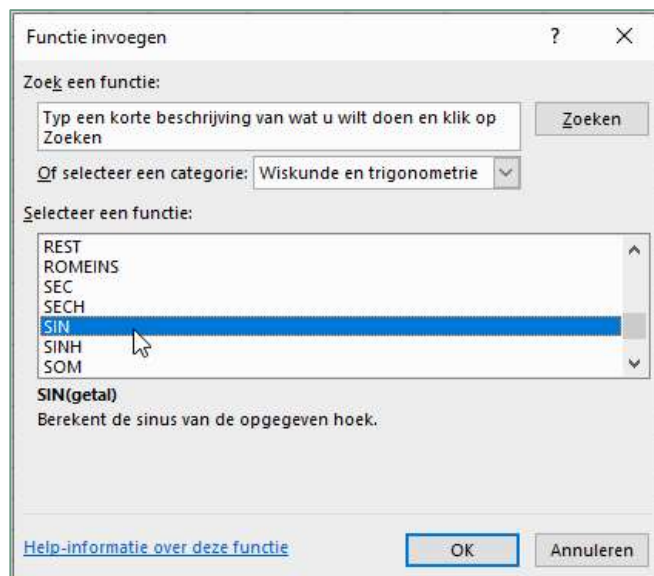
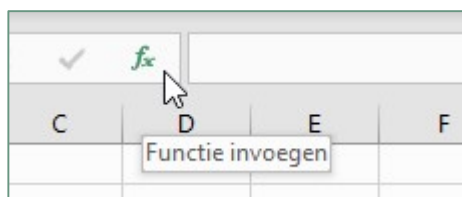


Maak je eigen functies in Excel met VBA

Het eerste voorbeeld van het gebruik van Visual Basic for Applications (VBA) in Excel is er een die heel eenvoudig is en heel snel hele grote stappen vooruit zet in de kwaliteit van de spreadsheets en tijdswinst bij het opstellen en controleren van spreadsheet (Excel). Het gaat om het schrijven van eigen functies in Excel, zie fragmenten Excel.

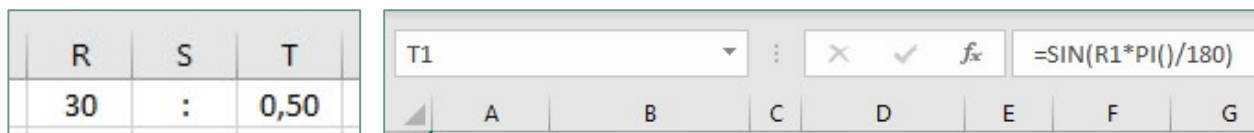


Functies in Excel

Excel kent heel veel functies waarvan enkele wel in elke spreadsheet gebruikt worden. Het gaat hier om allerlei wiskundige formules zoals: EXP, SOM, SIN, COS, MAX, MIN, GEMIDDELD, enz. Naast wiskundige formules zijn er voor allerlei andere categorieën functies beschikbaar, zoals als VERT.ZOEKEN, WEEKDAG, enz.

Wat minder bekend is dat het heel eenvoudig is om eigen functies te schrijven en te gebruiken. Deze eigen functies worden door Excel op exact dezelfde wijze gebruikt en verwerkt als de voorgaande standaard Excel-functies.

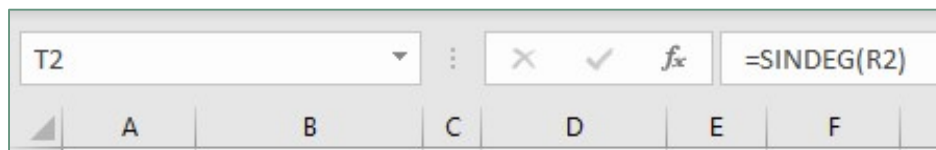
Neem als voorbeeld de Excel-functie "SIN()" in cel T1 met een hoek van 30 graden in cel R1.



De SIN() functie maakt standaard gebruik van een hoek in radialen. Voor een hoek in graden (DEG) is een kleine aanpassing nodig $R1 \times \pi/180$. Dit principe is voor iedere Excelgebruiker duidelijk. Een goed alternatief is een eigen geschreven functie SINDEG() als de SIN() heel vaak wordt gebruikt.



R	S	T
30	:	0,50
30	:	0,50



SINDEG() is in dit geval een eigen geschreven functie en werkt op basis van volgende VBA-code:

```
Public Function SINDEG(Hoek As Double) As Double
    SINDEG = Sin(Hoek * (Atn(1) * 4) / 180)
End Function
```

De Hoek wordt direct in graden (DEG) gegeven en de eigen functies maakt vervolgens gebruik van de Excel-functie SIN, maar nu met een automatische aanpassing van de Hoek in radialen naar graden: $\text{Hoek} \times \pi/180$. Let op VBA kent het getal π niet. Daarvoor wordt gebruik gemaakt van een VBA-functie voor arctangent ATN(), zo wordt π bepaald met $\text{ATN}(1) \times 4$.

Ogenschijnlijk lijkt dit niet zo'n grote stap vooruit, maar dit is een heel eenvoudig voorbeeld. De voordelen van eigen functies zijn echter heel groot. Enkele van de voordelen is hier al duidelijk zichtbaar:

- Het schrijven van eigen functies is super eenvoudig.
- Eigen functies werken naadloos met Excel, identiek aan standaard functies in Excel.
- De formule in T2 is veel eenvoudiger in gebruik dan de formule in T1
T2: = SINDEG(R2) versus T1: = SIN(R1 * PI() / 180)
Tevens is de functie SINDEG een perfecte aanvulling op de standaard bibliotheek van Excel
- De eigen functie hoeven we maar één keer goed te controleren en zijn verder altijd goed, in elk nieuw Excel document weer opnieuw. De functie is ook eenvoudig te controleren in VBA-code. De tekst is voor iedereen prima leesbaar en niet cryptisch door een leger aan celverwijzingen.

Deze eigen functies kunnen eenvoudig in een basis template van Excel worden opgenomen, zodat deze voor alle gebruikers (het hele bedrijf) direct beschikbaar zijn. Zo kan heel eenvoudig een bibliotheek met eigen functies worden opgesteld. Iedereen binnen het bedrijf maakt van deze functies gebruik en in de VBA-code kan de uitleg worden gegeven. Deze kan direct als handleiding worden gebruikt door het plaatsen van commentaren én het gebruik van duidelijke namen van variabelen. Alles is erg overzichtelijk en kan door iedereen worden gebruikt, gecontroleerd én worden aangevuld. Een bibliotheek van eigen functies kan in jaren groeien en verwerkt in een template is het onderhoud heel eenvoudig te realiseren en te handhaven

Voorbeelden van eigen functies

Een ander voorbeeld zal de kracht van eigen functies veel duidelijker maken. Neem de berekeningen van enkele geometrische profielgrootheden van een H-profiel. Stel dat een eenvoudig H-profiel onderdeel is van een omvangrijke berekening van een portaalconstructie en enkele gegevens moeten worden bepaald om de complexe portaalberekeningen uit te kunnen voeren.



	A	B	C	D
1	Invoer afmeting H-profiel			
2	Hoogte	=	100	
3	Breedte	=	100	
4	tf	=	6	
5	tl	=	4	
6				
7	Berekende profielgrootheden			
8	Omtrek	=	592	mm
9	Oppervlak	=	1552	mm ²
10	z	=	50	mm
11	W	=	57631	mm ³
12	I	=	2881557	mm ⁴

De functies voor de bepaling van de omtrek, oppervlak enz van een H-profiel zijn met Excel als volgt opgebouwd.

C8	=	$=(C3+C3+C4+C4-C5)*2+(C2-2*C4)*2$
C9	=	$=2*(C3*C4)+(C2-2*C4)*C5$
C10	=	$=(C3*C4*C4/2+C3*C4*(C2-C4/2)+(C2-2*C4)*C5*C2/2)/C9$
C11	=	$=C12/C10$
C12	=	$=(C3/12*C4^3+C3*C4*(C10-C4/2)^2)+(C3/12*C4^3+C3*C4*(C2-C4/2-C10)^2)+(C5/12*(C2-2*C4)^3+(C2-2*C4)*C5*(C2/2-C10)^2)$

voor de duidelijkheid:

Cel **Formule**

C8
$$=(C3+C3+C4+C4-C5)*2+(C2-2*C4)*2$$

C9
$$=2*(C3*C4)+(C2-2*C4)*C5$$

C10
$$=(C3*C4*C4/2+C3*C4*(C2-C4/2)+(C2-2*C4)*C5*C2/2)/C9$$

C11
$$=C12/C10$$

C12
$$=(C3/12*C4^3+C3*C4*(C10-C4/2)^2)+(C3/12*C4^3+C3*C4*(C2-C4/2-C10)^2)+(C5/12*(C2-2*C4)^3+(C2-2*C4)*C5*(C2/2-C10)^2)$$

Het mag duidelijk zijn dat deze formules alles behalve leesbaar zijn en fouten (onjuiste celverwijzingen) heel eenvoudig te maken zijn. Onjuiste celverwijzingen zijn doorgaans moeilijk te vinden in langere formules. Wat als we hier eigen functies gebruiken? Dan krijgen de formules in het werkblad een iets ander uiterlijk.

	A	B	C	D	E	F
1	Invoer afmeting H-profiel					
2	Hoogte	=	100			
3	Breedte	=	100			
4	tf	=	6			
5	tl	=	4			
6						
7	Berekende profielgrootheden				Eigen functie	
8	Omtrek	=	592	mm		592
9	Oppervlak	=	1552	mm ²		1552
10	z	=	50	mm		50
11	W	=	57631	mm ³		57631
12	I	=	2881557	mm ⁴		2881557

De inhoud van de cellen is nu gelijk aan:

F8	:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>fx</i>	=OmtrekH(C2;C3;C5;C4)
F9	:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>fx</i>	=OppervlakH(C2;C3;C5;C4)
F10	:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>fx</i>	=Zwaartelijnh(C2;C3;C5;C4)
F11	:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>fx</i>	=WeerstandsmomentH(C2;C3;C5;C4)
F12	:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>fx</i>	=TraagheidsmomentH(C2;C3;C5;C4)

De formules zijn een stuk eenvoudiger, simpel te onderhouden en simpel te controleren. Zie o.a. navolgende figuur, een controle op de standaard Excel manier met kleurtjes.

	A	B	C	D	E	F	G	H	
1	Invoer afmeting H-profiel								
2	Hoogte	=	100						
3	Breedte	=	100						
4	tf	=	6						
5	tl	=	4						
6									
7	Berekende profielgrootheden					Eigen functie			
8	Omtrek	=	592	mm		592			
9	Oppervlak	=	1552	mm ²		1552			
10	z	=	50	mm		50			
11	W	=	57631	mm ³		C3;C5;C4)			
12	I	=	2881557	mm ⁴		2881557			

De VBA-code van enkele eigen geschreven formules zijn:

```

'-----
Public Function OppervlakH(Hoogte As Double, Breedte As Double, _
                          TLijf As Double, TFlens As Double) As Double
'-----
'Oppervlak van een H-profiel
'
'      oppervlak van flens onder
OppervlakH = Breedte * TFlens
'      plus oppervlak van flens boven
OppervlakH = OppervlakH + Breedte * TFlens
'      plus oppervlak van lijf
OppervlakH = OppervlakH + (Hoogte - 2 * TFlens) * TLijf
End Function
  
```

De layout van de VBA-code is iets aangepast om deze duidelijk zichtbaar te maken in deze tekst. In onderstaande figuur is de werkelijk toegepaste layout aangegeven. Dit heeft overigens geen enkele invloed op het resultaat van de functie. De layout en het gebruik van commentaren (groen) wordt geheel door de programmeur bepaald.

Alle overige VBA-voorbeelden zijn “omgebouwd” om de tekst duidelijk weer te kunnen geven.

```

'-----
Public Function OppervlakH(Hoogte As Double, Breedte As Double, TLijf As Double, TFlens As Double) As Double
'-----
'Oppervlak van een H-profiel
OppervlakH = Breedte * TFlens                                'oppervlak van flens onder
OppervlakH = OppervlakH + Breedte * TFlens                  'plus oppervlak van flens boven
OppervlakH = OppervlakH + (Hoogte - 2 * TFlens) * TLijf    'plus oppervlak van lijf
End Function
  
```



```
'-----  
Public Function Zwaartelijnh(Hoogte As Double, Breedte As Double, _  
                             TLijf As Double, TFlens As Double) As Double  
'-----  
'Zwaartelijnh van een H-profiel met een flens als basis  
'Zwaartelijnh is de afstand van de basis van het H-profiel tot het  
'zwaartepunt van het profiel. Hierbij wordt gebruik gemaakt van een  
'andere functie voor de bepaling van het oppervlak van het H-profiel  
'Gebruikt functie: OppervlakH  
  
'           A x z = Statisch moment (S) flens onder  
Zwaartelijnh = Breedte * TFlens * TFlens / 2  
'           Plus statisch moment flens boven  
Zwaartelijnh = Zwaartelijnh + Breedte * TFlens * (Hoogte - TFlens / 2)  
'           Plus statisch moment lijf  
Zwaartelijnh = Zwaartelijnh + (Hoogte - 2 * TFlens) * TLijf * (Hoogte / 2)  
'           = S_totaal / A_totaal  
Zwaartelijnh = Zwaartelijnh / OppervlakH(Hoogte, Breedte, TLijf, TFlens)  
  
End Function
```

```
'-----  
Public Function TraagheidsmomentH(Hoogte As Double, Breedte As Double, _  
                                  TLijf As Double, TFlens As Double) As Double  
'-----  
'Bepaling (I) traagheidsmoment van een H-profiel met een flens als basis  
'Voor de bepaling van I van het H-profiel wordt gebruikt gemaakt van  
'de stelling van Steiner. Er wordt gebruik gemaakt van de functie: Zwaartelijnh  
  
Dim HulpZL As Double  
  
HulpZL = Zwaartelijnh(Hoogte, Breedte, TLijf, TFlens)  
  
'           I(flens onder) + Az^2  
TraagheidsmomentH = Breedte / 12 * (TFlens) ^ 3  
TraagheidsmomentH = TraagheidsmomentH + Breedte * TFlens * (HulpZL - TFlens / 2) ^ 2  
'           Plus I(flens boven) + Az^2  
TraagheidsmomentH = TraagheidsmomentH + Breedte / 12 * (TFlens) ^ 3  
TraagheidsmomentH = TraagheidsmomentH + Breedte * TFlens * (Hoogte - TFlens / 2 - HulpZL) ^ 2  
'           Plus I(lijf) (verplaatsing niet nodig, want excentriciteit=0  
TraagheidsmomentH = TraagheidsmomentH + TLijf / 12 * (Hoogte - 2 * TFlens) ^ 3  
  
End Function
```

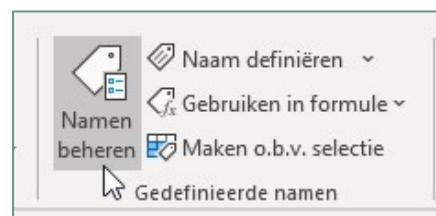
De eigen functies zijn prima te lezen en te controleren. De commentaren (groen) geven een perfecte mogelijkheid om de VBA-code van omvangrijke toelichtingen te voorzien. Naast toelichtingen kan ook informatie worden uitgewisseld in de vorm van beperkingen, bijvoorbeeld SINDEG heeft een hoek nodig in graden (DEG). Of de opmerking dat de functie nog niet helemaal gereed is. Of er ontbreken nog onderdelen aan de functie. Of de functie maakt gebruik van een andere (eigen) functie. Als een van die functies wijzigt (bijvoorbeeld door wijzigingen in de EC's), kan dit eenvoudig worden doorlopen, evenals de aansluitend gebruikte functies conform de verwijzingen in de commentaren.

Het gebruik van duidelijke omschrijvingen en variabelnamen maakt het zoveel eenvoudiger dan een cel verwijzing naar "C3" of "AB534".

Geef cellen namen

Een andere functionaliteit in Excel die zelden wordt toegepast is het gebruik van namen voor specifieke cellen of selecties van cellen. Hiermee kan een formule in Excel (ook eigen functies) veel duidelijker maken.

	A	B	C	D	E	F
1	Invoer afmeting H-profiel					
2	Hoogte	=	100			
3	Breedte	=	100			
4	tf	=	6			
5	tl	=	4			
6						
7	Berekende profielgrootheden				Eigen functie	
8	Omtrek	=	592	mm		592
9	Oppervlak	=	1552	mm ²		1552
10	z	=	50	mm		50
11	W	=	57631	mm ³		57631
12	I	=	2881557	mm ⁴		2881557



Selecteer de gewenste cel (C2) en geef deze cel een eenduidige en logische naam door deze links boven in de informatievenster (info geselecteerde cellen) te typen. Zo krijgt cel "C2" de naam "Hoogte" en kan in formules "C2" worden vervangen door "Hoogte". Dit geldt natuurlijk ook voor de cellen Breedte (C3), TFlens (C4) en TLIJf (C5). Beheer van de namen is mogelijk via de menuoptie in het lint van Excel onder "Formules".

De formules worden dan:

C8	:	\times	\checkmark	f_x	$=(\text{Breedte}+\text{Breedte}+\text{TFlens}+\text{TFlens}-\text{TLIJf}) * 2 + (\text{Hoogte}-2 * \text{TFlens}) * 2$
----	---	----------	--------------	-------	---

in plaats van

C8	:	\times	\checkmark	f_x	$=(\text{C3}+\text{C3}+\text{C4}+\text{C4}-\text{C5}) * 2 + (\text{C2}-2 * \text{C4}) * 2$
----	---	----------	--------------	-------	--

C9	:	\times	\checkmark	f_x	$=2 * (\text{Breedte} * \text{TFlens}) + (\text{Hoogte}-2 * \text{TFlens}) * \text{TLIJf}$
----	---	----------	--------------	-------	--

in plaats van

C9	:	\times	\checkmark	f_x	$=2 * (\text{C3} * \text{C4}) + (\text{C2}-2 * \text{C4}) * \text{C5}$
----	---	----------	--------------	-------	--

of (met celnamen i.p.v. celverwijzingen)

F8	:	\times	\checkmark	f_x	$=\text{OmtrekH}(\text{Hoogte};\text{Breedte};\text{TLIJf};\text{TFlens})$
----	---	----------	--------------	-------	--

in plaats van

F8	:	\times	\checkmark	f_x	$=\text{OmtrekH}(\text{C2};\text{C3};\text{C5};\text{C4})$
----	---	----------	--------------	-------	--

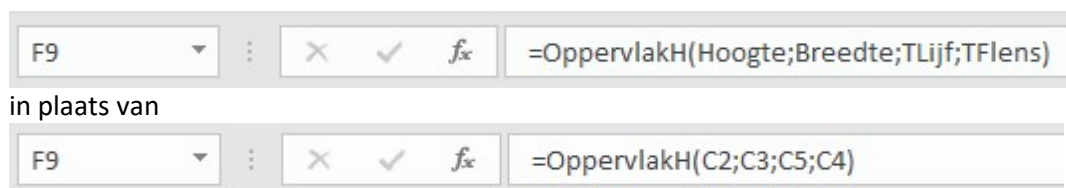


Rubriek Constructeurs programmeren

Item Maak eigen functies in Excel met VBA

Datum 01-02-2020

Pagina 8 | 8



Echter voor lange formules zoals voor de bepaling van het traagheidsmoment (I) van een H-profiel (C12) zijn namen niet voldoende om formules duidelijker te maken. De lange formules zijn nog steeds niet handig. Dan geeft een eigen functie altijd de beste randvoorwaarden voor een snelle, duidelijke, controleerbare en kwalitatief perfecte berekeningen.

Probeer het eens zelf en maak gebruik van de gegeven voorbeelden en de voorbeeld documenten op de website www.VullingsAdvies.nl (De constructeur).

Er zullen regelmatig andere voorbeelden worden gegeven voor het gebruik van VBA in Microsoft applicaties. Om deze zelf te kunnen schrijven gebruik de toetsen "Alt"+"F11" om de VBA editor te activeren in de betreffende applicatie. Om deze optie in het lint zichtbaar te maken zie de toelichting in "Activeer VBA.pdf".

-0-0-0-