

TI-Nspire™ Python Programmeerhandleiding

Belangrijke informatie

Tenzij expliciet anders vermeld in de bij een programma meegeleverde licentie, geeft Texas Instruments geen garantie, expliciet dan wel impliciet, met inbegrip van, maar niet beperkt tot willekeurig welke impliciete garanties van verhandelbaarheid en geschiktheid voor een bepaald doel met betrekking tot welke programma's of boekmaterialen dan ook, en stelt dergelijke materialen uitsluitend beschikbaar op een "as-is" basis. Texas Instruments is in geen enkel geval aansprakelijk voor speciale, indirecte, incidentele of voortvloeiende schade in verband met of voortkomend uit de aankoop of het gebruik van deze materialen, en de enige en uitsluitende aansprakelijkheid van Texas Instruments, ongeacht de actievorm, is niet hoger dan het in de licentie voor het programma vermelde bedrag. Voorts is Texas Instruments niet aansprakelijk voor welke eis van welke aard dan ook tegen het gebruik van deze materialen door enige andere partij.

© 2021 Texas Instruments Incorporated

"Python" en de Python-logo's zijn handelsmerken of geregistreerde handelsmerken van de Python Software Foundation, gebruikt door Texas Instruments Incorporated met toestemming van de Foundation.

Feitelijke producten kunnen enigszins afwijken van de getoonde afbeeldingen.

Inhoudsopgave

Aan de slag met Python Programmeren	1
Python-modules	1
Een Python-programma als een module installeren	2
Python-werkruimtes	4
Python Editor	4
Python Shell	8
Python Menu Map	11
Acties-menu	12
Run-menu	13
Tools-menu	14
Edit-menu	15
Menu met Built-ins (inbouwprogramma's)	16
Math-menu	19
Random-menu	21
TI PlotLib-menu	22
TI Hub-menu	24
TI Rover-menu	33
Complex Math-menu	41
Time-menu	42
TI System-menu	43
TI Draw-menu	44
TI Image-menu	47
Variables-menu	49
Bijlage	50
Python-trefwoorden	51
Python-sleuteltoewijzing	52
Voorbeelden van Python-programma's	54
Algemene informatie	61

Aan de slag met Python Programmeren

Bij het gebruiken van Python met TI-Nspire™-producten kunt u:

- Python-programma's toevoegen aan TNS-bestanden;
- Python-programma's maken met behulp van sjablonen;
- interactie hebben en gegevens delen met andere TI-Nspire™-apps;
- Communiceren met de TI-Innovator™ Hub en TI-Innovator™ Rover

De TI-Nspire™ Python-omgeving is gebaseerd op MicroPython, een kleine subset van de Python 3 standaardbibliotheek ontworpen om op microcontrollers te draaien. De originele MicroPython-omgeving is aangepast voor gebruik door TI.

Opmerking: Sommige numerieke antwoorden kunnen verschillen van de Rekenmachineresultaten vanwege verschillen in de onderliggende, geïmplementeerde, wiskundige instellingen.

Python is beschikbaar op de volgende TI-Nspire™-producten:

Rekenmachines	Desktop Software
TI-Nspire™ CX II	TI-Nspire™ CX Premium Teacher Software
TI-Nspire™ CX II CAS	Handleiding TI-Nspire™ CX CAS Premium-Docentensoftware
TI-Nspire™ CX II-T	TI-Nspire™ CX Student Software
TI-Nspire™ CX II-T CAS	TI-Nspire™ CX CAS Student-software
TI-Nspire™ CX II-C	
TI-Nspire™ CX II-C CAS	

Opmerking: In de meeste gevallen is de functionaliteit identiek op de rekenmachine en in de softwareweergaven, maar u zou enkele verschillen kunnen zien. Deze handleiding gaat ervan uit dat u rekenmachine of de rekenmachineweergave in de software gebruikt.

Python-modules

TI-Nspire™ Python bevat de volgende modules:

Standaardmodules	TI-modules
Wiskunde (math)	TI PlotLib (ti_plotlib)
Random (random)	TI Hub (ti_hub)
Complex Math (cmath)	TI Rover (ti_rover)
Time (time)	TI System (ti_system)
	TI Draw (ti_draw)
	TI Image (ti_image)

Opmerking: Als u bestaande Python-programma's hebt die zijn gemaakt in andere Python-ontwikkelomgevingen, moet u ze mogelijk bewerken om ze in de TI-Nspire™ Python-omgeving te laten uitvoeren. Modules kunnen verschillende methoden, argumenten en volgordes van methoden in een programma gebruiken in vergelijking met de TI-modules. Wees in het algemeen bewust van compatibiliteit bij het gebruik van een willekeurige versie van Python- en Python-modules.

Wanneer u Python-programma's overzet van een niet-TI-platform naar een TI-platform OF van het ene TI-product naar het andere, vergeet dan het volgende niet:

- Programma's die kerntaalfuncties en standaardbibliotheken (libs) gebruiken (wiskunde, willekeurig, enz.) kunnen zonder wijzigingen worden overgezet.
- Programma's die platformspecifieke bibliotheken gebruiken, zoals matplotlib voor PC- of TI-modules, vereisen bewerkingen voordat ze kunnen worden uitgevoerd op een ander platform. Dit kan zelfs tussen TI-platformen het geval zijn.

Zoals bij elke versie van Python moet u imports opnemen om functies, methoden of constanten te gebruiken die in een bepaalde module zijn opgenomen. Gebruik bijvoorbeeld de volgende opdrachten om de functie `cos()` uit de module `math` uit te voeren:

```
>>>from math import *
>>>cos(0)
1.0
```

Voor een lijst met menu's met hun items en beschrijvingen, zie het [Menukaart](#) hoofdstuk.

Een Python-programma als een module installeren

Om uw Python-programma als een module op te slaan:

- Selecteer in de Editor **Acties > Installeren als Python-module**.
- Selecteer in de Shell **Tools > Installeren als Python-module**.

Na selectie gebeurt het volgende:

- De Python-syntax wordt aangevinkt.
- Het bestand wordt opgeslagen en verplaatst naar de PyLib-map.
- Er verschijnt een dialoogvenster waarin wordt bevestigd dat het bestand is geïnstalleerd als een module.
- Het bestand is gesloten en de module is klaar voor gebruik.
- De modulenaam wordt toegevoegd aan het menu **Meer modules** met een menu-item **van <module> importeren ***.

Als u van plan bent om deze module met anderen te delen, is het aan te bevelen dat u deze richtlijnen volgt:

- Sla slechts één module per TNS-bestand op.
- De modulenaam komt overeen met de naam van het TNS-bestand (bijv. "my_program"-module bevindt zich in het bestand "my_program.tns").

- Voeg een Notities-pagina toe voor de Python-editor die de bedoeling van de module, de versie en de functies beschrijft.
- Gebruik de functie `ver()` om het versienummer van de module weer te geven.
- (Optioneel) Voeg een helpfunctie toe om de lijst met methoden in de functie weer te geven.

Python-werkruimtes

Er zijn twee werkruimtes voor uw Python-programmeren: De Python-editor en de Python-shell.

Python Editor	Python Shell
<ul style="list-style-type: none">• Python-programma's maken, bewerken en opslaan• Syntaxmarkering en automatisch inspringen• Inline-aanwijzingen voor het gebruik van functieargumenten• Tooltips om het bereik van geldige waarden weer te geven• <code>var</code>-toets toont globale gebruikersvariabelen en -functies gedefinieerd in het huidige programma• Sneltoetsen op het toetsenbord	<ul style="list-style-type: none">• Voer Python-programma's uit• Handig voor het testen van kleine stukjes programmacode• Interactie met Shell-geschiedenis om eerdere invoer en uitvoer te selecteren voor hergebruik• <code>var</code>-toets geeft een lijst met gebruikersvariabelen gedefinieerd in het laatste programma, dat is uitgevoerd in de gegeven opdracht

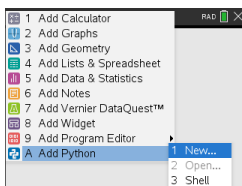
Opmerking: Meerdere Python-programma's en Shells kunnen aan een probleem worden toegevoegd.

Python Editor

In de Python Editor kunt u Python-programma's maken, bewerken en opslaan.

Een Python Editor-pagina toevoegen

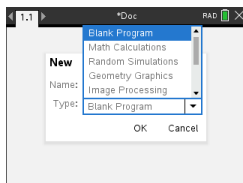
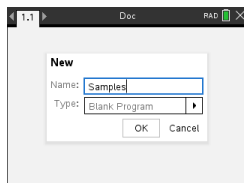
Om een nieuwe Python Editor-pagina toe te voegen in de huidige opdracht, drukt u op `menu` en selecteert u **Python toevoegen > Nieuw**.



U kunt een leeg programma maken of u kunt een sjabloon selecteren.

Leeg programma

Sjabloon



Nadat het programma is gemaakt, wordt de Python-editor weergegeven. Als u een sjabloon hebt geselecteerd, worden de noodzakelijke importopdrachten automatisch toegevoegd (zie hieronder).

Opmerking: U kunt meerdere programma's in één TNS-bestand hebben, net als andere apps. Als het Python-programma bedoeld is om als module te worden gebruikt, kan het TNS-bestand worden opgeslagen in de PyLib-map. Die module kan vervolgens worden gebruikt in andere programma's en documenten.

Wiskundige berekeningen

toeval simulaties (random)

```

1.1 | Templates.py | 5/5
# Math Calculations
=====
from math import *
=====
  
```

```

1.1 | Templates.py | 6/6
# Random Simulations
=====
from math import *
from random import *
=====
  
```

Geometrische grafiek

Beeldverwerking

```

1.1 | Templates.py | 5/5
# Geometry Graphics
=====
from ti_draw import *
=====
  
```

```

1.1 | Templates.py | 6/6
# Image Processing
=====
from ti_image import *
from ti_draw import get_screen_dim
=====
  
```

Plotting (x,y) & Text

Gegevens delen

```

1.1 | Templates.py | 5/5
# Plotting (x,y) & Text
=====
import ti_plotlib as plt
=====
  
```

```

1.1 | Templates.py | 5/5
# Data Sharing
=====
from ti_system import *
=====
  
```

TI-Innovator Hub-project

TI-Rover Programmeren


```
1.1 |> *Doc RAD 1/1
+ *Templates.py 9/9
# Hub Project
#=====
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at_cls
from ti_system import get_key
#=====
```

```
1.1 |> *Doc RAD 6/6
+ *Templates.py 6/6
# Rover Coding
#=====
import ti_rover as rv
from math import *
#=====
```

Een Python-programma openen

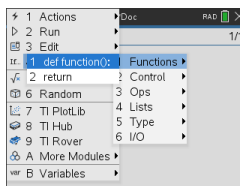
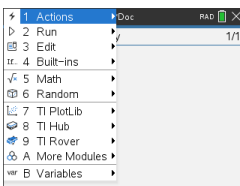
Om een bestaand Python-programma te openen drukt u op **[doc]** en selecteert u **Invoegen > Python toevoegen > Open**. Hierdoor wordt een lijst met programma's weergegeven die in het TNS-bestand zijn opgeslagen.

Als de Editor-pagina die werd gebruikt om het programma aan te maken is verwijderd, is het programma nog steeds beschikbaar in het TNS-bestand.

Werken in de Python-editor

Drukken op **[menu]** geeft het menu Documenttools weer. Met deze menuopties kunt u blokken programmacode voor uw programma toevoegen, verplaatsen en kopiëren.

Het menu Documenttools



Items die geselecteerd zijn uit de modulemenu's zullen automatisch een programmacode-sjabloon aan de Editor toevoegen met inline-aanwijzingen voor elk onderdeel van de functie of opdracht. U kunt van het ene argument naar het volgende navigeren door te drukken op **[tab]** (vooruit) of **[shift]+[tab]** (achteruit). Tooltips of pop-uplijsten verschijnen wanneer ze beschikbaar zijn, om u te helpen de juiste waarden te selecteren.

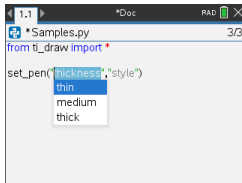
Inline-prompts

```
1.1 |> *Doc RAD 3/4
+ *Samples.py 3/4
from ti_draw import *
def function():
  +=block
```

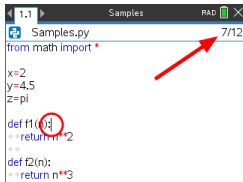
Knopinfo

```
1.1 |> *Doc RAD 3/3
+ *Samples.py 3/3
from ti_draw import *
set_color(red, green, blue)
  0-255
```

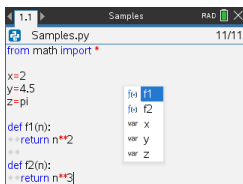
Pop-uplijsten



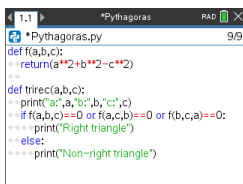
De nummers rechts van de programmaam geven het huidige regelnummer van de cursor aan en het totaal aantal regels in het programma.



Globale functies en variabelen die zijn gedefinieerd op de regels boven de huidige cursorpositie kunnen worden ingevoegd door te drukken op h en te selecteren uit de lijst.



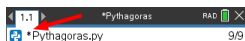
Terwijl u opdrachten (code) toevoegt aan uw programma, geeft de Editor trefwoorden, operatoren, opmerkingen, tekenreeksen en inspringing weer in verschillende kleuren om de verschillende elementen te identificeren.



Programma's opslaan en uitvoeren

Als u klaar bent met uw programma, drukt u op **menu** en selecteert u **Run > Check Syntax & Save**. Dit controleert de syntax van het Python-programma en slaat het op in het TNS-bestand.

Opmerking: Als u niet-opgeslagen wijzigingen in uw programma heeft, wordt naast de programmaam een asterisk (*) weergegeven.



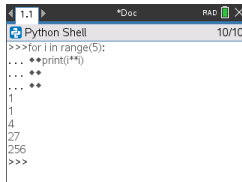
Om het programma uit te voeren drukt u op **[menu]** en selecteert u **Uitvoeren > Uitvoeren**. Dit zal het huidige programma uitvoeren op de volgende Python Shell-pagina of op een nieuwe als de volgende pagina geen Shell is.

Opmerking: Door het programma uit te voeren, controleert u automatisch de syntaxis en slaat u het programma op.

Python Shell

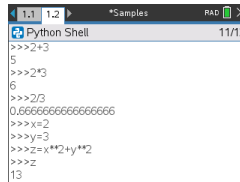
De Python Shell is de 'vertaler' die uw Python-programma's, andere stukjes Python-programmacode of eenvoudige opdrachten uitvoert.

Python-code



```
Python Shell 10/10
>>>for i in range(5):
...     **print(i**i)
...     **
...     **
1
1
4
27
256
>>>
```

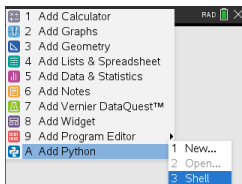
Eenvoudige opdrachten



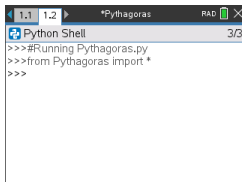
```
Python Shell 11/12
>>>2+3
5
>>>2*3
6
>>>2/3
0.6666666666666666
>>>x=2
>>>y=3
>>>z=x**2+y**2
>>>z
13
```

Een Python Shell-pagina toevoegen

Om een nieuwe Python Shell-pagina toe te voegen in de huidige opdracht, drukt u op **[menu]** en selecteert u **Python toevoegen > Shell**.



De Python Shell kan ook vanuit de Python Editor worden gestart door een programma uit te voeren door op **[menu]** te drukken en **Uitvoeren > Uitvoeren** te selecteren.

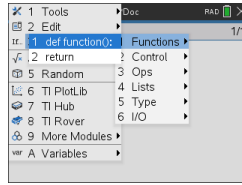
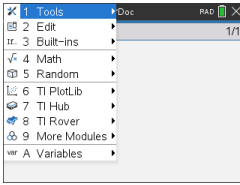


```
Python Shell 3/3
>>>#Running Pythagoras.py
>>>from Pythagoras import *
>>>
```

Werken in de Python Shell

Drukken op **[menu]** geeft het menu Documenttools weer. Met deze menuopties kunt u blokken programmacode toevoegen, verplaatsen en kopiëren.

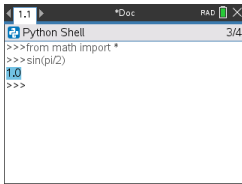
Het menu Documenttools



Opmerking: Als u een methode (functie of opdracht) gebruikt van een van de beschikbare modules, zorg er dan voor dat u eerst een importmodule-, opdracht uitvoert zoals in elke Python programmeeromgeving.

Interactie met de Shell-uitvoer is vergelijkbaar met de Rekenmachine-app waar u eerdere invoer en uitvoer kunt selecteren en kopiëren voor gebruik elders in de Shell, Editor of andere apps.

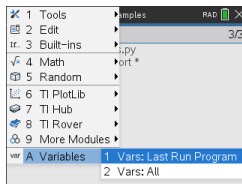
Pijltje omhoog om te selecteren, vervolgens kopiëren en plakken naar de gewenste locatie



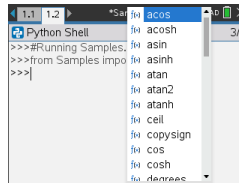
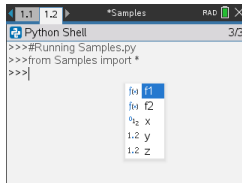
Algemene functies en variabelen van het laatst uitgevoerde programma kunnen worden ingevoegd door op **var** of **ctrl**+**L** te drukken en te selecteren uit de lijst of door op **menu** te drukken en **Variables > Vars te selecteren: Last Run Program**.

Om uit een lijst met globale functies en variabelen te kiezen van zowel het laatst uitgevoerde programma als elke geïmporteerde module, drukt u op **menu** en selecteert u **Variabelen > Vars All**.

Variables menu

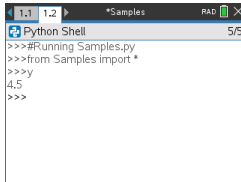


Variabelen uit laatst uitgevoerde programma **Alle variabelen**

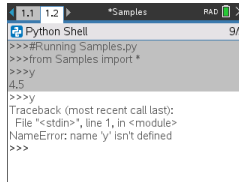


Alle Python Shell-pagina's in dezelfde opdracht delen dezelfde status (definities van door de gebruiker gedefinieerde en geïmporteerde variabelen). Wanneer u een Python-programma opslaat of uitvoert in die opdracht, of drukt op **[menu]** en **Gereedschappen > Shell opnieuw initialiseren** selecteert, dan heeft de Shell-geschiedenis een grijze achtergrond die aangeeft dat de vorige status niet meer geldig is.

Voordat u opslaat of opnieuw initialiseert *Na het opslaan of opnieuw initialiseren*



A screenshot of a Python Shell window titled '*Samples' with a tab 'Python Shell' and a status bar '5/5'. The shell prompt is '>>>'. The user has entered the following code: '#Running Samples.py', 'from Samples import *', 'y', and '4.5'. The output shows '4.5'.



A screenshot of a Python Shell window titled '*Samples' with a tab 'Python Shell' and a status bar '9/9'. The shell prompt is '>>>'. The user has entered the same code as in the previous screenshot. The output shows a traceback error: 'Traceback (most recent call last): File "<stdin>", line 1, in <module> NameError: name 'y' isn't defined'. The error message is displayed on a grey background.

Opmerking: De optie **[menu]** **Gereedschappen > Geschiedenis wissen** wist het scherm van elke vorige activiteit in de Shell, maar variabelen zijn nog steeds beschikbaar.

Meldingen

Foutmeldingen en andere informatieve berichten kunnen worden weergegeven terwijl u in een Python-sessie bent. Als er een fout wordt weergegeven in de Shell wanneer een programma wordt uitgevoerd, wordt een programmaregelnummer weergegeven. Druk op **[ctrl]** **[menu]** en selecteer **Go to Python Editor** (Ga naar Python Editor). Druk in de Editor op **[menu]** en selecteer vervolgens **Edit > Go to Line** (Bewerken Ga naar regel). Vul het regelnummer in en druk op **[enter]**. De cursor wordt weergegeven op het eerste teken van de regel waarin de fout is opgetreden.

Een lopend programma onderbreken

Wanneer een programma of functie wordt uitgevoerd, wordt de aanwijzer " bezig " **[stop]** weergegeven.

- ▶ Het programma of de functie stoppen,
 - Windows®: Druk op de toets **F12**.
 - Mac®: Druk op de toets **F5**.
 - Rekenmachine: Druk op de toets **[stop]**.

Python Menu Map

In dit hoofdstuk worden alle menu's en menu-items voor de Python Editor en Shell weergegeven met een korte beschrijving voor elk.

Opmerking: Voor de menu-items die sneltoetsen hebben, moeten Mac[®]-gebruikers ⌘ (CMD) gebruiken overal waar **Ctrl** staat. Zie de TI-Nspire™ Technologie eGuide voor een volledige lijst van TI-Nspire™ rekenmachine en softwaresnelkoppelingen.

Acties-menu	12
Run-menu	13
Tools-menu	14
Edit-menu	15
Menu met Built-ins (inbouwprogramma's)	16
Math-menu	19
Random-menu	21
TI PlotLib-menu	22
TI Hub-menu	24
TI Rover-menu	33
Complex Math-menu	41
Time-menu	42
TI System-menu	43
TI Draw-menu	44
TI Image-menu	47
Variables-menu	49

Acties-menu

Opmerking: Dit is alleen van toepassing op de Editor.

object	matrix
Nieuw	Opent het dialoogvenster New (Nieuw) waarin u een naam invult en een type selecteert voor uw nieuwe programma.
Openen	Opent een lijst met programma's die beschikbaar zijn in het huidige document.
Kopie creëren	Opent het dialoogvenster Create Copy (Kopie maken) waarin u het huidige programma onder een andere naam kunt opslaan.
Rename	Opent het dialoogvenster Rename (Naam wijzigen) waarin u de naam van het huidige programma kunt wijzigen.
Het script	Sluit het huidige programma af.
Instellingen	Opent het dialoogvenster Settings (Instellingen) waarin u de lettergrootte voor zowel de Editor als de Shell kunt wijzigen.
Installeren als Python-module	Hiermee controleert u de Python-syntax van het huidige TNS-bestand en verplaatst u dit naar de map PyLib.

Run-menu

Opmerking: Dit is alleen van toepassing op de Editor.

object	Sneltoets	matrix
Run	Ctrl+R	Controleert syntax, slaat programma op en voert uit in een Python Shell.
Check Syntax & Save	Ctrl+B	Controleert de syntaxis en slaat het programma op.
Go to Shell	N/A	Verschuift focus naar de Shell die gerelateerd is aan het huidige programma of opent een nieuwe Shell-pagina naast de Editor.

Tools-menu

Opmerking: Dit is alleen van toepassing op de Shell.

object	Sneltoets	matrix
Rerun Last Program	Ctrl+R	Voert het laatste programma gerelateerd aan de huidige Shell opnieuw uit.
Go to Python Editor	N/A	Opent de pagina Editor gerelateerd aan de huidige Shell.
Run	N/A	Opent een lijst met programma's die beschikbaar zijn in het huidige document. Na selectie wordt het gekozen programma uitgevoerd.
Geschiedenis wissen	N/A	Wist de geschiedenis van de huidige Shell maar initialiseert de Shell niet opnieuw.
Reinitialize Shell	N/A	Stelt de status van alle open Shell-pagina's in het huidige probleem opnieuw in. Alle gedefinieerde variabelen en geïmporteerde functies zijn niet langer beschikbaar.
dir()	N/A	Geeft de lijst met functies in de gespecificeerde module weer wanneer gebruikt na de import opdracht.
From PROGRAM import *	N/A	Opent een lijst met programma's die beschikbaar zijn in het huidige document. Na selectie wordt de importopdracht in de Shell geplakt.
Installeren als Python-module	N/A	Alleen ingeschakeld voor modules in binaire indeling. Hiermee verplaatst u het huidige TNS-bestand naar de map PyLib.

Edit-menu

Opmerking: Ctrl+A selecteert alle regels van een programma om ze te knippen of te verwijderen (alleen Editor), of te kopiëren en plakken (Editor en Shell).

object	Sneltoets	matrix
Inspringen	TAB*	Laat tekst op de huidige regel of geselecteerde regels inspringen. * Als er onvolledige inline-aanwijzingen (prompts) zijn, gaat TAB naar de volgende prompt.
Uitspringen	Shift+TAB**	Laat de tekst op de huidige regel of geselecteerde regels uitspringen. ** Als er onvolledige inline-aanwijzingen (prompts) zijn, gaat Shift+TAB naar de vorige prompt.
Opmerking/geen opmerking	Ctrl+T	Voegt het opmerkingensymbool toe of verwijdert dit aan het begin van de huidige regel.
String (tekenreeks) van meerdere regels invoegen	N/A	(Alleen editor) voegt sjabloon van meerdere regels in
Zoeken	Ctrl+F	(Alleen Editor) Opent het dialoogvenster Zoeken (Find) en zoekt naar de ingevoerde string in het huidige programma.
Vervangen	Ctrl+H	(Alleen editor) Opent het dialoogvenster Replace (Vervangen) en zoekt naar de ingevoerde tekenreeks in het huidige programma.
Ga naar regel	Ctrl+G	(Alleen Editor) Opent het dialoogvenster Go to line (Ga naar regel) en springt naar de gespecificeerde regel in het huidige programma.
Begin van de regel	Ctrl+8	Verplaatst de cursor naar het begin van de huidige regel.
Einde van de regel	Ctrl+2	Verplaatst de cursor naar het eind van de huidige regel.
Naar bovenste springen	Ctrl+7	Verplaatst de cursor naar het begin van de eerste regel in het programma.
Naar onderste springen	Ctrl+1	Verplaatst de cursor naar het eind van de laatste regel in het programma.

Menu met Built-ins (inbouwprogramma's)

functies

object	matrix
def function():	Definieert een functie afhankelijk van gespecificeerde variabelen.
return	Definieert de waarde die door een functie wordt geproduceerd.

Besturing

object	matrix
if..	Voorwaardelijke verklaring.
if..else..	Voorwaardelijke verklaring.
if..elif..else..	Voorwaardelijke verklaring.
voor index in reeks waarden (grootte):	Herhaalt (itereert) over een reeks waarden.
voor index in reeks waarden (start, stop):	Herhaalt (itereert) over een reeks waarden.
voor index in reeks waarden (start, stop, stap):	Herhaalt (itereert) over een reeks waarden.
voor index in lijst:	Herhaalt voor de elementen van de lijst.
while..	Voert opdrachten uit in een blok programmacode totdat een voorwaarde wordt uitgewerkt naar Onwaar.
elif:	Voorwaardelijke verklaring.
else:	Voorwaardelijke verklaring.

Ops

object	matrix
x=y	Stelt de waarde van de variabele in.
x==y	Plakt 'is gelijk aan' (==) vergelijkingsoperator.
x!=y	Plakt 'is niet gelijk aan' (!=) vergelijkingsoperator.

object	matrix
x>y	Plakt groter dan (>) vergelijkingsoperator.
x>=y	Plakt groter dan of gelijk aan (>=) vergelijkingsteken.
x<y	Plakt kleiner dan (<) vergelijkingsteken.
x<=y	Plakt kleiner dan of gelijk aan (<=) vergelijkingsteken.
en	Plakt and (en) logische operator.
of	Plakt or (of) logische operator.
niet	Plakt not (niet) logische operator.
Waar	Plakt True (waar) booleaanse waarde.
Onwaar	Plakt False (onwaar) booleaanse waarde.

Lijsten gedefinieerd

object	matrix
[]	Plakt rechte haken ([]).
list()	Zet een rij om in lijst" type."
len()	Geeft de som van de elementen in een lijst.
max()	Geeft de maximumwaarde in de lijst.
min()	Geeft de maximumwaarde in de lijst.
.append()	De opdracht voegt een element toe aan een lijst.
.remove()	De opdracht verwijdert de eerste instantie van een element uit een lijst.
range(start,stop,step)	Geeft een reeks getallen.
voor index in reeks waarden (start, stop, stap)	Wordt gebruikt om over een reeks waarden (bereik) te herhalen (itereren).
.insert()	De opdracht voegt een element toe op de gespecificeerde positie.
.split()	De opdracht geeft een lijst weer met elementen gescheiden door gespecificeerd scheidingsteken.
sum()	Geeft de som van de elementen in een lijst.

object	matrix
sorted()	Geeft een gesorteerde lijst.
.sort()	De opdracht sorteert een lijst.

Type

object	matrix
int()	Geeft een geheel deel.
float()	Geeft een waarde met drijvende komma.
round(x, ndigits)	Geeft een getal met drijvende komma dat afgerond is met het gespecificeerde aantal cijfers.
str()	Geeft een string (tekenreeks).
complex()	Geeft een complex getal.
type()	Geeft het type object.

I/O

object	matrix
print()	Geeft argument weer als tekenreeks.
input()	Vraagt de gebruiker om iets in te vullen.
eval()	Werkt een uitdrukking uit die als een tekenreeks wordt weergegeven.
.format()	De opdracht formatteert de gespecificeerde tekenreeks.

Math-menu

Opmerking: Bij het maken van een nieuw programma dat gebruikmaakt van deze module, wordt aanbevolen om het programmatype **Math Calculations** (Wiskundige berekeningen) te gebruiken. Dit zorgt ervoor dat alle relevante modules worden geïmporteerd.

object	matrix
uit wiskunde import*	Importeert alle methoden (opdrachten en functies) uit de wiskundemodule.
fabs()	Geeft de absolute waarde van een reëel getal.
sqrt()	Geeft de vierkantswortel van een reëel getal.
exp()	Geeft $e^{**}x$.
pow(x,y)	Geeft x tot de macht y
log(x,base)	Geeft $\log_{base}(x)$. log(x) zonder grondtal geeft de natuurlijke logaritme (ln) van x.
fmod(x,y)	Geeft de modulowaarde van x en y. Gebruik dit wanneer x en y getallen met een drijvende komma zijn.
ceil()	Geeft het kleinste geheel getal groter dan of gelijk aan een reëel getal.
floor()	Geeft het grootste geheel getal kleiner dan of gelijk aan een reëel getal.
trunc()	Kapt een reëel getal af op een geheel getal.
frexp()	Geeft een paar (y,n) waarbij $x == y * 2^{**}n$.

Const

object	matrix
e	Returns value for the constant e.
pi	Returns value for the constant pi.

Gonio

object	matrix
radians()	Converteert hoek in graden naar radialen.

object	matrix
degrees()	Converteert hoek in radialen naar graden.
sin()	Geeft de sinus van het argument in radialen.
cos()	Geeft de cosinus van het argument in radialen.
tan()	Geeft de tangens van het argument in radialen.
asin()	Geeft de arcsinus van een argument in radialen.
acos()	Geeft de arccosinus van een argument in radialen.
atan()	Geeft de arctangens van argument in radialen.
atan2(y,x)	Geeft de arctangens van y/x in radialen.

Random-menu

Opmerking: Bij het maken van een nieuw programma dat deze module gebruikt, wordt aanbevolen om het programmatype **Random Simulations** (Willekeurige simulaties) te gebruiken. Dit zorgt ervoor dat alle relevante modules worden geïmporteerd.

object	matrix
van toevalsgetallen import*	Importeert alle opdrachten en functies uit de module Toevalsgetallen.
random()	Geeft een toevalsgetal met drijvende komma tussen 0 en 1.0.
uniform(min,max)	Geeft een toevalsgetal x (float) zo dat $\text{min} \leq x \leq \text{max}$.
randint(min,max)	Geeft een willekeurig geheel getal tussen min en max.
choice(sequence)	Geeft een willekeurig element uit een niet-lege reeks.
randrange(start,stop,step)	Geeft een willekeurig getal tussen start en stop met stap.
seed()	Initialiseert de generator van toevalsgetallen.

TI PlotLib-menu

Opmerking: Bij het maken van een nieuw programma dat deze module gebruikt, wordt aanbevolen om het programmatype **Plotten (x, y) & Tekst** te gebruiken. Dit zorgt ervoor dat alle relevante modules worden geïmporteerd.

object	matrix
<code>import ti_plotlib as plt</code>	Importeert alle methoden (functies en opdrachten) van de <code>ti_plotlib</code> -module in de "plt"-naamruimte. Als gevolg hiervan worden alle functienamen geplakt uit de menu's voorafgegaan door "plt."

Instellen

object	matrix
<code>cls()</code>	Wist het plot-canvas.
<code>grid(x-scale,y-scale,"style")</code>	Geeft een rooster weer met gespecificeerde schaal voor x- en y-assen.
<code>window(xmin,xmax,ymin,ymax)</code>	Definieert het grafiekvenster door het gespecificeerde horizontale interval (<code>xmin</code> , <code>xmax</code>) en verticale interval (<code>ymin</code> , <code>ymax</code>) af te passen op het toegewezen plotgebied (pixels).
<code>auto_window(x-list,y-list)</code>	Hiermee wordt automatisch het grafiekvenster geschaald om te passen bij de gegevens in de x-lijst en y-lijst die in het programma zijn gespecificeerd vóórafgaand aan het <code>auto_window()</code> .
<code>axes("mode")</code>	Geeft assen weer op het gespecificeerde venster in het plotgebied.
<code>labels("x-label","y-label",x,y)</code>	Geeft "x-label" en "y-label" als labels weer op de plotassen op rij-posities <code>x</code> en <code>y</code> .
<code>title("title")</code>	Toont "title" gecentreerd op de bovenste regel van het venster.
<code>show_plot()</code>	Geeft de gebufferde teken-output weer. De opdrachten <code>use_buffer()</code> en <code>show_plot()</code> zijn nuttig in gevallen waarin het weergeven van meerdere objecten op het scherm vertragingen kunnen veroorzaken (niet nodig in de meeste gevallen).
<code>use_buffer()</code>	Maakt een off-screen buffer mogelijk om het tekenen te versnellen.

Tekenen

object	matrix
<code>color(red,green,blue)</code>	Stelt de kleur in voor alle volgende afbeeldingen/plots.
<code>cls()</code>	Wist het plot-canvas.
<code>show_plot()</code>	Voert de weergave van de grafiek (plot) uit zoals ingesteld in het programma.
<code>scatter(x-list,y-list,"mark")</code>	Plot een reeks van geordende paren uit (x-lijst, y-lijst) met de gespecificeerde markeerstijl.
<code>plot(x-list,y-list,"mark")</code>	Plot een lijn met behulp van de geordende paren uit gespecificeerde x-lijst en y-lijst.
<code>plot(x,y,"mark")</code>	Plot een punt met coördinaten x en y met de gespecificeerde markeerstijl.
<code>line(x1,y1,x2,y2,"mode")</code>	Plot een lijnsegment van (x1, y1) naar (x2, y2).
<code>lin_reg(x-list,y-list,"display")</code>	Berekent en tekent het lineaire regressiemodel, $ax+b$, van x-list,y-list.
<code>pen("size","style")</code>	Stelt de weergave van alle volgende regels in totdat de volgende <code>pen()</code> wordt uitgevoerd.
<code>text_at(row,"text","align")</code>	Geeft tekst" weer in plotgebied bij gespecificeerde "uitlijning".

Eigenschappen

object	matrix
<code>xmin</code>	Gespecificeerde variabele voor vensterargumenten gedefinieerd als <code>plt.xmin</code> .
<code>xmax</code>	Gespecificeerde variabele voor vensterargumenten gedefinieerd als <code>plt.xmax</code> .
<code>ymin</code>	Gespecificeerde variabele voor vensterargumenten gedefinieerd als <code>plt.ymin</code> .
<code>ymax</code>	Gespecificeerde variabele voor vensterargumenten gedefinieerd als <code>plt.ymax</code> .
<code>M</code>	Nadat <code>plt.linreg()</code> in een programma is uitgevoerd, worden de berekende waarden van helling, m en snijpunt met de y-as, b, opgeslagen in <code>plt.m</code> en <code>plt.b</code> .
<code>b</code>	Nadat <code>plt.linreg()</code> in een programma is uitgevoerd, worden de berekende waarden van helling, a en snijpunt met de y-as, b, opgeslagen in <code>plt.a</code> en <code>plt.b</code> .

TI Hub-menu

Opmerking: Bij het maken van een nieuw programma dat deze module gebruikt, wordt aanbevolen om het programmatype **Hub-project** te gebruiken. Dit zorgt ervoor dat alle relevante modules worden geïmporteerd.

object	matrix
<code>from ti_hub import *</code>	Importeert alle opdrachten en functies van de <code>ti_hub</code> module.

Ingebouwde Hub-apparaten > Kleuruitvoer

object	matrix
<code>rgb(red,green,blue)</code>	Stelt de kleur voor de RGB-led in.
<code>blink(frequency,time)</code>	Stelt de knipperfrequentie en-duur in voor de geselecteerde kleur.
<code>off()</code>	Zet de RGB LED uit.

Ingebouwde Hub-apparaten > Lichtuitvoer

object	matrix
<code>on()</code>	Zet de LED aan.
<code>off()</code>	Zet de LED uit.
<code>blink(frequency,time)</code>	Stelt de knipperfrequentie en -duur in voor de LED.

Ingebouwde Hub-apparaten > Geluidoutput

object	matrix
<code>tone(frequency,time)</code>	Speelt een toon met de gespecificeerde frequentie gedurende de gespecificeerde tijd.
<code>note("note",time)</code>	Speelt de gespecificeerde noot in de gespecificeerde tijd af. De noot wordt gespecificeerd met behulp van de naam van de noot en een octaaf. Bijvoorbeeld: A4, C5. De namen van de noten zijn C, CS, D, DS, E, F, FS, G, GS, A, AS en B. De octaafnummers lopen van 1 tot en met 9.

object	matrix
toon(frequentie,tijd, tempo)	Speelt een toon met de gespecificeerde frequentie gedurende de gespecificeerde tijd en in het gespecificeerde tempo. Het tempo definieert het aantal pieptonen per seconde, dat loopt van 0 tot en met 10.
noot("noot",tijd,tempo)	Speelt de gespecificeerde noot gedurende de gespecificeerde tijd en in het gespecificeerde tempo. De noot wordt gespecificeerd met behulp van de naam van de noot en een octaaf. Bijvoorbeeld: A4, C5. De namen van de noten zijn C, CS, D, DS, E, F, FS, G, GS, A, AS en B. De octaafnummers lopen van 1 tot en met 9. De tempowaarden lopen van 1 tot en met 10.

Ingebouwde hub-apparaten > Helderheidsingang

object	matrix
measurement()	Leest de ingebouwde HELDERHEID (lichtniveau) sensor uit en geeft een resultaat. Het standaardbereik is 0-100. Dit kan worden gewijzigd met de functie range().
range(min,max)	Stelt het bereik in voor het in kaart brengen van de metingen van de lichtniveausensor. Als beide ontbreken of zijn ingesteld op de waarde Geen, is het standaardhelderheidsbereik van 0 t/m 100 ingesteld.

Add Input Device-menu

Dit menu bevat een lijst met sensoren (invoerapparaten) die worden ondersteund door de ti_hub-module. Alle menu-items plakken de naam van het object en verwachten een variabele en een poort die met de sensor wordt gebruikt. Elke sensor heeft een measurement() (meetmethode) die de waarde van de sensor geeft.

object	matrix
DHT (Digital Humidity & Temp)	Geeft een lijst bestaande uit de huidige temperatuur, luchtvochtigheid, type sensor en de laatst opgeslagen uitlees-status.
Ranger	Geeft de huidige afstand vanaf de gespecificeerde ultrasone afstandsmeter.

object	matrix
	<ul style="list-style-type: none"> • tijd meting() - Geeft de tijd die het ultrasonische signaal nodig heeft om het object te bereiken (de "tijd van de vlucht").
Lichtniveau	Geeft het helderheidsniveau van de externe lichtniveau (helderheid) sensor.
Temperatuur	<p>Geeft de huidige temperatuur, gelezen van de externe temperatuursensor.</p> <p>De standaardconfiguratie is het ondersteunen van de Seed temperatuur sensor in de poorten IN1, IN 2 of IN 3.</p> <p>Om de TI LM19 temperatuursensor te gebruiken uit het breadboard-pakket van de TI-Innovator™ Hub, bewerkt u de poort naar de BB-pen die in gebruik is en gebruikt u een optioneel argument TIANALG".</p> <p>Voorbeeld: mylm19=temperature("BB 5","TIANALOG")</p>
Moisture	Geeft de waarde van de vochtigheidssensor.
Magnetic	<p>Detecteert de aanwezigheid van een magnetisch veld.</p> <p>De drempelwaarde om de aanwezigheid van het veld te bepalen, wordt ingesteld met de functie trigger().</p> <p>De standaardwaarde van de drempel is 150.</p>
Vernier	<p>Leest de waarde van de Vernier analoge sensor die in de opdracht is gespecificeerd.</p> <p>De opdracht ondersteunt de volgende Vernier-sensoren:</p> <ul style="list-style-type: none"> • temperature - Roestvrijstalen temperatuursensor. • lightlevel - TI-lichtniveausensor. • pressure - Originele gasdruksensor. • pressure - Nieuwere gasdruksensor. • pH - pH-sensor. • force10 - ±10 N instelling, Dubbele kracht-sensor. • force50 - ±50 N instelling, Dubbele kracht-sensor. • accelerometer - Low-G-accelerometer. • algemeen - Hiermee kunt u andere sensoren die niet rechtstreeks hierboven worden ondersteund, instellen en de calibrate () API van hierboven gebruiken om

object	matrix
	vergelijingscoëfficiënten in te stellen.
Analog In	Ondersteunt het gebruik van generieke apparaten met analoge ingang.
Digital In	Geeft de huidige status van de op het DIGITALE object aangesloten digitale pin, of de tijdelijk opgeslagen status van de digitale uitvoerwaarde die het laatst werd INGESTELD op het object.
Potentiometer	Ondersteunt een potentiometersensor. Het bereik van de sensor kan worden gewijzigd met de functie <code>range()</code> (Bereik).
Thermistor	Leest thermistorsensoren. De standaardcoëfficiënten zijn ontworpen om overeen te komen met de thermistor die is opgenomen in het Breadboard-pakket van de TI-Innovator™ Hub, bij gebruik met een vaste weerstand van 10KΩ. Een nieuwe set kalibratiecoëfficiënten en referentie weerstand voor de thermistor kan worden geconfigureerd met behulp van de functie <code>calibrate()</code> .
Loudness	Ondersteunt luidheidssensoren.
Color Input	Biedt interfaces voor een I2C-aangesloten kleurinvoersensor. De <code>bb_poort</code> pen wordt naast de I2C-poort gebruikt om de LED op de kleursensor te bedienen. <ul style="list-style-type: none"> • <code>color_number()</code>: Geeft een waarde van 1 t/m 9 die de kleur vertegenwoordigt die de sensor detecteert. De getallen vertegenwoordigen de kleuren volgens de volgende koppeling: <ol style="list-style-type: none"> 1: Rood 2: Groen 3: Blauw 4: Cyaan 5: Magenta 6: Geel 7: Zwart 8: Wit 9: Grijs • <code>red()</code>: Geeft een waarde van 0 t/m 255 die de

object	matrix
	<p>intensiteit van het gedetecteerde kleurniveau ROOD vertegenwoordigt.</p> <ul style="list-style-type: none"> • green(): Geeft een waarde van 0 t/m 255 die de intensiteit van het gedetecteerde kleurniveau GROEN vertegenwoordigt. • blue(): Geeft een waarde van 0 t/m 255 die de intensiteit van het gedetecteerde kleurniveau BLAUW vertegenwoordigt. • gray(): Geeft een waarde van 0 t/m 255 die de intensiteit van het gedetecteerde kleurniveau GRIJS vertegenwoordigt., waarbij 0 zwart is en 255 wit.
BB Port	<p>Biedt ondersteuning voor het gebruik van alle 10 BB poortpennen als een gecombineerde digitale ingangs-/uitgangspoort.</p> <p>De initialisatiefuncties hebben een optionele parameter "masker" waarmee de subset van de 10 pennen kan worden gebruikt.</p> <ul style="list-style-type: none"> • read_port(): Leest de huidige waarden op de ingangspennen van de BB-poort. • write_port(value): Stelt de output pin-waarden in op de gespecificeerde waarde, waarbij de waarde tussen 0 en 1023 ligt. Merk op dat de waarde ook wordt aangepast aan de maskerwaarde in de var=bbport(mask)-bewerking, als een masker is aangeboden.
Hub Time	Biedt toegang tot de interne milliseconde-timer.
TI-RGB Array	<p>Biedt opdrachten en functies voor het programmeren van de TI-RGB-array.</p> <p>De initialisatiefunctie accepteert een optionele "LAMP"-parameter om een modus met hoge helderheid mogelijk te maken voor de TI-RGB-array die een externe voeding vereist.</p> <ul style="list-style-type: none"> • set(led_position, r,g,b): Stelt een specifieke led_position (0-15) in op de gespecificeerde waarde voor r, g, b, waarbij r, g, b waarden zijn van 0 t/m 255. • set(led_list,rood,groen,blauw): Stelt de LED's gedefinieerd in de "led_list" in op de kleur gespecificeerd als "rood", "groen", "blauw". De "led_list" is een Pythonlijst die indices van de LED's van 0 tot 15 bevat. De opdracht set ([0,2,4,6,15], 0, 0, 255) stelt bijvoorbeeld LED's

object	matrix
	<p>0, 2, 4, 6 en 15 in op blauw.</p> <ul style="list-style-type: none"> • set_all(r,g,b): Stelt alle RGB-LED's in de array in op dezelfde waarde voor r, g, b. • all_off(): Zet alle RGBs in de array uit. • measurement(): Geeft de geschatte stroom die de RGB-array gebruikt van de TI-Innovator™ in milliAmps. • pattern(pattern): Door de waarde van het argument als binaire waarde in het bereik 0 tot 65535 te gebruiken, schakelt u pixels in waarbij waarde 1 betekent dat de pixel weergegeven wordt. LEDs worden ingeschakeld als ROOD met een pwm-waarde van 255. • De opdracht patroon (waarde,rood,groen,blauw): Stelt de LED's gedefinieerd door het "patroon" in op de kleur opgegeven als "rood", "groen", "blauw".

Add Output Device-menu

Dit menu bevat een lijst met uitvoerapparaten die worden ondersteund door de `ti_hub`-module. Alle menu-items plakken de naam van het object en verwachten een variabele en een poort die met het apparaat wordt gebruikt.

object	matrix
LED	Functies voor het aansturen van extern aangesloten leds.
RGB	Ondersteuning voor het aansturen van externe RGB LED's.
TI-RGB Array	Biedt opdrachten en functies voor het programmeren van de TI-RGB-array.
Luidspreker	Functies en opdrachten voor het ondersteunen van een externe luidspreker met de TI-Innovator™ Hub. De functies en opdrachten zijn hetzelfde als die voor "geluid" hierboven.
Macht	Functies en opdrachten voor het aansturen van externe stroom met de TI-Innovator™ Hub. <ul style="list-style-type: none"> • set(value): Stelt het niveau van het vermogen (stroom) in op de gespecificeerde waarde tussen 0 en 100. • on(): Stelt het vermogensniveau in op 100. • off(): Stelt het vermogensniveau in op 0.
Continuous Servo	Functies en opdrachten voor het bedienen van continue servomotoren.

object	matrix
	<ul style="list-style-type: none"> • set_cw(speed,time): De servo draait met de klok mee met de aangegeven snelheid (0-255) en voor de specifieke duur in seconden. • set_ccw(speed,time): De servo draait tegen de klok in met de aangegeven snelheid (0-255) en voor de specifieke duur in seconden. • stop(): Stopt de continue servo.
Analog Out	Functies voor het gebruik van generieke apparaten met analoge ingang.
Vibratiemotor	Functies en opdrachten voor het aansturen van vibratiemotoren. <ul style="list-style-type: none"> • set(val): Stelt de intensiteit van de vibratiemotor in op "val" (0-255). • off(): Schakelt de vibratiemotor uit. • on(): Zet de vibratiemotor aan op het hoogste niveau.
Relais	Stuurt interfaces aan voor het aansturen van relais. <ul style="list-style-type: none"> • on(): Zet het relais op AAN. • off(): Zet het relais op UIT.
Servo	Functies en opdrachten voor het aansturen van servomotoren. <ul style="list-style-type: none"> • set_position(pos): Stelt de 'sweep' (zwaai) servo positie in binnen een bereik van -90 tot +90. • zero(): Stelt de sweep (zwaai) servo in op de nulpositie.
Squarewave	Functies en opdrachten voor het genereren van een blok golf. <ul style="list-style-type: none"> • set(frequency,duty,time): Stelt de output blok golf in met een standaard arbeidscyclus (duty cycle) van 50% (als de duty-factor niet gespecificeerd is) en een outputfrequentie gespecificeerd door frequentie". De frequentie kan een waarde van 1 tot 500 Hz hebben. De arbeidscyclus kan, indien gespecificeerd, van 0 tot 100% zijn. • off(): Schakelt de blok golf uit.
Digital Out	Interfaces voor het aansturen van een digitale uitvoer. <ul style="list-style-type: none"> • set(val): Stelt de digitale uitgang in op de waarde gespecificeerd door "val" (0 of 1). • on(): Stelt de status van de digitale uitvoer in op hoog (1). • off(): Stelt de status van de digitale uitvoer in op laag (0).
BB Port	Biedt opdrachten en functies voor het programmeren van de TI-RGB-array. Zie de details hierboven.

Opdrachten

object	matrix
<code>sleep(seconds)</code>	Pauzeert de uitvoering van een programma gedurende een ingesteld aantal seconden. Geïmporteerd uit de 'time'-module.
<code>text_at(row,"text","align")</code>	Geeft de gespecificeerde "tekst" in het plot-gebied weer volgens de gespecificeerde "uitlijning". Onderdeel van de <code>ti_plotlib</code> -module.
<code>cls()</code>	Wist het Shell-scherm voor plotten. Onderdeel van de <code>ti_plotlib</code> -module.
<code>while get_key() != "esc":</code>	Voert de opdrachten uit in de "while"-lus totdat de "esc"-toets wordt ingedrukt.
<code>get_key()</code>	Geeft een string (tekenreeks) die de ingedrukt toets vertegenwoordigt. De '1'-toets geeft "1", 'esc' geeft "esc", enzovoort. Wanneer <code>get_key()</code> wordt aangeroepen zonder parameters, geeft het onmiddellijk een resultaat. Wanneer <code>get_key()</code> wordt aangeroepen met een parameter, bijvoorbeeld <code>get_key(1)</code> - wacht het tot een toets wordt ingedrukt. Onderdeel van de <code>ti_system</code> -module.

Poorten

Dit zijn de in- en uitpoorten die beschikbaar zijn op de TI-Innovator™ Hub.

object
UIT 1
UIT 2
UIT 3
IN 1
IN 2
IN 3
BB 1
BB 2
BB 3

object

BB 4

BB 5

BB 6

BB 7

BB 8

BB 9

BB 10

I2C

TI Rover-menu

Opmerking: Bij het maken van een nieuw programma dat deze module gebruikt, wordt aanbevolen om het programmatype **Rover Programmeren** te gebruiken. Dit zorgt ervoor dat alle relevante modules worden geïmporteerd.

object	matrix
import ti_rover as rv	Importeert alle methoden (functies en opdrachten) van de ti_rover module in de "rv" naamruimte. Als gevolg daarvan worden functie- en opdrachtnamen die uit de menu's zijn geplakt voorafgegaan door "rv".

Drive

object	matrix
forward(distance)	Beweegt de Rover vooruit over de gespecificeerde afstand in roostereenheden.
backward(distance)	Beweegt de rover achteruit over de gespecificeerde afstand in roostereenheden.
left(angle_degrees)	Draait de Rover naar links volgens de gespecificeerde hoek in graden.
right(angle_degrees)	Draait de Rover naar rechts volgens de gespecificeerde hoek in graden.
stop()	Stopt elke huidige beweging onmiddellijk.
stop_clear()	Stopt elke huidige beweging onmiddellijk en wist alle lopende opdrachten.
resume()	Hervat de verwerking van opdrachten.
stay(time)	Rover blijft 'op de plaats' gedurende een optioneel gespecificeerd aantal seconden. Als er geen tijd is gespecificeerd, blijft de Rover 30 seconden lang op zijn plaats staan.
to_xy(x,y)	Beweegt Rover naar positie met coördinaten (x, y) op virtueel rooster.
to_polar(r,theta_degrees)	Beweegt Rover naar positie met poolcoördinaten (r, theta) op virtueel rooster. De hoek wordt gespecificeerd in graden.
to_angle(angle,"unit")	Draait Rover naar de gespecificeerde hoek in het virtuele rooster. De hoek is bepaald ten opzichte van een hoek van nul

object	matrix
	graden die omlaag wijst vanaf de x-as in het virtuele rooster.

Drive > Drive with Options

object	matrix
forward_time(time)	Beweegt Rover vooruit gedurende de gespecificeerde tijdsduur.
backward_time(time)	Beweegt Rover achteruit gedurende de gespecificeerde tijdsduur.
forward(distance,"unit")	Beweegt Rover vooruit met de standaardsnelheid over de gespecificeerde afstand. De afstand kan worden gespecificeerd in roostereenheden, meters of wielomwentelingen.
backward(distance,"unit")	Beweegt Rover achteruit met de standaardsnelheid over de gespecificeerde afstand. De afstand kan worden gespecificeerd in roostereenheden, meters of wielomwentelingen.
left(angle,"unit")	Draait Rover naar links volgens gespecificeerde hoek. De hoek kan in graden, radialen of decimale graden gegeven zijn.
right(angle,"unit")	Draait Rover naar rechts volgens de gespecificeerde hoek. De hoek kan in graden, radialen of decimale graden gegeven zijn.
forward_time(time,speed,"rate")	Beweegt Rover vooruit gedurende de gespecificeerde tijdsduur met de gespecificeerde snelheid. De snelheid kan worden gespecificeerd in roostereenheden/s, meters/s of wielomwentelingen.
backward_time(time,speed,"rate")	Beweegt de rover achteruit gedurende de gespecificeerde tijd met de gespecificeerde snelheid. De snelheid kan worden gespecificeerd in

object	matrix
	roostereenheden/s, meters/s of wielomwentelingen.
forward(distance,"unit",speed,"rate")	<p>Beweegt de rover vooruit over de gespecificeerde afstand met de gespecificeerde snelheid.</p> <p>De afstand kan worden gespecificeerd in roostereenheden, meters of wielomwentelingen.</p> <p>De snelheid kan worden gespecificeerd in roostereenheden/s, meters/s of wielomwentelingen.</p>
backward(distance,"unit",speed,"rate")	<p>Beweegt de rover achteruit over de gespecificeerde afstand met de gespecificeerde snelheid.</p> <p>De afstand kan worden gespecificeerd in roostereenheden, meters of wielomwentelingen.</p> <p>De snelheid kan worden gespecificeerd in roostereenheden/s, meters/s of wielomwentelingen.</p>

invoer

object	matrix
ranger_measurement()	Leest de ultrasone afstandssensor aan de voorkant van de Rover en geeft de huidige afstand in meters.
color_measurement()	<p>Geeft een waarde van 1 tot en met 9, die de dominante kleur aangeeft die "gezien" wordt door de kleurinvoersensor van de Rover.</p> <p>1 = rood 2 = groen 3 = blauw 4 = cyaan 5 = magenta 6 = geel 7 = zwart 8 = grijs 9 = wit</p>
red_measurement()	Geeft een waarde tussen 0 en 255 die het door de kleurinvoersensor waargenomen rode niveau

object	matrix
	aan geeft.
green_measurement()	Geeft een waarde tussen 0 en 255 die het door de kleurinvoersensor waargenomen groene niveau aan geeft.
blue_measurement()	Geeft een waarde tussen 0 en 255 die het door de kleurinvoersensor waargenomen blauwe niveau aan geeft .
gray_measurement()	Geeft een waarde tussen 0 en 255 die het door de kleurinvoersensor waargenomen grijsniveau aan geeft.
encoders_gyro_measurement()	Geeft een lijst met waarden die de tellingen van de linker- en rechterwiel encoder en de huidige gyro-kop bevat.
gyro_measurement()	Geeft een waarde die de huidige gyro-aflezing vertegenwoordigt, waaronder de drift, in de graden.
ranger_tijd()	Geeft de tijd die het ultrasone signaal van de TI-Rover ranger nodig heeft om het object te bereiken (de "tijd van de vlucht").

Uitvoer

object	matrix
color_rgb(r,g,b)	Stelt de kleur van de RGB LED van de Rover in op de specifieke waarden van rood, groen en blauw.
color_blink(frequency,time)	Stelt de knipperfrequentie en-duur in voor de geselecteerde kleur.
color_off()	Zet de RGB LED van de Rover uit.
motor_left(speed,time)	<p>Stelt het vermogen van de linkermotor in op de gespecificeerde waarde voor de gespecificeerde duur.</p> <p>De snelheid ligt in het bereik -255 tot 255 waarbij 0 stilstand betekent. De positieve snelheidswaarden zijn linksom (tegen de klok in), en de negatieve snelheidswaarden zijn rechtsom (met de klok mee).</p> <p>De optionele tijdparameter, indien gespecificeerd, heeft een geldig bereik van 0,05 t/m 655,35 seconden. Indien</p>

object	matrix
	niet gespecificeerd, wordt een standaardwaarde van 5 seconden gebruikt.
motor_right(speed,time)	<p>Stelt het vermogen van de linkermotor in op de gespecificeerde waarde voor de gespecificeerde duur.</p> <p>De snelheid ligt in het bereik -255 tot 255 waarbij 0 stilstand betekent. De positieve snelheidswaarden zijn linksom (tegen de klok in), en de negatieve snelheidswaarden zijn rechtsom (met de klok mee).</p> <p>De optionele tijdparameter, indien gespecificeerd, heeft een geldig bereik van 0,05 t/m 655,35 seconden. Indien niet gespecificeerd, wordt een standaardwaarde van 5 seconden gebruikt.</p>
motors("ldir",left_val,"rdir",right_val,time)	<p>Stelt het linker- en rechterwiel in op de gespecificeerde snelheden, voor een optionele tijdsduur in seconden.</p> <p>De snelheid (left_val, right_val) heeft waarden die liggen in het bereik 0 tot 255 waarbij 0 stilstand is. De parameters ldir en rdir specificeren de richting van de draaiing: CW- (met de klok mee) of CCW (tegen de klok in) van de betreffende wielen.</p> <p>De optionele tijdparameter, indien gespecificeerd, heeft een geldig bereik van 0,05 t/m 655,35 seconden. Indien niet gespecificeerd, wordt een standaardwaarde van 5 seconden gebruikt.</p>

Pad

object	matrix
waypoint_xythdrn()	Leest de x-coördinaat, y-coördinaat, tijd, koers (richting), afgelegde afstand, aantal toeren (omwentelingen) van het wiel, opdracht nummer van het huidige knooppunt (waypoint) Geeft een lijst met al deze waarden als elementen.
waypoint_prev	Leest de x-coördinaat, y-coördinaat, tijd, koers (richting),

object	matrix
	afgelegde afstand, aantal toeren (omwentelingen) van het wiel, opdrachtnummer van het vorige knooppunt (waypoint).
waypoint_eta	Geeft de geschatte tijd om naar een knooppunt (waypoint) te rijden.
path_done()	Geeft een waarde van 0 of 1, afhankelijk van of de Rover beweegt (0) of klaar is met alle bewegingen (1).
pathlist_x()	Geeft een lijst met X-waarden vanaf het begin tot en met de X-waarde van het huidige knooppunt (waypoint).
pathlist_y()	Geeft een lijst met Y-waarden vanaf het begin tot en met de Y-waarde van het huidige knooppunt (waypoint).
pathlist_time()	Geeft een lijst met de tijd in seconden vanaf het begin tot en met de waarde van de tijd van het huidige knooppunt (waypoint).
pathlist_heading()	Geeft een lijst met de koersen vanaf het begin tot en met de koerswaarde van het huidige knooppunt (waypoint).
pathlist_distance()	Geeft een lijst met afgelegde afstanden vanaf het begin tot en met de waarde van de afstand van het huidige knooppunt (waypoint).
pathlist_revs()	Geeft een lijst met het aantal gemaakte omwentelingen vanaf het begin tot en met de waarde van het aantal omwentelingen van het huidige knooppunt (waypoint).
pathlist_cmdnum()	Geeft een lijst met opdrachtnummers voor het pad.
waypoint_x ()	Geeft de x-coördinaat van het huidige knooppunt (waypoint).
waypoint_y()	Geeft de y-coördinaat van het huidige knooppunt (waypoint).
waypoint_time()	Geeft de reistijd vanaf het vorige tot het huidige knooppunt (waypoint).
waypoint_heading()	Geeft de absolute koers van het huidige knooppunt (waypoint).
waypoint_distance()	Geeft de afgelegde afstand tussen het vorige en het huidige knooppunt (waypoint).
waypoint_revs()	Geeft het aantal omwentelingen dat nodig is om te verplaatsen vanaf het vorige tot het huidige knooppunt (waypoint).

Instellingen

object	matrix
units/s	Optie voor snelheid in roostereenheden per seconde.

object	matrix
m/s	Optie voor snelheid in meters per seconde.
revs/s	Optie voor snelheid in wielomwentelingen per seconde.
Eenheden	Optie voor afstand in roostereenheden.
M	Optie voor afstand in meters.
revs	Optie voor afstand in wielomwentelingen.
graden	Optie voor draaiing in graden.
radialen	Optie voor draaiing in radialen.
gradians	Optie voor draaiing in decimale graden.
Rechtsom	Mogelijkheid om wielrichting te specificeren.
Linksom	Mogelijkheid om wielrichting te specificeren.

Opdrachten

Deze opdrachten zijn een verzameling functies en opdrachten zowel van andere modules als van de TI Rover module.

object	matrix
sleep(seconds)	Pauzeert de uitvoering van een programma gedurende een ingesteld aantal seconden. Geïmporteerd uit de "time"-module.
text_at(row,"text", "align")	Geeft tekst" weer in plotgebied bij gespecificeerde "uitlijning". Geïmporteerd uit de ti_plotlib-module.
cls()	Wist het Shell-scherm voor plotten. Geïmporteerd uit de ti_plotlib-module.
while get_key() != "esc":	Voert de opdrachten uit in de "while"-lus totdat de "esc"-toets wordt ingedrukt.
wait_until_done()	Pauzeert het programma totdat de Rover het huidige commando heeft voltooid. Dit is een handige manier om niet-Rover opdrachten te synchroniseren met de beweging van de Rover.
while not path_done()	Voert de opdrachten in de "while"-lus uit, totdat de rover klaar is met alle bewegingen. De functie path_done() geeft een waarde van 0 of 1, afhankelijk van of de Rover beweegt (0) of klaar is met alle bewegingen (1).

object	matrix
position(x,y)	Stelt de positie van de Rover in op het virtuele rooster op de gespecificeerde x, y-coördinaat.
position(x,y,heading,"unit")	Stelt de positie van de Rover in op het virtuele rooster op de gespecificeerde x, y-coördinaat, en de virtuele koers, ten opzichte van de virtuele x-as, wordt ingesteld als een koers wordt gespecificeerd (in de gespecificeerde eenheden voor hoeken). Positieve hoeken van 0 tot 360 graden worden verondersteld tegen de klok in te liggen vanaf de positieve x-as. Negatieve hoeken van 0 tot -360 graden worden verondersteld met de klok mee te liggen vanaf de positieve x-as.
grid_origin()	Stelt RV in op het oorspronkelijke punt van (0,0) van het huidige raster.
grid_m_unit(scale_value)	Stelt de virtuele rasterafstand in meter per eenheid (m/eenheid) in op de gespecificeerde waarde. 0,1 is de standaard m/eenheid en vertaalt zich naar 1 eenheid = 100 mm of 10 cm of 1 dm of 0,1 m. Het bereik van geldige scale_value is van 0,01 t/m 10,0.
path_clear()	Wist alle reeds bestaande pad- of knooppuntinformatie.
zero_gyro()	Zet de Rover gyro terug een hoek van 0, 0 en wist de tellingen van de linker- en rechterwielenencoder.

Complex Math-menu

Dit submenu bevindt zich onder **More modules** (Meer modules).

object	matrix
uit wiskunde import*	Importeert alle opdrachten uit de cwis-module.
complex(real, imag)	Geeft een complex getal.
rect(modulus, argument)	Converteert poolcoördinaten naar de rechthoekige vorm van een complex getal.
.real	Geeft het reële deel van het complexe getal.
.imag	Geeft een denkbeeldig deel van een complex getal.
polar()	Converteert de rechthoekige vorm van een complex getal naar poolcoördinaten.
phase()	Geeft de fase van een complex getal.
exp()	Geeft e^{**x} .
cos()	Geeft de cosinus van een complex getal.
sin()	Geeft de sinus van een complex getal.
log()	Geeft de natuurlijke logaritme (ln) van een complex getal.
log10()	Geeft het logaritme met grondtal 10 (log) van een complex getal.
sqrt()	Geeft de wortel van een complex getal.

Time-menu

Dit submenu bevindt zich onder **More modules** (Meer modules).

object	matrix
vanuit time import*	Importeert alle methoden (functies en opdrachten) van de tijdmodule.
sleep(seconds)	Pauzeert de uitvoering van een programma gedurende een ingesteld aantal seconden.
clock()	Geeft de huidige processortijd als een getal met drijvende komma, uitgedrukt in seconden.
localtime()	Zet een tijd uitgedrukt in seconden sinds 1 januari 2000 om in een negen-tupel met jaar, maand, dag van de maand, uur, minuut, seconde, dag van de week, dag van het jaar, dag en zomertijd (DST) vlag. Als het optionele argument (seconden) niet is gespecificeerd, wordt de real-time klok gebruikt.
ticks_cpu()	Geeft een processorspecifieke toenemende milliseconde-teller met willekeurige referentiepunt. Gebruik ticks_ms() om tijd consistent te meten voor verschillende systemen.
ticks_diff()	Meet de periode tussen opeenvolgende aanroepen naar ticks_cpu() of ticks_ms(). Deze functie moet niet worden gebruikt om willekeurig lange tijdsperiodes te meten.

Ti System-menu

Dit submenu bevindt zich onder **More modules** (Meer modules).

Opmerking: Bij het maken van een nieuw programma dat deze module gebruikt, wordt aanbevolen om het programmatype **Delen van gegevens** te gebruiken. Dit zorgt ervoor dat alle relevante modules worden geïmporteerd.

object	matrix
<code>from ti_system import *</code>	Importeert alle methoden (functies en opdrachten) uit de <code>ti_systeemmodule</code> .
<code>recall_value("name")</code>	Roept een vooraf gedefinieerde OS-variabele (waarde) genaamd "naam" opnieuw aan.
<code>store_value("name",value)</code>	Slaat een Pythonvariabele (waarde) op in een OS-variabele met de naam "naam".
<code>recall_list("name")</code>	Roept een vooraf gedefinieerde OS-lijst met de naam "naam" opnieuw aan.
<code>store_list("name",list)</code>	Slaat een pythonlijst (lijst) op in een OS lijstvariabele met de naam "naam".
<code>eval_function("name",value)</code>	Werkt een vooraf gedefinieerde OS-functie uit voor de gespecificeerde waarde.
<code>get_platform()</code>	Geeft "rm" voor rekenmachine en "dt" voor desktop.
<code>get_key()</code>	Geeft een string (tekenreeks) die de ingedrukt toets vertegenwoordigt. De '1'-toets geeft "1", 'esc' geeft "esc", enzovoort. Wanneer <code>get_key()</code> wordt aangeroepen zonder parameters, geeft het onmiddellijk een resultaat. Wanneer <code>get_key()</code> wordt aangeroepen met een parameter, bijvoorbeeld <code>get_key(1)</code> - wacht het tot een toets wordt ingedrukt.
<code>get_mouse()</code>	Geeft de muiscoördinaten als een tuple met 2 elementen, ofwel de pixelpositie op het canvas ofwel (-1, -1) als die zich buiten het canvas bevindt.
<code>while get_key() != "esc":</code>	Voert de opdrachten uit in de "while"-lus totdat de "esc"-toets wordt ingedrukt.
<code>clear_history()</code>	Wist de Shell-geschiedenis.
<code>get_time_ms()</code>	Geeft de tijd in milliseconden met een precisie van milliseconden. Deze functionaliteit kan worden gebruikt om een tijdsduur te berekenen in plaats van de werkelijke kloktijd te bepalen.

Tl Draw-menu

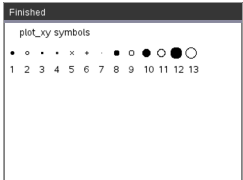
Dit submenu bevindt zich onder **More modules** (Meer modules).

Opmerking: Bij het maken van een nieuw programma dat deze module gebruikt, wordt aanbevolen om het programmatype] **Meetkundige afbeeldingen** te gebruiken Dit zorgt ervoor dat alle relevante modules worden geïmporteerd.

object	matrix
<code>from ti_draw import *</code>	Importeert alle methoden (functies en opdrachten) uit de ti_draw module.

Vorm

object	matrix
<code>draw_line()</code>	Tekent een lijn vanaf de gespecificeerde x1, y1 coördinaat naar x2, y2.
<code>draw_rect()</code>	Tekent een rechthoek die beginnend op de gespecificeerde x, y-coördinaat met de gespecificeerde breedte en hoogte.
<code>fill_rect()</code>	Tekent een rechthoek beginnend bij de gespecificeerde x, y coördinaat met de gespecificeerde breedte en hoogte en die is opgevuld met de gespecificeerde kleur (met behulp van <code>set_color</code> of zwart indien niet gedefinieerd).
<code>draw_circle()</code>	Tekent een cirkel beginnend bij het gespecificeerde x, y-middelpunt met de gespecificeerde straal.
<code>fill_circle()</code>	Tekent een cirkel beginnend bij het gespecificeerde x, y-middelpunt met de gespecificeerde straal en opgevuld met de gespecificeerde kleur (met behulp van <code>set_color</code> of zwart indien niet gedefinieerd).
<code>draw_text()</code>	Tekent een tekststring beginnend bij de gespecificeerde x, y-coördinaat.
<code>draw_arc()</code>	Tekent een boog beginnend bij de gespecificeerde x, y-coördinaat met de gespecificeerde breedte, hoogte en hoeken.
<code>fill_arc()</code>	Tekent een boog beginnend bij de gespecificeerde x, y coördinaat met de gespecificeerde breedte, hoogte en hoeken en opgevuld met de gespecificeerde kleur (met behulp van <code>set_color</code> of zwart indien niet gedefinieerd).
<code>draw_poly()</code>	Tekent een veelhoek met behulp van de gespecificeerde x-lijst, y-lijstwaarden.
<code>fill_poly()</code>	Tekent een veelhoek met behulp van de gespecificeerde x-lijst, y-lijstwaarden opgevuld met de gespecificeerde kleur (met behulp van <code>set_color</code> of zwart indien niet gedefinieerd).

object	matrix
plot_xy()	<p>Tekent een vorm met behulp van de gespecificeerde x, y-coördinaat en het gespecificeerde nummer van 1-13 die verschillende vormen en symbolen vertegenwoordigen (zie hieronder).</p> 

Besturing

object	matrix
clear()	Wist de volledige schermweergave. Kan worden gebruikt met de parameters x, y, breedte en hoogte om een bestaande rechthoek te wissen.
clear_rect()	Wist de rechthoek op de gespecificeerde x, y-coördinaat met de gespecificeerde breedte en hoogte.
set_color()	Stelt de kleur in van de vorm(en) die in het programma volgen totdat een andere kleur is ingesteld.
set_pen()	Stelt de gespecificeerde dikte en stijl van de rand in bij het tekenen van vormen (niet van toepassing bij het gebruik van vulopdrachten).
set_window()	<p>Hiermee stelt u de grootte van het venster in waarin eventuele vormen worden getekend.</p> <p>Deze functie is handig om de grootte van het venster aan te passen aan de gegevens of om de oorsprong (0, 0) van het tekencanvas te wijzigen.</p>
get_screen_dim()	Geeft de xmax en ymax van de schermafmetingen.
use_buffer()	Maakt een off-screen buffer mogelijk om het tekenen te versnellen.
paint_buffer()	<p>Geeft de gebufferde teken-output weer.</p> <p>De opdrachten use_buffer() en paint_buffer() zijn handig in gevallen waarin het weergeven van meerdere objecten op het scherm vertragingen kan veroorzaken.</p>

Notities

- De standaardconfiguratie heeft (0,0) in de linkerbovenhoek van het scherm. De positieve x-as wijst naar rechts en de positieve y-as wijst naar beneden Dit kan worden aangepast met de functie `set_window()`.
- Opdrachten in de tekenmodule `tj_draw` zijn alleen beschikbaar op de rekenmachine en in de rekenmachineweergave op het bureaublad

Tl Image-menu

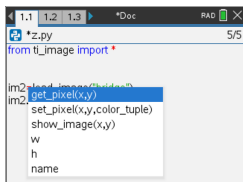
Dit submenu bevindt zich onder **More modules** (Meer modules).

Opmerking: Bij het maken van een nieuw programma dat deze module gebruikt, wordt aanbevolen om het programmatype **Image Processing** te gebruiken. Dit zorgt ervoor dat alle relevante modules worden geïmporteerd.

object	matrix
<code>from ti_image import *</code>	Importeert alle opdrachten uit de <code>ti_image</code> module.
<code>new_image(width,height,(r,g,b))</code>	Creëert een nieuwe afbeelding met de gespecificeerde breedte en hoogte voor gebruik in het Python-programma. De kleur van de nieuwe afbeelding is gedefinieerd door de waarden <code>(r, g, b)</code> .
<code>load_image("name")</code>	Hiermee wordt de afbeelding geladen die wordt gespecificeerd door <code>naam</code> , voor gebruik in het Python-programma. De afbeelding moet deel uitmaken van het TNS-document in de toepassing Notities of Grafieken. De <code>"naam"</code> -prompt geeft de afbeeldingsnamen weer (als ze eerder zijn benoemd) of een nummer dat hun invoegvolgorde aangeeft.
<code>copy_image(image)</code>	Hiermee wordt een kopie gemaakt van de afbeelding die is gespecificeerd door de variabele <code>"afbeelding"</code> .

Functies en opdrachten van het afbeeldingsobject

Aanvullende functies en opdrachten met betrekking tot de afbeeldingsobjecten zijn beschikbaar in de Editor en Shell door de variabelenaam in te typen gevolgd door een `.` (punt).



- **get_pixel(x,y):** Haalt de `(r, g, b)` waarde op van de pixel op positie gedefinieerd door het `(x, y)` coördinatenpaar.

```
px_val = get_pixel(100,100)
print(px_val)
```
- **set_pixel(x,y,color_tuple):** Stelt de pixel op positie `(x, y)` in op de kleur die is gespecificeerd in de `kleur_tuple`.

```
set_pixel(100,100, (0,0,255))
```

Stelt de pixel op (100,100) in op de kleur (0,0,255).

- **show_image(x,y):** Geeft de afbeelding weer met de linkerbovenhoek op positie (x, y).
- **w, h, name:** Haalt de parameters breedte, hoogte en naam van het afbeelding op.

Voorbeeld

```
from ti_image import *

# An image has been previously inserted into the TNS document in a
Notes application and named "bridge"
im1=load_image("bridge")
px_val = im1.get_pixel(100,100)
print(px_val)

# Set the pixel at 100,100 to blue (0,0,255)
im1.set_pixel(100,100, (0,0,255))
new_px = im1.get_pixel(100,100)
print(new_px)

# Print the width, height and name of the image
print(im1.w, im1.h, im1.name)
```

Variables-menu

Opmerking: Deze lijsten bevatten geen variabelen die in andere TI-Nspire™-apps zijn gedefinieerd.

object	matrix
Vars:Current Program	(Alleen Editor) Geeft een lijst weer met globale functies en variabelen die in het huidige programma zijn gedefinieerd
Vars: Last Run Program	(Alleen shell) Geeft een lijst weer met algemene functies en variabelen die zijn gedefinieerd in het laatst uitgevoerde programma
Vars: Alles	(Alleen shell) Geeft een lijst met globale functies en variabelen weer uit zowel het laatst uitgevoerde programma als alle geïmporteerde modules

Bijlage

Python-trefwoorden	51
Python-sleuteltoewijzing	52
Voorbeelden van Python-programma's	54

Python-trefwoorden





















De volgende trefwoorden zijn ingebouwd in de TI-Nspire™ Python-omgeving.

False	elif	lambda
None	else	nonlocal
True	except	not
and	finally	or
as	for	pass
assert	from	raise
break	global	return
class	if	try
continue	import	while
def	in	with
del	is	yield

Python-sleuteltoewijzing

Bij het invoeren van programmacode in de Editor of in de Shell is het toetsenbord zo ontworpen dat het de juiste Python-bewerkingen of open menu's plakt, voor het eenvoudig invoeren van functies, trefwoorden, methoden, operatoren, enz.

Toets	Toewijzing
<code>var</code>	Opent het menu Variabelen
<code>sto→</code>	Plakt het =-teken
<code>del</code>	Wist het teken links van de cursor
<code>clear</code>	Geen actie
<code>=</code>	Plakt het =-teken
<code>[#>]</code>	Plakt de geselecteerde symbolen: <ul style="list-style-type: none">• >• <• !=• >=• <=• ==• en• of• niet• • &• ~
<code>trig</code>	Plakt de geselecteerde functie: <ul style="list-style-type: none">• sin• cos• tan• atan2• asin• acos• atan
<code>[?]</code>	Geeft hints weer
<code>:=</code>	Plakt :=
<code>^</code>	Plakt **
<code>[x→]</code>	Geen actie
<code>x²</code>	Plakt **2

Toets	Toewijzing
	Plakt sqrt()
	Plakt vermenigvuldigingsteken (*)
	Plakt een dubbel aanhalingsteken (")
	Plakt deeltteken (/)
	Geen actie
	Plakt exp()
	Plakt log()
	Plakt 10**
	Plakt log (waarde, basis)
	Plakt (
	Plakt **
	Plakt []
	Plakt { }
	Plakt minteken (-)
	Voegt een nieuwe regel toe na de huidige regel
	Plakt E
	Plakt de geselecteerde symbolen: <ul style="list-style-type: none"> • ? • ! • \$ • ° • ' • % • " • : • ; • _ • \ • #
	Plakt "pi"
	Gedrag bestaande vlag
	Voegt een nieuwe regel toe na de huidige regel

Voorbeelden van Python-programma's

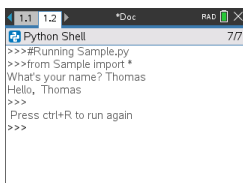
Gebruik de volgende voorbeeldprogramma's om vertrouwd te raken met Python-methoden. Ze zijn ook beschikbaar in het bestand **Aan de slag met Python.tns**, dat zich bevindt in de **Voorbeelden** map.

Opmerking: Als u regels uit een voorbeeldprogramma kopieert en plakt die tab-inspringindicatoren bevatten (••) naar de TI-Nspire™-software, moet u deze vervangen door de werkelijke tabs om in te springen.

Hallo

```
# This program asks for your name and uses
# it in an output message.
# Run the program here by typing "Ctrl R"
```

```
name=input("What's your name? ")
print("Hello, ", name)
print("\n Press ctrl+R to run again")
```



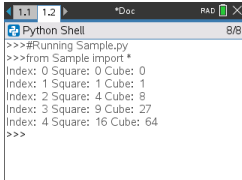
The screenshot shows a TI-Nspire Python Shell window titled "Python Shell" with a file icon and "7/7" in the top right corner. The window contains the following text:

```
>>>#Running Sample.py
>>>from Sample import *
What's your name? Thomas
Hello, Thomas
>>>
Press ctrl+R to run again
>>>
```

Voorbeeld van lus

```
# This program uses a "for" loop to calculate
# the squares and cubes of the first 5 numbers
# 0,1,2,3,4
# Note: Python starts counting at 0
```

```
for index in range(5):
    **square = index**2
    **cube = index**3
    **print("Index: ", index, "Square: ", square,
    ****"Cube: ", cube)
```



The screenshot shows a Python Shell window with the following output:

```
>>>#Running Sample.py
>>>from Sample import *
Index: 0 Square: 0 Cube: 0
Index: 1 Square: 1 Cube: 1
Index: 2 Square: 4 Cube: 8
Index: 3 Square: 9 Cube: 27
Index: 4 Square: 16 Cube: 64
>>>
```

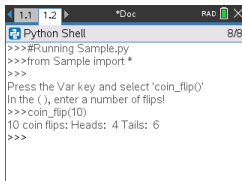
Kop of munt

```
# Use random numbers to simulate a coin flip
# We will count the number of heads and tails
# Run the program here by typing "Ctrl R"

# Import all the functions of the "random" module
from random import *

# n is the number of times the die is rolled
def coin_flip(n):
    **heads = tails = 0
    **for i in range(n):
# Generate a random integer - 0 or 1
# "0" means head, "1" means tails
    **side=randint(0,1)
    **if (side == 0):
    *****heads = heads + 1
    **else:
    *****tails = tails + 1
# Print the total number of heads and tails
    **print(n, "coin flips: Heads: ", heads, "Tails: ", tails)

print("\nPress the Var key and select 'coin_flip()')")
print("In the ( ), enter a number of flips!")
```



The screenshot shows a Python Shell window titled "Python Shell" with a file path of "*Doc: Sample.py" and a page indicator "8/8". The shell contains the following text:

```
>>>#Running Sample.py
>>>from Sample import *
>>>
Press the Var key and select 'coin_flip()'
In the ( ), enter a number of flips!
>>>coin_flip(10)
10 coin flips: Heads: 4 Tails: 6
>>>
```

Plotten

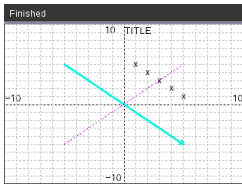
```
# Plotting example
import ti_plotlib as plt

# Set up the graph window
plt.window(-10,10,-10,10)
plt.axes("on")
plt.grid(1,1,"dashed")
# Add leading spaces to position the title
plt.title("          TITLE")

# Set the pen style and the graph color
plt.pen("medium","solid")
plt.color(28,242,221)
plt.line(-5,5,5,-5,"arrow")

plt.pen("thin","dashed")
plt.color(224,54,243)
plt.line(-5,-5,5,5,"")

# Scatter plot from 2 lists
plt.color(0,0,0)
xlist=[1,2,3,4,5]
ylist=[5,4,3,2,1]
plt.scatter(xlist,ylist, "x")
```



Tekenen

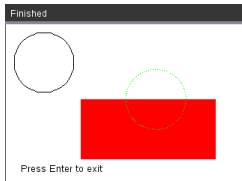
```
from ti_draw import *

# (0,0) is in top left corner of screen
# Let's draw some circles and squares
# Circle with center at (50,50) and radius 40
draw_circle(50,50,40)

# Set color to red (255,0,0) and fill a rectangle of
# of width 180, height 80 with top left corner at
# (100,100)
set_color(255,0,0)
fill_rect(100,100,180,80)

# Set color to green and pen style to "thin"
# and "dotted".
# Then, draw a circle with center at (200,100)
# and radius 40
set_color(0,255,0)
set_pen("thin","dotted")
draw_circle(200,100,40)

set_color(0,0,0)
draw_text(20,200,"Press Enter to exit")
```



Beeld

```
# Image Processing
#=====
from ti_image import *
from ti_draw import *
#=====

# Load and show the 'manhole_cover' image
# It's in a Notes app
# Draw a circle on top
im1=load_image("manhole_cover")
im1.show_image(0,0)
set_color(0,255,0)
set_pen("thick","dashed")
draw_circle(140,110,100)
```



Hub

Dit programma gebruikt Python voor het besturen van de TI-Innovator™ Hub, een programmeerbare microcontroller. Als u het programma uitvoert zonder een TI-Innovator™ Hub aan te sluiten, wordt een foutmelding weergegeven.

Ga naar www.education.ti.com voor meer informatie over TI-Innovator™ Hub.

```
##### Import Section #####
from ti_hub import *
from math import *
from random import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
##### End of Import Section #####

print("Connect the TI-Innovator Hub and hit 'enter'")
input()
print("Blinking the RGB LED for 4 seconds")
# Set the RGB LED on the Hub to purple
color.rgb(255,0,255)

# Blink the LED 2 times a second for 4 seconds
color.blink(2,4)

sleep(5)

print("The brightness sensor reading is: ", brightness.measurement())

# Generate 10 random colors for the RGB LED
# Play a tone on the Hub based on the random
# color
print("Generate 10 random colors on the Hub & play a tone")
for i in range(10):
    *r=randint(0,255)
    *b=randint(0,255)
    *g=randint(0,255)
    *color.rgb(r,g,b)
    *sound.tone((r+g+b)/3,1)
    *sleep(1)

color.off()
```

Algemene informatie

Online Help

education.ti.com/eguide

Selecteer uw land voor meer productinformatie.

Neem contact op met TI Ondersteuning

education.ti.com/ti-cares

Selecteer uw land voor technische en andere ondersteuningsbronnen.

Service- en garantie-informatie

education.ti.com/warranty

Selecteer uw land voor meer informatie over de duur en voorwaarden van de garantie of over de productservice.

Beperkte garantie. Deze garantie heeft geen invloed op uw wettelijke rechten.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243