

# TI-Nspire™ Python

## Vejledning til programmering

## **Vigtige oplysninger**

Med mindre andet er anført i den Licens, som følger med programmet, stiller Texas Instruments ingen garantier, udtrykte eller underforståede, herunder, men ikke begrænset til enhver underforstået garanti om salgbarhed og egnethed til et bestemt formål for nogen programmer eller bøger, og stiller udelukkende disse materialer til rådighed "som de forefindes." Texas Instruments kan under ingen omstændigheder holdes ansvarlige for nogen særlige, indirekte, påløbne eller følgeskader i forbindelse med eller som følge af købet eller anvendelsen af disse materialer, og det eneste erstatningsansvar, Texas Instruments kan pådrage sig, uanset handlingen, kan ikke overstige købsprisen for dette produkt, som den er angivet i licensen. Texas Instruments kan endvidere ikke holdes ansvarlig for nogen form for krav i forbindelse med nogen tredjeparts anvendelse af disse materialer.

© 2021 Texas Instruments Incorporated

"Python" og Python-logoer er varemærker eller registrerede varemærker tilhørende Python Software Foundation, der bruges af Texas Instruments Incorporated med tilladelse fra fonden.

De faktiske produkter kan variere let fra de viste billeder.

# Indholdsfortegnelse

<b>Kom godt i gang med Python-programmering</b> .....	<b>1</b>
Python-moduler .....	1
Sådan installeres et Python-program som et modul .....	2
<b>Python-arbejdsområder</b> .....	<b>3</b>
Python Editor .....	3
Python Shell .....	6
<b>Python-menukort</b> .....	<b>10</b>
Menuen Handlinger .....	11
Kørselsmenu .....	12
Værktøjsmenu .....	13
Menuen Rediger .....	14
Menuen Indbyggede .....	15
Matematikmenu .....	18
Tilfældigheds-menu .....	20
TI Menuen PlotLib .....	21
Menuen TI Hub .....	23
Menuen TI Rover .....	31
Menu for kompleks matematik .....	39
Tidsmenu .....	40
TI systemmenu .....	41
TI tegnemenu .....	42
TI-billedmenu .....	44
Menuen Variabel .....	46
<b>Tillæg</b> .....	<b>47</b>
Python Nøgleord .....	48
Python-tastaturoversigt .....	49
Prøveprogrammer fra Python .....	51
<b>Generelle oplysninger</b> .....	<b>58</b>

# Kom godt i gang med Python-programmering

Ved at bruge Python med TI-Nspire™-produkter kan du:

- tilføje Python-programmer til TNS-filer
- oprette Python-programmer ved hjælp af skabeloner
- interagere og dele data med andre TI-Nspire™ apps
- interagere med TI-Innovator™ Hub og TI-Innovator™ Rover

TI-Nspire™ Python-implementeringen er baseret på MicroPython, som er en lille delmængde af Python 3-standardbiblioteket, der er designet til at køre på mikrocontrollere. Den originale MicroPython-implementering er blevet tilpasset til brug af TI.

**Bemærk:** Nogle numeriske svar kan variere fra beregningsresultaterne på grund af forskelle i de underliggende matematiske implementeringer.

Python er tilgængelig på disse TI-Nspire™-produkter:

Håndholdte	Desktop Software
TI-Nspire™ CX II	TI-Nspire™ CX Premium Teacher Software
TI-Nspire™ CX II CAS	TI-Nspire™ CX CAS Premium Teacher Software
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	

**Bemærk:** I de fleste tilfælde er funktionaliteten identisk mellem den håndholdte og softwarevisningerne, men du kan muligvis se nogle forskelle. Denne vejledning antager, at du bruger den håndholdte enhed eller håndholdt visning i softwaren.

## Python-moduler

TI-Nspire™ Python inkluderer følgende moduler:

Standardmoduler	TI-moduler
Matematik (math)	TI PlotLib (ti_plotlib)
Tilfældig (random)	TI Hub (ti_hub)
Kompleks matematik (cmath)	TI Rover (ti_rover)
Tid (time)	TI System (ti_system)
	TI Draw (ti_draw)
	TI-billede (ti_image)

**Bemærk:** Hvis du har eksisterende Python-programmer oprettet i andre Python-udviklingsmiljøer, kan det være nødvendigt at redigere dem for at køre på TI-Nspire™ Python-løsningen. Moduler kan anvende forskellige metoder, argumenter og bestilling

af metoder i et program sammenlignet med TI-modulerne. Vær generelt opmærksom på kompatibilitet ved brug af en hvilken som helst version af Python og Python-moduler.

Når du overfører Python-programmer fra en ikke-TI-plattform til en TI-plattform ELLER fra et TI-produkt til et andet, skal du huske:

- Programmer, der bruger kernesprogfunktioner og standardbiblioteker (matematik, tilfældig, osv.), kan overføres uden ændringer.
- Programmer, der bruger platformspecifikke biblioteker såsom matplotlib til PC eller TI-moduler, vil kræve redigeringer, før de køres på en anden platform. Dette kan ske selv mellem TI-plattformer.

Som med enhver version af Python skal du inkludere importvarer for at bruge alle funktioner, metoder eller konstanter indeholdt i et givent modul. Brug f.eks. følgende kommandoer til at udføre `cos()`-funktionen fra matematikmodulet:

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

For en liste over menuer med deres elementer og beskrivelser se venligst afsnittet [Menukort](#).

## ***Sådan installeres et Python-program som et modul***

**Sådan gemmer du dit Python-program som et modul:**

- I Editor skal du vælge **Handlinger > Installér som Python-modul**.
- I Shell skal du vælge **Værktøjer > Installér som Python-modul**.

Efter valget sker følgende:

- Der er kryds ved Python-syntaks.
- Filen gemmes og flyttes til PyLib-mappen.
- Der vises en dialogboks, der bekræfter, at filen er blevet installeret som et modul.
- Filen lukkes, og modulet er klar til brug.
- Modulnavnet vil blive føjet til menuen **Flere moduler med et fra <module> importmenuen\***.

Hvis du planlægger at dele dette modul med andre, anbefales det, at du følger disse retningslinjer:

- Gem kun ét modul pr. TNS-fil.
- Modulnavnet svarer til navnet på TNS-filen (f.eks. er "my\_program" modulet i filen "my\_program.tns").
- Tilføj en Noter-side før Python-redigeringsprogrammet, der beskriver hensigten med modulet, versionen og funktionerne.
- Brug funktionen `ver()` til at vise modulets versionsnummer.
- (Valgfrit) Tilføj en hjælpefunktion for at vise listen over metoder i funktionen.

# Python-arbejdsområder

Der er to arbejdsområder til din Python-programmering: Python Editor og Python Shell.

Python Editor	Python Shell
<ul style="list-style-type: none"><li>• Opret, rediger og gem Python-programmer</li><li>• Fremhævelse af syntaks og auto-indrykning</li><li>• Indbyggede prompter til vejledning med funktionsargumenter</li><li>• Værktøjstip til visning af gyldige værdier</li><li>• Tasten <code>var</code> viser globale brugervariable og funktioner defineret i det aktuelle program</li><li>• Tastaturgenveje</li></ul>	<ul style="list-style-type: none"><li>• Kør Python-programmer</li><li>• Praktisk til test af små kodefragmenter</li><li>• Interaktion med Shell-historik for at vælge tidligere input og output til genbrug</li><li>• Tasten <code>var</code> viser globale brugervariable defineret i det sidste program, der blev kørt i det givne problem</li></ul>

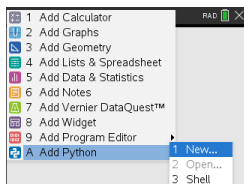
**Bemærk:** Flere Python-programmer og Shells kan tilføjes til et problem.

## Python Editor

Python Editor er, hvor du kan oprette, redigere og gemme Python-programmer.

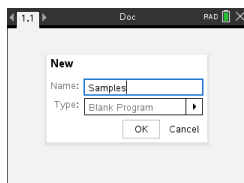
### Tilføj en Python Editor-side

Hvis du vil tilføje en ny Python Editor-side i det aktuelle problem, skal du trykke på `menu` og vælge **Tilføj Python > Ny**.

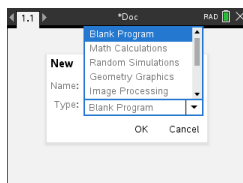


Du kan oprette et tomt program, eller du kan vælge en skabelon.

#### Tomt program



#### Skabelon



Når programmet er oprettet, vises Python Editor. Hvis du valgte en skabelon, tilføjes de nødvendige importerklæringer automatisk (se nedenfor).

**Bemærk:** Du kan have flere programmer i en enkelt TNS-fil ligesom andre apps. Hvis Python-programmet er beregnet til at blive brugt som et modul, kan TNS-filen gemmes i PyLib-mappen. Dette modul kan derefter bruges i andre programmer og dokumenter.

### Matematiske beregninger

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

### Vilkårlige simuleringer

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

### Geometrigrafik

```
1.1 |> *Doc RAD 5/5
+ *Templates.py
# Geometry Graphics
#=====
from t_draw import *
#=====
```

### Billedbehandling

```
1.1 |> *Doc RAD 6/6
+ *Templates.py
# Image Processing
#=====
from t_image import *
from t_draw import get_screen_dim
#=====
```

### Plotning (x,y) & Tekst

```
1.1 |> *Doc RAD 5/5
+ *Templates.py
# Plotting (x,y) & Text
#=====
import t_plotlib as plt
#=====
```

### Datadeling

```
1.1 |> *Doc RAD 5/5
+ *Templates.py
# Data Sharing
#=====
from t_system import *
#=====
```

### TI-Innovator Hub-projekt

```
1.1 |> *Doc RAD 9/9
+ *Templates.py
# Hub Project
#=====
from t_hub import *
from math import *
from time import sleep
from t_plotlib import text_at_cls
from t_system import get_key
#=====
```

### TI-Rover-kodning

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

## Åbning af et Python-program

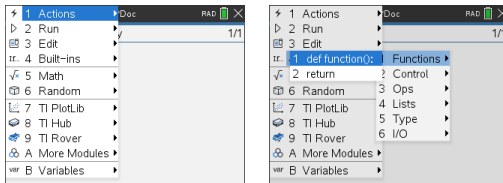
For at åbne et eksisterende Python-program, tryk på **doc** og vælg **Indsæt > Tilføj Python > Åbn**. Dette vil vise en liste over programmer, der er blevet gemt i TNS-filen.

Hvis Editor-siden bruges til at oprette programmet, er programmet stadig tilgængeligt i TNS-filen.

## Arbejde i Python Editor

Tryk på **menu** for at vise menuen Dokumentværktøjer. Med disse menuindstillinger kan du tilføje, flytte og kopiere blokke af kode til dit program.

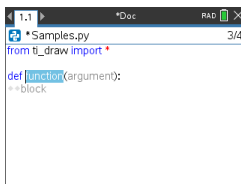
### Dokumentværktøjsmenu



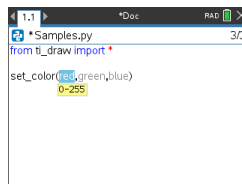
De elementer, der vælges i modulmenuerne, vil automatisk tilføje en kodeskabelon til editoren med indbyggede prompter for hver del af funktionen. Du kan navigere fra et argument til det næste ved at trykke på **tab** (fremad) eller **shift+tab** (bagud).

Værktøjstips eller pop op-lister vises, når de er tilgængelige, for at hjælpe dig med at vælge de korrekte værdier.

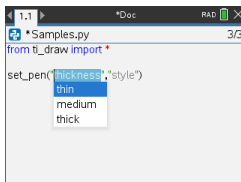
### Indbyggede prompter



### Værktøjstip



### Pop op-lister



Tallene til højre for programnavnet afspejler det aktuelle linjenummer for markøren og det samlede antal linjer i programmet.



```
1.1 |> Samples RAD 7/12
Samples.py
from math import *
x=2
y=4.5
z=pi
def f1(n):
    =>return n**2
def f2(n):
    =>return n**3
```

Globale funktioner og variable defineret i linjerne over den aktuelle markøroposition kan indsættes ved at trykke på `var` og vælge fra listen.

```
1.1 |> Samples RAD 11/11
Samples.py
from math import *
x=2
y=4.5
z=pi
def f1(n):
    =>return n**2
def f2(n):
    =>return n**3
```

Når du tilføjer kode til dit program, viser Editor nøgleord, operatører, kommentarer, strenge og inddeler i forskellige farver for at hjælpe med at identificere de forskellige elementer.

```
1.1 |> *Pythagoras RAD 9/9
*Pythagoras.py
def f(a,b,c):
    =>return(a**2+b**2-c**2)
def ttriec(a,b,c):
    =>print("a:",a,"b:",b,"c:",c)
    =>if f(a,b,c)==0 or f(a,c,b)==0 or f(b,c,a)==0:
        =>print("Right triangle")
    =>else:
        =>print("Non-right triangle")
```

## Gemme og køre programmer

Når du er færdig med programmet, skal du trykke på `menu` og vælge **Kør > Kontrollér syntaks & Gem**. Dette vil kontrollere syntaksen for Python-programmet og gemme det i TNS-filen.

**Bemærk:** Hvis du har ændringer, der ikke er gemt i programmet, vises en stjerne ved siden af programnavnet.

```
1.1 |> *Pythagoras RAD 9/9
*Pythagoras.py
```

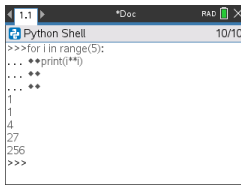
For at køre programmet skal du trykke på `menu` og vælge **Kør > Kør**. Dette vil køre det aktuelle program på den næste Python Shell-side eller på en ny, hvis den næste side ikke er en Shell.

**Bemærk:** Kørsel af programmet kontrollerer automatisk syntaksen og gemmer programmet.

## Python Shell

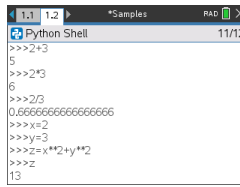
Python Shell er den tolk, der udfører dine Python-programmer, andre stykker Python-kode eller enkle kommandoer.

## Python-kode




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

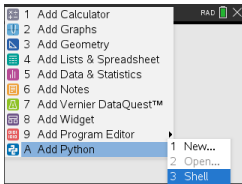
## Enkle kommandoer



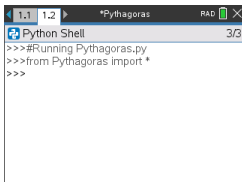
```
Python Shell 11/12
>>>2+3
5
>>>2*3
6
>>>2/3
0.6666666666666666
>>>x=2
>>>y=3
>>>z=x**2+y**2
>>>z
13
```

## Tilføjelse af Python Shell-side

Hvis du vil tilføje en ny Python Shell-side i det aktuelle problem, skal du trykke på  og vælge **Tilføj Python > Shell**.




Python Shell kan også startes fra Python Editor ved at udføre et program ved at trykke på  og vælge **Kør > Kør**.

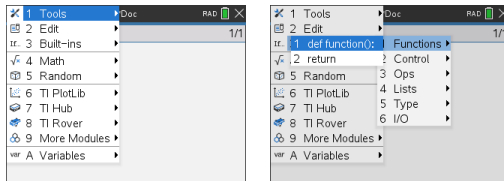


```
Python Shell 3/3
>>>#Running Pythagoras.py
>>>from Pythagoras import *
>>>
```

## Arbejde i Python Shell

Tryk på  for at vise menuen Dokumentværktøjer. Med disse menuindstillinger kan du tilføje, flytte og kopiere blokke af kode.

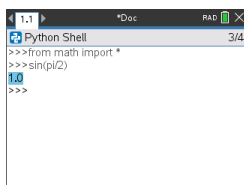
### Dokumentværktøjsmenu



**Bemærk:** Hvis du bruger en hvilken som helst metode fra et af de tilgængelige moduler, skal du sørge for at udføre en importmodulerklæring først som i ethvert Python-kodningsmiljø.

Interaktion med Shell-output minder om lommeregner-appen, hvor du kan vælge og kopiere tidligere input og output til brug andetsteds i Shell, Editor eller andre apps.

*Pil op for at vælge, så kopiér og indsæt i den ønskede placering*

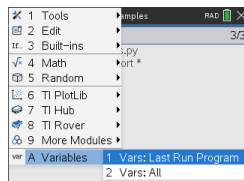


```
Python Shell 3/4
>>>from math import *
>>>sin(pi/2)
1.0
>>>
```

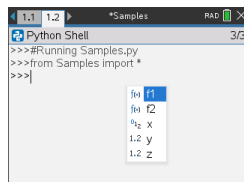
Globale funktioner og variable fra det sidst kørte program kan indsættes ved at trykke på **var** eller **ctrl+L** og vælge fra listen eller trykke på **menu** og vælg **Variabler > Variabler: Sidst kørte program**.

Du kan vælge mellem en liste over globale funktioner og variable fra både det sidst kørte program og eventuelle importerede moduler ved at trykke på **menu** og vælg **Variabler > Variabler: Alle**.

### Menuen Variabel

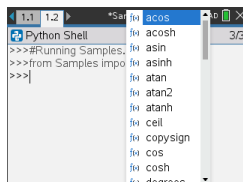


### Sidst kørte program variabler



```
Python Shell 3/4
>>>#Running Samples.py
>>>from Samples import *
>>>|
  f1
  f2
  x
  1.2 y
  1.2 z
```

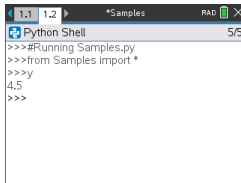
### Alle variable



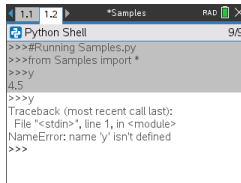
```
Python Shell 3/4
  acos
  acosh
  asin
  asinh
  atan
  atan2
  atanh
  ceil
  copysign
  cos
  cosh
  denrepe
```

Alle Python Shell-sider i samme opgave deler den samme tilstand (brugerdefinerede og importerede variabeldefinitioner). Når du gemmer eller kører et Python-program i dette problem, eller trykker på **menu** og vælger **Værktøjer > Geninitialiser Shell**, vil Shell-historikken derefter have en grå baggrund, der indikerer, at den tidligere tilstand ikke længere er gyldig.

*Før du gemmer eller genstarter*    *Efter du gemmer eller genstarter*



```
Python Shell
5/5
>>>#Running Samples.py
>>>from Samples import *
>>>y
4,5
>>>
```



```
Python Shell
9/9
>>>#Running Samples.py
>>>from Samples import *
>>>y
4,5
>>>y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'y' isn't defined
>>>
```

**Bemærk:** Indstillingen **☰** **Værktøjer > Ryd historik** rydder skærmen for enhver tidligere aktivitet i Shell, men variable er stadig tilgængelige.

## Meddelelser

Fejl og andre meddelelser kan vises, mens du er i Python-session. Hvis der vises en fejl i Shell, når et program eksekveres, vises et programlinjenummer. Tryk på **ctrl** **☰** og vælg **Gå til Python Editor**. I Editor trykker du på **☰**, og vælger derefter **Rediger > Gå til linje**. Indtast linjenummeret, og tryk på **enter**. Markøren vises på det første tegn i den linje, hvor fejlen opstod.

## Afbrydelse af et program, der kører

Mens et program eller en funktion kører, vises markøren ☹.

- ▶ Afbrydelse af programmet eller funktionen
  - Windows®: Tryk på tasten **F12**.
  - Mac®: Tryk på tasten **F5**.
  - Håndholdt: Tryk på tasten **⏏**.

# Python-menumkort

Dette afsnit viser alle menuer og menupunkter for Python Editor og Shell samt en kort beskrivelse for hver enkelt.

**Bemærk:** For menupunkter med tastaturgenveje skal Mac® brugere erstatte ⌘ (cmd) hvor som helst **Ctrl** anvendes. En fuldstændig liste over TI-Nspire™-håndholdte og softwaregenveje findes i e-vejledningen til TI-Nspire™-teknologien.

---

Menuen Handlinger .....	11
Kørselsmenu .....	12
Værktøjsmenu .....	13
Menuen Rediger .....	14
Menuen Indbyggede .....	15
Matematikmenu .....	18
Tilfældigheds-menu .....	20
TI Menuen PlotLib .....	21
Menuen TI Hub .....	23
Menuen TI Rover .....	31
Menu for kompleks matematik .....	39
Tidsmenu .....	40
TI systemmenu .....	41
TI tegnmenu .....	42
TI-billedmenu .....	44
Menuen Variabel .....	46

## Menuen Handlinger

**Bemærk:** Dette gælder kun for editoren.

Menupunkt	Beskrivelse
Nyt (New)	Åbner dialogboksen <b>Ny</b> , hvor du kan indtaste et navn og vælge en type til dit nye program.
Åbn	Åbner en liste med programmer, der er tilgængelige i det aktuelle dokument.
Lav kopi (Create Copy)	Åbner dialogboksen <b>Opret kopi</b> , hvor du kan gemme det aktuelle program under et andet navn.
Omdøb	Åbner dialogboksen <b>Omdøb</b> , hvor du kan omdøbe det aktuelle program.
Luk (Close)	Lukker det aktuelle program
Indstillinger	Åbner dialogboksen <b>Indstillinger</b> , hvor du kan ændre skriftstørrelsen for både Editor og Shell.
Installer som Python-modul	Kontrollerer Python-syntaksen for den aktuelle TNS-fil og flytter den til PyLib-mappen.

## Kørselsmenu

Bemærk: Dette gælder kun for Editor.

Menupunkt	genveje	Beskrivelse
Kør	Ctrl+R	Kontrollerer syntaks, gemmer program og udfører i Python Shell.
Kontrollér syntaks & gem	Ctrl+B	Kontrollerer syntaks og gemmer program
Gå til Shell	N/A	Skifter fokus til Shell relateret til det aktuelle program eller åbner en ny Shell-side ved siden af Editor.

## Værktøjsmenu

Bemærk: Dette gælder kun for Shell.

Menupunkt	genveje	Beskrivelse
Kør det sidste program igen	Ctrl+R	Kører det sidste program, der er relateret til den aktuelle Shell.
Gå til Python-editoren	N/A	Åbner siden Editor, der er relateret til den aktuelle Shell.
Kør	N/A	Åbner en liste med programmer, der er tilgængelige i det aktuelle dokument. Efter valget køres det valgte program.
Slet Historik	N/A	Rydder historikken i den aktuelle Shell, men genstarter ikke Shell.
Genstarter Shell	N/A	Nulstiller status for alle åbne Shell-sider i den aktuelle opgave. Alle definerede variabler og importerede funktioner er ikke længere tilgængelige.
dir()	N/A	Viser en liste over funktioner i det angivne modul, når det anvendes efter importerklæringen.
Fra PROGRAM-import *	N/A	Åbner en liste med programmer, der er tilgængelige i det aktuelle dokument. Efter valget indsættes importerklæringen i Shell.
Installér som Python-modul	N/A	Kun aktiveret for moduler i binært format. Flytter den aktuelle TNS-fil til PyLib-mappen.



## Menuen Rediger

**Bemærk:** Ctrl+A vælger alle linjer eller output til skæring eller sletning (kun Editor), eller kopiering og indsættelse (Editor og Shell).

Menupunkt	genveje	Beskrivelse
Indryk	TAB*	Indrykker tekst på den aktuelle linje eller valgte linjer. * Hvis der er ufuldstændige indbyggede prompter, vil TAB navigere til næste prompt.
Udrykning	Shift+Tab**	Udrykker tekst på den aktuelle linje eller valgte linjer. ** Hvis der er ufuldstændige indbyggede prompter, vil Shift+TAB navigere til forrige prompt.
Kommenter/fjern kommentar	Ctrl+T	Tilføjer/fjerner kommentarsymbol til/fra begyndelsen af den aktuelle linje.
Indsæt multilinjestreng	N/A	(Kun Editor) Indsætter flerlinjet strengskabelon.
Søg	Ctrl+F	(Kun editor) Åbner <b>Find</b> -dialogboks og søger efter den indtastede streng i det aktuelle program.
Erstat	Ctrl+H	(Kun editor) Åbner <b>Erstat</b> -dialogboks og søger efter den indtastede streng i det aktuelle program.
Gå til linje (Go to Line)	Ctrl+G	(Kun redaktør) Åbner <b>Gå til linje</b> -dialogboks og hopper til den specificerede streng i det aktuelle program.
Start på linje	Ctrl+8	Flytter markøren til starten af den aktuelle linje.
Afslutning af linje	Ctrl+2	Flytter markøren til afslutningen af den aktuelle linje.
Gå til top	Ctrl+7	Flytter markøren til starten af den første linje af programmet.
Gå til bund	Ctrl+1	Flytter markøren til slutningen af den sidste linje af programmet.

## Menuen Indbyggede

### funktioner

Menupunkt	Beskrivelse
def function():	Definerer en funktion, der er afhængig af de angivne variable.
return	Definerer værdien produceret af en funktion.

### Kontrol (Control)

Menupunkt	Beskrivelse
if..	Betinget erklæring.
if..else..	Betinget erklæring.
if..elif..else..	Betinget erklæring.
for index in range(størrelse):	Itererer over et interval
for indeks i interval(start/stop):	Itererer over et interval
for indeks i interval(start,stop,step):	Itererer over et interval
for indeks på liste	Itererer over over liste af elementer.
while..	Eksekverer erklæringer i en kodeblok, indtil en betingelse evalueres som Falsk.
elif:	Betinget erklæring.
else:	Betinget erklæring.

### Indst.

Menupunkt	Beskrivelse
x=y	Indstiller variabel værdi.
x==y	Indsætter lig med (==) sammenligningshandling.
x!=y	Indsætter er ikke lig med (!=) sammenligningshandling.
x>y	Indsætter større end (>) sammenligningsoperator.
x>=y	Indsætter større end eller lig med (>=) sammenligningsoperator.
x<y	Indsætter større end (<) sammenligningsoperator.
x<=y	Indsætter mindre end eller lig med (<=) sammenligningsoperator.

Menupunkt	Beskrivelse
og	Indsætter og (og) logisk operator.
—eller—	Indsætter eller (eller) logisk operator.
not	Indsætter ikke (ikke) logisk operator.
Sand	Indsætter Sand Boolesk værdi.
Falsk	Indsætter Falsk Boolesk værdi.

## Lister

Menupunkt	Beskrivelse
[]	Indsætter klammer ([]).
liste()	Konverterer talfølge til "liste"-type.
len()	Returnerer antallet af elementer i listen.
max()	Returnerer det største element i listen.
min()	Returnerer det mindste element i listen.
.append()	Metoden tilføjer et element til en liste.
.remove()	Metoden fjerner første forekomst af et element fra en liste.
range(start,stop,trin)	Returnerer et sæt tal.
for index in range(start,stop,trin)	Bruges til at gentage over et interval.
.insert()	Metoden tilføjer et element i den angivne position.
.split()	Metoden returnerer en liste med elementer adskilt af specificeret skilletegn.
sum()	Returnerer summen af elementer i en liste.
sorted()	Returnerer en sorteret liste.
.sort()	Metoden sorterer en liste på plads.

## Type

Menupunkt	Beskrivelse
int()	Returnerer en heltalsdel.
float()	Returnerer en flydende værdi.

Menupunkt	Beskrivelse
round(x,ncifre)	Returnerer et flydende decimaltal afrundet til det angivne antal cifre.
str()	Returnerer en streng.
complex()	Returnerer et komplekst tal.
type()	Returnerer typen af objektet.

## I/O

Menupunkt	Beskrivelse
print()	Viser argumentet som streng.
input()	Prompter bruger om input
eval()	Evaluerer et udtryk repræsenteret som en streng.
.format()	Metoden formaterer den angivne streng.

## Matematikmenu

**Bemærk:** Når der oprettes et nyt program, der bruger dette modul, anbefales det at bruge **Matematiske beregninger** programtype. Dette vil sikre, at alle relevante moduler importeres.

Menupunkt	Beskrivelse
fra math import *	Importerer alle metoder (funktioner) fra matematik-modulet.
fabs()	Returnerer den absolutte værdi af et reelt tal.
sqrt()	Returnerer kvadratroden af et reelt tal.
exp()	Returnerer $e^{**x}$ .
pow(x,y)	Returnerer x opløftet til potensen y.
log(x,base)	Returnerer $\log_{\text{base}}(x)$ . log(x) uden base returnerer den naturlige logaritme x.
fmod(x,y)	Returnerer modulværdien for x og y. Bruges, når x og y flyder.
ceil()	Returnerer det mindste heltal, der er større end eller lig med et reelt tal.
floor()	Returnerer det største heltal, der er mindre end eller lig med et reelt tal.
trunc()	Trunkerer et reelt tal til et heltal.
frexp()	Returnerer et par (y,n), hvor $x == y * 2^{**n}$ .

## Konstant

Menupunkt	Beskrivelse
e	Returns value for the constant e.
pi	Returns value for the constant pi.

## Trigonometri

Menupunkt	Beskrivelse
radians()	Konverterer vinkel i grader til radianer.
degrees()	Konverterer vinkel i radianer til grader.
sin()	Returnerer sinus af argumentet i radianer.

Menupunkt	Beskrivelse
cos()	Returnerer cosinus af argumentet i radianer.
tan()	Returnerer tangens af argumentet i radianer.
asin()	Returnerer arc-sinus af argumentet i radianer.
acos()	Returnerer arc-cosinus af argumentet i radianer.
atan()	Returnerer arc-tangens af argumentet i radianer.
atan2(y,x)	Returnerer arc-tangent for $y/x$ i radianer.

## Tilfældigheds-menu

**Bemærk:** Når der oprettes et nyt program, der bruger dette modul, anbefales det at bruge **Tilfældige simuleringer** programtype. Dette vil sikre, at alle relevante moduler importeres.

Menupunkt	Beskrivelse
from random import *	Importerer alle metoder fra tilfældigheds-modulet.
random()	Returnerer et tal med flydende komma mellem 0 og 1,0.
uniform(min,maks)	Returnerer et tilfældigt tal x (float) så min. $\leq x \leq$ maks.
randint(min,maks)	Returnerer et tilfældigt heltal mellem min. og maks.
choice(talfølge)	Returnerer et vilkårligt element fra en ikke-tom talfølge.
randrange(start,stop,step)	Returnerer et tilfældigt tal fra start til stop trin for trin.
seed()	Initialiserer tilfældighedsgeneratoren

## Ti Menuen PlotLib

**Bemærk:** Når der oprettes et nyt program, der bruger dette modul, anbefales det at bruge programtypen **Plotning (x,y) & Tekst**. Dette vil sikre, at alle relevante moduler importeres.

Menupunkt	Beskrivelse
import ti_plotlib as plt	Importerer alle metoder (funktioner) fra ti_plotlib-modulet i "plt"-navnerummet. Som et resultat vil alle funktionsnavne, der indsættes i menuerne, komme forud for "plt".

### Indst

Menupunkt	Beskrivelse
cls()	Rydder lærred med plotning.
grid(x-skala,y-skala,"stil")	Viser et gitter ved hjælp af den angivne skala for x- og y-akser.
window(xmin,xmaks,ymin,ymax)	Definerer plotvinduet ved at kortlægge det angivne horisontale interval (xmin, xmax) og det lodrette interval (ymin, ymax) til det tildelte plotningsområde (pixel).
auto_window(x-liste,y-liste)	Autoskalér vinduet til plotning, så det passer til dataområderne i x-liste og y-liste, der er angivet i programmet, før auto_window().
axes("tilstand")	Viser akser i det specificerede vindue i plotningsområdet.
labels("x-mærke","y-mærke",x,y)	Viser "x-mærke"- og "y-mærke"-etiketter på plotakserne ved rækkepositionerne x og y.
title("titel")	Viser "titel" centreret i vinduets øverste linje.
show_plot()	Viser den bufferede tegningsoutput. Funktionerne use_buffer() og show_plot() er nyttige i tilfælde, hvor visning af flere objekter på skærmen kan forårsage forsinkelser (ikke nødvendigt i de fleste tilfælde).
use_buffer()	Aktiverer en buffer for gøre tegning hurtigere.



## Tegn

Menupunkt	Beskrivelse
<code>color(rød,grøn,blå)</code>	Indstiller farven for alle følgende grafik/plotning.
<code>cls()</code>	Rydder lærred med plotning.
<code>show_plot()</code>	Udfører visningen af plottet som konfigureret i programmet.
<code>scatter(x-liste,y-liste,"mærke")</code>	Plotter en sekvens af et bestilt par fra (x-liste,y-liste) med den angivne mærketype.
<code>plot(x-liste,y-liste,"mærke")</code>	Tegner en linje ved hjælp af bestilt par fra den angivne x-liste og y-liste.
<code>plot(x,y,"mærke")</code>	Plotter et punkt ved brug af koordinater x og y med den angivne mærketype.
<code>line(x1,y1,x2,y2,"tilstand")</code>	Tegner et linjestykke fra (x1,y1) til (x2,y2).
<code>lin_reg(x-liste,y-liste,"vis")</code>	Beregner og tegner den lineære regressionsmodel, $ax+b$ , af x-liste,y-liste.
<code>pen("størrelse","stil")</code>	Indstiller udseendet af alle følgende linjer, indtil den næste <code>pen()</code> udføres.
<code>text_at(række,"tekst","align")</code>	Viser "tekst" i plotningsområde ved angivet "align".

## Egenskaber

Menupunkt	Beskrivelse
<code>xmin</code>	Angivet variabel for vinduesargumenter defineret som <code>plt.xmin</code> .
<code>xmax</code>	Angivet variabel for vinduesargumenter defineret som <code>plt.xmax</code> .
<code>ymin</code>	Angivet variabel for vinduesargumenter defineret som <code>plt.ymin</code> .
<code>ymax</code>	Angivet variabel for vinduesargumenter defineret som <code>plt.ymax</code> .
<code>m</code>	Efter <code>plt.linreg()</code> udføres i et program, lagres de beregnede værdier for hældning, m og skæringspunkt, b, i <code>plt.m</code> og <code>plt.b</code> .
<code>b</code>	Efter <code>plt.linreg()</code> udføres i et program, lagres de beregnede værdier for hældning, a, og skæring, b, i <code>plt.a</code> og <code>plt.b</code> .

## Menuen *Ti Hub*

**Bemærk:** Når du opretter et nyt program, der bruger dette modul, anbefales det at bruge programtypen **Hub Project**. Dette vil sikre, at alle relevante moduler importeres.

Menupunkt	Beskrivelse
fra import af <code>ti_hub *</code>	Importerer alle metoder fra <code>ti_hub</code> -modulet.

### Hub-indbyggede enheder > Farveoutput

Menupunkt	Beskrivelse
<code>rgb(rød, grøn, blå)</code>	Angiver farven for RGB-lysdioden.
<code>blink(frekvens,tid)</code>	Angiver blinkefrekvensen og varigheden for den valgte farve.
<code>slukket()</code>	Slukker RGB-lysdioden.

### Hub-indbyggede enheder > Lysoutput

Menupunkt	Beskrivelse
<code>tændt()</code>	Tænder lysdioden.
<code>slukket()</code>	Slukker lysdioden.
<code>blink(frekvens,tid)</code>	Angiver lysdiodens blinkefrekvens og varighed.

### Hub-indbyggede enheder > Lydudgang

Menupunkt	Beskrivelse
<code>tone(frekvens,tid)</code>	Afspiller en tone med den angivne frekvens i det angivne tidsrum.
<code>note("note",tid)</code>	Afspiller den angivne tone i det angivne tidsrum. Noten er angivet ved hjælp af notetnavnet og en oktav. For eksempel: A4, C5. Tonenavnene er C, CS, D, DS, E, F, FS, G, GS, A, AS og B. Oktavnumrene spænder fra 1 til 9 (inklusive).
<code>tone(frekvens,tid,tempo)</code>	Afspiller en tone med den angivne frekvens i det angivne tidsrum og med det angivne tempo. Tempoet definerer antallet af bip pr. sekund fra 0 til 10

Menupunkt	Beskrivelse
	(inklusive).
note("tone",tid,tempo)	Afspiller den angivne tone i det angivne tidsrum og med det angivne tempo. Noten er angivet ved hjælp af notetnavnet og en oktav. For eksempel: A4, C5. Tonenavnene er C, CS, D, DS, E, F, FS, G, GS, A, AS og B. Oktavnumrene spænder fra 1 til 9 (inklusive). Tempotallene spænder fra 0 til 10 (inklusive).

## Hub-indbyggede enheder > lysstyrkeinput

Menupunkt	Beskrivelse
målinger()	Aflæser den indbyggede sensors LYSSTYRKE (lysniveau) og returnerer aflæsningen. Standardintervallet er 0 til 100. Dette kan ændres ved hjælp af funktionen interval().
interval(min.,maks.)	Angiver intervallet for kortlægning af aflæsninger fra lysniveausensoren. Hvis begge mangler eller er angivet til værdien Ingen, er standardlystyrkeintervallet sat til 0 til 100.

## Tilføj inputenhed

Denne menu har en liste over sensorer (inputenheder), der understøttes af ti\_hub-modulet. Alle menupunkterne indsætter objektets navn og forventer en variabel og en port, der bruges med sensoren. Hver sensor har en målemetode (), der returnerer sensorens værdi.

Menupunkt	Beskrivelse
DHT (digital & fugtighedstemperatur)	Returnerer en liste, der består af den aktuelle temperatur, fugtighed, sensortype og sidste cachelagrede aflæsningstatus.
Bevægelsessensor	Returnerer den aktuelle afstandsmåling fra den angivne ultralydsafstandsmåler. <ul style="list-style-type: none"> <li><b>measurement_time()</b> - Returnerer den tid, det tager ultralydssignalet at nå objektet ("flyvetiden").</li> </ul>
Lysniveau	Returnerer lysstyrkeniveauet fra den eksterne lysstyrkesensor (lysstyrke).

Menupunkt	Beskrivelse
temperatur	<p>Returnerer temperatur aflæsningen fra den eksterne temperatursensor.</p> <p>Standardkonfigurationen er at understøtte Seed-temperatursensoren i IN 1-, IN 2- eller IN 3-porten.</p> <p>Hvis du vil bruge TI LM19-temperatursensoren fra TI-Innovator™ Hub breadboard-pakken, skal du redigere porten til den anvendte BB-stift og bruge et valgfrit argument "TIANALOG".</p> <p>Eksempel: <code>mylm19=temperatur("BB 5","TIANALOG")</code></p>
Fugtighed	Returnerer aflæsningen af fugtighedssensoren.
Magnetisk	<p>Registrerer tilstedeværelsen af et magnetfelt. Tærskelværdien til bestemmelse tilstedeværelsen af feltet angives via <code>udløser()</code>-funktionen.</p> <p>Standardværdien for tærsklen er 150.</p>
Vernier	<p>Aflæser værdien fra den analoge Verniersensor, der er angivet i kommandoen. Kommandoen understøtter følgende Verniersensorer:</p> <ul style="list-style-type: none"> <li>• <b>temperatur</b> – Temperaturføler af rustfrit stål.</li> <li>• <b>lysniveau</b> – TI-lysstyrkesensor.</li> <li>• <b>tryk</b> – Oprindelig gastryksensor</li> <li>• <b>tryk</b> – Ny gastryksensor.</li> <li>• <b>Ph</b> – pH-sensor.</li> <li>• <b>force10</b> – <math>\pm 10</math> N-indstilling, Dual Force-sensor.</li> <li>• <b>kraft50</b> - <math>\pm 50</math> N indstilling, Dual Force Sensor.</li> <li>• <b>accelerometer</b> – Low-g-accelerometer.</li> <li>• <b>generisk</b> – Muliggør angivelse af andre sensorer, der ikke understøttes direkte ovenfor, og brug af API'en til kalibrering() ovenfor til angivelse af ligningskoefficienter.</li> </ul>
Analog ind	Understøtter brugen af analoge input-generiske enheder.
Digital ind	Returnerer status på den digitale pin, der er

Menupunkt	Beskrivelse
	forbundet til det DIGITALE objekt, eller den cachelagrede tilstand på den digitale outputværdi, der sidst blev ANGIVET til objektet.
Potentiometer	Understøtter en potentiometersensor. Sensorens interval kan ændres med funktionen interval().
Termistor	Aflæser termistorsensorer. Standardkoefficienterne er designet til at matche den termistor, der er inkluderet i Breadboard-pakken på TI-Innovator™-hubben, når de bruges med en 10 kΩ fast modstand. Et nyt sæt kalibreringskoefficienter og referencemodstand for termistoren kan konfigureres ved hjælp af kalibreringsfunktionen ().
Lydstyrke	Understøtter lydlydstyrkesensorer.
Farveinput	Giver grænseflader til en I2C-forbundet farveinputsensor. Stiften bb_port bruges ud over I2C-porten til at styre lysdioden på farvesensoren. <ul style="list-style-type: none"> <li>• <b>farve_nummer():</b> Returnerer en værdi fra 1 til 9, der repræsenterer den farve, sensoren detekterer. Tallene repræsenterer farverne i henhold til følgende kortlægning: 1: Rød 2: Grøn 3: Blå 4: Cyan 5: Magenta 6: Gul 7: Sort 8: Hvid 9: Grå</li> <li>• <b>rød():</b> Returnerer en værdi fra 0 til 255, der repræsenterer intensiteten af det RØDE farveniveau, der detekteres.</li> <li>• <b>grøn():</b> Returnerer en værdi fra 0 til 255, der repræsenterer intensiteten af det GRØNNE farveniveau, der detekteres.</li> </ul>

Menupunkt	Beskrivelse
	<ul style="list-style-type: none"> <li>• <b>blå():</b> Returnerer en værdi fra 0 til 255, der repræsenterer intensiteten af det BLÅ farveniveau, der registreres.</li> <li>• <b>grå():</b> Returnerer en værdi fra 0 til 255, der repræsenterer det gråtoneniveau, der registreres, hvor 0 er sort og 255 er hvid.</li> </ul>
BB-port	<p>Giver understøttelse til brug af alle 10 BB-portstifter som en kombineret digital indgangs-/udgangsport.</p> <p>Initialiseringsfunktionerne har en valgfri "maske"-parameter, som gør det muligt at bruge undersættet med 10 stifter.</p> <ul style="list-style-type: none"> <li>• <b>read_port():</b> Læser de aktuelle værdier på BB-portens indgangsstikben.</li> <li>• <b>write_port(værdi):</b> Angiver outputstiftværdierne til den specificerede værdi, hvor værdien er mellem 0 og 1023. Bemærk, at værdien også justeres i forhold til maskeværdien i funktionen var=bbport (maske), hvis der blev givet en maske.</li> </ul>
Hub-tid	<p>Giver adgang til den interne timer i millisekunder.</p>
TI-RGB-array	<p>Indeholder funktioner til programmering af TI-RGB-array.</p> <p>Initialiseringsfunktionen accepterer en valgfri parameter "LAMPE" for at aktivere en højlysstyrketilstand for TI-RGB-arrayet, som kræver en ekstern strømforsyning.</p> <ul style="list-style-type: none"> <li>• <b>angiv(led_position, r,g,b):</b> Indstiller en specifik led_position (0-15) til den angivne r,g,b-værdi, hvor r,g,b er værdier fra 0 til 255.</li> <li>• <b>angiv(led_list,red,green,blue):</b> Indstiller de LED'er, der er defineret i "led_list" til den farve, der er angivet med "rød", "grøn", "blå". "led_list" er en Python-liste, der indeholder indekser for LED'er fra 0 til 15. F.eks. vil ([0,2,4,6,15], 0, 0, 255) indstille LED'er 0, 2, 4, 6 og 15 til blå.</li> <li>• <b>set_all(r,g,b):</b> Indstiller alle RGB-lysdioder i rækken til den samme r,g,b-værdi.</li> <li>• <b>all_off():</b> Slukker for alle RGB-enheder i modulet.</li> </ul>

Menupunkt	Beskrivelse
	<ul style="list-style-type: none"> <li>• <b>måliger():</b> Returnerer det omtrentlige strømforbrug, som RGB-arrayet bruger, fra TI-Innovator™ i milliampere.</li> <li>• <b>mønster(mønster):</b> Ved at bruge værdien af argumentet som en binær værdi i området 0 til 65535 aktiveres pixel, hvor en 1-værdi i repræsentationen ville være. LED'erne tænder som RØD med en pwm-niveauværdi på 255.</li> <li>• <b>mønster(værdi,rød,grøn,blå):</b> Indstiller LED'erne defineret af "mønsteret" til farven angivet med "rød", "grøn", "blå".</li> </ul>

### Tilføj udgangsenhed

Denne menu har en liste over de outputenheder, der understøttes af ti\_hub-modulet. Alle menupunkterne indsætter objektets navn og forventer en variabel og en port, der bruges med apparatet.

Menupunkt	Beskrivelse
LED	Funktioner til kontrol af eksternt tilsluttede LED'er.
RGB	Understøttelse af kontrol af eksterne RGB-lysdioder.
TI-RGB-array	Indeholder funktioner til programmering af TI-RGB-array.
Højttaler	Funktioner til understøttelse af en ekstern højttaler med TI-Innovator™ Hub. Funktionerne er de samme som for "lyd" ovenfor.
Strøm	Funktioner til styring af ekstern strøm med TI-Innovator™ Hub. <ul style="list-style-type: none"> <li>• <b>angiv(værdi):</b> Angiver effektniveauet til den specificerede værdi, mellem 0 og 100.</li> <li>• <b>tænd():</b> Indstiller strømniveauet til 100.</li> <li>• <b>slået fra():</b> Angiver strømniveauet til 0.</li> </ul>
Kontinuerlig servo	Funktioner til styring af kontinuerlige servomotorer. <ul style="list-style-type: none"> <li>• <b>set_cw(hastighed,tid):</b> Servoen drejer med uret ved den specificerede hastighed (0-255) og i den specifikke varighed i sekunder.</li> <li>• <b>set_ccw(hastighed,tid):</b> Servoen drejer mod uret ved den specificerede hastighed (0-255) og i den specifikke varighed i sekunder.</li> <li>• <b>stop():</b> Stopper den kontinuerlige servo.</li> </ul>
Analog ud	Funktioner til brug af analoge input-generiske enheder.

Menupunkt	Beskrivelse
Vibrationsmotor	Funktioner til styring af vibrationsmotorer. <ul style="list-style-type: none"> <li>• <b>angiv(val)</b>: Angiver vibrationsmotorens intensitet til "val" (0-255).</li> <li>• <b>slået fra()</b>: Slukker vibrationsmotoren.</li> <li>• <b>tændt()</b>: Tænder vibrationsmotoren på det højeste niveau.</li> </ul>
Relæ	Styrer grænseflader til styring af relæer. <ul style="list-style-type: none"> <li>• <b>tændt()</b>: Angiver relæet til TÆNDT.</li> <li>• <b>slået fra()</b>: Angiver relæet til SLUKKET.</li> </ul>
Servo	Funktioner til styring af servomotorer. <ul style="list-style-type: none"> <li>• <b>set_position(pos)</b>: Indstiller sweep-servopositionen inden for et interval på -90 til +90.</li> <li>• <b>nulpunkt()</b>: Angiver sweep-servoens til nulposition.</li> </ul>
Firkantssvingninger	Funktioner til generering af firkantssvingning. <ul style="list-style-type: none"> <li>• <b>angiv(frekvens,drift,tid)</b>: Angiver firkantssvingens output med en standarddriftscyklus på 50 % (hvis der ikke er angivet en driftstid) og en udgangsfrekvens, der er angivet af "frekvens". Frekvensen kan være fra 1 til 500 Hz. Driftscyklussen, hvis den er angivet, kan være fra 0 til 100 %.</li> <li>• <b>slået fra()</b>: Slukker for firkantssvingning.</li> </ul>
Digital udgang	Grænseflader til styring af en digital udgang. <ul style="list-style-type: none"> <li>• <b>angiv(val)</b>: Angiver den digitale udgang til den værdi, der er angivet med "val" (0 eller 1).</li> <li>• <b>tændt()</b>: Angiver tilstanden for den digitale udgang til høj (1).</li> <li>• <b>slået fra()</b>: Angiver tilstanden for den digitale udgang til lav (0).</li> </ul>
BB-port	Indeholder funktioner til programmering af TI-RGB-array. Se oplysningerne ovenfor.

## Kommandoer

Menupunkt	Beskrivelse
dvale(sekunder)	Sætter programmet på pause i det angivne antal sekunder. Importeret fra 'tid'-modulet.
text_at(række,"tekst","align")	Viser den angivne "tekst" i plotteområdet ved den angivne "justering". Del af ti_plotlib-modulet.
cls()	Rydder Shell-skærbilledet for plotning.



Menupunkt	Beskrivelse
	Del af ti_plotlib-modulet.
<code>while get_key() != "esc":</code>	Kører kommandoerne i "mens"-løkken, indtil der trykkes på "esc"-tasten.
<code>get_key()</code>	Returnerer en streng, der repræsenterer den tast, der trykkes på. Tasten "1" returnerer "1", "esc" returnerer "esc" osv. Når den kaldes uden parametre, returnerer - <code>get_key()</code> - med det samme. Når den kaldes med en parameter, venter - <code>get_key(1)</code> -, indtil der trykkes på en tast. Del af ti_system-modulet.

## Porte

Dette er de indgangs- og udgangsporte, der er tilgængelige på TI-Innovator™-hubben.

Menupunkt
UD 1
UD 2
UD 3
IND 1
IND 2
IND 3
BB 1
BB 2
BB 3
BB 4
BB 5
BB 6
BB 7
BB 8
BB 9
BB 10
I2C

## Menuen Ti Rover

**Bemærk:** Når du opretter et nyt program, der bruger dette modul, anbefales det at bruge programtypen **Rover Coding** (Rover-kodning). Dette vil sikre, at alle relevante moduler importeres.

Menupunkt	Beskrivelse
importér ti_rover som rv	Importerer alle metoder (funktioner) fra ti_rover-modulet i "rv"-navneområdet. Som et resultat vil alle funktionsnavne, der indsættes fra menuerne, være efter "rv."

## Drev

Menupunkt	Beskrivelse
fremad(afstand)	Flytter rover'en fremad i den angivne afstand i gitterenheder.
baglæns(afstand)	Flytter rover'en den angivne afstand tilbage i gitterenheder.
venstre(angle_degrees)	Drejer Rover til venstre med den angivne vinkel i grader.
højre(vinkel_degrees)	Drejer Rover mod højre med den angivne vinkel i grader.
stop()	Stopper øjeblikkeligt enhver aktuel bevægelse.
stop_clear()	Stopper øjeblikkeligt enhver aktuel bevægelse og rydder alle ventende kommandoer.
genoptag()	Genoptager behandlingen af kommandoer.
ophold(tid)	Rover forbliver på plads i det angivne tidsrum i sekunder (valgfrit). Hvis der ikke er angivet en tid, forbliver Roveren samme sted i 30 sekunder.
to_xy(x,y)	Flytter Rover til koordinatposition (x,y) på det virtuelle gitter.
to_polar(r,theta_degrees)	Flytter Roveren til den polære koordinatposition (r, theta) på det virtuelle gitter. Vinklen er angivet i grader.
to_angle(vinkel,"enhed")	Drejer Roveren til den angivne vinkel i det virtuelle gitter. Vinklen er i forhold til en nulvinkel, der peger ned ad x-aksen i det virtuelle gitter.

## Drev > Drev med valgmuligheder

Menupunkt	Beskrivelse
<code>forward_time(tidspunkt)</code>	Flytter Roveren fremad i det angivne tidsrum.
<code>backward_time(tid)</code>	Flytter Roveren bagud i det angivne tidsrum.
<code>fremad(afstand,"enhed")</code>	Flytter Roveren fremad med standardhastigheden for den angivne afstand. Afstanden kan angives i gitterenheder, meter eller hjulomdrejninger.
<code>tilbage(afstand,"enhed")</code>	Flytter Roveren tilbage med standardhastigheden for den angivne afstand. Afstanden kan angives i gitterenheder, meter eller hjulomdrejninger.
<code>venstre(vinkel,"enhed")</code>	Drejer Roveren til venstre med den angivne vinkel. Vinklen kan være i grader, radianer eller radianer.
<code>højre(vinkel,"enhed")</code>	Drejer Roveren mod højre med den angivne vinkel. Vinklen kan være i grader, radianer eller radianer.
<code>forward_time(tid,hastighed,"hastighed")</code>	Flytter Roveren fremad i den angivne tid ved den angivne hastighed. Hastigheden kan specificeres i gitterenheder, meter eller hjulomdrejninger.
<code>backward_time(tid,hastighed,"hastighed")</code>	Flytter Roveren bagud i det angivne tidsrum ved den angivne hastighed. Hastigheden kan specificeres i gitterenheder, meter eller hjulomdrejninger.
<code>fremad(afstand,"enhed",hastighed,"hastighed")</code>	Flytter Roveren fremad i den angivne afstand ved den angivne hastighed. Afstanden kan angives i gitterenheder, meter eller

Menupunkt	Beskrivelse
	<p>hjulomdrejninger.</p> <p>Hastigheden kan specificeres i gitterenheder, meter eller hjulomdrejninger.</p>
baglæns(afstand,"enhed",hastighed,"hastighed")	<p>Flytter Roveren bagud i den angivne afstand ved den angivne hastighed.</p> <p>Afstanden kan angives i gitterenheder, meter eller hjulomdrejninger.</p> <p>Hastigheden kan specificeres i gitterenheder, meter eller hjulomdrejninger.</p>

## Inputs

Menupunkt	Beskrivelse
ranger_measurement()	Aflæser ultralydsafstandssensoren på forsiden af Roveren, og returnerer den aktuelle afstand i meter.
color_measurement()	<p>Returnerer en værdi fra 1 til 9, der angiver den dominerende farve, der "ses" af Roveren farveinputsensor.</p> <p>1 = rød</p> <p>2 = grøn</p> <p>3 = blå</p> <p>4 = cyan</p> <p>5 = magenta</p> <p>6 = gul</p> <p>7 = sort</p> <p>8 = grå</p> <p>9 = hvid</p>
red_measurement()	Returnerer en værdi mellem 0 og 255, der angiver det opfattede røde niveau, der ses af farveinputsensoren.
green_measurement()	Returnerer en værdi mellem 0 og 255, der angiver det opfattede grønne niveau, der ses af farveinputsensoren.
blue_measurement()	Returnerer en værdi mellem 0 og 255, der angiver det opfattede blå niveau, der ses af farveinputsensoren.

Menupunkt	Beskrivelse
gray_measurement()	Returnerer en værdi mellem 0 og 255, der angiver det opfattede gråniveau, der ses af farveinputsensoren.
encoders_gyro_measurement()	Returnerer en liste med værdier, der indeholder venstre og højre hjulkoderantal samt den aktuelle gyro-overskrift.
gyro_measurement()	Returnerer en værdi, der repræsenterer den aktuelle gyro-af læsning, herunder skred, i grader.
ranger_time()	Returnerer den tid, som det tager ultralydssignalet fra TI-Rover-afstandsenheden at nå objektet ("flyvetiden").

## Output

Menupunkt	Beskrivelse
color_rgb(r,g,b)	Angiver farven på Roverens RGB-lysdiode til de specifikke røde, grønne, blå værdier.
color_blink(frekvens,tid)	Angiver blinkefrekvensen og varigheden for den valgte farve.
color_off()	Slukker for Roverens RGB-lysdiode.
motor_left(hastighed,tid)	<p>Indstiller den venstre motoreffekt den angivne værdi i den angivne varighed. Hastigheden er i intervallet -255 til 255, hvor 0 er stop. Positive hastighedsværdier er rotation mod uret, og negative hastighedsværdier er med uret.</p> <p>Det valgfrie tidsparameter, hvis specificeret, har et gyldigt interval på 0,05 til 655,35 sekunder. Hvis det ikke er angivet, bruges en standard på 5 sekunder.</p>
motor_right(hastighed,tid)	<p>Indstiller den venstre motoreffekt den angivne værdi i den angivne varighed. Hastigheden er i intervallet -255 til 255, hvor 0 er stop. Positive hastighedsværdier er rotation mod uret, og negative hastighedsværdier er med uret.</p>

Menupunkt	Beskrivelse
	Det valgfrie tidsparameter, hvis specificeret, har et gyldigt interval på 0,05 til 655,35 sekunder. Hvis det ikke er angivet, bruges en standard på 5 sekunder.
<code>motorer("ldir",left_val,"rdir",højre_val,tid)</code>	<p>Indstiller venstre og højre hjul til de angivne hastighedsniveauer i sekunder (valgfrit).</p> <p>Værdierne for hastighed (<code>left_val</code>, <code>right_val</code>) er i intervallet 0 til 255, hvor 0 er stop. <code>ldir</code>- og <code>rdir</code>-parametrene angiver rotation med uret eller mod uret for de respektive hjul.</p> <p>Det valgfrie tidsparameter, hvis specificeret, har et gyldigt interval på 0,05 til 655,35 sekunder. Hvis det ikke er angivet, bruges en standard på 5 sekunder.</p>

## Sti

Menupunkt	Beskrivelse
<code>waypoint_xythdrn()</code>	Aflæser x-koordinat, y-koordinat, tid, overskrift, tilbagelagt afstand, antal hjulomdrejninger og kommandonummer for det aktuelle referencepunkt. Returnerer en liste med alle disse værdier som elementer.
<code>waypoint_prev</code>	Aflæser x-koordinat, y-koordinat, tid, overskrift, tilbagelagt afstand, antal hjulomdrejninger, det forrige referencepunkts kommandonummer.
<code>waypoint_eta</code>	Returnerer den anslåede tid til at køre til et referencepunkt.
<code>path_done()</code>	Returnerer en værdi på 0 eller 1 afhængigt af, om Roveren bevæger sig (0) eller har udført alle bevægelser (1).
<code>pathlist_x()</code>	Returnerer en liste med X-værdier fra begyndelsen til og med den aktuelle værdi for referencepunkt X.
<code>pathlist_y()</code>	Returnerer en liste med Y-værdier fra begyndelsen til og med den aktuelle værdi for referencepunkt Y.
<code>pathlist_time()</code>	Returnerer en liste over tiden i sekunder fra begyndelsen til og med den aktuelle værdi for tidspunkt for referencepunkt.
<code>pathlist_heading()</code>	Returnerer en liste med retninger fra begyndelsen til og med den aktuelle værdi for referencepunktsretning.

Menupunkt	Beskrivelse
pathlist_distance()	Returnerer en liste over de afstande, der tilbagelægges fra begyndelsen til og med det aktuelle referencepunkts afstandsværdi.
pathlist_revs()	Returnerer en liste over antallet af omdrejninger, der er tilbagelagt fra start til og med den aktuelle værdi for referencepunkts-omdrejninger.
pathlist_cmdnum()	Returnerer en liste med kommandonumre for stien.
waypoint_x()	Returnerer x-koordinat af aktuelt viapunkt.
waypoint_y()	Returnerer y-koordinat af det aktuelle referencepunkt.
waypoint_time()	Returnerer tid brugt på at rejse fra forrige til nuværende referencepunkt.
waypoint_heading()	Returnerer absolut retning for det aktuelle referencepunkt.
waypoint_distance()	Returnerer den tilbagelagte afstand mellem det forrige og det aktuelle referencepunkt.
waypoint_revs()	Returnerer antal omdrejninger, der er nødvendige for at bevæge sig mellem det forrige og det aktuelle referencepunkt.

## Indstillinger

Menupunkt	Beskrivelse
enhed/er	Indstilling for hastighed i gitterenheder pr. sekund.
m/s	Indstilling for hastighed i meter pr. sekund.
revs/s	Indstilling for hastighed i hjulomdrejninger pr. sekund.
Enheder	Indstilling for afstand i gitterenheder.
m	Indstilling for afstand i meter.
revs	Indstilling for afstand i hjulomdrejninger.
Grader	Indstilling for at dreje i grader.
radian	Indstilling for at dreje radianer.
gradianer	Indstilling for at dreje i gradianer.
med uret	Indstilling for angivelse af hjulretning.
mod uret	Indstilling for angivelse af hjulretning.

## Kommandoer

Disse kommandoer er samling af funktioner fra andre moduler samt fra TI Rover-modulet.

Menupunkt	Beskrivelse
<code>dvale(sekunder)</code>	Sætter programmet på pause i det angivne antal sekunder. Importeret fra tidsmodulet.
<code>text_at(række,"tekst","align")</code>	Viser "tekst" i plotteområdet ved specificeret "juster". Importeret fra <code>ti_plotlib</code> -modulet.
<code>cls()</code>	Rydder Shell-skærbilledet for ploting. Importeret fra <code>ti_plotlib</code> -modulet.
<code>while get_key() != "esc":</code>	Kører kommandoerne i "mens"-løkken, indtil der trykkes på "esc"-tasten.
<code>wait_until_done()</code>	Afbryder programmet midlertidigt, indtil Roveren afslutter den aktuelle kommando. Dette er en nyttig måde at synkronisere ikke-Rover-kommandoer med Rovers bevægelse.
<code>while not path_done()</code>	Kører kommandoerne i "while"-løkken, indtil Roveren er færdig med alle bevægelser. Funktionen <code>path_done()</code> returnerer en værdi på 0 eller 1 afhængigt af, om Roveren bevæger sig (0) eller er færdig med alle bevægelser (1).
<code>stilling(x,y)</code>	Angiver Roverens position på det virtuelle gitter til den angivne x,y-koordinat.
<code>position(x,y,heading,"enhed")</code>	Angiver Roverens position på det virtuelle gitter til den angivne x,y-koordinat og den virtuelle retning i forhold til den virtuelle x-akse, hvis der er angivet en retning (i enhederne for de angivne vinkler). Positive vinkler fra 0 til 360 antages at være mod uret fra den positive x-akse. Negative vinkler fra 0 til -360 antages at være med uret fra den positive x-akse.
<code>grid_origin()</code>	Indstiller RV som værende ved det aktuelle gitters begyndelsepunkt (0,0).
<code>grid_m_unit(scale_value)</code>	Angiver afstanden mellem det virtuelle gitter i meter pr. enhed (m/enhed) til den angivne værdi. 0,1 er standard m/enhed og oversættes til 1 enhed = 100 mm eller 10 cm eller 1 dm eller 0,1 m. Intervalleret for gyldig <code>scale_value</code> er fra 0,01 til 10,0.
<code>path_clear()</code>	Rydder alle allerede eksisterende sti- eller



Menupunkt	Beskrivelse
	viapunktoplysninger.
zero_gyro()	Nulstiller Roverens gyro til en vinkel på 0,0 og rydder venstre og højre hjulkodertællinger.

## Menu for kompleks matematik

Denne undermenu er placeret under **Flere moduler**.

Menupunkt	Beskrivelse
fra cmath import *	Importerer alle metoder fra cmath-modulet.
kompleks(reel,imag)	Returnerer et komplekst tal.
rekt(modulus,argument)	Konverterer polære koordinater til rektangulær form af et komplekst tal.
.real	Returnerer den reelle del af det komplekse tal.
.imag	Returnerer imaginær del af et komplekst tal.
polar()	Konverterer rektangulær form til polære koordinater for et komplekst tal.
phase()	Returnerer fase af et komplekst tal.
exp()	Returnerer $e^{**x}$ .
cos()	Returnerer cosinus af et komplekst tal.
sin()	Returnerer sinus af et komplekst tal.
log()	Returnerer den naturlige logaritme til et komplekst tal.
log10()	Returnerer 10-tals logaritmen af et komplekst tal.
sqrt()	Returnerer kvadratroden af et komplekst tal.

## Tidsmenu

Denne undermenu er placeret under **Flere moduler**.

Menupunkt	Beskrivelse
<code>from time import *</code>	Importerer alle metoder fra tidsmodulet.
<code>sleep(sekunder)</code>	Sætter programmet på pause i det angivne antal sekunder.
<code>clock()</code>	Returnerer den aktuelle processortid som et flydende tal, der udtrykkes i sekunder.
<code>localtime()</code>	Konverterer en tid, der er udtrykt i sekunder siden 1. januar 2000 til et ni-tupel indeholdende år, måned, måned, dag, minut, sekund, ugedag, år og sommertid (DST) flag. Hvis det valgfrie (sekunder) argument ikke er angivet, bruges uret i realtid.
<code>ticks_cpu()</code>	Returnerer en processorspecifik millisekund-tæller, der tæller stigende i millisekunder med et vilkårligt udgangspunkt. Brug <code>ticks_ms()</code> til at måle tid konsistent på tværs af forskellige systemer.
<code>ticks_diff()</code>	Måler perioden mellem fortløbende opkald til <code>ticks_cpu()</code> eller <code>ticks_ms()</code> . Denne funktion bør ikke bruges til at måle vilkårligt lange tidsperioder.

## Ti systemmenu

Denne undermenu er placeret under **Flere moduler**.

**Bemærk:** Når der oprettes et nyt program, der bruger dette modul, anbefales det at bruge **Deling af data** programtype. Dette vil sikre, at alle relevante moduler importeres.

Menupunkt	Beskrivelse
<code>from ti_system import *</code>	Importerer alle metoder (funktioner) fra <code>ti_system</code> modulet.
<code>recall_value("navn")</code>	Genkalder en foruddefineret OS-variabel (værdi) med navnet "navn".
<code>store_value("navn",værdi)</code>	Gemmer en Python-variabel (værdi) til en OS-variabel med navnet "navn".
<code>recall_list("navn")</code>	Genkalder en foruddefineret OS-liste med navnet "navn".
<code>store_list("navn", liste)</code>	Gemmer en Python-liste (liste) til en OS-listevariabel med navnet "navn".
<code>eval_function("navn",værdi)</code>	Evaluerer en foruddefineret OS-funktion ved den angivne værdi.
<code>get_platform()</code>	Returnerer "hh" for håndholdt og "dt" til skrivebordet.
<code>get_key()</code>	Returnerer en streng, der repræsenterer den tast, der er trykket på. '1'-tasten returnerer "1", "esc" returnerer "esc" osv. Når det aktiveres uden nogen parametre – <code>get_key()</code> – returnerer det straks. Når aktiveres med et parameter – <code>get_key(1)</code> – venter det, indtil der trykkes på en tast.
<code>get_mouse()</code>	Returnerer musekoordinater som to element-tupel, enten lærred-pixelpositionen eller (-1,-1), hvis det er uden for lærredet.
<code>while get_key() != "esc":</code>	Kører kommandoerne i "while"-løkken, indtil der trykkes på "esc"-tasten.
<code>clear_history()</code>	Rydder Shell-historikken.
<code>get_time_ms()</code>	Returnerer tid i millisekunder med millisekund præcision. Denne funktionalitet kan bruges til at beregne en varighed i stedet for at bestemme det faktiske klokkeslæt.

## Ti tegnmenu

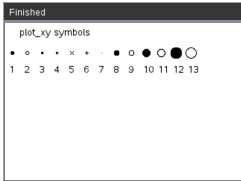
Denne undermenu er placeret under **Flere moduler**.

**Bemærk:** Når der oprettes et nyt program, der bruger dette modul, anbefales det at bruge **Geometri grafik** programtype. Dette vil sikre, at alle relevante moduler importeres.

Menupunkt	Beskrivelse
fra ti_draw import *	Importerer alle metoder fra ti_tegn-modulet.

### Form

Menupunkt	Beskrivelse
draw_line()	Tegner en linje fra den angivne x1,y1-koordinat til x2,y2.
draw_rect()	Tegner et rektangel fra de angivne (x,y)-koordinater med den angivne bredde og højde.
fill_rect()	Tegner et rektangel, der starter ved de angivne (x,y)-koordinater med den angivne bredde og højde, og fyldt med den angivne farve (ved brug af set_color eller sort, hvis ikke defineret).
draw_circle()	Tegner en cirkel, der starter ved de angivne (x,y)-koordinater for centrum og med den angivne radius.
fill_circle()	Tegner en cirkel, der starter ved de angivne (x,y)-koordinater for centrum og med den angivne bredde og højde, og fyldt med den angivne farve (ved brug af set_color eller sort, hvis ikke defineret).
draw_text()	Tegner en tekststreng, der starter ved de angivne (x,y)-koordinater.
draw_arc()	Tegner en bue fra de angivne (x,y)-koordinater med den angivne bredde, højde og vinkler.
fill_arc()	Tegner en bue fra de angivne (x,y)-koordinater med den angivne bredde, højde og vinkler, der er fyldt med den angivne farve (ved hjælp af set_color eller sort, hvis ikke defineret).
draw_poly()	Tegner en polygon ved hjælp af værdierne i den angivne x-liste,y-liste.
fill_poly()	Tegner en polygon ved hjælp af værdierne i den angivne x-liste,y-liste, der er fyldt med den angivne farve (ved hjælp af set_color eller sort, hvis ikke defineret).
plot_xy()	Tegner en figur ved hjælp af de angivne (x,y)-koordinater og det angivne tal fra 1-13, der repræsenterer forskellige figurer og symboler (se nedenfor).

Menupunkt	Beskrivelse
	

## Kontrol (Control)

Menupunkt	Beskrivelse
<code>clear()</code>	Rydder hele skærmen Kan bruges med <code>x,y,bredde,højde</code> -parametre til at rydde et eksisterende rektangel.
<code>clear_rect()</code>	Rydder rektanglet ved den angivne <code>x,y</code> -koordinat med den angivne bredde og højde.
<code>set_color()</code>	Indstiller farven på de(n) figur(er), der følger i programmet, indtil en anden farve er indstillet.
<code>set_pen()</code>	Indstiller kantens og typografiens tykkelse ved tegning af figurer (ikke relevant ved brug af udfyldningskommandoer).
<code>set_window()</code>	Indstiller størrelsen på det vindue, hvor der vil blive tegnet figurer. Denne funktion er nyttig til at ændre størrelsen på vinduet, så det passer til dataene eller til at ændre placeringen af <code>(0,0)</code> på tegnelærredet.
<code>get_screen_dim()</code>	Returnerer <code>xmax</code> og <code>ymax</code> for skærmdimensionerne.
<code>use_buffer()</code>	Aktiverer en buffer for gøre tegning hurtigere.
<code>paint_buffer()</code>	Viser den bufferede tegningsoutput. Funktionerne <code>use_buffer()</code> og <code>paint_buffer()</code> er nyttige i tilfælde, hvor visning af flere objekter på skærmen kan forårsage forsinkelser.

## Noter

- Standardkonfigurationen har `(0,0)` i øverste venstre hjørne af skærmen. Den positive `x`-akse peger mod højre og den positive `y`-akse peger mod bunden Dette kan ændres ved at bruge funktionen `set_window()`.
- Funktionerne i modulet `ti_draw` er kun tilgængelige på den håndholdte og i håndholdt visning på skrivebordet.

## Ti-billedmenu

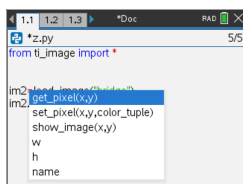
Denne undermenu er placeret under **Flere moduler**.

**Bemærk:** Når der oprettes et nyt program, der bruger dette modul, anbefales det at bruge **Billedbehandling** programtype. Dette vil sikre, at alle relevante moduler importeres.

Menupunkt	Beskrivelse
<code>from ti_image import *</code>	Importerer alle metoder fra <code>ti_billed</code> modulet.
<code>new_image(bredde,højde,(r,g,b))</code>	Opretter et nyt billede med den angivne bredde og højde til brug i Python-programmet. Farven på det nye billede defineres af værdierne <code>(r,g,b)</code> .
<code>load_image("navn")</code>	Indlæser det billede, der er angivet af "navn" til brug i Python-programmet. Billedet skal være en del af TNS-dokumentet enten i en Noter eller Grafer-applikation. Prompten "navn" viser billednavnene (hvis de er blevet navngivet tidligere) eller et tal, der angiver deres indsættelsesrækkefølge.
<code>copy_image(billede)</code>	Opretter en kopi af det billede, der er angivet af "billede"-variablen.

### Billedobjektets metoder

Yderligere funktioner relateret til billedobjekter er tilgængelige i Editor og Shell ved at skrive variabelnavnet efterfulgt af en `.`(dot).



- **get\_pixel(x,y):** Henter værdien `(r,g,b)` af pixlen på placeringen defineret af `(x,y)` koordinatparret.

```
px_val = get_pixel(100,100)
print(px_val)
```

- **set\_pixel(x,y,farve\_tupel):** Indstiller pixel ved placering `(x,y)` til farven angivet i `farve_tupel`.

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

Indstiller pixel ved `(100,100)` til `(0,0,255)` farve.

- **show\_image(x,y):** Viser billedet med det øverste venstre hjørne på lokationen `(x,y)`.

- **w, h, name:** Henter billedets bredde-, højde- og navn-parametre.

### For eksempel

```
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)
```



## Menuen Variabel

**Bemærk:** Disse lister omfatter ikke variabler defineret i andre TI-Nspire™-apps.

Menupunkt	Beskrivelse
Variabler: Aktuelt program	(Kun Editor) Viser en liste over globale funktioner og variabler defineret i det aktuelle program
Variabler: Senest kørte program	(Kun Shell) Viser en liste over globale funktioner og variabler defineret i det sidst kørte program
Variabler: Alle	(Kun Shell) Viser en liste over globale funktioner og variable fra både det sidst kørte program og eventuelle importerede moduler

## Tillæg

---

Python Nøgleord .....	48
Python-tastaturoversigt .....	49
Prøveprogrammer fra Python .....	51













## Python Nøgleord

Følgende nøgleord er indbygget i TI-Nspire™ Python-implementeringen.

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-tastaturoversigt

Når du indtaster kode i Editor eller i Shell, er tastaturet designet til at indsætte de relevante Python-handlinger eller åbne menuer for nem indtastning af funktioner, nøgleord, metoder, operatører osv.

Prøve	Kortlægning
	Åbner menuen Variable
	Indsætter = tegn
	Sletter karakter til venstre for markøren
	Ingen handling
	Indsætter = tegn
	Indsætter de(t) valgte symbol(er): <ul style="list-style-type: none"><li>• &gt;</li><li>• &lt;</li><li>• !=</li><li>• &gt;=</li><li>• &lt;=</li><li>• ==</li><li>• og</li><li>• —eller—</li><li>• not</li><li>•  </li><li>• &amp;</li><li>• ~</li></ul>
	Indsætter den valgte funktion: <ul style="list-style-type: none"><li>• sin</li><li>• cos</li><li>• tan</li><li>• atan2</li><li>• asin</li><li>• acos</li><li>• atan</li></ul>
	Viser tip
	Indsætter :=
	Indsætter **
	Ingen handling
	Indsætter **2

Prøve	Kortlægning
$\sqrt{\quad}$	Indsætter kvrod()
$\times$	Indsætter gange-tegn (*)
$\text{"}\text{"}$	Indsætter dobbelt anførselstegn (")
$\div$	Indsætter divisionstegn (/)
$\frac{\quad}{\quad}$	Ingen handling
$e^x$	Indsætter exp()
$\ln$	Indsætter log()
$10^x$	Indsætter $10^{**}$
$\log$	Indsætter log(værdi,base)
(	Indsætter (
)	Indsætter )
[ ]	Indsætter [ ]
{ }	Indsætter { }
(-)	Indsætter tegn for subtraktion (-)
$\approx$	Tilføjer en ny linje efter den aktuelle linje
EE	Indsætter E
$\text{?}$	Indsætter de(t) valgte symbol(er): <ul style="list-style-type: none"> <li>• ?</li> <li>• !</li> <li>• \$</li> <li>• °</li> <li>• '</li> <li>• %</li> <li>• "</li> <li>• :</li> <li>• ;</li> <li>• -</li> <li>• \</li> <li>• #</li> </ul>
$\pi$	Indsætter "pi"
$\text{E}$	Eksisterende flagadfærd
$\text{~}$	Tilføjer en ny linje efter den aktuelle linje

## Prøveprogrammer fra Python

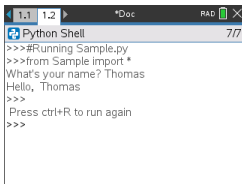
Brug følgende prøveprogrammer til at blive fortrolig med Python-metoder. De er også tilgængelige i **Kom godt i gang Python.tns**-filen, der er placeret i mappen **Eksempler**.

**Bemærk:** Hvis du kopierer og indsætter en prøvekode, der indeholder tabulatorindikatorer (••) til TI-Nspire™-softwaren, skal du erstatte disse forekomster med de faktiske faneindenter.

### Hej

```
# 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")
```



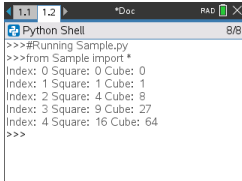
The screenshot shows a TI-Nspire Python Shell window with the following text:

```
Python Shell 7/7
>>>#Running Sample.py
>>>from Sample import *
What's your name? Thomas
Hello, Thomas
>>>
Press ctrl+R to run again
>>>
```

## Løkke-eksempel

```
# 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 titled "Python Shell" with a file path of "1.1 1.2" and a file name of "Sample.py". The shell displays 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
>>>
```

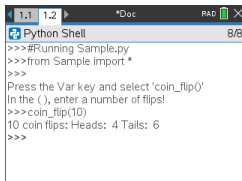
## Plat eller krone

```
# 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
>>>
```



## Plot

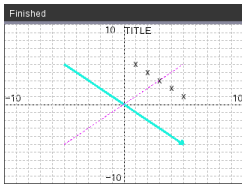
```
# Plotting example
import ti_plotlib as plt

# 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")
```



## Tegning

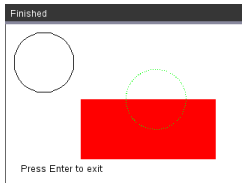
```
from ti_draw import *

# (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)

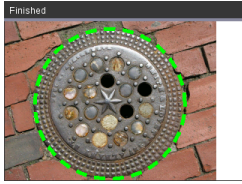
set_color(0,0,0)
draw_text(20,200,"Press Enter to exit")
```



## Billede

```
# 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

Dette program bruger Python til at styre TI-Innovator™ Hub, en programmerbar mikrocontroller. Hvis du kører programmet uden at vedhæfte en TI-Innovator™ Hub, vises en fejlmeddelelse.

For yderligere oplysninger om TI-Innovator™ Hub kan du besøge [education.ti.com](http://education.ti.com).

```
##### 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()
```

## Generelle oplysninger

### **Online hjælp**

[education.ti.com/eguide](http://education.ti.com/eguide)

Vælg dit land for at få flere produktoplysninger.

### **Kontakt TI-Support**

[education.ti.com/ti-cares](http://education.ti.com/ti-cares)

Vælg dit land for at finde ressourcer for teknisk support og andre supportressourcer.

### **Service og garanti**

[education.ti.com/warranty](http://education.ti.com/warranty)

Vælg dit land for at få oplysninger om varigheden og betingelserne for garantien, eller om produktservice.

Begrænset reklamationsret. Denne garanti påvirker ikke dine lovbestemte rettigheder.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243