



# xxter scripts handleiding

<b>De basis</b>	<b>2</b>
<b>Scripts beheren</b>	<b>3</b>
<b>Scripts editor</b>	<b>4</b>
<b>Commando's</b>	<b>5</b>
Componentsturing	5
Het bedienen van (andere) scripts	6
Vertragingen toepassen	8
Scenario's, waarschuwingsservice en planner	8
Opdrachten	9
<b>Waarden</b>	<b>9</b>
<b>Variabelen en berekeningen</b>	<b>10</b>
Variabelen	10
Berekeningen	11
<b>Functies</b>	<b>12</b>
Rekenkundig	12
Tijdfuncties	13
Tekstfuncties	15
<b>Programmeerstructuren</b>	<b>16</b>
IF en WHILE	16
BEGIN en END	17
<b>Overig</b>	<b>17</b>



## xxter scripts

Met xxter scripts kunt u eenvoudig zelf kleine programma's maken binnen xxter. Deze scripts zijn erg flexibel en kunnen vele mogelijkheden toevoegen aan een bestaande domotica-installatie. Zo kunt u logische functies maken, acties vertraagd laten uitvoeren, scènes uitbreiden met verlopende RGB verlichting en veel meer. Scripts kunnen worden geactiveerd door een scène, een planner, een actie (trigger) of een ander script.

Buiten deze handleiding staan op de website verschillende voorbeelden die u kunt downloaden en gebruiken om eigen scripts te maken in xxter. Deze voorbeelden kunnen ook erg handig zijn om te leren hoe xxter scripts werken.

### De basis

xxter scripts zijn geschreven programma's die uit een of meerdere regels bestaan. Elke regel staat voor een specifiek commando. Het is niet mogelijk meerdere commando's op één regel te zetten. U kunt commentaar toevoegen aan een script door gebruik te maken van het “#” karakter. U kunt dit toepassen op een aparte regel of na een commando aan het einde van een regel.

Een voorbeeld van een xxter script:

```
1 #Voorbeeld script
2 #De volgende regel zet een dimmer op 100%
3 SET BYTE(547/Voorbeeld lamp) TO 100
4 #De volgende regel deactiveert een planner
5 SET SCHEDULER(3/Wekker) TO 0
6 #De volgende regel bevat een fout
7 STOP [SCRIPT]
8
```

Scripts kunnen worden gestart of gestopt op vele manieren, bijvoorbeeld door middel van een scène of een planner. Scripts kunnen “oneindig” zijn en op herhalende intervallen acties uitvoeren of worden gedefinieerd als een eenmalige reeks commando's die achtereenvolgens worden uitgevoerd, steeds wanneer het script wordt gestart.



## Scripts beheren

xxter scripts kunnen worden gemaakt en aangepast via de xxter editor. Deze editor kunt u vinden in de projectconfiguratie van “Mijn xxter”. Selecteer het project waarvoor u scripts wilt beheren en klik in de menubalk op “Scripts”. Alle huidige scripts worden hier weergegeven en kunnen worden bewerkt en verwijderd.

Test Project Project is gewijzigd

Instellingen | Componenten | Camera's | Opdrachten | Statistieken/Grafieken data | Aanwezigheidsdetectie | Acties | **Scripts**

Filter  x

Actief Naam			
Ja	Testscript 1		
Nee	Testscript 2	In scene's en planners: Niet beschikbaar	

**Belangrijk:** Als u scripts toevoegt of aanpast, moeten deze worden ingeladen op het xxter apparaat voordat ze kunnen worden gebruikt.

Wanneer er een wijziging is uitgevoerd, ziet u rechtsboven het project de tekst “Project is gewijzigd”. Als u hierop klikt en vervolgens op “Verzoek apparaat om configuratie op te halen”, dan zal het xxter apparaat de nieuwe configuratie ophalen. Dit kunt u alleen gebruiken wanneer xxter is ingesteld om buitenshuis te gebruiken (zie Installatiehandleiding hoofdstuk 13). Als alternatief kunt u ook het profiel opnieuw inladen vanuit de app (zie gebruikershandleiding hoofdstuk 11). Uiteraard kunt u ook het profiel opnieuw inladen vanaf het xxter apparaat zelf, door in te loggen op het apparaat en linksboven in het menu op de knop “Configuratie laden” te drukken.

Wanneer u op het xxter apparaat bent ingelogd, kunt u de ingeladen scripts zien, door in het menu de optie “Scripts” te kiezen. Hier kunnen scripts ook handmatig worden gestart, herstart of gestopt voor testdoeleinden.

### Scripts

Om acties en scripts te wijzigen ga naar '[Mijn xxter](#)'.

Actief	Naam		Status	
Ja	Testscript 1		Gestopt	<input type="button" value="Start"/> <input type="button" value="Herstart"/> <input type="button" value="Stop"/>
Nee	Testscript 2	In scenes en planners: Niet beschikbaar	Niet actief	<input type="button" value="Start"/> <input type="button" value="Herstart"/> <input type="button" value="Stop"/>



## Scripts editor

Als u een script toevoegt of bewerkt, kunt u aangeven of deze actief is of niet. Ook kunt u aangeven of het script voor de eindgebruiker beschikbaar is voor het gebruik in scènes of de planner. Daarnaast kunt u aangeven of het scripts automatisch moet worden gestart, wanneer het xxter apparaat (opnieuw) wordt gestart.

**Let op:** Gedeactiveerde scripts kunnen niet worden uitgevoerd, ook niet als ze worden aangeroepen vanuit een scène, actie of planner. Dit kan handig zijn bij het testen van scripts of bij het verhelpen van problemen. Houd er echter rekening mee dat een script wel kan worden geactiveerd of gedeactiveerd door een ander script.

Test Project Project is gewijzigd

Instellingen | Componenten | Camera's | Opgavten | Statistieken/Grafieken data | Aanwezigheidsdetectie | Acties | **Scripts**

### Script - Bewerken

Actief:

Naam:

Bij opstarten:

In scene's en planners:

Help

COMMANDS

- RESTART
- START [script]
- START AT BEGIN [script]
- STOP
- STOP [script]
- WAIT [time]
- WAIT [time] RANDOM [time]
- CALL [scenario]
- CALL [alert]
- CALL [alert] WITH [value]
- SET [script] TO [on/off]
- SET [scheduler] TO [on/off]
- SET [bit component] TO [value]
- SET [byte component] TO [value]
- SET [2byte component] TO [value]

1

2

De regelnummers die naast het script worden weergegeven staan er alleen ter informatie en worden in het script niet gebruikt. Commando's kunnen worden ingetypt, maar ook worden ingevoegd via het commandovenster, rechtsboven op het scherm. Let bij het invoegen van commando's op dat de cursor zich op de juiste plek in de script editor bevindt. Wanneer de commando's worden ingevoegd zullen de velden, die kunnen worden opgezocht in de xxter configuratie, als link worden ingevoegd. U kunt hierop klikken om direct de juiste waarde te selecteren. Hiermee is het mogelijk snel en gemakkelijk commando's in te voeren.

1 SET [SCHEDULER\(1/Afwezig\)](#) TO OFF

2

3



Nadat een of meerdere commando's zijn toegevoegd via de editor, kan worden gecontroleerd of deze juist zijn ingegeven door op de "Opslaan en controleren" knop te drukken. Alle regels die correct zijn worden in zwart weergegeven. Regels die in rood worden weergegeven hebben ergens een fout in die regel. Commentaar wordt altijd in groen weergegeven.

```
1 # Vertraagd alles uitzetten
2 WAIT 5000
3 SET BIT(601/ALLES UIT) TO [0-1]
4
```

Wanneer u een regel heeft gecorrigeerd of toegevoegd kunt u eenvoudig met dezelfde knop het script nogmaals controleren. Met de "Opslaan en sluiten" knop kan de editor worden afgesloten.

Opmerking: scripts die een fout bevatten kunnen wel worden opgeslagen en gestart. De regels met fouten (in rood aangegeven) worden niet uitgevoerd bij het uitvoeren van het script. xxter slaat deze over en gaat verder met de volgende regel.

```
1 #Voorbeeld script
2 #De volgende regel zet een dimmer op 100%
3 SET BYTE(547/Voorbeeld lamp) TO 100
4 #Het is niet nodig een STOP commando te gebruiken aan
5 #het einde van een script.
6
```

Wanneer het laatste commando aan het einde van een script is uitgevoerd, stop het script automatisch.

Bij het gebruik van het commando-venster zullen sommige delen van commando's tussen [ ] rechte haken staan. Deze delen moeten worden vervangen door geldige waarden. De helptekst tussen de haken geeft een indicatie van de vereiste waarde.

Scripts worden opgebouwd door commando's en waarden. Daarnaast kunnen scripts verder worden uitgebreid door middel van variabelen, functies, berekeningen en programmeerstructuren zoals IF-statements en WHILE-lussen.

## Commando's

De commando's zijn verdeeld in verschillende types.

### Componentsturing

```
SET [component] TO [value]
```

Hiermee is het mogelijk een component van type bit, byte of kommagetal (2byte) naar een bepaalde waarde te zetten. Gebruik een "." (punt) om komma getallen aan te geven, niet de komma. Als



waarde kunnen ook andere componenten worden gebruikt, waardoor het mogelijk is een component de waarde te geven van een ander component.

```
SET [RGB component] TO [red], [green], [blue]
```

Hiermee is het mogelijk een RGB component naar een bepaalde kleur te zetten. De kleur moet worden omschreven met 3 waarden: rood, groen en blauw, allemaal met een waarde van 0 tot 255. In plaats van een vaste waarde kan ook een ander component gebruikt worden voor elk van de 3 kleuren. Ook kunt u in plaats van de drie verschillende kleuren één ander RGB component opgeven. Hiermee wordt de kleur van het ene RGB component gelijk gezet aan die van een ander RGB component.

```
FADE [component] TO [value] IN [time]
```

Dit commando is vergelijkbaar met het SET commando. Dit commando zal de waarde niet direct gelijk zetten, maar er langzaam heen dimmen of veranderen. Met de tijdwaarde kunt u opgeven hoelang er over gedaan moet worden om de betreffende waarde te bereiken. U kunt hier als tijdseenheden MSEC/SEC/MIN/HOUR/DAY toepassen. Indien er geen eenheid wordt opgegeven, wordt uitgegaan van milliseconden. Dit commando zal elke seconde de waarde van het component veranderen. Net als het SET commando is het ook hier mogelijk RGB componenten te gebruiken.

```
FADE [component] TO [value] in [time] STEP [time]
```

Dit commando is erg vergelijkbaar met het commando hierboven, echter is het mogelijk zelf de stap-tijd op te geven waarin de waarden worden veranderd. Ook hier kunt u RGB componenten gebruiken.

```
READ [component]
```

Met dit commando kunt u de huidige status van het betreffende component uitvragen op de bus. U kunt het commando ook gebruiken voor een scenario, waarbij een READ wordt uitgevoerd op alle componenten in de betreffende scène.

### Het bedienen van (andere) scripts

```
STOP
```

Dit commando stopt het huidige script.

```
RESTART
```



Dit commando herstart het huidige script. Met andere woorden: het script wordt opnieuw vanaf het begin uitgevoerd.

**BELANGRIJK:** Wanneer een script automatisch herstart wordt zonder ergens gebruik te maken van een WAIT commando (zie de volgende sectie) wordt het script in een directe oneindige lus gebracht, wat moet worden voorkomen.

```
START [script]
```

Hiermee zal een ander script worden gestart, indien het nog niet loopt. Vervang [script] door het betreffende script dat u wilt starten.

```
START AT BEGIN [script]
```

Hiermee zal een ander script worden gestart, indien het nog niet loopt. Wanneer het script wel al loopt, wordt het opnieuw gestart vanaf het begin (restart). Vervang [script] door het betreffende script dat u wilt (her)starten.

```
STOP [script]
```

Hiermee wordt een ander script gestopt. Vervang [script] door het betreffende script dat u wilt stoppen.

```
SET [script] TO [on/off]
```

Hiermee is het mogelijk een script te activeren of te deactiveren. Vervang [script] door het betreffende script dat u wilt (de)activeren. Alle waarden groter dan 0 worden als aan / activeren geïnterpreteerd en alle waarden van 0 of kleiner als uit / deactiveren. Dit kan handig zijn wanneer een component wordt gebruikt voor het (de)activeren van een script.

Het deactiveren van een draaiend script zal er voor zorgen dat het script direct wordt gestopt.



## Vertragingen toepassen

WAIT [time] of WAIT [time] MSEC/SEC/MIN/HOUR/DAY

Bij dit commando zal het script de opgegeven tijd wachten. Wanneer er geen tijdseenheid is opgegeven wordt uitgegaan van milliseconden. De maximale tijd is ongeveer 20 dagen. In plaats van een vaste waarde op te geven, is het mogelijk een component in te vullen (BYTE of 2BYTE)

```
1 #Voorbeeld script
2 #De volgende regel wacht voor de waarde van de BYTE aan minuten.
3 WAIT BYTE(542/Wachttijd) MIN
4
```

WAIT [time] RANDOM [time]

Dit commando is vergelijkbaar met het vorige commando echter wacht dit commando de opgegeven tijd met daarbij een willekeurig gekozen tijd in het interval tussen 0 en de opgegeven tweede tijd. Bijvoorbeeld: **WAIT 1 HOUR RANDOM 30 MIN**. Deze regel zal wachten voor een periode die ligt tussen 1 uur en 1,5 uur. Elke keer dat deze regel wordt uitgevoerd wordt de vertraging opnieuw berekend, met een andere tijd als resultaat.

## Scenario's, waarschuwingsservice en planner

Bestaande scènes, waarschuwingsservice en planners kunnen ook in scripts worden gebruikt.

CALL [scenario]

Hiermee wordt de opgegeven scène aangeroepen. Vervang [scenario] door de betreffende scène.

LEARN [scenario]

Hiermee wordt de scène aangepast en worden alle componenten in de scène ingesteld op de huidige waarden.

CALL [alert] of CALL [alert] WITH [value]

Hiermee is het mogelijk de waarschuwingsservice aan te roepen vanuit scripts. Vervang [alert] door de betreffende waarschuwingsservice. Wanneer geen aanvullende waarde wordt opgegeven, wordt de waarde 0 gebruikt.





```
SET [scheduler] TO [on/off]
```

Hiermee is het mogelijk een planner te activeren of deactiveren. Vervang [scheduler] door de betreffende planner. Alles groter dan 0 wordt geïnterpreteerd als aan / activeren en 0 en kleiner als uit / deactiveren. Let op dat dit alleen werkt voor planners die handmatige kunnen worden geactiveerd. Planners die automatisch geactiveerd worden door een component kunnen niet via een script worden geactiveerd of gedeactiveerd.

### Opdrachten

```
EXECUTE [network command]
```

Hiermee is het mogelijk een netwerkcommando aan te roepen vanuit scripts. Vervang [network command] door het betreffende commando.

```
EXECUTE [ir command]
```

Hiermee is het mogelijk een infraroodcommando aan te roepen vanuit scripts. Vervang [ir command] door het betreffende commando.

```
EXECUTE ON [upnp device] [upnp command]
```

Hiermee is het mogelijk een commando aan te roepen op een uPnP apparaat vanuit scripts. Vervang [upnp device] door het betreffende uPnP apparaat en [upnp command] door het betreffende commando.

```
EXECUTE ON [diva] [diva macro]
```

Hiermee is het mogelijk een commando aan te roepen op een diva apparaat vanuit scripts. Vervang [diva] door de betreffende diva server en [diva macro] door het betreffende macro.

### Waarden

Wanneer u een commando invoert via het commando-venster, worden de waarden en componenten weergegeven tussen [ ] rechte haken. Tussen de haken wordt een indicatie van de mogelijke waarden weergegeven. Indien mogelijk wordt een verwijzing gemaakt naar de juiste lijst van componenten, scripts, waarschuwingsservices, planners of scènes, waaruit u kunt kiezen.

Geldige componenten worden geschreven in de vorm **SCENARIO(123/name)**. In dit voorbeeld gaat het om een scène met identificatienummer 123. De naam wordt alleen weergegeven voor de leesbaarheid. Wanneer een verkeerde naam wordt ingevoerd, zal deze automatisch worden verbeterd nadat een script is gecontroleerd.



In veel gevallen zullen vaste waarden worden gebruikt, zoals **WAIT 20 SEC**. In sommige gevallen is het echter handig om de waarde van een component te gebruiken in plaats van een vaste waarde. Bij alle commando's is het mogelijk om de vaste waarde door een component te vervangen.

Bij tijdwaarden wordt uitgegaan van milliseconden wanneer er geen eenheid wordt opgegeven. Geldige eenheden zijn: MSEC, SEC, MIN, HOUR en DAY.

## Variabelen en berekeningen

Deze basis functionaliteit kan worden uitgebreid met het gebruik van variabelen en berekeningen.

### Variabelen

U kunt uw eigen variabelen definiëren in scripts. Deze variabelen worden xxter-breed gedefinieerd en zijn dus ook te gebruiken om informatie door te geven aan scripts onderling. Een variabele begint altijd met een dollar teken "\$". Bijvoorbeeld: \$GEMIDDELD. De naam van een variabele mag bestaan uit alfanumerieke tekens (A-Z en 0-9).

Variabelen kunnen worden gebruikt om iedere waarde op te slaan. Bijvoorbeeld:

```
$LICHTPERCENTAGE = 0  
SET BYTE (21/Spots) TO $LICHTPERCENTAGE
```

Of

```
$LICHTSPOTS = BYTE (21/Spots)  
SET BYTE (24/Spots Keuken) TO $LICHTSPOTS
```



## Berekeningen

U kunt variabelen ook gebruiken voor het berekenen van waarden, door gebruik te maken van eenvoudige wiskundige operaties. Berekeningen worden als volgt ingevoerd:

```
$MAXLICHT = ( 2 * 10 ) + 10
```

Uiteraard kunt u hierbij ook de waarde van een component gebruiken, bijvoorbeeld:

```
$MAXLICHT = ( 2 * BYTE(21/Spots) ) + 10
```

**BELANGRIJK:** gebruik bij berekeningen altijd haakjes. Wanneer er geen haakjes staan, wordt de berekening uitgevoerd in de volgorde dat het is geschreven! Zo geeft bijvoorbeeld  $2 * 10 + 10$  het antwoord 30, maar  $10 + 2 * 10$  geeft het antwoord 24. U kunt dit voorkomen door  $10 + (2*10)$  te schrijven. Houd hier altijd rekening mee, als u berekeningen uitvoert.

Gebruik voor kommagetallen de "." (punt), niet de komma. Bijvoorbeeld: 2.42

**LET OP:** Houd er rekening mee dat variabelen xxter-breed gelden. Dit betekent dat een variabele in een script, gebruikt kan worden in een ander script. Indien u een variabele alleen maar binnen één script wilt toepassen, zorg dan dat dezelfde variabele naam niet ook in andere scripts wordt gebruikt.

U kunt gebruik maken van de volgende operatoren:

+	plus	
-	min	
*	vermenigvuldigen	
/	delen	
%	modulo (of deelrest)	
^	tot de macht	
&	logische en	(resultaat = 1 als beide waarden groter zijn dan 0, anders 0)
	logische of	(resultaat = 1 als een van de waarden groter is dan 0, anders 0)
=	is gelijk aan	(resultaat = 1 indien waar, 0 indien onwaar)
<	is kleiner dan	(resultaat = 1 indien waar, 0 indien onwaar)
>	is groter dan	(resultaat = 1 indien waar, 0 indien onwaar)
<>	is niet gelijk aan	(resultaat = 1 indien waar, 0 indien onwaar)

Worteltrekken kan worden geschreven met de machtsoperator:  $^ (1/2)$  = tot de macht  $\frac{1}{2}$  = de wortel van de waarde.



## Funcities

In scripts kunt u gebruik maken van verschillende functies om te rekenen met waarden. Deze functies kunt u gebruiken in combinatie met de commando's en met variabelen.

### Rekenkundig

`ROUND (waarde)`                      of            `ROUND (waarde, precisie)`

Geeft de afgeronde waarde. Wanneer er geen precisie wordt opgegeven wordt "0" gebruikt. Bijvoorbeeld `ROUND(4.3) = 4` en `ROUND(4.32, 1) = 4.3`

`FLOOR (waarde)`                      of            `FLOOR (waarde, precisie)`

Geeft de naar beneden afgeronde waarde. Wanneer er geen precisie wordt opgegeven wordt "0" gebruikt. Bijvoorbeeld `FLOOR(4.7) = 4` en `FLOOR(4.38, 1) = 4.3`

`CEIL (waarde)`                      of            `CEIL (waarde, precisie)`

Geeft de naar boven afgeronde waarde. Wanneer er geen precisie wordt opgegeven wordt "0" gebruikt. Bijvoorbeeld `CEIL(4.2) = 5` en `FLOOR(4.33, 1) = 4.4`

`ABS (waarde)`

Geeft de absolute waarde. Bijvoorbeeld `ABS(-7.2) = 7.2` en `ABS(4.33) = 4.33`

`MIN (waarde, waarde, waarde, ..)`

Geeft de minimale waarde van de ingegeven waarden. Vereist minimaal één waarde als input.

`MAX (waarde, waarde, waarde, ..)`

Geeft de maximale waarde van de ingegeven waarden. Vereist minimaal één waarde als input.

`AVG (waarde, waarde, waarde, ..)`

Geeft de gemiddelde waarde van de ingegeven waarden. Vereist minimaal één waarde als input.

`RANDOM (maxwaarde)`

Geeft een random (willekeurige) waarde tussen 0 en de opgegeven maxwaarde.



## Tijdfuncties

NOW ( )

Geeft de huidige tijd als timestamp in seconden sinds 1 januari 1970 (Unix datum). Dit tijdstip kan bijvoorbeeld worden vergeleken met een eerdere waarde om te bepalen hoeveel tijd is verstreken.

DAYTIME ( )                      of      DAYTIME (timestamp)

Geeft een 1 als de zon op is en een 0 als de zon onder is. Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt.

SECOND ( )                      of      SECOND (timestamp)

Geeft het aantal seconden (0-59). Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt.

MINUTE ( )                      of      MINUTE (timestamp)

Geeft het aantal minuten (0-59). Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt.

HOUR ( )                      of      HOUR (timestamp)

Geeft het aantal uren (0-23). Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt.

DAYOFWEEK ( )                      of      DAYOFWEEK (timestamp)

Geeft de dag van de week (1-7). Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt. 1 = zondag, 2 = maandag, .. 7 = zaterdag.

DAYOFMONTH ( )                      of      DAYOFMONTH (timestamp)

Geeft de dag van de maand (1-31). Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt.

DAYOFYEAR ( )                      of      DAYOFYEAR (timestamp)



Geeft de dag van het jaar (1-366). Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt.

MONTH ( )                      of      MONTH (timestamp)

Geeft de maand (1-12). Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt.

YEAR ( )                      of      YEAR (timestamp)

Geeft het jaar, bijvoorbeeld "2015". Als er een timestamp wordt meegegeven wordt deze tijd gebruikt, anders wordt de huidige tijd gebruikt.



## Tekstfuncties

`CONCAT ("abc", "def", ....)`

Plakt teksten aan elkaar, bijvoorbeeld `CONCAT("abc", "def", "ghi")` geeft "abcdefghi". Als alternatief kunt u ook de punt (".") als text operator gebruiken om tekst aan elkaar te plakken, bijvoorbeeld `"abc"."def"."ghi"` geeft "abcdefghi".

`FORMAT (format, parameters....)`

Kan een tekst opmaken. Het *format* veld geeft aan hoe, daarna kunnen parameters meegegeven worden die erin worden gebruikt. Standaard tekst wordt direct overgenomen en daarnaast heeft u de volgende bijzondere format opties:

- Gebruik %s om een waarde als tekst in te voegen, bijvoorbeeld:  
`FORMAT("Het is nu %s.", "droog")` geeft "Het is nu droog."
- Gebruik %d om een geheel getal in de tekst in te voegen, bijvoorbeeld:  
`FORMAT("Het is nu %d graden.", 10)` geeft "Het is nu 10 graden."
- Gebruik %f om een komma getal in de tekst in te voegen, bijvoorbeeld:  
`FORMAT("Het is nu %f graden.", 10.12345)` geeft "Het is nu 10.12345 graden."  
%f kent een optionele parameter voor het aantal decimalen, deze gebruikt u in het formaat `%.xd` waarbij de x het aantal decimalen is. NB: let op de "." na het procentteken. Bijvoorbeeld:  
`FORMAT("Het is nu %.2f graden.", 10.12345)` geeft "Het is nu 10.12 graden."
- Gebruik %% om een % teken aan te geven, bijvoorbeeld:  
`FORMAT("Het raam is nu %d%% dicht.", 10)` geeft "Het raam is nu 10% dicht."

`SUBSTR (text, startindex) of`  
`SUBSTR (text, startindex, lengte)`

Kan een deel van een tekst uit een andere tekst halen. NB: let hierbij op dat de startindex bij 0 begint. U kunt deze functie met of zonder lengte gebruiken. Zonder lengte wordt de deeltekst altijd tot het einde van de tekst gebruikt.

Voorbeeld: `SUBSTR("abcdef", 3)` geeft "def"

Voorbeeld: `SUBSTR("abcdef", 3, 1)` geeft "d"

`STRLEN (text)`

Deze functie geeft de lengte van de tekst terug, bijvoorbeeld `STRLEN("abcdef")` geeft 6.



## Programmeerstructuren

### IF en WHILE

U kunt IF statements en WHILE lussen toepassen om bepaalde acties situationeel afhankelijk te maken, zodat ze alleen worden uitgevoerd onder bepaalde condities. De conditie waaronder de actie moet worden uitgevoerd kan worden geformuleerd door gebruik te maken van de wiskundige operatoren zoals zojuist beschreven. Bijvoorbeeld:

```
IF $LICHT > 33.33
  $LICHT = 33.33
```

De actie die beschreven staat op de tweede regel zal alleen worden uitgevoerd, wanneer de berekening achter IF waar is (groter dan 0). De actie zal worden overgeslagen als de berekening onwaar is (0 of kleiner).

U kunt de IF constructie ook uitbreiden met een ELSE constructie:

```
IF $LICHT > 33.33
  $X = 100
ELSE
  $X = 0
```

De ELSE constructie moet altijd op een aparte regel staan, direct na de actie die wordt uitgevoerd als de IF constructie waar is.

Behalve de IF constructie, kunt u ook gebruik maken van de WHILE lus. Het verschil tussen IF en WHILE, is dat bij IF een actie altijd slechts één maal wordt uitgevoerd, terwijl de WHILE lus wordt herhaald net zo lang tot de conditie niet langer geldig (waar) is. Bijvoorbeeld:

```
$TELLER = 0
WHILE $TELLER < 10
  $TELLER = $TELLER+1
```

In dit voorbeeld zal in de WHILE lus de variabele TELLER steeds worden opgehoogd, totdat deze 10 of hoger is. In de volgende sectie wordt uitgelegd hoe u meerdere commando's in een lus kunt plaatsen.

**BELANGRIJK:** wanneer de conditie in een WHILE lus altijd waar blijft, zal de lus oneindig doorlopen. Als dit wordt toegepast zonder in de acties een WAIT commando op te nemen, wordt het script in een directe oneindige lus gebracht, wat moet worden voorkomen.





## BEGIN en END

Indien u meerdere acties uitgevoerd wilt hebben in een IF, WHILE of ELSE statement, kunt u gebruik maken van de BEGIN en END constructie.

Ook deze moeten op een aparte regel worden geschreven. Bijvoorbeeld:

```
$X = 0
WHILE $X < 11
  BEGIN
    $Y = $X * 10
    SET BYTE (22/EXAMPLE BYTE) TO $Y
    WAIT 2 SEC
    $X = $X + 1
  END
```

En een ander voorbeeld:

```
IF $LICHT > 33.33
  BEGIN
    $X = 100
    $Y = 0
  END
ELSE
  BEGIN
    $X = 0
    $Y = 100
  END
```

## Overig

```
SEND WAKEONLAN TO [ma:c :ad:re:ss]
```



Met dit commando kunt u een wake on lan commando verzenden naar een netwerkapparaat. Vervang [ma:c :ad:re:ss] door het mac-adres van het betreffende apparaat.

OPEN KNXTUNNEL

Met dit commando kunt u de knx-tunnelingang op het xxter apparaat openstellen voor programmering vanuit ETS voor een periode van 8 uur.

CLOSE KNXTUNNEL

Met dit commando kunt u de knx-tunnelingang op het xxter apparaat weer dichtzetten voor programmering vanuit ETS.

Door gebruik te maken van alle beschreven commando's, waarden, functies en programmeerstructuren is het mogelijk bijna alle benodigde functionaliteit in xxter in te bouwen.