

TI-Nspire™ Python Programmeringshåndbok

Viktig informasjon

Unntatt som uttrykkelig oppgitt i lisensen som medfølger et program, gir ikke Texas Instruments noen garantier, verken uttrykte eller implisitte, inkludert men ikke begrenset til implisitte garantier for salgbarhet eller egnethet til et bestemt formål, med hensyn til noen programmer eller bokmateriale, og gjør slike materialer tilgjengelige utelukkende på en "som det er"-basis ("as-is") Texas Instruments skal under ingen omstendigheter holdes ansvarlig overfor noen for spesielle, kollaterale, tilfeldige eller følgeskader i forbindelse med eller med bakgrunn i kjøp eller bruk av disse materialene, og det eneste og eksklusive økonomiske ansvaret til Texas Instruments, uavhengig av søksmålsform, skal ikke overskride prisen som er angitt i lisensen til programmet. Videre skal ikke Texas Instruments holdes økonomisk ansvarlig for noen form for krav mot bruk av dette materialet av noen annen part

© 2021 Texas Instruments Incorporated

«Python» og Python-logoene er varemerker eller registrerte varemerker for Python Software Foundation, brukt av Texas Instruments Incorporated med tillatelse fra stiftelsen.

Faktiske produkter kan være litt annerledes enn bilder.

Innhold

Komme i gang med Python-programmering	1
Python-moduler	1
Installere et Python-program som en modul	2
Python-arbeidsområder	3
Python-redigeringsprogram	3
Python Shell	6
Menykart for Python	10
Handlinger-menyen	11
Kjør-menyen	12
Verktøy-menyen	13
Rediger-menyen	14
Innebygde-menyen	15
Matematikk-menyen	18
Tilfeldig-menyen	20
TI PlotLib-menyen	21
TI Hub-menyen	23
TI Rover-menyen	32
Kompleks matematikk-menyen	39
Tid-menyen	40
TI System-menyen	41
TI Draw-menyen	42
TI Image-menyen	44
Variabel-menyen	46
Tillegg	47
Nøkkelord for Python	48
Tastetilordning for Python	49
Eksempelprogrammer i Python	51
Generell informasjon	58

Komme i gang med Python-programmering

Når du bruker Python med TI-Nspire™-produkter kan du:

- legge til Python-programmer i TNS-filer
- opprette Python-programmer ved hjelp av maler (sjabloner)
- kommunisere og dele data med andre TI-Nspire™-applikasjoner
- samhandle med TI-Innovator™ Hub og TI-Innovator™ Rover

TI-Nspire™ Python-implementeringen er basert på MicroPython, som er et lite delsett av standardbiblioteket i Python 3 som er utformet for å kjøre på mikrokontrollere. Den originale MicroPython-implementeringen er tilpasset for bruk av TI.

Merk: Noen numeriske svar kan variere fra Kalkulator-resultatene på grunn av forskjeller i de underliggende matematiske implementeringene.

Python er tilgjengelig på disse TI-Nspire™-produktene:

Håndholdte	Skrivebordsprogramvare
TI-Nspire™ CX II	TI-Nspire™ CX Premium Teacher Software
TI-Nspire™ CX II CAS	TI-Nspire™ CX CAS premium lærerprogramvare
TI-Nspire™ CX II-T	TI-Nspire™ CX Student Software
TI-Nspire™ CX II-T CAS	TI-Nspire™ CX CAS Student-programvare
TI-Nspire™ CX II-C	
TI-Nspire™ CX II-C CAS	

Merk: I de fleste tilfeller er funksjonaliteten identisk mellom den håndholdte enheten og programvarevisningene, men du kan se noen forskjeller. Denne veiledningen forutsetter at du bruker den håndholdte enheten eller Håndholdt-visningen i programvaren.

Python-moduler

TI-Nspire™ Python inneholder følgende moduler:

Standardmoduler	TI-moduler
Matematikk (math)	TI PlotLib (ti_plotlib)
Tilfeldig (random)	TI Hub (ti_hub)
Kompleks matematikk (cmath)	TI Rover (ti_rover)
Tid (time)	TI System (ti_system)
	TI Draw (ti_draw)
	TI Image (ti_image)

Merk: Hvis du har eksisterende Python-programmer opprettet i andre Python-utviklingsmiljøer, kan det hende du må redigere dem for å kjøre på TI-Nspire™ Python-løsningen. Moduler kan bruke forskjellige metoder, argumenter og rekkefølge for

metoder i et program sammenlignet med TI-modulene. Vær generelt oppmerksom på kompatibilitet ved bruk av en hvilken som helst versjon av Python og Python-moduler.

Når du overfører Python-programmer fra en plattform som ikke er fra TI til en TI-plattform ELLER fra ett TI-produkt til en annen, husk:

- Programmer som bruker kjernespråkfunksjoner og standardbiblioteker (math, random osv.) kan porteres uten endringer.
- Programmer som bruker plattformspeifikke biblioteker, f.eks. matplotlib for PC eller TI-moduler, vil kreve redigering før de kjøres på en annen plattform. Dette kan være tilfelle selv mellom TI-plattformer.

Som med enhver versjon av Python må du inkludere importert for å bruke funksjoner, metoder eller konstanter i en gitt modul. For å utføre cos()-funksjonen fra math-modulen, bruker du for eksempel følgende kommandoer:

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

For en liste over menyer med elementer og beskrivelser, se avsnittet [Menykart](#).

Installere et Python-program som en modul

Slik lagrer du Python-programmet ditt som en modul:

- I Rediger, velg **Handlinger> Installer som Python-modul**.
- I Shell, velg **Verktøy> Installer som Python-modul**.

Følgende skjer etter valget:

- Python-syntaksen blir kontrollert.
- Filen lagres og flyttes til PyLib-mappen.
- En dialog vises og bekrefter at filen er installert som en modul.
- Filen lukkes og modulen er klar til bruk.
- Modulnavnet vil bli lagt til **Flere moduler**-menyen med en **fra <module> import***-menyelement.

Hvis du planlegger å dele denne modulen med andre, anbefales det at du følger disse retningslinjene:

- Lagre kun én modul per TNS-fil.
- Modulnavnet samsvarer med navnet på TNS-filen (f.eks. «my_program»-modulen er i filen «my_program.tns»).
- Legg til en Notat-side før Python-redigeringsprogrammet som beskriver hensikten med modulen, versjonen og funksjonene.
- Bruk funksjonen ver() til å vise versjonsnummeret til modulen.
- (Valgfritt) Legg til en hjelpefunksjon for å vise listen over metodene i funksjonen.

Python-arbeidsområder

Det er to arbeidsområder for Python-programmering: Python-redigeringsprogram og Python Shell.

Python-redigeringsprogram	Python Shell
<ul style="list-style-type: none">• Opprett, rediger og lagre Python-programmer• Syntaksutheving og automatisk innrykk• Innebygde ledetekster for å veilede med funksjonsargumenter• Verktøytips for å vise gyldige verdier• <code>var</code>-tasten lister opp globale brukervariabler og funksjoner som er definert i det aktuelle programmet• Hurtigtaster	<ul style="list-style-type: none">• Kjør Python-programmer• Praktisk for testing av små kodefragmenter• Interaksjon med Shell-historikk for å velge tidligere inndata og utdata for gjenbruk• <code>var</code>-tasten lister opp globale brukervariabler som er definert i programmet som ble sist kjørt i det gitte problemet

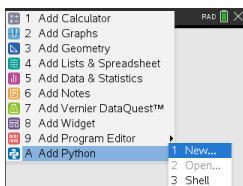
Merk: Flere Python-programmer og Shell kan legges til i et problem.

Python-redigeringsprogram

Med Python-redigeringsprogram kan du opprette, redigere og lagre Python-programmer.

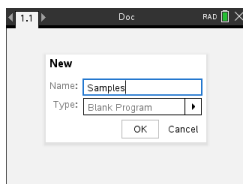
Legge til en side i et Python-redigeringsprogram

Hvis du vil legge til en ny side i et Python-redigeringsprogram i den aktuelle oppgaven, trykker du på `menu` og velger **Legg til Python > Ny**.

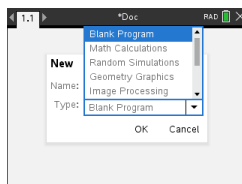


Du kan opprette et tomt program, eller du kan velge en mal.

Tomt program



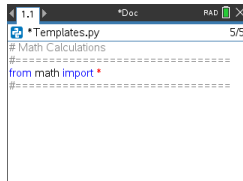
Mal



Når programmet er opprettet, vises Python-redigeringsprogram. Hvis du valgte en mal, legges de nødvendige importutsagnene automatisk til (se nedenfor).

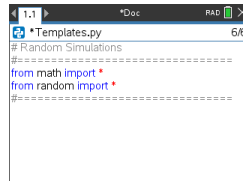
Merk: Du kan ha flere programmer i én enkelt TNS-fil akkurat som andre apper. Hvis Python-programmet er ment å brukes som en modul, kan TNS-filen lagres i mappen PyLib. Denne modulen kan deretter brukes i andre programmer og dokumenter.

Matematiske beregninger



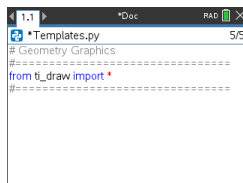
```
*Templates.py 5/5
# Math Calculations
#=====
from math import *
#=====
```

Tilfældige simuleringer



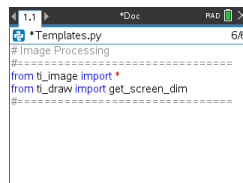
```
*Templates.py 6/6
# Random Simulations
#=====
from math import *
from random import *
#=====
```

Geometrigrafikk




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

Bildebehandling



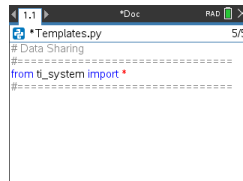
```
*Templates.py 6/6
# Image Processing
#=====
from ti_image import *
from ti_draw import get_screen_dim
#=====
```

Plotting (x,y) & og tekst



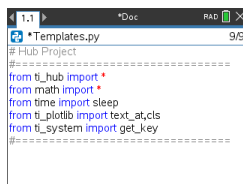
```
*Templates.py 5/5
# Plotting (x,y) & Text
#=====
import ti_plotlib as plt
#=====
```

Datadeling



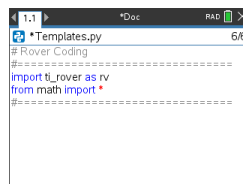
```
*Templates.py 5/5
# Data Sharing
#=====
from ti_system import *
#=====
```

TI-Innovator Hub-prosjekt



```
*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
#=====
```

TI-Rover-koding



```
*Templates.py 6/6
# Rover Coding
#=====
import ti_rover as rv
from math import *
#=====
```

Åpne et Python-program

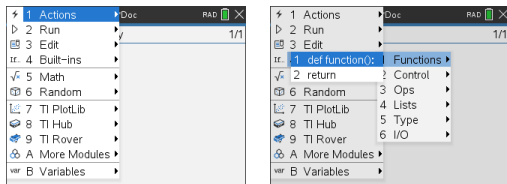
Hvis du vil åpne et eksisterende Python-program, trykker du på **[doc]** og velger **Sett inn > Legg til Python > Åpne**. Dette vil vise en liste over programmer som er lagret i TNS-filen.

Hvis redigeringsprogrammsiden som ble brukt til å opprette programmet er slettet, er programmet fremdeles tilgjengelig i TNS-filen.

Arbeide i Python-redigeringsprogram

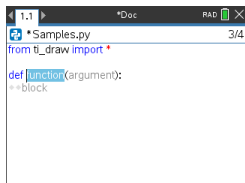
Å trykke på **[menu]** viser Dokumentverktøy-menyen. Med disse menyalternativene kan du legge til, flytte og kopiere kodeblokker for programmet ditt.

Dokumentverktøy-meny

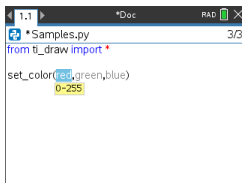


Elementer valgt fra modulmenyene vil automatisk legge til en kodemal i redigeringsprogrammet med ledetekster på linjen for hver del av funksjonen. Du kan navigere fra ett argument til det neste ved å trykke på **[tab]** (forover) eller **[shift]+[tab]** (bakover). Verktøytips eller pop-up-lister blir vist når de er tilgjengelige for å hjelpe deg med å velge riktige verdier.

Ledetekst på linjen



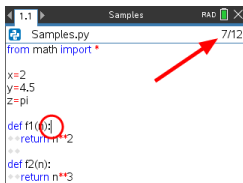
Verktøytips



Pop-up-lister

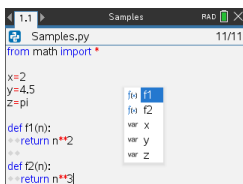


Tallene til høyre for programnavnet viser gjeldende linjenummer og det totale antallet linjer i programmet.



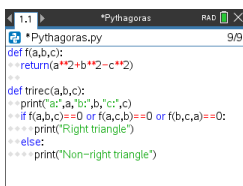
```
1.1 | Samples.py | RAD | 7/12
Samples.py
from math import *
x=2
y=4.5
z=pi
def f1(n):
    =>return n**2
def f2(n):
    =>return n**3
```

Globale funksjoner og variabler definert i linjene over aktuell markørposisjon kan settes inn ved å trykke på **var** og velge fra listen.



```
1.1 | Samples.py | 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 legger til kode i programmet, viser redigeringsprogrammet nøkkelord, operatører, kommentarer, strenger og innrykk i forskjellige farger for å hjelpe deg med å identifisere de ulike elementene.



```
1.1 | *Pythagoras.py | 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")
```

Lagre og kjøre programmer

Når du er ferdig med programmet, trykker du på **menu** og velger **Kjør > Kontroller syntaks & Lagre**. Dette vil kontrollere syntaksen til Python-programmet og lagre det i TNS-filen.

Merk: Hvis du har ulagrede endringer i programmet, vises en stjerne ved siden av programnavnet.



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

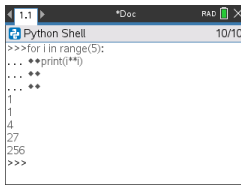
Hvis du vil kjøre programmet, trykker du på **menu** og velger **Kjør > Kjør**. Dette vil kjøre det gjeldende programmet i neste Python Shell-side eller en ny hvis neste side ikke er Shell.

Merk: Når programmet kjøres, blir syntaksen automatisk sjekket og programmet lagres.

Python Shell

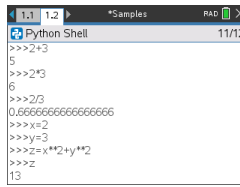
Python Shell er tolken som kjører Python-programmer, andre biter av Python-kode eller enkle kommandoer.

Python-kode



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

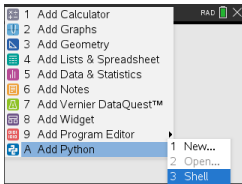
Enkle kommandoer



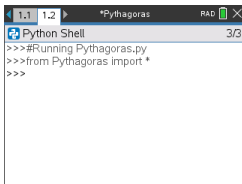
```
Python Shell 11/12
>>>2+3
5
>>>2*3
6
>>>2/3
0.6666666666666666
>>>x=2
>>>y=3
>>>z=x**2+y**2
>>>z
13
```

Legge til en side med Python Shell

Hvis du vil legge til en ny Python Shell-side i den aktuelle oppgaven, trykker du på **menu** og velger **Legg til Python > Shell**.



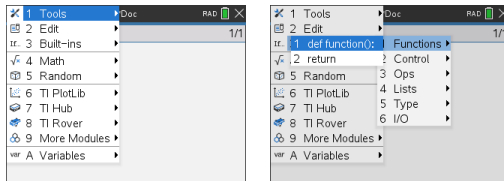
Python Shell kan også startes fra Python-redigeringsprogram ved å kjøre et program ved å trykke på **menu** og velge **Kjør > Kjør**.



Arbeide i Python Shell

Å trykke på **menu** viser Dokumentverktøy-menyen. Med disse menyalternativene kan du legge til, flytte og kopiere kodeblokker.

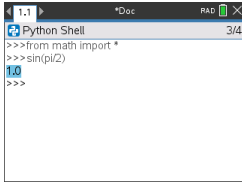
Dokumentverktøy-meny



Merk: Hvis du bruker en metode fra en av de tilgjengelige modulene, må du først sørge for å kjøre en setning for å importere modul, som i ethvert Python-kodemiljø.

Interaksjon med Shell-utdata ligner på Kalkulator-appen der du kan velge og kopiere tidligere inndata og utdata for bruk andre steder i Shell, redigeringsprogram eller andre apper.

Bruk opp-pilen for å velge, deretter kopierer du og limer inn på ønsket sted

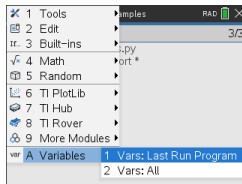


```
Python Shell 3/3
>>>from math import *
>>>sin(pi/2)
1.0
>>>
```

Globale funksjoner og variabler fra programmet som ble kjørt sist kan settes inn ved å trykke på **var** eller **(ctrl)+L** og velge fra listen eller trykke på **(menu)** og velge **Variabler > Vars: Programmet som ble kjørt sist**.

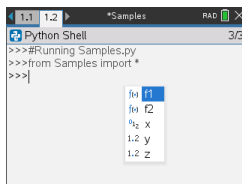
Hvis du vil velge fra en liste over globale funksjoner og variabler fra både programmet som ble kjørt sist og eventuelle importerte moduler, trykker du på **(menu)** og velger **Variabler > Vars: Alle**.

Variabler-menyen

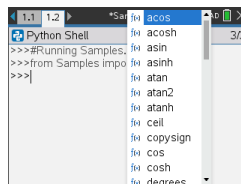


```
1 Tools
2 Edit
3 Built-ins
4 Math
5 Random
6 TI PlotLib
7 TI Hub
8 TI Rover
9 More Modules
A Variables 1 Vars: Last Run Program
2 Vars: All
```

Variabler fra programmet som ble kjørt sist *Alle variabler*



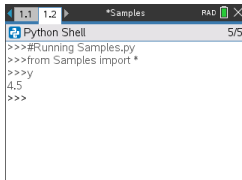
```
Python Shell 3/3
>>>#Running Samples.py
>>>from Samples import *
>>>]
f1 f1
f2 f2
%2 x
1.2 y
1.2 z
```



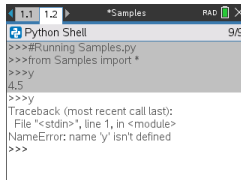
```
Python Shell 3/3
>>>#Running Samples.py
>>>from Samples impo
>>>]
f1 acos
f2 asin
f3 asinh
f4 atan
f5 atan2
f6 atanh
f7 ceil
f8 copysign
f9 cos
f10 cosh
f11 degrees
```

Alle sidene med Python Shell i samme problem deler samme tilstand (brukerdefinerte og importerte variabeldefinisjoner). Når du lagrer eller kjører et Python-program i problemet, eller trykker på **(menu)** og velger **Verktøy > Reinitialiser Shell**, vil Shell-historikken deretter ha en grå bakgrunn som angir at den forrige tilstanden ikke lenger er gyldig.

Før du lagrer eller initialiserer på nytt *Etter at du lagrer eller initialiserer på nytt*



```
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
>>>
```

Merk: Alternativet **menu** **Verktøy** > **Slett historikk** sletter skjermbildet for tidligere aktivitet i Shell, men variabler er fremdeles tilgjengelige.

Meldinger

Feil og andre informasjonsmeldinger kan vises mens du er i en Python-økt. Hvis en feil vises i Shell når et program kjøres, vises et programlinjenummer. Trykk på **ctrl** **menu** og velg **Gå til Python-redigeringsprogram**. I redigeringsprogrammet trykker du på **menu**, deretter velger du **Rediger** > **Gå til linje**. Skriv inn linjenummeret og trykk på **enter**. Markøren vises på det første tegnet i linjen der feilen oppstod.

Avbryte et program som kjører

Mens et program eller en funksjon kjører, vises opptatt-pekeren **☹**.

- ▶ For å stoppe programmet eller funksjonen,
 - Windows®: Trykk på **F12**-tasten.
 - Mac®: Trykk på **F5**-tasten.
 - Grafregner: Trykk på **on**-tasten.

Menykart for Python

Denne delen lister opp alle menyene og menyelementene for Python-redigeringsprogram og Shell og en kort beskrivelse for hver av dem.

Merk: For menyelementer som har hurtigtaster, bør Mac®-brukere sette inn \mathcal{H} (Cmd) overalt hvor **Ctrl** brukes. Du finner en komplett liste over hurtigtaster for TI-Nspire™ håndholdt enhet og programvare i TI-Nspire™-teknologi eGuide.

Handlinger-menyen	11
Kjør-menyen	12
Verktøy-menyen	13
Rediger-menyen	14
Innebygde-menyen	15
Matematikk-menyen	18
Tilfeldig-menyen	20
TI PlotLib-menyen	21
TI Hub-menyen	23
TI Rover-menyen	32
Kompleks matematikk-menyen	39
Tid-menyen	40
TI System-menyen	41
TI Draw-menyen	42
TI Image-menyen	44
Variabel-menyen	46

Handlinger-menyen

Merk: Dette gjelder kun for redigeringsprogrammet.

Objekt	Beskrivelse
Ny	Åpner dialogboksen Ny , der du legger inn et navn og velger en type for det nye programmet.
Åpne	Åpner en liste over programmer som er tilgjengelige i det gjeldende dokumentet.
Opprett kopi	Åpner dialogboksen Opprett kopi , der du kan lagre det gjeldende programmet under et annet navn.
Gi nytt navn	Åpner dialogboksen Gi nytt navn , der du kan gi nytt navn til det gjeldende programmet.
Lukk	Lukker det gjeldende programmet.
Innstillinger	Åpner dialogboksen Innstillinger , der du kan endre skriftstørrelsen for både redigeringsprogrammet og Shell.
Installer som Python-modul	Kontrollerer Python-syntaksen til den gjeldende TNS-filen og flytter den til PyLib-mappen.

Kjør-menyen

Merk: Dette gjelder kun for redigeringsprogrammet.

Objekt	hurtigtaster	Beskrivelse
Kjør	Ctrl+R	Sjekker syntaks, lagrer program og kjører i et Python Shell.
Kontroller syntaks & og lagre	Ctrl+B	Kontrollerer syntaks og lagrer program.
Gå til Shell	Ikke tilgjengelig	Flytter fokus til Shell relatert til det gjeldende programmet eller åpner en ny Shell-side ved siden av redigeringsprogrammet.

Verktøy-menyen

Merk: Dette gjelder kun Shell.

Objekt	hurtigtaster	Beskrivelse
Kjør siste program på nytt	Ctrl+R	Kjører det siste programmet på nytt relatert til gjeldende Shell.
Gå til Python-redigeringsprogram	Ikke tilgjengelig	Åpner redigeringsprogrammet relatert til gjeldende Shell.
Kjør	Ikke tilgjengelig	Åpner en liste over programmer som er tilgjengelige i det gjeldende dokumentet. Etter valget, kjøres det valgte programmet.
Slett logg	Ikke tilgjengelig	Sletter historikken i gjeldende Shell, men initialiserer ikke Shell på nytt.
Reinitialiser Shell	Ikke tilgjengelig	Tilbakestill tilstanden til alle åpne Shell-sider i gjeldende problem. Ingen definerte variabler og importerte funksjoner er lenger tilgjengelige.
dir()	Ikke tilgjengelig	Viser listen over funksjoner i den spesifiserte modulen når den brukes etter import-setningen.
From PROGRAM import *	Ikke tilgjengelig	Åpner en liste over programmer som er tilgjengelige i det gjeldende dokumentet. Etter valg limes den importerte setningen inn i Shell.
Installer som Python-modul	Ikke tilgjengelig	Aktivert kun for moduler i binært format. Flytter den gjeldende TNS-filen til PyLib-mappen.

Rediger-menyen

Merk: Ctrl+A velger alle linjer med kode eller utdata for utklipping eller sletting (kun redigeringsprogram), eller kopiering og liming (redigeringsprogram og Shell).

Objekt	hurtigtaster	Beskrivelse
Innrykk	TAB*	Lager tekstinnrykk på gjeldende linje eller valgte linjer. * Hvis det er ufullstendige ledetekster på linjen, vil TAB navigere til neste ledetekst.
Dedent	Skift+TAB**	Lager tekstutrykk på gjeldende linje eller valgte linjer. ** Hvis det er ufullstendige ledetekster på linjen, vil Skift+TAB navigere til forrige ledetekst.
Legg til / fjern kommentar	Ctrl+T	Legger til / fjerner kommentarsymbol til/fra begynnelsen av gjeldende linje.
Sett inn flerlinjestreng	Ikke tilgjengelig	(Kun redigeringsprogram) Setter inn flerlinjers strengmal.
Finn	Ctrl+F	(Kun redigeringsprogram) Åpner dialogboksen Finn og søker etter den angitte strengen i gjeldende program.
Erstatt	Ctrl+H	(Kun redigeringsprogram) Åpner dialogboksen Erstatt og søker etter den angitte strengen i gjeldende program.
Gå til linje	Ctrl+G	(Kun redigeringsprogram) Åpner dialogboksen Gå til linje og hopper til den angitte linjen i det aktuelle programmet.
Begynnelsen av linjen	Ctrl+8	Flytter markøren til begynnelsen av gjeldende linje.
Slutten av linjen	Ctrl+2	Flytter markøren til slutten av gjeldende linje.
Gå til toppen	Ctrl+7	Flytter markøren til begynnelsen av den første linjen i programmet.
Gå til bunnen	Ctrl+1	Flytter markøren til slutten av den siste linjen i programmet.

Innebygde-menyen

funksjoner

Objekt	Beskrivelse
def function():	Definerer en funksjon avhengig av spesifiserte variabler.
return	Definerer verdien som produseres av en funksjon.

Kontroll

Objekt	Beskrivelse
if..	Betingelsesuttrykk.
if..else..	Betingelsesuttrykk.
if..elif..else..	Betingelsesuttrykk.
for indeks in range(størrelse):	Itererer over et intervall.
for indeks in range(start,stop):	Itererer over et intervall.
for indeks in range(start,stop,trinn):	Itererer over et intervall.
for indeks in liste:	Itererer over listeelementer.
while..	Utfører erklæringer i en kodeblokk til en betingelse evalueres til False.
elif:	Betingelsesuttrykk.
else:	Betingelsesuttrykk.

Operatører

Objekt	Beskrivelse
x=y	Angir variabelverdi.
x==y	Limer inn sammenligningsoperatoren er lik (==).
x!=y	Limer inn sammenligningsoperatoren er ikke lik (!=).
x>y	Limer inn sammenligningsoperatoren større enn (>).
x>=y	Limer inn sammenligningsoperatoren større enn eller lik (>=).
x<y	Limer inn sammenligningsoperatoren mindre enn (<).
x<=y	Limer inn sammenligningsoperatoren mindre enn eller lik (<=).

Objekt	Beskrivelse
og	Limer inn den logiske operatoren og (and).
—eller—	Limer inn den logiske operatoren eller (or).
ikke	Limer inn den logiske operatoren ikke (not).
Sann	Limer inn boolsk verdi True.
Usann	Limer inn boolsk verdi False.

Lister

Objekt	Beskrivelse
[]	Limer inn hakeparenteser ([]).
list() (liste)	Konverterer sekvens til «list»-typen.
len()	Returnerer antall elementer i listen.
maks ()	Returnerer maksimumsverdien i listen.
min() (minimum)	Returnerer minimumsverdien i listen.
.append()	Metoden legger til et element i en liste.
.remove()	Metoden fjerner den første forekomsten av et element fra en liste.
range(start,stop,trinn)	Returnerer et sett med tall.
for indeks in range(start,stop,trinn)	Brukes til å iterere over et intervall.
.insert()	Metoden legger til et element i den angitte posisjonen.
.split()	Metoden returnerer en liste med elementer separert med spesifisert skilletegn.
sum()	Returnerer summen av elementene i en liste.
sorted()	Returnerer en sortert liste.
.sort()	Metoden sorterer en liste på stedet.

Type

Objekt	Beskrivelse
int()	Returnerer en heltallsdel.
float()	Returnerer en flyttallsverdi.

Objekt	Beskrivelse
round(x,nsifre)	Returnerer et flyttall avrundet til det angitte antallet sifre.
str()	Returnerer en streng.
complex()	Returnerer et komplekst tall.
type()	Returnerer objekttypen.

I/U

Objekt	Beskrivelse
print()	Viser argument som streng.
input()	Ber brukeren om inndata.
eval()	Evaluerer et uttrykk representert som en streng.
.format()	Metoden formaterer den spesifiserte strengen.

Matematikk-menyen

Merk: Når du oppretter et nytt program som bruker denne modulen, anbefales det å bruke programtypen **Matematiske beregninger**. Dette vil sikre at alle relevante moduler importeres.

Objekt	Beskrivelse
from math import *	Importerer alle metoder (funksjoner) fra math-modulen.
fabs()	Returnerer absolutt verdi av et reelt tall.
sqrt() (kvdrt)	Returnerer kvadratroten av et reelt tall.
exp()	Returnerer $e^{**}x$.
pow(x,y)	Returnerer x opphøyd til potens y .
log(x,grunntall)	Returnerer $\log_{\text{grunntall}}(x)$. $\log(x)$ uten grunntall returnerer den naturlige logaritmen x .
fmod(x,y)	Returnerer modulusverdi av x og y . Brukes når x og y er flyttall.
ceil()	Returnerer det minste heltallet større enn eller lik et reelt tall.
floor () (nedre)	Returnerer det største heltallet mindre enn eller lik et reelt tall.
trunc()	Avkorter et reelt tall til et heltall.
fexp()	Returnerer et par (y,n) der $x == y * 2^{**}n$.

Konst

Objekt	Beskrivelse
E	Returns value for the constant e.
pi	Returns value for the constant pi.

Trigonometri

Objekt	Beskrivelse
radians()	Konverterer vinkel i grader til radianer.
degrees()	Konverterer vinkel i radianer til grader.
sin()	Returnerer sinus av argument i radianer.

Objekt	Beskrivelse
<code>cos ()</code>	Returnerer cosinus av argument i radianer.
<code>tan()</code>	Returnerer tangens av argument i radianer.
<code>asin()</code>	Returnerer arcus sinus til argument i radianer.
<code>acos()</code>	Returnerer arcus cosinus av argument i radianer.
<code>atan()</code>	Returnerer arcus tangens til argument i radianer.
<code>atan2(y,x)</code>	Returnerer arcus tangens til y/x i radianer.

Tilfeldig-menyen

Merk: Når du oppretter et nytt program som bruker denne modulen, anbefales det å bruke programtypen **Tilfeldige simuleringer**. Dette vil sikre at alle relevante moduler importeres.

Objekt	Beskrivelse
<code>from random import *</code>	Importerer alle metoder fra random-modulen.
<code>random()</code>	Returnerer et flyttall mellom 0 og 1,0.
<code>uniform(min,maks)</code>	Returnerer et tilfeldig tall x (flyttall) slik at $\text{min} \leq x \leq \text{maks}$.
<code>randint(min,maks)</code>	Returnerer et tilfeldig heltall mellom min og maks.
<code>choice(sekvens)</code>	Returnerer et tilfeldig element fra en ikke-tom sekvens.
<code>randrange(start,stop,trinn)</code>	Returnerer et tilfeldig tall fra start til slutt etter trinn.
<code>seed()</code>	Initialiserer slumptallsgenerator.

Ti PlotLib-menyen

Merk: Når du oppretter et nytt program som bruker denne modulen, anbefales det å bruke programtypen **Plotting (x,y) & og tekst**. Dette vil sikre at alle relevante moduler importeres.

Objekt	Beskrivelse
<code>import ti_plotlib as plt</code>	Importerer alle metoder (funksjoner) fra <code>ti_plotlib</code> -modulen i navneområdet «plt». Som følge av dette vil alle funksjonsnavn som er limt inn fra menyene, bli innledet med «plt.».

Oppsett

Objekt	Beskrivelse
<code>cls()</code>	Sletter plottlerretet.
<code>grid(x-skala,y-skala,"stil")</code>	Viser et rutenett med spesifisert skala for x- og y-akser.
<code>window(xmin,xmaks,ymin,ymaks)</code>	Definerer plottvinduet ved å tilordne det spesifiserte horisontale intervallet (<code>xmin</code> , <code>xmaks</code>) og vertikale intervallet (<code>ymin</code> , <code>ymaks</code>) til det tilordnede plottingsområdet (piksler).
<code>auto_window(x-liste,y-liste)</code>	Autoskalerer plottvinduet for å tilpasse dataområdene innenfor x-liste og y-liste som er spesifisert i programmet før <code>auto_window()</code> .
<code>axes("modus")</code>	Viser akser på angitt vindu i ploteområdet.
<code>labels("x-etikett","y-etikett",x,y)</code>	Viser etikettene «x-etikett» og «y-etikett» på plottaksene ved radposisjoner <code>x</code> og <code>y</code> .
<code>title("tittel")</code>	Viser «tittel» sentrert på øverste linje i vinduet.
<code>show_plot()</code>	Viser utdata for bufret tegning. Funksjonene <code>use_buffer()</code> og <code>show_plot()</code> er nyttige i tilfeller der visning av flere objekter på skjermen kan forårsake forsinkelser (ikke nødvendig i de fleste tilfeller).
<code>use_buffer()</code>	Aktiverer en buffer utenfor skjermen for å øke hastigheten på tegning.

Skilletegn

Objekt	Beskrivelse
<code>color(rød,grønn,blå)</code>	Angir fargen for all følgende grafikk/plotting.
<code>cls()</code>	Sletter plottlerretet.
<code>show_plot()</code>	Utfører visningen av plottet som konfigurert i programmet.
<code>scatter(x-liste,y-liste,"merke")</code>	Plotter en sekvens med par i rekkefølge fra (x-liste,y-liste) med den angitte merkestilen.
<code>plot(x-liste,y-liste,"merke")</code>	Tegner en linje ved hjelp av par i rekkefølge fra spesifisert x-liste og y-liste.
<code>plot(x,y,"merke")</code>	Plotter et punkt med koordinater x og y med den angitte merkestilen.
<code>line(x1,y1,x2,y2,"modus")</code>	Plotter et linjesegment fra (x1,y1) til (x2,y2).
<code>lin_reg(x-liste,y-liste,"visning")</code>	Beregner og tegner lineær regresjonsmodell, $ax+b$, av x-liste,y-liste.
<code>pen("størrelse","stil")</code>	Angir utseendet til alle følgende linjer til neste <code>pen()</code> kjøres.
<code>text_at(rad,"tekst","juster")</code>	Viser teksten «tekst» i området der det er angitt «juster».

Egenskaper

Objekt	Beskrivelse
<code>xmin</code>	Spesifisert variabel for vinduargumenter definert som <code>plt.xmin</code> .
<code>xmax</code>	Spesifisert variabel for vinduargumenter definert som <code>plt.xmax</code> .
<code>ymin</code>	Spesifisert variabel for vinduargumenter definert som <code>plt.ymin</code> .
<code>ymax</code>	Spesifisert variabel for vinduargumenter definert som <code>plt.ymax</code> .
<code>M</code>	Etter at <code>plt.linreg()</code> er kjørt i et program, blir de beregnede verdiene for stigningstall, m, og skjæringspunkt, b, lagret i <code>plt.m</code> og <code>plt.b</code> .
<code>b</code>	Etter at <code>plt.linreg()</code> er kjørt i et program, blir de beregnede verdiene for stigningstall, a, og skjæringspunkt, b, lagret i <code>plt.a</code> og <code>plt.b</code> .

Ti Hub-menyen

Merk: Når du oppretter et nytt program som bruker denne modulen, anbefales det å bruke programtypen **Hub Project**. Dette vil sikre at alle relevante moduler importeres.

Objekt	Beskrivelse
fra ti_hub-import *	Importerer alle metoder fra ti_hub-modulen.

Enheter som er innebygget i hub > Fargeeffekt

Objekt	Beskrivelse
rgb(rød,grønn,blå)	Angir fargen for RGB LED-lys.
blink(frekvens,tid)	Angir frekvensen for blink og varigheten for den valgte fargen.
off()	Slår av RGB LED.

Enheter som er innebygget i hub > Lyseffekt

Objekt	Beskrivelse
on()	Slår på LED-lyset.
off()	Slår av LED-lyset.
blink(frekvens,tid)	Angir frekvensen for blink og varigheten for LED-lys.

Enheter som er innebygget i hub > Lyduttgang

Objekt	Beskrivelse
tone(frekvens,tid)	Spiller av en tone for den angitte frekvensen i det angitte tidsrommet.
note("note",tid)	Spiller av den spesifiserte merknaden for det angitte tidsrommet. Noten spesifiseres ved bruk av notenavn og en oktav. For eksempel: A4, C5. Notenavnene er C, CS, D, DS, E, F, FS, G, GS, A, AS og B. Oktav-tallene varierer fra 1 til 9 (inkludert).
tone (frekvens,tid,tempo)	Spiller av en tone for den angitte frekvensen i det angitte tidsrommet og tempoet.

Objekt	Beskrivelse
	Tempotallene definerer antall pip per sekund fra 0 til 10 (inkludert).
note("note",tid,tempo)	Spiller av den angitte noten for det angitte tidsrommet og tempoet. Noten spesifiseres ved bruk av notenavn og en oktav. For eksempel: A4, C5. Notenavnene er C, CS, D, DS, E, F, FS, G, GS, A, AS og B. Oktav-tallene varierer fra 1 til 9 (inkludert). Tempotallene varierer fra 0 til 10 (inkludert).

Enheter som er innebygget i hub > Lysstyrkeinnngang

Objekt	Beskrivelse
måling()	Leser den innebygde sensoren for LYSSTYRKE (lysnivå) og returnerer en måling. Standardområdet er 0 til 100. Dette kan endres med funksjonen range().
range(min,maks)	Angir intervallet for avleste målinger fra lysnivåsensoren. Hvis begge mangler eller er satt til en verdi av Ingen, stilles standard lysstyrkeområde fra 0 til 100.

Legg til innmatingsenhet

Denne menyen har en liste over sensorene (innmatingsenhetene) som støttes av ti_hub-modulen. Alle menyelementene vil lime inn navnet på objektet og forvente en variabel og en port som brukes med sensoren. Hver sensor har en measurement()-metode som returnerer verdien til sensoren.

Objekt	Beskrivelse
DHT (digital fuktighet og & temperatur)	Returnerer en liste som består av gjeldende temperatur, fuktighet, sensortype og siste avlesingsstatus i hurtigbuffer.
Avstandsmåler	Returnerer den gjeldende avstandsmålingen fra den spesifiserte ultralydmåleren. <ul style="list-style-type: none"> measurement_time() – Returnerer tiden det tar for ultralydsignalet å nå objektet («flyvningstiden»).
Lysnivå	Returnerer lysstyrkenivået fra den eksterne lysnivåsensoren (lysstyrke).

Objekt	Beskrivelse
temperatur	<p>Returnerer temperaturavlesningen fra den eksterne temperatursensoren.</p> <p>Standardkonfigurasjonen er å støtte Seeed-temperatursensoren i portene IN 1, IN 2 eller IN 3.</p> <p>For å bruke TI LM19-temperatursensoren fra TI-Innovator™ Hub-koblingskortet for plugger, rediger porten til BB-pinnen som er i bruk og bruk et valgfritt argument "TIANALOG".</p> <p>Eksempel: <code>mylm19=temperature("BB 5","TIANALOG")</code></p>
Fuktighet	Returnerer fuktighetssensoravlesningen.
Magnetisk	<p>Oppdager tilstedeværelsen av et magnetfelt.</p> <p>Terskelverdien for å bestemme om feltet er til stede blir satt med funksjonen <code>trigger()</code>.</p> <p>Standardverdien for terskelen er 150.</p>
Vernier	<p>Leser verdien fra Vernier analog sensor spesifisert i kommandoen.</p> <p>Kommandoen støtter følgende Vernier-sensorer:</p> <ul style="list-style-type: none"> • temperature – Temperatursensor i rustfritt stål. • lightlevel – TI lysnivåsensor. • pressure – Original gasstrykksensor • pressure – Nyere gasstrykksensor. • pH – pH-sensor. • force10 – ±10 N-innstilling, dobbelt kraftsensor. • force50 – ±50 N-innstilling, dobbelt kraftsensor. • accelerometer – Lav-G akselerometer. • generic – Tillater innstilling av andre sensorer som ikke støttes direkte ovenfor, og bruk av API-et <code>calibrate()</code> ovenfor for å angi ligningskoeffisienter.
Analog inngang	Støtter bruk av generiske enheter med analog inngang.
Digital inngang	Returnerer den nåværende tilstanden til den digitale pinnen som er koblet til

Objekt	Beskrivelse
	DIGITAL-objektet, eller den hurtigbufrede tilstanden til den digitale utgangsverdien som sist er SET på objektet.
Potensiometer	Støtter en potensiometersensor. Sensorens intervall kan endres av funksjonen <code>range()</code> .
Varmeleder	Leser varmeledersensorer. Standardkoeffisientene er utformet for å samsvare med varmelederen som er inkludert i koblingsbordpakken til TI-Innovator™ Hub, når den brukes med en 10 KΩ fast resistor. Et nytt sett med kalibreringskoeffisienter og referansemotstand for varmeleder kan konfigureres med <code>calibrate()</code> -funksjonen.
Lydstyrke	Støtter lydstyrkesensorer.
Fargeinput	Gir grensesnitt til en I2C-tilkoblet fargeinputsensor. <code>bb_port</code> -pinnen brukes i tillegg til I2C-porten for å kontrollere LED-lampen på fargesensoren. <ul style="list-style-type: none"> • color_number(): Returnerer en verdi fra 1 til 9 som representerer fargen sensoren oppdager. Tallene representerer fargene i henhold til følgende tilordning: <ol style="list-style-type: none"> 1: Rød 2: Grønn 3: Blå 4: Cyan 5: Magenta 6: Gul 7: Svart 8: Hvit 9: Grå • red(): Returnerer en verdi fra 0 til 255 som representerer intensiteten til det RØDE fargenivået som oppdages. • green(): Returnerer en verdi fra 0 til 255 som representerer intensiteten til det GRØNNE fargenivået som oppdages.

Objekt	Beskrivelse
	<ul style="list-style-type: none"> • blue(): Returnerer en verdi fra 0 til 255 som representerer intensiteten til det BLÅ fargenivået som oppdages. • gray(): Returnerer en verdi fra 0 til 255 som representerer grånivået som oppdages, der 0 er svart og 255 er hvit.
BB-port	<p>Gir støtte for bruk av alle 10 BB-portpinner som en kombinert digital inngangs-/utgangsport.</p> <p>Initialiseringsfunksjonene har en valgfri «mask»-parameter som gjør det mulig å bruke delsettet av de 10 pinnene.</p> <ul style="list-style-type: none"> • read_port(): Leser de aktuelle verdiene på inngangspinnene til BB-porten. • write_port(verdi): Stiller inn utgangspinneverdiene til den angitte verdien, hvor verdien er mellom 0 og 1023. Vær oppmerksom på at verdien også er justert mot maskeverdien i operasjonen <code>var=bbport(maske)</code>, hvis en maske ble angitt.
Hubtid	Gir tilgang til den interne millisekund-timeren.
TI-RGB-matrise	<p>Inneholder funksjoner for programmering av TI-RGB-matrise.</p> <p>Initialiseringsfunksjonen godtar en valgfri «LAMP»-parameter for å aktivere en modus med høy lysstyrke for TI-RGB-matrisen som krever en ekstern strømforsyning.</p> <ul style="list-style-type: none"> • set(led_position, r,g,b): Angir en spesifikk <code>led_position</code> (0–15) til den angitte <code>r,g,b</code>-verdien, der <code>r,g,b</code> er verdier fra 0 til 255. • set(led_list,rød,grønn,blå): Angir lysdiodene som er definert i «<code>led_list</code>» til fargen spesifisert av «<code>rød</code>», «<code>grønn</code>», «<code>blå</code>». «<code>led_list</code>» er en Python-liste som inkluderer indekser for LED-er fra 0 til 15. Settet([0,2,4,6,15], 0, 0, 255) vil for eksempel sette LED-er 0, 2, 4, 6 og 15 til blå. • set_all(r,g,b): Stiller inn alle RGB-lysdioder i matrisen til samme <code>r,g,b</code>-

Objekt	Beskrivelse
	<p>verdi.</p> <ul style="list-style-type: none"> • all_off(): Slår av alle RGB-lysene i matrisen. • measurement(): Returnerer det omtrentlige strømforbruket som RGB-matrisen benytter fra TI-Innovator™ i milliampere. • pattern(mønster): Bruker verdien av argumentet som en binær verdi i området 0 til 65535, og slår på piksler der en verdi 1 i representasjonen ville være. LED-er slått på som RØD med pwm-nivå på 255. • pattern(verdi,rød,grønn,blå): Angir lysdiodene som er definert av «pattern» til fargen spesifisert av «rød», «grønn», «blå».

Legg til utgangsenhet

Denne menyen har en liste over enhetene som støttes av ti_hub-modulen. Alle menyelementene vil lime inn navnet på objektet og forventet en variabel og en port som brukes med enheten.

Objekt	Beskrivelse
Indikatorlampe	Funksjoner for styring av eksternt tilkoblede LED-lamper.
RGB	Støtte for styring av eksterne RGB LED-lys.
TI-RGB-matrise	Inneholder funksjoner for programmering av TI-RGB-matrise.
Høytaler	Funksjoner for å støtte en ekstern høytaler med TI-Innovator™ Hub. Funksjonene er de samme som dem for «lyd» ovenfor.
Effekt	Funksjoner for styring av ekstern strøm med TI-Innovator™ Hub. <ul style="list-style-type: none"> • set(verdi): Angir effektnivået til den angitte verdien mellom 0 og 100. • on(): Angir effektnivået til 100. • off(): Angir effektnivået til 0.
Kontinuerlig servo	Funksjoner for styring av kontinuerlige servomotorer. <ul style="list-style-type: none"> • set_cw(hastighet,tid): Servoen roterer med klokken ved spesifisert hastighet (0–255) og i den spesifikke varigheten i sekunder.

Objekt	Beskrivelse
	<ul style="list-style-type: none"> • set_ccw(hastighet,tid): Servoen roterer i retning mot klokken med spesifisert hastighet (0–255) og i den spesifikke varigheten i sekunder. • stop(): Stopper den kontinuerlige servoen.
Analog utgang	Funksjoner for bruk av generiske enheter med analog inngang.
Vibrasjonsmotor	<p>Funksjoner for styring av vibrasjonsmotorer.</p> <ul style="list-style-type: none"> • set(verdi): Stiller inn vibrasjonsmotorintensiteten til «val» (0-255). • off(): Slår av vibrasjonsmotoren. • on(): Slår på vibrasjonsmotoren på høyeste nivå.
Relé	<p>Kontrollerer grensesnitt for styring av releer.</p> <ul style="list-style-type: none"> • on(): Setter reléet til PÅ-tilstanden. • off(): Setter reléet til AV-tilstanden.
Servo	<p>Funksjoner for styring av servomotorer.</p> <ul style="list-style-type: none"> • set_position(pos): Stiller inn svingservoposisjonen innenfor et område på -90 til +90. • zero(): Stiller inn svingservoen til nullstillingen.
Firkantbølge	<p>Funksjoner for å generere en firkantbølge.</p> <ul style="list-style-type: none"> • set(frekvens,arbeidssyklus,tid): Angir utgangsfirkantbølgen med en standard arbeidssyklus på 50 % (hvis arbeidssyklus ikke er spesifisert) og en utgangsfrekvens som er spesifisert ved «frekvens». Frekvensen kan være fra 1 til 500 Hz. Arbeidssyklusen, hvis spesifisert, kan være fra 0 til 100 %. • off(): Slår av firkantkurven.
Digital utgang	<p>Grensesnitt for styring av digital utgang.</p> <ul style="list-style-type: none"> • set(verdi): Angir den digitale utgangen til verdien spesifisert av «verdi» (0 eller 1). • on(): Angir tilstanden til den digitale utgangen til høy (1). • off(): Angir tilstanden til den digitale utgangen til lav (0).
BB-port	Inneholder funksjoner for programmering av TI-RGB-matrise. Se detaljene ovenfor.

Kommandoer

Objekt	Beskrivelse
sleep(sekunder)	Stanser programmet i det angitte antallet sekunder. Importert fra «time»-modulen.

Objekt	Beskrivelse
<code>text_at(rad,"tekst","juster")</code>	Viser den spesifiserte «tekst» i plotting-området ved spesifisert «juster». Del av <code>ti_plotlib</code> -modulen.
<code>cls()</code>	Tømmer skjermbildet i Shell for plotting. Del av <code>ti_plotlib</code> -modulen.
<code>while get_key() != "esc":</code>	Kjører kommandoene i «while»-løkken til «esc»-tasten trykkes.
<code>get_key()</code>	Returnerer en streng som representerer tasten som ble trykket. Tasten «1» returnerer «1», «esc» og så videre. Når den anropes uten parametre – <code>get_key()</code> – returneres den umiddelbart. Når den anropes med en parameter – <code>get_key(1)</code> – venter den til en tast trykkes. Del av <code>ti_system</code> -modulen.

Porter

Dette er inngangs- og utgangsportene som er tilgjengelige på TI-Innovator™ Hub.

Objekt
OUT 1
OUT 2
OUT 3
IN 1
IN 2
IN 3
BB 1
BB 2
BB 3
BB 4
BB 5
BB 6
BB 7
BB 8

Objekt

BB 9

BB 10

I2C

Ti Rover-menyen

Merk: Ved oppretting av et nytt program som bruker denne modulen, er det anbefalt å bruke programtypen **Rover Coding**. Dette vil sikre at alle relevante moduler importeres.

Objekt	Beskrivelse
import ti_rover som rv	Importerer alle metoder (funksjoner) fra ti_mobile-modulen i navneområdet «rv». Som følge av dette vil alle funksjonsnavn som er limt inn fra menyene bli innledet med «rv.».

Drive

Objekt	Beskrivelse
forward(avstand)	Flytter Rover forover den angitte avstanden i rutenettenheter.
backward(avstand)	Flytter Rover bakover den angitte avstanden i rutenettenheter.
left(angle_degrees)	Snur Rover til venstre den angitte vinkelen i grader.
right(angle_degrees)	Snur Rover til høyre den angitte vinkelen i grader.
stop()	Stopper enhver nåværende bevegelse umiddelbart.
stop_clear()	Stopper enhver nåværende bevegelse umiddelbart og fjerner alle ventende kommandoer.
resume()	Fortsetter behandlingen av kommandoer.
stay(time)	Rover blir på plass i den angitte tidsperioden målt i sekunder (valgfritt). Hvis det ikke er angitt noe tid, forblir Rover i 30 sekunder.
to_xy(x,y)	Flytter Rover til koordinatposisjon (x,y) på virtuelt rutenett.
to_polar(r,theta_degrees)	Flytter Rover til polarkoordinatposisjonen (r, theta) på virtuelt rutenett. Vinkelen angis i grader.
to_angle(angle,"unit")	Snur Rover til den angitte vinkelen i det virtuelle rutenettet. Vinkelen er relativ til en nullvinkel som peker ned x-aksen i det virtuelle rutenettet.

Kjør > Kjør med alternativer

Objekt	Beskrivelse
forward_time(tid)	Flytter Rover forover i den angitte tiden.
backward_time(tid)	Flytter Rover bakover i den angitte tiden.
forward(avstand,"enhet")	Flytter Rover forover med standardhastigheten den angitte avstanden. Avstanden kan angis i rutenettenheter, meter eller hjulomdreininger.
backward(avstand,"enhet")	Flytter Rover bakover med standardhastigheten den angitte avstanden. Avstanden kan angis i rutenettenheter, meter eller hjulomdreininger.
left(vinkel,"enhet")	Snur Rover til venstre den angitte vinkelen. Vinkelen kan være i grader, radianer eller gradianer.
right(vinkel,"enhet")	Snur Rover til høyre i den angitte vinkelen. Vinkelen kan være i grader, radianer eller gradianer.
forward_time(tid,hastighet,"hastighet")	Flytter Rover forover i den angitte tiden med spesifisert hastighet. Hastigheten kan angis i rutenettenheter/s, meter/s eller hjulomdreininger/s.
backward_time(tid,hastighet,"hastighet")	Flytter Rover bakover i den angitte tiden med spesifisert hastighet. Hastigheten kan angis i rutenettenheter/s, meter/s eller hjulomdreininger/s.
forward(avstand,"enhet",hastighet,"hastighet")	Flytter Rover forover den angitte avstanden med spesifisert hastighet. Avstanden kan angis i rutenettenheter, meter eller hjulomdreininger.

Objekt	Beskrivelse
	Hastigheten kan angis i rutenettenheter/s, meter/s eller hjulomdreininger/s.
backward(avstand,"enhet",hastighet,"hastighet")	Flytter Rover bakover den angitte avstanden med spesifisert hastighet. Avstanden kan angis i rutenettenheter, meter eller hjulomdreininger. Hastigheten kan angis i rutenettenheter/s, meter/s eller hjulomdreininger/s.

Innganger

Objekt	Beskrivelse
ranger_measurement()	Leser ultralydavstandssensoren på forsiden av Rover, og returnerer gjeldende avstand i meter.
color_measurement()	Returnerer en verdi fra 1 til 9, noe som indikerer den dominerende fargen som «ses» av fargeinputsensoren på Rover. 1 = rød 2 = grønn 3 = blå 4 = cyan 5 = magenta 6 = gul 7 = svart 8 = grå 9 = hvit
red_measurement()	Returnerer en verdi mellom 0 og 255 som indikerer det oppfattede nivået av rød som ses av fargeinputsensoren.
green_measurement()	Returnerer en verdi mellom 0 og 255 som indikerer det oppfattede nivået av grønn som ses av fargeinputsensoren.
blue_measurement()	Returnerer en verdi mellom 0 og 255 som indikerer det oppfattede nivået av blå som ses av fargeinputsensoren.

Objekt	Beskrivelse
gray_measurement()	Returnerer en verdi mellom 0 og 255 som indikerer det oppfattede nivået av grå som ses av fargeinputsensoren.
encoders_gyro_measurement()	Returnerer en liste over verdier som inneholder venstre og høyre hjulkoderantall, samt gjeldende gyroretning.
gyro_measurement()	Returnerer en verdi som representerer gjeldende gyrolesning, inkludert drift, i grader.
ranger_time()	Returnerer tiden som ultralydsignalet fra TI-Rover-avstandsmåleren bruker for å nå objektet («flyvningstiden»).

Utdata

Objekt	Beskrivelse
color_rgb(r,g,b)	Stiller inn fargen på RGB LED på Rover til de spesifikke røde, grønne og blå verdiene.
color_blink(frekvens,tid)	Angir frekvensen for blink og varigheten for den valgte fargen.
color_off()	Slår av Rover RGB LED.
motor_left(hastighet,tid)	Stiller inn kraften på venstre motor til den angitte verdien i den angitte varigheten. Hastigheten er i intervallet -255 til 255 hvor 0 betyr stopp. Positive hastighetsverdier er mot klokken, og negative hastighetsverdier er med klokken. Den valgfrie tidsparameteren, hvis spesifisert, har et gyldig intervall på 0,05 til 655,35 sekunder. Hvis det ikke spesifisert, brukes en standard på 5 sekunder.
motor_right(hastighet,tid)	Stiller inn kraften på venstre motor til den angitte verdien i den angitte varigheten. Hastigheten er i intervallet -255 til 255 hvor 0 betyr stopp. Positive hastighetsverdier er mot klokken, og negative hastighetsverdier er

Objekt	Beskrivelse
	<p>med klokken.</p> <p>Den valgfrie tidsparameteren, hvis spesifisert, har et gyldig intervall på 0,05 til 655,35 sekunder. Hvis det ikke spesifisert, brukes en standard på 5 sekunder.</p>
motors(«ldir»,venstre_verdi,«rdir»,right_val,tid)	<p>Stiller inn venstre og høyre hjul til de angitte hastighetsnivåene i en valgfri tidsperiode i sekunder.</p> <p>Verdiene for hastighet (venstre_verdi, høyre_verdi) er i intervallet 0 til 255 hvor 0 betyr stopp. ldir- og rdir-parameterne angir CW- eller CCW-rotasjon av de respektive hjulene.</p> <p>Den valgfrie tidsparameteren, hvis spesifisert, har et gyldig intervall på 0,05 til 655,35 sekunder. Hvis det ikke spesifisert, brukes en standard på 5 sekunder.</p>

Bane

Objekt	Beskrivelse
waypoint_xythdrn()	Leser x-koordinat, y-koordinat, tid, retning, avstand kjørt, antall hjulomdreininger, kommandonummer til gjeldende veipunkt. Returnerer en liste med alle disse verdiene som elementer.
waypoint_prev	Leser x-koordinat, y-koordinat, tid, retning, avstand kjørt, antall hjulomdreininger, kommandonummer til forrige veipunkt.
waypoint_eta	Returnerer anslått tid til å kjøre til et veipunkt.
path_done()	Returnerer en verdi på 0 eller 1 avhengig av om Rover beveger seg (0) eller er ferdig med all bevegelse (1).
pathlist_x()	Returnerer en liste med X-verdier fra begynnelsen til og med gjeldende veipunkts X-verdi.
pathlist_y()	Returnerer en liste med Y-verdier fra begynnelsen til og med gjeldende veipunkts Y-verdi.
pathlist_time()	Returnerer en liste over tiden i sekunder fra begynnelsen til og med gjeldende veipunkts tidsverdi.
pathlist_heading()	Returnerer en liste over retningene fra begynnelsen til og med gjeldende veipunkts retningsverdi.

Objekt	Beskrivelse
pathlist_distance()	Returnerer en liste over avstander som er kjørt fra begynnelsen til og med gjeldende veipunkts avstandsverdi.
pathlist_revs()	Returnerer en liste over antall omdreininger som er kjørt fra begynnelsen til og med gjeldende veipunkts omdreiningsverdi.
pathlist_cmdnum()	Returnerer en liste over kommandonumre for banen.
waypoint_x()	Returnerer x-koordinat for gjeldende veipunkt.
waypoint_y()	Returnerer y-koordinat for gjeldende veipunkt.
waypoint_time()	Returnerer tid brukt på å kjøre fra forrige til nåværende veipunkt.
waypoint_heading()	Returnerer absolutt retning for gjeldende veipunkt.
waypoint_distance()	Returnerer avstand kjørt mellom forrige og nåværende veipunkt.
waypoint_revs()	Returnerer antall omdreininger som trengs for å kjøre mellom forrige og nåværende veipunkt.

Innstillinger

Objekt	Beskrivelse
enheter/s	Alternativ for hastighet i rutenettenheter per sekund.
m/s	Alternativ for hastighet i meter per sekund.
omdreininger/s	Alternativ for hastighet i hjulomdreininger per sekund.
grafegnere	Alternativ for avstand i rutenettenheter.
M	Alternativ for avstand i meter.
omdreininger	Alternativ for avstand i hjulomdreininger.
grader	Alternativ for dreieing i grader.
radianer	Alternativ for dreieing i radianer.
gradianer	Alternativ for dreieing i gradianer.
med klokken	Alternativ for angivelse av hjulretning.
mot klokken	Alternativ for angivelse av hjulretning.

Kommandoer

Disse kommandoene er en samling av funksjoner fra andre moduler samt fra TI Rovermodulen.

Objekt	Beskrivelse
sleep(sekunder)	Stanser programmet i det angitte antallet sekunder. Importert fra time-modulen.
text_at (rad,"tekst","juster")	Viser teksten «tekst» i området der det er angitt «juster». Importert fra ti_plotlib-modulen.
cls()	Tømmer skjermbildet i Shell for plotting. Importert fra ti_plotlib-modulen.
while get_key() != "esc":	Kjører kommandoene i «while»-løkken til «esc»-tasten trykkes.
wait_until_done()	Stanser programmet til Rover fullfører den nåværende kommandoen. Dette er en nyttig metode for å synkronisere kommandoer som ikke er fra Rover med bevegelsene til Rover.
while not path_done()	Kjører kommandoene i «while»-løkken til Rover er ferdig med all bevegelse. Funksjonen path_done() returnerer en verdi på 0 eller 1 avhengig av om Rover beveger seg (0) eller er ferdig med all bevegelse (1).
position(x,y)	Angir posisjon for Rover på det virtuelle rutenettet til den spesifiserte x,y-koordinaten.
position(x,y,heading,"unit")	Angir posisjon for Rover på det virtuelle rutenettet til den spesifiserte x,y-koordinaten, og den virtuelle retningen relativt til den virtuelle x-aksen blir angitt hvis en retning er gitt (i de spesifiserte enhetene for vinkler). Positive vinkler fra 0 til 360 antas å være mot klokken fra den positive x-aksen. Negative vinkler fra 0 til 360 antas å være med klokken fra den positive x-aksen.
grid_origin()	Angir at RV er ved gjeldende rutenetts origo i (0,0).
grid_m_unit(scale_value)	Angir den virtuelle rutenettavstanden i meter per enhet (m/enhet) til den spesifiserte verdien. 0,1 er standard m/enhet og oversettes til 1 enhet = 100 mm eller 10 cm eller 1 dm eller 0,1 m. Gyldig skalaverdiområde er fra 0,01 til 10,0.
path_clear()	Sletter all eksisterende bane- eller veipunktinformasjon.
zero_gyro()	Tilbakestill gyro på Rover til 0,0-vinkelen, og tømmer de venstre og høyre hjulkodertallene.

Kompleks matematikk-menyen

Denne undermenyen er plassert under **Flere moduler**.

Objekt	Beskrivelse
<code>from cmath import *</code>	Importerer alle metoder fra cmath-modulen.
<code>complex(reelt,imag)</code>	Returnerer et komplekst tall.
<code>rect(modulus,argument)</code>	Konverterer polarkoordinater til rektangulær form av et komplekst tall.
<code>.real</code>	Returnerer reell del av det komplekse tallet.
<code>.imag</code>	Returnerer imaginær del av et komplekst tall.
<code>polar()</code>	Konverterer rektangulær form til polarkoordinater av et komplekst tall.
<code>phase()</code>	Returnerer fase av et komplekst tall.
<code>exp()</code>	Returnerer e^{**x} .
<code>cos ()</code>	Returnerer cosinus til et komplekst tall.
<code>sin()</code>	Returnerer sinus til et komplekst tall.
<code>log()</code>	Returnerer naturlig logaritme av et komplekst tall.
<code>log10()</code>	Returnerer logaritme med grunntall 10 av et komplekst tall.
<code>sqrt()</code> (kvdrtr)	Returnerer kvadratrot av et komplekst tall.

Tid-menyen

Denne undermenyen er plassert under **Flere moduler**.

Objekt	Beskrivelse
<code>from time import *</code>	Importerer alle metoder fra time-modulen.
<code>sleep(sekunder)</code>	Stopper programmet i det angitte antallet sekunder.
<code>clock()</code>	Returnerer gjeldende prosessortid som et flyttall uttrykt i sekunder.
<code>localtime()</code>	Omregner en tid uttrykt i sekunder siden 1. januar 2000 til et nittuppel som inneholder år, måned, dag i måneden, time, minutt, sekund, ukedag, år, og flagg for sommertid (DST). Hvis det valgfrie argumentet (sekunder) ikke er oppgitt, brukes sanntidsklokken.
<code>ticks_cpu()</code>	Returnerer en prosessorspesifikk økende millisekundteller med vilkårlig referansepunkt. For å måle tiden konsekvent på tvers av ulike systemer, bruker du <code>ticks_ms()</code> .
<code>ticks_diff()</code>	Måler perioden mellom påfølgende kall til <code>ticks_cpu()</code> eller <code>ticks_ms()</code> . Denne funksjonen skal ikke brukes til å måle tidsperioder av vilkårlig lengde.

Ti System-menyen

Denne undermenyen er plassert under **Flere moduler**.

Merk: Når du oppretter et nytt program som bruker denne modulen, anbefales det å bruke programtypen **Datadeling**. Dette vil sikre at alle relevante moduler importeres.

Objekt	Beskrivelse
<code>from ti_system import *</code>	Importerer alle metoder (funksjoner) fra <code>ti_system</code> -modulen.
<code>recall_value("navn")</code>	Henter inn en forhåndsdefinert OS-variabel (verdi) kalt navnet «navn».
<code>store_value("navn",verdi)</code>	Lagrer en Python-variabel (verdi) til en OS-variabel med navnet «navn».
<code>recall_list("navn")</code>	Henter inn en forhåndsdefinert OS-liste med navnet «navn».
<code>store_list("navn",liste)</code>	Lagrer en Python-liste (liste) til en OS-listevariabel med navnet «navn».
<code>eval_function("navn",verdi)</code>	Evaluere en forhåndsdefinert OS-funksjon med den angitte verdien.
<code>get_platform()</code>	Returere «hh» for håndholdt enhet og «dt» for skrivebordet.
<code>get_key()</code>	Returere en streng som representerer tasten som ble trykket. «1»-tasten returere «1», «esc» returere «esc» og så videre. Når den kalles uten noen parametere – <code>get_key()</code> – returere den umiddelbart. Når den kalles med en parameter – <code>get_key(1)</code> – venter det til en tast trykkes.
<code>get_mouse()</code>	Returere musekoordinater som et tuppel med to elementer, enten pikselposisjonen på lerretet eller (-1,-1) hvis det er utenfor lerretet.
<code>while get_key() != "esc":</code>	Kjør kommandoene i «while»-løkken til «esc»-tasten trykkes.
<code>clear_history()</code>	Sletter historikken i Shell.
<code>get_time_ms()</code>	Returere tid i millisekunder med millisekundpresisjon. Denne funksjonaliteten kan brukes til å beregne en varighet i stedet for å bestemme den faktiske klokketiden.

Ti Draw-menyen

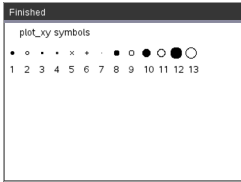
Denne undermenyen er plassert under **Flere moduler**.

Merk: Når du oppretter et nytt program som bruker denne modulen, anbefales det å bruke programtypen **Geometrigrafikk**. Dette vil sikre at alle relevante moduler importeres.

Objekt	Beskrivelse
<code>from ti_draw import *</code>	Importerer alle metoder fra <code>ti_draw</code> -modulen.

Figur

Objekt	Beskrivelse
<code>draw_line()</code>	Tegner en linje fra spesifisert koordinat <code>x1,y1</code> til <code>x2,y2</code> .
<code>draw_rect()</code>	Tegner et rektangel som starter ved spesifisert koordinat <code>x,y</code> med spesifisert bredde og høyde.
<code>fill_rect()</code>	Tegner et rektangel som starter ved spesifisert koordinat <code>x,y</code> med spesifisert bredde og høyde og fylt med spesifisert farge (med <code>set_color</code> eller svart hvis ikke definert).
<code>draw_circle()</code>	Tegner en sirkel som starter ved spesifisert senterkoordinat <code>x,y</code> med angitt radius.
<code>fill_circle()</code>	Tegner en sirkel som starter ved spesifisert senterkoordinat <code>x,y</code> med angitt radius og fylt med spesifisert farge (med <code>set_color</code> eller svart hvis ikke definert).
<code>draw_text()</code>	Tegner en tekststreng som starter på spesifisert koordinat <code>x,y</code> .
<code>draw_arc()</code>	Tegner en bue som starter ved spesifisert koordinat <code>x,y</code> med spesifisert bredde, høyde og vinkler.
<code>fill_arc()</code>	Tegner en bue som starter ved spesifisert koordinat <code>x,y</code> med spesifisert bredde, høyde og vinkler fylt med spesifisert farge (med <code>set_color</code> eller svart hvis ikke definert).
<code>draw_poly()</code>	Tegner et polygon med spesifiserte verdier <code>x-liste,y-liste</code> .
<code>fill_poly()</code>	Tegner et polygon med spesifiserte verdier <code>x-liste,y-liste</code> fylt med spesifisert farge (med <code>set_color</code> eller svart hvis ikke definert).
<code>plot_xy()</code>	Tegner en figur med spesifisert koordinat <code>x,y</code> og spesifisert nummer fra 1–13 som representerer forskjellige former og symboler (se nedenfor).

Objekt	Beskrivelse
	

Kontroll

Objekt	Beskrivelse
<code>clear()</code>	Tømmer hele skjermen. Kan brukes med parametrene <code>x,y,bredde,høyde</code> for å fjerne et eksisterende rektangel.
<code>clear_rect()</code>	Sletter rektangelet ved spesifisert koordinat <code>x,y</code> med spesifisert bredde og høyde.
<code>set_color()</code>	Angir fargen på figuren(e) som følger i programmet til en annen farge blir angitt.
<code>set_pen()</code>	Angir den angitte tykkelsen og stilen til kanten når figuren tegnes (gjelder ikke ved bruk av fyllkommandoer).
<code>set_window()</code>	Angir størrelsen på vinduet der alle former vil bli tegnet. Denne funksjonen er nyttig for å endre størrelsen på vinduet til å samsvare med dataene eller endre utgangspunktet (0,0) på tegnerretet.
<code>get_screen_dim()</code>	Returnerer <code>xmaks</code> og <code>ymaks</code> av skjermdimensjoner.
<code>use_buffer()</code>	Aktiverer en buffer utenfor skjermen for å øke hastigheten på tegning.
<code>paint_buffer()</code>	Viser utdata for bufret tegning. Funksjonene <code>use_buffer()</code> og <code>paint_buffer()</code> er nyttige i tilfeller der visning av flere objekter på skjermen kan forårsake forsinkelser.

Notater

- Standardkonfigurasjonen har (0,0) øverst til venstre på skjermen. De positive x-aksepunktene går mot høyre og de positive y-aksepunktene mot bunnen Dette kan endres ved å bruke `set_window()`-funksjonen.
- Funksjonene i `ti_draw`-modulen er bare tilgjengelig på håndholdt enhet og i håndholdtvisning på skrivebordet.

Ti Image-menyen

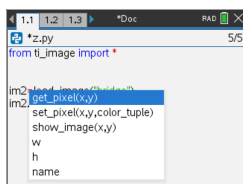
Denne undermenyen er plassert under **Flere moduler**.

Merk: Når du oppretter et nytt program som bruker denne modulen, anbefales det å bruke programtypen **Bildebehandling**. Dette vil sikre at alle relevante moduler importeres.

Objekt	Beskrivelse
<code>from ti_image import *</code>	Importerer alle metoder fra <code>ti_image</code> -modulen.
<code>new_image(bredde,høyde,(r,g,b))</code>	Oppretter et nytt bilde med spesifisert bredde og høyde for bruk i Python-programmet. Fargen på det nye bildet defineres av verdiene <code>(r,g,b)</code> .
<code>load_image("navn")</code>	Laster bildet som er spesifisert av «navn» til bruk i Python-programmet. Bildet må være en del av TNS-dokumentet enten i en Merknader- eller Grafer-applikasjon. «navn»-ledeteksten vil vise bildeneavnene (hvis de har blitt navngitt tidligere) eller et tall som indikerer deres innsetningsrekkefølge.
<code>copy_image(bilde)</code>	Oppretter en kopi av bildet som er spesifisert av variabelen «bilde».

Bildeobjektets metoder

Tilleggsfunksjoner relatert til bildeobjekter er tilgjengelige i redigeringsprogram og Shell ved å skrive variabelnavnet etterfulgt av `.` (punktum).



- **get_pixel(x,y):** Mottar verdien `(r,g,b)` av pikselen på stedet som er definert av koordinatparet `(x,y)`.

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

- **set_pixel(x,y,farge_tupple):** Setter pikselen på stedet `(x,y)` til fargen som er spesifisert i `farge_tupple`.

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

Setter pikselen på `(100 100)` til fargen `(0,0,255)`.

- **show_image(x,y):** Viser bildet med øvre venstre hjørne på stedet `(x,y)`.

- **w, h, name:** Henter parametrene for bildets bredde, høyde og navn.

Eksempel

```
from ti_image import *

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

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

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


Variabel-menyen

Merk: Disse listene inkluderer ikke variabler definert i andre TI-Nspire™-applikasjoner.

Objekt	Beskrivelse
Vars: Gjeldende program	(Kun redigeringsprogram) Viser en liste over globale funksjoner og variabler som er definert i det aktuelle programmet
Vars: Programmet som ble kjørt sist	(Kun Shell) Viser en liste over globale funksjoner og variabler som er definert i programmet som ble kjørt sist
Vars: Alle	(Kun Shell) Viser en liste over globale funksjoner og variabler fra både programmet som ble kjørt sist og importerte moduler

Tillegg

Nøkkelord for Python	48
Tastetilordning for Python	49
Eksempelprogrammer i Python	51













Nøkkelord for Python





















Følgende nøkkelord er bygget inn 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

Tastetilordning for Python

Når du skriver inn kode i redigeringsprogrammet eller i Shell, er tastaturet utformet for å lime inn riktige Python-operasjoner eller åpne menyer for enkel innlegging av funksjoner, nøkkelord, metoder, operatører osv.

tasten	Tilordning
	Åpner variabler-meny
	Limer inn tegnet =
	Sletter tegnet til venstre for markøren
	Ingen handling
	Limer inn tegnet =
	Limer inn de valgte symbolene: <ul style="list-style-type: none">• >• <• !=• >=• <=• ==• og• —eller—• ikke• • &• ~
	Limer inn den valgte funksjonen: <ul style="list-style-type: none">• 0• cos• 1.619775191• atan2• asin• acos• atan
	Viser tips
	Limer inn :=
	Limer inn **
	Ingen handling
	Limer inn **2

tasten	Tilordning
	Limer inn sqrt()
	Limer inn multiplikasjonstegn (*)
	Limer inn ett dobbelt anførselstegn (")
	Limer inn divisjonstegn (/)
	Ingen handling
	Limer inn exp()
	Limer inn log()
	Limer inn 10**
	Limer inn log(verdi,base)
	Limer inn (
	Limer inn)
	Limer inn []
	Limer inn { }
	Limer inn subtraksjonstegn (-)
	Legger til en ny linje etter gjeldende linje
	Limer inn E
	Limer inn de valgte symbolene: <ul style="list-style-type: none"> • ? • ! • \$ • ° • ' • % • " • : • ; • _ • \ • #
	Limer inn «pi»
	Eksisterende flaggatferd
	Legger til en ny linje etter gjeldende linje

Eksempelprogrammer i Python

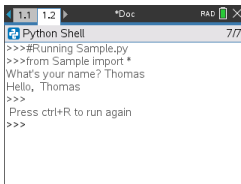
Bruk følgende eksempelprogrammer for å gjøre deg kjent med Python-metoder. De er også tilgjengelige i filen **Getting Started Python.tns** i mappen **Examples**.

Merk: Hvis du kopierer og limer inn eksempelkode som inneholder tabulatorindikatorer (••) i TI-Nspire™-programvaren, må du erstatte disse forekomstene med faktiske tabulatorinnrykk.

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")
```

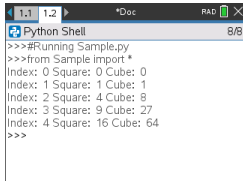
A screenshot of a Python Shell window titled "Python Shell" with a file icon and "7/7" in the top right corner. The window shows the following text:

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

Løkkeeksempel

```
# 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)
```



```
*Doc
Python Shell
>>>#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
>>>
```

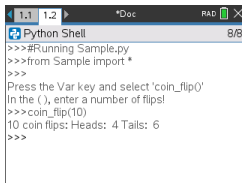
Mynt eller krone

```
# 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:" 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
>>>
```


Plotting

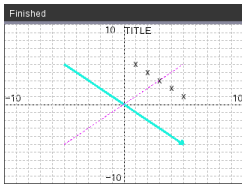
```
# Plotting example
import ti_plotlib as plt

# 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

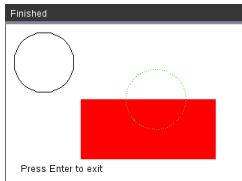
```
from ti_draw import *

# (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")
```



Bilde

```
# 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 programmet bruker Python til å kontrollere TI-Innovator™ Hub, en programmerbar mikrokontroller. Hvis du kjører programmet uten å koble til en TI-Innovator™-hub, vises en feilmelding.

Du finner mer informasjon om TI-Innovator™ Hub på 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()
```

Generell informasjon

Hjelp på nettet (online)

education.ti.com/eguide

Velg ditt land for mer produktinformasjon.

Kontakt TIs brukerstøtte

education.ti.com/ti-cares

Velg ditt land for tekniske og andre støtteressurser.

Service og garantiinformasjoner

education.ti.com/warranty

Velg landet ditt for informasjon om lengden og vilkårene for garantien eller om produkttjenester.

Begrenset garanti. Denne garantien påvirker ikke dine lovmessige rettigheter.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243