berekend. Ook de oppervlakte en de inhoud van een bol komen aan bod. Ik bespreek functies die één getal verwerken, exemplaren die er twee verwerken, functies die met If...Then werken, tot de BMI die met Select Case uit een rijtje afleest hoe het staat met uw gezondheid. Deze functies voegt u zelf toe aan Excel. Die kunt u invoeren via Functie invoegen en de Help-tekst die u daarbij wilt weergeven stelt u zelf op. Wilt u tellen hoeveel cellen in een bepaald gebied rood zijn, getallen in gele cellen optellen of cellen die een formule bevatten automatisch laten verkleuren, dan schrijft u daarvoor ook uw eigen functie. U kunt zelfs de belasting over uw inkomen berekenen met één formule. Hebt u een namenlijst met de voor- en achternaam in dezelfde cel, dan kunt u die lijst niet meteen sorteren op achternamen. Dat kan wel als u het laatste deel van de naam eruit haalt met een eigen functie. Ten slotte laat ik u zien hoe u uw eigen functies zo opslaat, dat u er in al uw werkmappen van profiteert, namelijk door er een invoegtoepassing, een add-in van te maken.

19. Programmeren met datums

In VBA kunt u werken met datums. Hoe u die noteert, vraagt enige aandacht, want VBA volgt het Amerikaanse systeem. Er zijn ingebouwde functies voor een datum. Wilt u een weeknummer opzoeken of op grond van iemands geboortedatum diens leeftijd berekenen, dan

kunt u daarvoor een datumfunctie gebruiken. Maar die werken niet perfect, die gaan we verbeteren. U kunt een datum op allerlei manieren weergeven. Laat u ze verschijnen in een berichtvenster, dan plaatst u daar een zin omheen. Met een zelfgemaakte functie kunt u van een jaartal opvragen of dat een schrikkeljaar is. U kunt ook functies maken voor de feestdagen. Geef een jaartal op en u ziet meteen op welke datum Pasen valt; daar worden andere feestdagen van afgeleid. Vanaf ramadan kunnen we de datum van het Suikerfeest berekenen. Werkt u parttime en wilt u weten hoeveel dinsdagen een bepaalde maand heeft, dan maakt u daarvoor een functie. Met dezelfde formule kunt u ook twee verschillende weekdagen tellen. Wilt u berekeningen uitvoeren met datums voor 1900, dan kan dat in Excel niet, maar in VBA wel. U kunt berekeningen maken tussen de jaren 100 en 9999. En daarvoor kunt u in VBA zelf functies maken. Zo kunt u in Excel bijvoorbeeld vinden op welke dag van de week een belangrijke historische gebeurtenis plaatsvond. We houden ook rekening met de overgang naar de Gregoriaanse kalender in 1582. Voor stamboomonderzoek kunt u op grond van geboorte- en sterfdatum de leeftijd berekenen. En met een andere functie telt u vanaf een bepaalde datum een aantal jaren, maanden of dagen terug of vooruit.

20. Programmeren met tijd

Dit hoofdstuk gaat over de functies voor tijd. Werken met tijd in VBA maakt uw werkmap levendig. U kunt tijdstippen op allerlei manieren weergeven met de functie Format. Komt een tijdstip in een werkblad, dan moet u mogelijk de cel daarvoor instellen in Excel. U kunt een dialoogvenster maken met een keuzelijst erin, waarmee u uit een serie tijdstippen kunt kiezen. U kunt een procedure op één vast tijdstip op een dag laten uitvoeren, of met vaste tussenpozen laten herhalen. U kunt tijd inzetten om de macro automatisch te laten starten. Met een lus maakt u een digitale klok die u met knopjes aan en uit kunt zetten, of automatisch. U kunt een cel laten knippen als iemand een waarde invoert buiten waarden die zijn toegestaan. U kunt een werkmap automatisch laten sluiten als er een uur niet aan is gewerkt. Hiervoor bouwt u een timer in. Als u gegevens van een website ophaalt met een query, gaat dat op zich automatisch. Wilt u de veranderde gegevens vastleggen in een lijst, dan kan dat met een macro.

21. Grafieken maken met VBA

U kunt in Excel grafieken maken, zoals u weet. Dat kan ook met VBA en daarmee kunt u de grafieken ook verder instellen. In dit hoofdstuk leest u welke code u daarvoor gebruikt. U kunt grafiekelementen zoals de titel en de legenda toevoegen. De plaats waar de grafiek in het werkblad komt en de afmetingen kunt u instellen. Nadat we losse opdrachtregels hebben verkend, maken we complete grafieken: als voorbeeld dienen een kolomgrafiek en een lijngrafiek. U bepaalt de kleur van het grafiekgebied, van het tekengebied, van de kolommen en van de lijn. Met code stelt u de horizontale en de verticale as in, en voegt u rasterlijnen toe. U maakt een combinatiegrafiek van kolommen en een lijn. Houdt u een lijst bij die aangroeit, dan staat het fraai als de grafiek steeds over de volle breedte gevuld is, hoe lang de lijst ook is.

22. Word aansturen vanuit Excel

Word en Excel horen beide tot het pakket Office en in beide werkt VBA. Daardoor kan een macro vanuit Excel bewerkingen in Word uitvoeren. Als uw werkmap opmerkingen bevat, kunt u de tekst van alle opmerkingen van het actieve werkblad afdrukken. Daarvoor moet u ook het werkblad zelf afdrukken. Met een macro kunt u de opmerkingen afdrukken zonder het werkblad, ze worden eerst gekopieerd naar een Word-document. Een ander voorbeeld van samenwerking tussen Word en Excel laat ik zien aan de hand van een rapport van een business case. In een document in Word staat de standaard tekst van het rapport. In Excel stelt

u de cijfers van een bepaald bedrijf op. Vervolgens legt u vanuit dit Excel-bestand verbindingen met het Word-rapport en een macro vult de juiste velden in Word met de bedrijfsnaam, de bedragen en grafieken. Ten slotte laat ik zien hoe u vanuit Excel kunt nagaan of Word is gestart.

A. Vertaling Nederlands - Engels

In dit boek worden de Nederlandse aanduidingen gehanteerd van de tabs, de knoppen en de menu-opdrachten in VBA en in Excel. Werkt u met een Engelstalige versie van Excel, dan vindt u hier een lijst met de Engelse vertaling van de Nederlandse aanduidingen. Alle knoppen die in dit boek worden genoemd, vindt u hier vertaald terug.

Verder is het prettig om een aantal vaak voorkomende termen in het Engels te kennen. Ook die vindt u in de volgende woordenlijst naast elkaar.

U kunt vanuit de Engelstalige versie de hele interface (knoppen, tabs, vensters en dergelijke) in het Nederlands weergeven (en andersom). Hoe dat werkt, leest u aan het eind van deze Appendix.

De Engelse vertalingen van foutmeldingen staan in Appendix B.

B. Foutmeldingen

Het is niet de vraag of er een foutmelding verschijnt, maar wanneer deze verschijnt. In deze Appendix staan de meest voorkomende foutmeldingen bij elkaar. U leest de melding, wat de oorzaak is en hoe u de fout oplost. Sommige fouten worden al tijdens het programma opgemerkt door de Visual Basic Editor; zo'n fout is een zogeheten compileerfout (compile error). Andere fouten komen pas aan het licht tijdens het uitvoeren van de macro of procedure; die heten een Fout tijdens uitvoering (Run-time error) en hebben meestal een nummer.

C. Sneltoetsen in de Visual Basic Editor

Op verschillende plaatsen in het boek wordt verwezen naar sneltoetsen. Die vormen het alternatief voor de muis. Als u ze in de vingers hebt, werkt u sneller; bovendien voorkomt u dat u een muisarm oploopt. De tabel geeft de sneltoetsen die u in de Visual Basic Editor kunt gebruiken.

D. Codes voor opmaak

In de betreffende hoofdstukken staan de diverse mogelijkheden voor opmaak van getallen, datums en tijd. Hier vindt u de opmaakcodes overzichtelijk bij elkaar.

E. VBA-functies

In deze appendix vindt u de meest voorkomende functies van VBA. Aan de hand van de Index vindt u waar in het boek een bepaalde functie wordt uitgelegd.

F. VBA in plaats van Excel-functie

Excel heeft rekenfuncties, WorksheetFunctions genaamd. VBA heeft ook eigen rekenfuncties. Bestaat er voor uw berekening een ingebouwde VBA-functie, dan moet u die gebruiken. Bestaat die niet, dan gebruikt u de werkbladfunctie van Excel door middel van `WorksheetFunction`, waarna u de Engelse naam van de functie opgeeft. De volgende functies kunt u niet gebruiken als `WorksheetFunction`, omdat u de ingebouwde VBA-functie moet gebruiken.

G. Snel werken zonder macro's

Dit boek gaat over macro's, waardoor u sneller met Excel kunt werken. Er zijn ook zaken waarvoor u geen macro nodig hebt en waardoor u toch sneller werkt.