

# TI-Nspire™ Python Programmeringsguide

## **Viktig information**

Texas Instruments lämnar inga garantier, vare sig uttryckliga eller underförstådda, inklusive men inte begränsade till underförstådda garantier i fråga om säljbarhet eller lämplighet för ett speciellt syfte, rörande program eller bokmaterial och gör endast sådant material tillgängligt på en "i befintligt skick"-grund. Under inga omständigheter kommer Texas Instruments att vara skyldigt för speciella, omgivande, oavsiktliga eller följdaktiga skador i samband med eller uppkomna genom köpet eller användandet av dessa material och det enda och exklusiva åtagande som tillfaller Texas Instruments, oavsett formen av åtgärd, skall inte överstiga summan som anges i programmets licens. Texas Instruments är inte heller skadeståndsskyldigt för några krav som uppstått genom användning av denna produkt av annan användare.

© 2021 Texas Instruments Incorporated

"Python" och Python-logotyperna är varumärken eller registrerade varumärken som tillhör Python Software Foundation, som används av Texas Instruments Incorporated med tillstånd från stiftelsen.

De faktiska produkterna kan variera något från de visade bilderna.

# Innehållsförteckning

<b>Komma igång med Python-programmering</b> .....	<b>1</b>
Python-moduler .....	1
Installera ett Python-program som en modul .....	2
<b>Python Workspaces</b> .....	<b>3</b>
Python Editor .....	3
Python Shell .....	7
<b>Python-menükarta</b> .....	<b>10</b>
Menyn Åtgärder .....	11
Menyn Kör .....	12
Verktygsmeny .....	13
Menyn Redigera .....	14
Inbyggda menyer .....	15
Menyn Matematik .....	18
menyn Slump .....	20
TI PlotLib Menu .....	21
TI hubb-meny .....	23
TI Rover-menyn .....	31
Menyn Komplex matematik .....	38
Menyn Tid .....	39
TI systemmeny .....	40
TI Draw-meny .....	41
TI bildmeny .....	43
Menyn Variabler .....	45
<b>Bilaga</b> .....	<b>46</b>
Python-nyckelord .....	47
Python tangentmappning .....	48
Python-exempelprogram .....	50
<b>Allmän information</b> .....	<b>57</b>

# Komma igång med Python-programmering

Om du använder Python med TI-Nspire™-produkter kan du:

- lägga till Python-program till TNS-filer
- skapa Python-program med mallar
- interagera och dela data med andra TI-Nspire™-appar
- interagera med TI-Innovator™ Hub+ och TI-Innovator™ Rover

Implementeringen av TI-Nspire™ Python baseras på MicroPython, som är en liten undergrupp av Python 3:s standardbibliotek, som utformats för att köras på mikrostyrenheter. Den ursprungliga MicroPython-implementeringen har anpassats för användning av TI.

**Obs:** Vissa numeriska svar kan skilja sig från Räknarens resultat på grund av skillnader i underliggande matematiska implementeringar.

Python finns tillgängligt på dessa TI-Nspire™-produkter:

Handenheter	Skrivbordsprogramvara
TI-Nspire™ CX II	TI-Nspire™ CX Premium Teacher Software
TI-Nspire™ CX II CAS	TI-Nspire™ CX CAS Premium lärarprogramvara
TI-Nspire™ CX II-T	TI-Nspire™ CX Student Software
TI-Nspire™ CX II-T CAS	TI-Nspire™ CX CAS elevprogramvara
TI-Nspire™ CX II-C	
TI-Nspire™ CX II-C CAS	

**Obs:** I de flesta fall är funktionaliteten identisk mellan handenheter och programvaruversionerna, men du kan se vissa skillnader. Denna guide förutsätter att du använder den handenheter eller att du använder handenheters vy i programvaran.

## Python-moduler

TI-Nspire™ Python inkluderar följande moduler:

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

**Obs:** Om du har befintliga Python-program skapade i andra Python-miljöer kan du behöva redigera dem för att köra dem med TI-Nspire™ Python-lösningen. Moduler kan använda olika metoder, argument och ordning på metoder i ett program jämfört med

TI-modulerna. Generellt sett, tänk på kompatibiliteten när du använder någon version av Python och Python-moduler.

Vid överföring av Python-program från en icke TI-plattform till en TI-plattform ELLER från en TI-produkt till en annan, kom ihåg:

- Program som använder huvudspråk och standard-bibliotek (matematik, slump osv.) kan importeras utan ändringar.
- Program som använder plattformsspecifika bibliotek såsom matplotlib för PC- eller TI-moduler kräver redigeringar innan de körs på en annan plattform. Detta kan behövas även mellan TI-plattformar.

Som med alla versioner av Python måste du inkludera import för att använda alla funktioner, metoder eller konstanter som finns i en viss modul. För att exempelvis exekvera funktionen `cos()` från den matematiska modulen, använd följande kommandon:

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

För en lista över menyer med deras artiklar och beskrivningar, se avsnittet [Menykarta](#).

## ***Installera ett Python-program som en modul***

**För att spara ditt Python-program som en modul:**

- I redigeraren väljer du **Åtgärder > Installera som Python-modul**.
- I Shell väljer du **Verktyg > Installera som Python-modul**.

Efter valet sker följande:

- Python-syntaxen är markerad.
- Filen sparas och flyttas till PyLib-mappen.
- En dialogruta visas som bekräftar att filen har installerats som en modul.
- Filen är stängd och modulen är klar att användas.
- Modulnamnet läggs till i menyn **Fler moduler** med ett **från menyn <module> Importera\***.

Om du planerar att dela denna modul med andra rekommenderar vi att du följer dessa riktlinjer:

- Lagra endast en modul per TNS-fil.
- Modulnamnet matchar namnet på TNS-filen (t.ex. "my\_program"-modulen finns i "my\_program.tns"-filen).
- Lägg till en anteckningssida före Python Editor som beskriver avsikten med modulen, versionen och funktionerna.
- Använd funktionen `ver()` för att visa modulens versionsnummer.
- (Valfritt) Lägg till en hjälpfunktion för att visa listan över metoder i funktionen.

# Python Workspaces

Det finns två arbetsytor för din Python-programmering: Python Editor och Python Shell.

Python Editor	Python Shell
<ul style="list-style-type: none"><li>• Skapa, redigera och spara Python-program</li><li>• Syntaxmarkering och automatisk indentering</li><li>• Infogade uppmaningar till handledning med funktionsargument</li><li>• Verktygstips för att visa giltiga värden</li><li>• <code>var</code>-tangenten listar globala användarvariabler och funktioner som definierats i det aktuella programmet</li><li>• Kortkommandon</li></ul>	<ul style="list-style-type: none"><li>• Kör Python-program</li><li>• Praktisk för test av små kodfragment</li><li>• Interaktion med Shell-historik för att välja tidigare inmatningar och resultat för återanvändning</li><li>• <code>var</code>-tangenten listar globala användarvariabler som definierats i det senaste programmet som kördes i det givna problemet</li></ul>

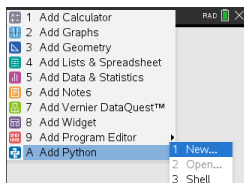
**Obs:** Flera Python- och Shell-program kan läggas till i ett problem.

## Python Editor

Python Editor är programmet där du kan skapa, redigera och spara Python-program.

### Lägga till en sida i Python Editor

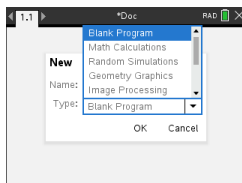
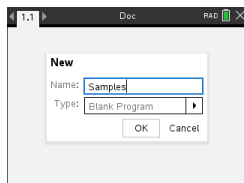
För att lägga till en ny Python Editor-sida i det aktuella problemet, tryck på `menu` och välj **Add Python > New**.



Du kan skapa ett tomt program eller välja en mall.

*Tomt program*

*Mall*



När programmet har skapats visas Python Editor. Om du har valt en mall läggs nödvändiga importdeklarationer automatiskt till (se nedan).

**Obs:** Du kan ha flera program i en enda TNS-fil, precis som andra appar. Om Python-programmet är avsett att användas som en modul kan TNS-filen sparas i mappen PyLib. Den modulen kan sedan användas i andra program och dokument.

### Matematiska beräkningar

### Slumpmässiga simuleringar

```

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

```

```

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

```

### Geometri grafik

### Bildbehandling

```

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

```

```

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

```

### Plotning (x,y) & text

### Datadelning

```

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

```

```

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

```

### TI-Innovator hubb-projekt

### TI-Rover-kodning

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

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

## Öppna ett Python-program

För att öppna ett befintligt Python-program, tryck på **[doc]** och välj **Insert> Add Python > Open**. Detta visar en lista över program som har sparats i TNS-filen.

Om den Editor-sida som används för att skapa programmet har raderats finns programmet fortfarande tillgängligt i TNS-filen.

## Arbeta i Python Editor

Om du trycker på **[menu]** visas menyn Dokumentverktyg. Med dessa menyalternativ kan du lägga till, flytta och kopiera kodblock för ditt program.

### Menyn Dokumentverktyg

```
1.1 |> *Doc RAD | X
1 Actions
2 Run
3 Edit
4 Built-ins
5 Math
6 Random
7 TI PlotLib
8 TI Hub
9 TI Rover
A More Modules
B Variables
```

```
1.1 |> *Doc RAD | X
1 Actions
2 Run
3 Edit
4 Built-ins
5 Math
6 Random
7 TI PlotLib
8 TI Hub
9 TI Rover
A More Modules
B Variables
```

Artiklar som valts från modulmenyerna lägger automatiskt till en kodmall i redigeraren med infogade prompter för varje del av funktionen. Du kan navigera från ett argument till nästa genom att trycka på **[tab]** (framåt) or **[shift]+[tab]** (bakåt). Verktvgstips eller popup-listor visas när de är tillgängliga för att hjälpa dig att välja rätt värden.

### Infogade prompter

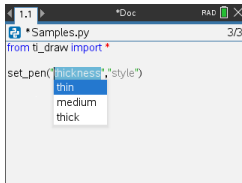
```
1.1 |> *Doc RAD | X
*Samples.py 3/4
from ti_draw import *
def function(argument):
    =block
```

### Verktvgstips

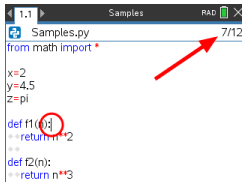
```
1.1 |> *Doc RAD | X
*Samples.py 3/3
from ti_draw import *
set_color(red,green,blue)
0-255
```

### Popup-listor

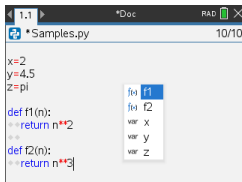




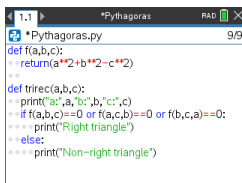
Siffrorna till höger om programnamnet återspeglar den aktuella raden för markören och det totala antalet rader i programmet.



Globala funktioner och variabler som definieras i raderna ovanför den aktuella markörpositionen kan infogas genom att trycka på `var` och välja från listan.



När du lägger till kod i ditt program, visar redigeraren nyckelord, operatörer, kommentarer, strängar och indenteringar i olika färger för att hjälpa till att identifiera de olika elementen.




## Spara och köra program

När du är klar med ditt program, tryck på `menu` och välj **Kör > Kontrollera syntax och Spara**. Detta kontrollerar syntaxen för Python-programmet och sparar den i TNS-filen.

**Obs:** Om du har osparade ändringar i programmet visas en asterisk bredvid programnamnet.



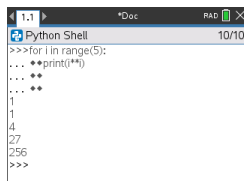
Om du vill köra programmet trycker du på  och väjer **Kör > Kör**. Detta kommer att köra det aktuella programmet på nästa Python Shell-sida eller på en ny sida om det inte är en Shell-sida.

**Obs:** När programmet körs kontrolleras syntaxen automatiskt och programmet sparas.

## Python Shell

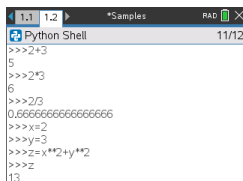
Python Shell är den tolk som kör dina Python-program, andra bitar av Python-kod eller enkla kommandon.

### Python-kod




```
>>>for i in range(5):
...     **print(i*)
...     **
...     **
1
1
1
1
1
27
256
>>>
```

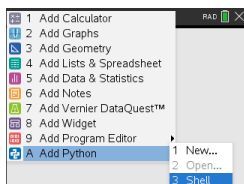
### Enkla kommandon




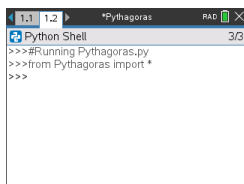
```
>>>2+3
5
>>>2*3
6
>>>2/3
0.6666666666666666
>>>x=2
>>>y=3
>>>z=x**2+y**2
>>>z
13
```

## Lägga till en Python Shell-sida

För att lägga till en ny Python Shell-sida i det aktuella problemet, tryck på  och välj **Lägg till Python > Ny**.




Python Shell kan också startas från Python Editor genom att köra ett program genom att trycka på  och välja **Kör > Kör**.

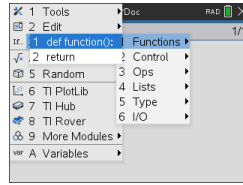
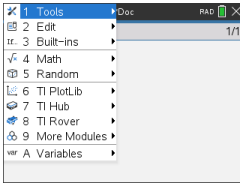


```
>>>#Running Pythagoras.py
>>>from Pythagoras import *
>>>
```

## Arbeta i Python Shell

Om du trycker på  visas menyn Dokumentverktyg. Med dessa menyalternativ kan du lägga till, flytta och kopiera kodblock.

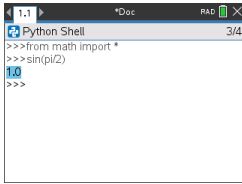
### Menyn Dokumentverktyg



**Obs:** Om du använder någon metod från en av de tillgängliga modulerna, se till att först importera moduluttryck, som i en vanlig Python-kodningsmiljö.

Interaktion med Shell-utdata liknar applikationen Räkare där du kan välja och kopiera tidigare inmatningar och utdata för användning på andra ställen i Shell, Editor eller andra appar.

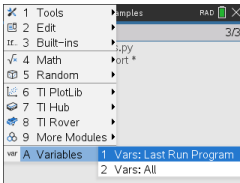
*Uppåt-pil för att välja, kopiera och klistra in på önskad plats*



Globala funktioner och variabler från det senaste körda programmet kan infogas genom att trycka på **var** eller **ctrl+l** och välja från listan eller genom att trycka på **menu** och välja **Variabler > Variabler: Köra om senaste programmet**.

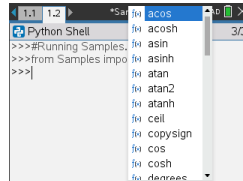
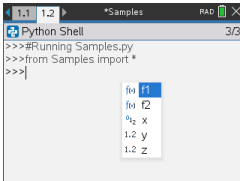
För att välja från en lista över globala funktioner och variabler från både det senaste körda programmet och alla importerade moduler, tryck på **menu** och välj **Variabler > Variabler: Alla**.

*Menyn Variabler*



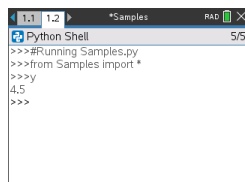
*Senaste programvariabler*

*Alla variabler*



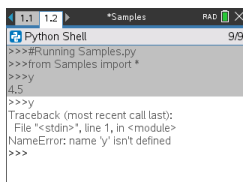
Alla Python Shell-sidor i samma problem delar samma status (användardefinierade och importerade variabeldefinitioner). När du sparar eller kör ett Python-program i det problemet eller trycker på **[menu]** och väljer **Verktyg > Starta om Shell** kommer Shell-historiken att ha en grå bakgrund som indikerar att den tidigare statusen inte längre är giltig.

*Innan du sparar eller startar om*



```
Python Shell 5/5
>>>#Running Samples.py
>>>from Samples import *
>>>y
4.5
>>>
```

*Efter att ha sparat eller initierat igen*



```
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
>>>
```

**Obs:** Alternativet **[menu] Verktyg > Rensa historik** rensar skärmen från all tidigare aktivitet i Shell, men variablerna är fortfarande tillgängliga.

### Meddelanden

Fel och andra informationsmeddelanden kan visas medan du är i en Python-session. Om ett fel visas i Shell när ett program exekveras, visas ett programradsnummer. Tryck på **[ctrl] [menu]** och välj **Gå till Python Editor**. I redigeraren trycker du på **[menu]** och väljer sedan **Redigera > Gå till linje**. Skriv in radnumret och tryck på **[enter]**. Markören visas på det första tecknet i raden där felet uppstod.

### Avbryta ett pågående program

Medan ett program eller en funktion exekveras visas upptagetsymbolen **⏏**.

- ▶ För att stoppa programmet eller funktionen,
  - Windows®: Tryck på knappen **F12**.
  - Mac®: Tryck på knappen **F5**.
  - Handenhet: Tryck på knappen **[stop]**.

# Python-menykarta

Detta avsnitt listar alla menyer och menyalternativ för Python Editor och Shell och en kort beskrivning för var och en av dem.

**Obs:** För menyobjekt som har tangentbordsgenvägar ska Mac<sup>®</sup>-användare ersätta ⌘ (Ctrl) till Cmd när den funktionen används. För en komplett lista över kortkommandon för TI-Nspire™ handenheter och programvara, se e-Guide för TI-Nspire™-teknologi.

---

Menyn Åtgärder .....	11
Menyn Kör .....	12
Verktöymeny .....	13
Menyn Redigera .....	14
Inbyggda menyer .....	15
Menyn Matematik .....	18
menyn Slump .....	20
TI PlotLib Menu .....	21
TI hubb-meny .....	23
TI Rover-meny .....	31
Menyn Komplex matematik .....	38
Menyn Tid .....	39
TI systemmeny .....	40
TI Draw-meny .....	41
TI bildmeny .....	43
Menyn Variabler .....	45

## Menyn Åtgärder

Obs: Detta gäller endast Editorn.

Post	matris
Nytt	Öppnar dialogrutan <b>Ny</b> där du anger ett namn och väljer en typ för ditt nya program.
Öppna	Öppnar en lista över program som finns tillgängliga i det aktuella dokumentet.
Skapa kopia	Öppnar dialogrutan <b>Skapa kopia</b> där du kan spara det aktuella programmet under ett annat namn.
Byt namn	Öppnar dialogrutan <b>Byt namn</b> där du kan byta namn på det aktuella programmet.
Stäng	Stänger det aktuella programmet.
Settings	Öppnar dialogrutan <b>Inställningar</b> där du kan ändra teckenstorleken för både Editor och Shell.
Installera som Python-modul	Kontrollerar Python-syntaxen för den aktuella TNS-filen och flyttar den till PyLib-mappen.

## Menyn Kör

Obs: Detta gäller endast Editorn.

Post	genvägar	matris
Kör	Ctrl+R	Kontrollera syntax, sparar program och exekverar i Python Shell.
Kontrollera syntax & spara	Ctrl+B	Kontrollerar syntax och sparar program.
Gå till Shell	Ej tillämpligt	Växlar fokus till Shell relaterat till det aktuella programmet eller öppnar en ny Shell-sida intill redigeraren.

## Verktysmeny

Obs: Detta gäller endast Shell.

Post	genvägar	matris
Kör om senaste programmet	Ctrl+R	Kör om det senaste programmet relaterat till aktuellt Shell.
Gå till Python Editor	Ej tillämpligt	Öppnar redigerarsidan relaterad till det aktuella Shell-programmet.
Kör	Ej tillämpligt	Öppnar en lista över program som finns tillgängliga i det aktuella dokumentet. Efter val körs det valda programmet.
Rensa historik	Ej tillämpligt	Rensar historiken i aktuellt Shell-fönster men initierar inte Shell igen.
Starta om Shell	Ej tillämpligt	Återställer status för alla öppna Shell-sidor i det aktuella problemet. Alla definierade variabler och importerade funktioner är inte längre tillgängliga.
dir()	Ej tillämpligt	Visar en lista över funktioner i den angivna modulen när den används efter importuttrycket.
From PROGRAM import *	Ej tillämpligt	Öppnar en lista över program som finns tillgängliga i det aktuella dokumentet. Efter val klistras importuttrycket in i Shell.
Installera som Python-modul	Ej tillämpligt	Aktiverad endast för moduler i binärt format. Flyttar den aktuella TNS-filen till PyLib-mappen.



## Menyn Redigera

**Obs:** Ctrl+A väljer alla kodlinjer eller utdata som ska skäras ut eller raderas (endast Editor) eller kopierar och klistrar in (Editor och Shell).

Post	genvägar	matris
Indentera	TAB*	Indenterar text på aktuell rad eller valda rader. * Om det finns ofullständiga direktmeddelanden kommer TAB att navigera till nästa uppmaning.
Dedentera	Shift+TAB**	Dedenterar text på aktuell rad eller valda rader. ** Om det finns ofullständiga direktmeddelanden kommer TAB att navigera till nästa uppmaning.
Kommentar(er)	Ctrl+T	Lägger till/tar bort kommentarssymbolen till/från början av den aktuella linjen.
Infoga flerradssträng	Ej tillämpligt	(Endast Editor) Infogar mall för strängar över flera linjer.
Sök	Ctrl+F	(Endast Editor) Öppnar <b>Sök</b> och söker efter den angivna strängen i det aktuella programmet.
Ersätt	Ctrl+H	(Endast Editor) Öppnar <b>Ersätt</b> och söker efter den angivna strängen i det aktuella programmet.
Gå till rad	Ctrl+G	(Endast Editor) Öppnar <b>Gå till rad</b> och hoppar till den angivna raden i det aktuella programmet.
Början av raden	Ctrl+8	Flyttar markören till början av den aktuella linjen.
Slutet på raden	Ctrl+2	Markören flyttas till slutet på den översta raden.
Hoppa till toppen	Ctrl+7	Flyttar markören till början av den första raden i programmet.
Hoppa till botten	Ctrl+1	Markören flyttas till slutet av programmets sista rad.

## Inbyggda menyer

### funktioner

Post	matris
def funktion():	Definierar en funktion som är beroende av specificerade variabler.
return	Definierar värdet som produceras av en funktion.

### Kontroller

Post	matris
if..	Villkorsuttryck.
if..else..	Villkorsuttryck.
if..elif..else..	Villkorsuttryck.
for index in range(size):	Itererar över ett område.
for index in range(start,stop):	Itererar över ett område.
for index in range(start,stop,step):	Itererar över ett område.
for index in list:	Itererar över listelement.
while..	Exekverar påståenden i ett kodblock tills ett villkor utvärderas till Falskt.
elif:	Villkorsuttryck.
else:	Villkorsuttryck.

### Ops

Post	matris
x=y	Anger variabelvärde.
x==y	Klistrar in lika med (==) jämförelseoperator.
x!=y	Klistrar in inte lika med (!=) jämförelseoperator.
x>y	Klistrar in större än (>) jämförelseoperator.
x>=y	Klistrar in större än eller lika med (>=) jämförelseoperator.
x<y	Klistrar in mindre än (<) jämförelseoperator.
x<=y	Klistrar in mindre än eller lika med (<=) jämförelseoperator.

Post	matris
respektive	Klistrar in och (och) logisk operator.
eller	Klistrar in eller (eller) logisk operator.
icke	Klistrar in inte (inte) logisk operator.
Sant	Klistrar in sant booleskt värde.
Falskt	Klistrar in falskt booleskt värde.

## Listor - definition

Post	matris
[]	Klistrar in parenteser ([]).
list()	Omvandlar sekvens till "list"-typ.
len()	Ger summan av elementen i listan.
max()	Returnerar maxvärdet i listan.
min()	Returnerar minsta värdet i listan.
.append()	Metoden lägger till ett element i en lista.
.remove()	Metoden tar bort den första instansen av ett element från en lista.
range(start,stop,step)	Returnerar en talrupsättning.
for index in range(start,stop,step)	Används för att iterera över ett område.
.insert()	Metoden lägger till ett element vid den angivna positionen.
.split()	Metoden returnerar en lista med element separerade med specificerad avgränsare.
sum()	Ger summan av elementen i en lista.
sorted()	Returnerar en sorterad lista.
.sort()	Metoden sorterar en öppnad lista.

## Typ

Post	matris
int()	Returnerar en heltalsdel.
float()	Returnerar ett flyttalsvärde.

Post	matris
round(x,ndigits)	Returnerar ett flyttal avrundat till specificerat antal siffror.
str()	Returnerar en sträng.
complex()	Ger ett komplext tal.
type()	Returnerar objektets typ.

## I/O

Post	matris
print()	Visar argument som strängar.
input()	Uppmanar användaren till inmatning.
eval()	Utvärderar ett uttryck representerat som en sträng.
.format()	Metoden formaterar den specificerade strängen.

## Menyn Matematik

**Obs:** När du skapar ett nytt program som använder den här modulen rekommenderas användning av program med **matematiska beräkningar**. Detta säkerställer att alla relevanta moduler importeras.

Post	matris
<code>from math import *</code>	Importerar alla metoder (funktioner) från den matematiska modulen.
<code>fabs()</code>	Returnerar absolutvärdet för ett reellt tal.
<code>sqrt()</code>	Returnerar kvadratroten för ett reellt tal.
<code>exp()</code>	Returnerar $e^{**x}$ .
<code>pow(x,y)</code>	Ger x upphöjt till potensen y.
<code>log(x,base)</code>	Returnerar $\log_{\text{bas}}(x)$ . $\log(x)$ utan bas ger den naturliga logaritmen x.
<code>fmod(x,y)</code>	Returnerar resten när x divideras med y. Använd när x och y är flyttal.
<code>ceil()</code>	Returnerar det minsta heltal som är större än eller lika med ett reellt tal.
<code>floor()</code>	Returnerar det största heltal mindre än eller lika med ett reellt tal.
<code>trunc()</code>	Trunkerar ett reellt tal till ett heltal.
<code>frexp()</code>	Returnerar ett par (y,n) där $x == y * 2^{**n}$ .

## Konst

Post	matris
<code>e</code>	Returns value for the constant e.
<code>pi</code>	Returns value for the constant pi.

## Trigonometri

Post	matris
<code>radians()</code>	Omvandlar vinkel i grader till radianer.

<b>Post</b>	<b>matris</b>
degrees()	Omvandlar vinkel i radianer till grader.
sin()	Returnerar sinus för argument i radianer.
cos()	Returnerar cosinus för argument i radianer.
tan()	Returnerar tangens för argument i radianer.
asin()	Returnerar arcsinus för argument i radianer.
acos()	Returnerar arccosinus för argument i radianer.
atan()	Returnerar arctangens för argument i radianer.
atan2(y,x)	Returnerar arctangens av y/x i radianer.

## menyn Slump

**Obs:** När du skapar ett nytt program som använder den här modulen rekommenderas användning av program med **sluppmässiga simuleringar**. Detta säkerställer att alla relevanta moduler importeras.

Post	matris
<code>from random import *</code>	Importerar alla metoder från modulen Slump.
<code>random()</code>	Returnerar ett flyttal från 0 till 1.0.
<code>uniform(min,max)</code>	Returnerar ett sluppmässigt tal $x$ (flyttal) så att $\min \leq x \leq \max$ .
<code>randint(min,max)</code>	Returnerar ett sluppmässigt heltal mellan min och max.
<code>choice(sequence)</code>	Returnerar ett sluppmässigt element från en talföljd som inte är tom.
<code>randrange(start,stop,step)</code>	Returnerar ett sluppmässigt tal från start till med angivet steg.
<code>seed()</code>	Initierar slumptalsgeneratoren.

## Ti PlotLib Menu

**Obs:** När du skapar ett nytt program som använder den här modulen rekommenderas användning av program med **plotting (x,y) & text**. Detta säkerställer att alla relevanta moduler importeras.

Post	matris
<code>import ti_plotlib as plt</code>	Importerar alla metoder (funktioner) från ti_plotlib-modulen i namnutrymmet "plt". Därför kommer alla funktionsnamn som klistras in från menyerna att föregås av "plt".

## Inställning

Post	matris
<code>cls()</code>	Rensar arbetsytan för plottning
<code>grid(x-scale,y-scale,"style")</code>	Visar ett rutnät med specificerad skala för x- och y-axlar.
<code>window(xmin,xmax,ymin,ymax)</code>	Definierar plottningsfönstret genom att kartlägga det angivna horisontella intervallet (xmin, xmax) och det vertikala intervallet (ymin, ymax) till det tilldelade plottningsområdet (pixlar).
<code>auto_window(x-list,y-list)</code>	Ritar automatiskt upp fönstret för att passa dataområdena i x-listan och y-listan som specificeras i programmet före <code>auto_window()</code> .
<code>axes("mode")</code>	Visar axlar i specificerat fönster i plottningsområdet.
<code>labels("x-label","y-label",x,y)</code>	Visar "x-label"- och "y-label"-etiketter på diagramaxlarna vid linjepositionerna x och y.
<code>title("title")</code>	Visar en centrerad titel på fönstrets övre linje.
<code>show_plot()</code>	Visar det buffrade ritningsresultatet. Funktionerna <code>use_buffer()</code> och <code>show_plot()</code> är användbara i fall där visning av flera objekt på skärmen kan orsaka fördröjningar (behövs normalt inte).
<code>use_buffer()</code>	Aktiverar en buffert utanför skärmen för att påskynda ritningsprocessen.



## Rita

Post	matris
color(red,green,blue)	Ställer in färgen för alla följande grafer/plottningar.
cls()	Rensar arbetsytan för plottning
show_plot()	Exekverar visningen av diagrammet som är inställt i programmet.
scatter(x-list,y-list,"mark")	Plottar en sekvens av ordnade par från (x-/y-lista) med den specificerade markeringsstilen.
plot(x-list,y-list,"mark")	Plottar en rad med ordnade par från specificerad x-/y-lista.
plot(x,y,"mark")	Plottar en punkt med koordinaterna x och y med den specificerade markeringsstilen.
line(x1,y1,x2,y2,"mode")	Plottar ett linjesegment från (x1,y1) till (x2,y2).
lin_reg(x-list,y-list,"display")	Beräknar och ritar den linjära regressionsmodellen, $ax+b$ , of x-list,y-list.
pen("size","style")	Ställer in alla följande linjer tills nästa pen() (penna) exekveras.
text_at(row,"text","align")	Visar "text" i plottningsområdet med specificerad "align" (justering).

## Egenskaper

Post	matris
xmin	Specificerad variabel för fönsterargument definierad som plt.xmin.
xmax	Specificerad variabel för fönsterargument definierad som plt.xmax.
ymin	Specificerad variabel för fönsterargument definierad som plt.ymin.
ymax	Specificerad variabel för fönsterargument definierad som plt.ymax.
m	Efter att plt.linreg() exekverats i ett program lagras de beräknade värdena för lutning, m, och skärningspunkt, b, i plt.m och plt.b.
b	Efter att plt.linreg() exekverats i ett program lagras de beräknade värdena för lutning, a, och skärningspunkt, b, i plt.m och plt.b.

## Ti hubb-meny

**Obs:** När du skapar ett nytt program som använder den här modulen rekommenderas användning av program med **hubbprojekt**. Detta säkerställer att alla relevanta moduler importeras.

Post	matris
<code>from ti_hub import *</code>	Importerar alla metoder från ti_hub-modulen.

### Hubb-inbyggda enheter > Färgutgång

Post	matris
<code>rgb(red,green,blue)</code>	Ställer in färgen för RGB LED-lampan (röd, grön, blå).
<code>blink(frequency,time)</code>	Ställer in blinkande frekvens och varaktighet för den valda färgen.
<code>off()</code>	Stänger av RGB LED-lampan.

### Hubb-inbyggda enheter > Ljusutgång

Post	matris
<code>on()</code>	Slår på LED-lampan.
<code>off()</code>	Slår av LED-lampan.
<code>blink(frequency,time)</code>	Ställer in blinkande frekvens och varaktighet för LED-lampan.

### Hubb-inbyggda enheter > Ljudutgång

Post	matris
<code>tone(frequency,time)</code>	En ton med specificerad frekvens spelas upp under specificerad tid.
<code>note("note",time)</code>	Spelar upp den angivna anteckningen för den angivna tiden. Tonen specificeras med hjälp av namnet och en oktav. Till exempel: A4, C5. Beteckningar är C, CS, D, DS, E, F, FS, G, GS, A, AS och B. Oktavnumren varierar från 1 till 9.

Post	matris
tone(frequency,time,tempo)	En ton med specificerad frekvens spelas upp under specificerad tid i specificerat tempo. Tempot definierar antalet pip per sekund från 0 till 10.
note("note",time,tempo)	Spelar upp specificerad ton under specificerad tid. Tonen specificeras med hjälp av namnet och en oktav. Till exempel: A4, C5. Beteckningar är C, CS, D, DS, E, F, FS, G, GS, A, AS och B. Oktavnumren varierar från 1 till 9. Tempovärde från 0 till 10.

## Hubb-inbyggda enheter > Ljusingång

Post	matris
measurement()	Läser den inbyggda BRIGHTNESS-sensor (ljusnivå) och returnerar en avläsning. Standardintervallet är 0 till 100. Detta kan ändras med funktionen range().
range(min,max)	Ställer in intervallet för mappning av värdena från ljusnivåsensorn. Om båda saknas eller ställs in på värdet None (inget), ställs standardljusintervallet på 0 till 100 in.

## Lägg till inmatningsenhet

Denna meny har en lista över de sensorer (inmatningsenheter) som stöds av `ti_hub`-modulen. Alla menyposter kommer att klistra in namnet på objektet och förvänta sig en variabel och en port som används med sensorn. Varje sensor har en `measurement()` method (mätmetod) som returnerar sensorns värde.

Post	matris
DHT (Digital fuktighet & temp)	Svarar med en lista som består av aktuell temperatur, fuktighet, typ av givare och senast lagrade avläsning.
Ranger	Svarar med aktuellt uppmätt avstånd från den specificerade avståndsmätaren via ultraljud. <ul style="list-style-type: none"> <li><b>measurement_time()</b> - Returnerar den tid det tar för ultraljudssignalen att nå objektet ("time of flight").</li> </ul>
Ljusnivå	Svarar med ljusstyrkan från den externa ljusnivåsensorn (ljusstyrka).

Post	matris
Temperatur	<p>Svarar med aktuell temperaturavläsning från den externa temperatursensorn.</p> <p>Standardkonfigurationen är att stödja Seeed temperatursensor i portarna IN 1, IN 2 eller IN 3.</p> <p>För att använda temperatursensorn TI LM19 från TI-Innovator™ Hub kopplingsplattepaketen, redigera porten till det BB-stift som används och använd ett valfritt argument "TIANALOG".</p> <p>Exempel: mylm19=temperature("BB 5","TIANALOG")</p>
Fuktighetsnivå	Svarar med en avläsning av luftfuktighetssensorn.
Magnetic	<p>Detekterar närvaro av ett magnetfält.</p> <p>Tröskelvärdet för att bestämma närvaro av fältet ställs in genom funktionen trigger().</p> <p>Standardtröskelvärdet är 150.</p>
Vernier	<p>Avläser värdet från den analoga Vernier-sensorn som angetts i kommandot.</p> <p>Kommandot stöder följande Vernier-sensor:</p> <ul style="list-style-type: none"> <li>• <b>temperatur</b> – IR-temperatursensor i rostfritt stål.</li> <li>• <b>ljusnivå</b> – TI-sensor för ljusstyrka.</li> <li>• <b>tryck</b> – ursprunglig gastrycksensor</li> <li>• <b>tryck</b> – nyare gastrycksensor.</li> <li>• <b>pH</b> – pH-sensor.</li> <li>• <b>kraft10</b> – ±10 N-inställning, dubbel kraftsensor.</li> <li>• <b>kraft50</b> – ±50 N-inställning, dubbel kraftsensor.</li> <li>• <b>accelerometer</b> – accelerometer för lågt G.</li> <li>• <b>generisk</b> – Möjliggör inställning av andra sensorer som inte stöds direkt ovan, och användning av calibrate() API (kalibrera API) ovan för att ställa in ekvationskoefficienter.</li> </ul>
Analog In	Stöder användning av enheter med analoga ingångar.
Digital In	Svarar med aktuellt tillstånd för digitalt stift anknutet till det digitala objektet eller det lagrade tillståndet för den digitala utgångsvärdet som senast sattes för objektet.
Potentiometer	<p>Stöder en potentiometersensor.</p> <p>Sensorns intervall kan ändras med funktionen range().</p>
Termistor	<p>Läser termistorsensorer.</p> <p>Standardkoefficienterna är utformade för att matcha den termistor som ingår i paketet med</p>

Post	matris
	<p>kopplingsplattan på TI-Innovator™ hubb, när den används med en fixerad resistor på 10 KΩ.</p> <p>En ny uppsättning kalibreringskoefficienter och referensresistans för termistorn kan konfigureras med funktionen <code>calibrate()</code>.</p>
Ljudstyrka	Med stöd för ljudnivåsensor.
Color Input	<p>Tillhandahåller gränssnitt till en I2C-ansluten färgångssensor.</p> <p>BB-portens stift används utöver I2C-porten för att kontrollera LED-lampan på färgsensorn.</p> <ul style="list-style-type: none"> <li>• <b>color_number()</b>: Returnerar ett värde från 1 till 9 som representerar den färg som sensorn detekterar. Siffrorna representerar färgerna enligt följande mappning: 1: Röd 2: Grön 3: Blå 4: Cyan 5: Magenta 6: Gul 7: Svart 8: Vit 9: Grå</li> <li>• <b>red()</b>: Returnerar ett värde från 0 till 255 som representerar intensiteten hos den RÖDA färgnivån som detekteras.</li> <li>• <b>green()</b>: Returnerar ett värde från 0 till 255 som representerar intensiteten hos den GRÖNA färgnivån som detekteras.</li> <li>• <b>blue()</b>: Returnerar ett värde från 0 till 255 som representerar intensiteten hos den BLÅ färgnivån som detekteras.</li> <li>• <b>gray()</b>: Returnerar ett värde från 0 till 255 som representerar den grå nivå som detekteras, där 0 är svart och 255 är vit.</li> </ul>
BB Port	<p>Ger stöd åt användning av alla 10 BB-portlås som en kombinerad digital ingångs-/utgångsport.</p> <p>Initieringsfunktionerna har en valfri "mask"-parameter som tillåter användning av undergruppen med 10 stift.</p>

Post	matris
	<ul style="list-style-type: none"> <li>• <b>read_port():</b> Läser aktuella värden på ingångsstiften i BB-porten.</li> <li>• <b>write_port(value):</b> Ställer in utgångsstifens värden till det angivna värdet, där värdet är mellan 0 och 1023. Observera att värdet också justeras mot maskeringsvärdet i var=bbport(mask)-operation om en maskering har angivits.</li> </ul>
Hub-tid	Ger åtkomst till den interna millisekundstimern.
TI-RGB-Array	<p>Innehåller funktioner för programmering av TI-RGB Array.</p> <p>Initieringsfunktionen accepterar en valfri "LAMP"-parameter för att aktivera läget med hög ljusstyrka för TI-RGB Array som kräver en extern strömförsörjning.</p> <ul style="list-style-type: none"> <li>• <b>set(led_position, r,g,b):</b> Ställer in en specifik led_position (0–15) till det specificerade r,g,b-värdet, där r,g,b är värden från 0 till 255.</li> <li>• <b>set(led_list,red,green,blue):</b> Ställer in lysdioderna som definieras i "led_list" till färgen som anges av "röd", "grön", "blå". "led_list" är en Python-lista som innehåller index för lysdioderna från 0 till 15. Till exempel kommer set ([0,2,4,6,15], 0, 0, 255) att ställa in lysdioderna 0, 2, 4, 6 och 15 på blått.</li> <li>• <b>set_all(r,g,b):</b> Ställer in alla RGB LED-lampor i matrisen till samma r,g,b-värde.</li> <li>• <b>all_off():</b> Stänger av alla RGB:er i matrisen.</li> <li>• <b>measurement():</b> Svarar med den ungefärliga strömförbrukning som RGB Array använder från TI-Innovator™ i milliampere.</li> <li>• <b>pattern(pattern):</b> Att använda värdet för argumentet som ett binärt värde i intervallet 0 till 65535 aktiverar pixlar där ett 1-värdet i representationen skulle vara. LED-lampor som är aktiverade som RÖDA med pwm-värdet på 255.</li> <li>• <b>pattern(value,red,green,blue):</b> Ställer in lysdioderna som definieras i "led_list" till färgen som anges av "röd", "grön", "blå".</li> </ul>

## Lägg till utmaningsenhet

Denna meny har en lista över de sensorer (utmaningsenheter) som stöds av ti\_hub-modulen. Alla menyposter kommer att klistra in namnet på objektet och förvänta sig en variabel och en port som används med enheten.

Post	matris
Lysdiod	Funktioner för kontroll av externt anslutna LED-lampor.
RGB-diod	Stöd för kontroll av externa RGB LED-lampor.
TI-RGB-Array	Innehåller funktioner för programmering av TI-RGB Array.
Högtalare	Funktioner för att stödja en extern högtalare med TI-Innovator™ Hub. Funktionerna är desamma som de för "ljud" ovan.
Effekt	Funktioner för kontroll av extern ström med TI-Innovator™ Hub. <ul style="list-style-type: none"> <li>• <b>set(value)</b>: Ställer in effektnivån på det angivna värdet, mellan 0 och 100.</li> <li>• <b>on()</b>: Ställer in effektnivån på 100.</li> <li>• <b>off()</b>: Ställer in effektnivån på 0.</li> </ul>
Kontinuerlig servo	Funktioner för kontroll av kontinuerliga servomotorer. <ul style="list-style-type: none"> <li>• <b>set_cw(speed,time)</b>: Servomotorn roterar medurs med angiven hastighet (0–255) och för den specifika varaktigheten på sekunder.</li> <li>• <b>set_ccw(speed,time)</b>: Servomotorn roterar moturs med angiven hastighet (0–255) och för den specifika varaktigheten i sekunder.</li> <li>• <b>stop()</b>: Stoppar den kontinuerliga servon.</li> </ul>
Analog Out	Funktion som stöder användning av enheter med analoga ingångar.
Vibrationsmotor	Funktioner för styrning av vibrationselement. <ul style="list-style-type: none"> <li>• <b>set(val)</b>: Ställer in intensiteten hos vibrationselement på "val" (0–255).</li> <li>• <b>off()</b>: Stänger av vibrationselementet.</li> <li>• <b>on()</b>: Slår på vibrationselementet på den högsta nivån.</li> </ul>
Relä	Styr gränssnittet för kontroll av reläer. <ul style="list-style-type: none"> <li>• <b>on()</b>: Ställer in reläet på läget ON.</li> <li>• <b>off()</b>: Ställer in reläet på läget OFF.</li> </ul>
Servo	Funktioner för kontroll av servomotorer. <ul style="list-style-type: none"> <li>• <b>set_position(pos)</b>: Ställer in svepservopositionen inom intervallet -90 till +90.</li> <li>• <b>zero()</b>: Ställer in svepservon på nollläget.</li> </ul>
Squarewave	Funktioner för att generera en fyrkantsvåg. <ul style="list-style-type: none"> <li>• <b>set(frequency,duty,time)</b>: Ställer in fyrkantsvågens form med en standardarbetscykel på 50 % (om ingen uppgift anges) och en utgångsfrekvens specificerad av "frequency" (frekvens). Frekvensen kan vara från 1 till 500 Hz. Arbetscykeln, om</li> </ul>

Post	matris
	specificerad, kan vara från 0 till 100 %. <ul style="list-style-type: none"> <li>• <b>off()</b>: Stänger av fyrkantsvågen.</li> </ul>
Digital utgång	Gränssnitt för kontroll av digital utgång. <ul style="list-style-type: none"> <li>• <b>set(val)</b>: Ställer in den digitala utgången på det värde som anges av "val" (0 eller 1).</li> <li>• <b>on()</b>: Ställer in statusen för den digitala utgången till hög (1).</li> <li>• <b>off()</b>: Ställer in statusen för den digitala utgången till låg (0).</li> </ul>
BB Port	Innehåller funktioner för programmering av TI-RGB Array. Se informationen ovan.

## Kommandon

Post	matris
sleep(seconds)	Pausar programkörningen under ett visst antal sekunder. Imported from the 'time' module (Importerad från tidsmodulen).
text_at(row,"text","align")	Visar angiven "text" i plottningsområdet vid specificerad "align" (justering). Del av ti_plotlib-modulen.
cls()	Rensar Shell-skärmen för plottning. Del av ti_plotlib-modulen.
medan get_key() != "esc":	Kör kommandona i "while"-slingan tills "esc"-tangents trycks in.
get_key()	Returnerar en sträng som representerar den tangent som trycks ned. Knappen "1" returnerar "1", "esc" returnerar "esc" och så vidare. När den anropas utan några parametrar, get_key(), så svarar den omedelbart. När den anropas med en parameter, get_key(1), så väntar den tills en tangent trycks ned. Del av ti_system-modulen.

## Portar

Det här är de ingångs- och utgångsportar som finns på TI-Innovator™ Hub.



Post
------

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
------

BB 9
------

BB 10
-------

I2C
-----

## Ti Rover-menyn

**Obs:** När du skapar ett nytt program som använder den här modulen rekommenderas användning av program med **Rover-kodning**. Detta säkerställer att alla relevanta moduler importeras.

Post	matris
import ti_rover as rv	Importerar alla metoder (funktioner) från ti_rover-modulen i namnutrymmet "rv". Därför kommer alla funktionsnamn som klistras in från menyerna att föregås av "rv".

## Drive

Post	matris
forward(distance)	Flyttar Rovern framåt det angivna avståndet i rutnätsenheter.
backward(distance)	Flyttar Rovern bakåt det angivna avståndet i rutnätsenheter.
left(angle_degrees)	Vrider Rovern vänster i den angivna vinkeln i grader.
right(angle_degrees)	Vrider Rovern höger i den angivna vinkeln i grader.
stop()	Stoppar pågående rörelse omedelbart.
stop_clear()	Stoppar all aktuell rörelse omedelbart och raderar alla väntande kommandon.
resume()	Återupptar behandlingen av kommandon.
stay(time)	Rover står still under en specificerad tid i sekunder (valfritt). Om ingen tid specificeras stannar Rover i 30 sekunder.
to_xy(x,y)	Flyttar Rover till koordinatposition (x,y) på virtuellt rutnät.
to_polar(r,theta_degrees)	Flyttar Rover till polär koordinatposition (r,theta) på virtuellt rutnät. Vinkeln anges i grader.
to_angle(angle,"unit")	Vrider Rover till den angivna vinkeln i det virtuella rastret. Vinkeln är relativ till en nollvinkel som pekar nedåt x-axeln i det virtuella rutnätet.

## Kör > Kör med alternativ

Post	matris
<code>forward_time(time)</code>	Flyttar Rover framåt under den angivna tiden.
<code>backward_time(time)</code>	Flyttar Rover bakåt under den angivna tiden.
<code>forward(distance,"unit")</code>	Flyttar Rover framåt vid standardhastigheten för det angivna avståndet. Avståndet kan specificeras i rasterenheter, meter eller hjulvarv.
<code>backward(distance,"unit")</code>	Flyttar Rover bakåt vid standardhastigheten för det angivna avståndet. Avståndet kan specificeras i rasterenheter, meter eller hjulvarv.
<code>left(angle,"unit")</code>	Vrider Rover vänster i den angivna vinkeln. Vinkeln kan vara i grader, radianer eller nygrader.
<code>right(angle,"unit")</code>	Vrider Rover höger i den angivna vinkeln. Vinkeln kan vara i grader, radianer eller nygrader.
<code>forward_time(time,speed,"rate")</code>	Flyttar Rover framåt i angivna hastigheten vid angiven tidpunkt. Hastigheten kan specificeras i rutnätsenheter, meter eller hjulvarv.
<code>backward_time(time,speed,"rate")</code>	Flyttar Rover bakåt i angivna hastigheten vid angiven tidpunkt. Hastigheten kan specificeras i rutnätsenheter, meter eller hjulvarv.
<code>forward(distance,"unit",speed,"rate")</code>	Flyttar Rover framåt i angivna hastigheten till angivet avstånd. Avståndet kan specificeras i rasterenheter, meter eller hjulvarv. Hastigheten kan specificeras i rutnätsenheter, meter eller hjulvarv.
<code>backward(distance,"unit",speed,"rate")</code>	Flyttar Rover bakåt i angivna hastigheten till angivet avstånd. Avståndet kan specificeras i rasterenheter, meter eller hjulvarv. Hastigheten kan specificeras i rutnätsenheter, meter eller hjulvarv.

## Inmatningar

Post	matris
ranger_measurement()	Läser av avståndssensorn (ultraljud) på Roverns front och returnerar det aktuella avståndet i meter.
color_measurement()	Svarar med ett värde från 1 till 9, vilket indikerar att den dominerande färgen har detekterats av den Roverns färginmatningssensorn. 1 = röd 2 = grön 3 = blå 4 = cyan 5 = magenta 6 = gul 7 = svart 8 = grå 9 = vit
red_measurement()	Returnerar ett värde mellan 0 och 255 som indikerar den uppfattade röda nivån som detekterats av färginmatningssensorn.
green_measurement()	Returnerar ett värde mellan 0 och 255 som indikerar den uppfattade gröna nivån som detekterats av färginmatningssensorn.
blue_measurement()	Returnerar ett värde mellan 0 och 255 som indikerar den uppfattade blå nivån som detekterats av färginmatningssensorn.
gray_measurement()	Returnerar ett värde mellan 0 och 255 som indikerar den uppfattade grå nivån som detekterats av färginmatningssensorn.
encoders_gyro_measurement()	Returnerar en lista med värden som innehåller vänster och höger hjulkodare samt den aktuella gyro-rubriken.
gyro_measurement()	Returnerar ett värde som representerar den aktuella gyro-avläsningen, inklusive driften, i grader.
ranger_time()	Returnerar den tid som ultraljudssignalen från avståndsmätaren för TI-Rover tar för att nå objektet ("time of flight").

## Utdata

Post	matris
<code>color_rgb(r,g,b)</code>	Ställer in färgen på Roverns RGB LED-lampa på de specifika röda, gröna och blåa värdena.
<code>color_blink(frequency,time)</code>	Ställer in blinkande frekvens och varaktighet för den valda färgen.
<code>color_off()</code>	Stänger av Roverns RGB LED-lampa.
<code>motor_left(speed,time)</code>	Ställer in den vänstra motoreffekten till det angivna värdet för den angivna varaktigheten. Hastigheten ligger inom intervallet -255 till 255 där 0 är stopp. Positiva hastighetsvärden är moturs rotation och negativa hastighetsvärden är medurs. Den valfria tidsparametern, om specificerad, har ett giltigt intervall på 0,05 till 655,35 sekunder. Om inget anges används en standard på 5 sekunder.
<code>motor_left(speed,time)</code>	Ställer in den vänstra motoreffekten till det angivna värdet för den angivna varaktigheten. Hastigheten ligger inom intervallet -255 till 255 där 0 är stopp. Positiva hastighetsvärden är moturs rotation och negativa hastighetsvärden är medurs. Den valfria tidsparametern, om specificerad, har ett giltigt intervall på 0,05 till 655,35 sekunder. Om inget anges används en standard på 5 sekunder.
<code>motors("ldir",left_val,"rdir",right_val,tid)</code>	Ställer in vänster och höger hjul på de angivna hastighetsnivåerna under en valfri tid i sekunder. Hastighetsvärdena ( <code>left_val</code> , <code>right_val</code> ) är i intervallet 0 till 255 där 0 är stopp. Parametrarna <code>ldir</code> och <code>rdir</code> specificerar medurs eller moturs rotation för respektive hjul. Den valfria tidsparametern, om specificerad, har ett giltigt intervall på 0,05 till 655,35 sekunder. Om inget anges används en standard på 5 sekunder.

## Sökväg

Post	matris
waypoint_xythdrn()	Avläser x-koordinat, y-koordinat, tid, riktning, förflyttning, antal hjulvarv, kommandonummer för tidigare position. Svarar med en lista av alla dessa värden som element.
waypoint_prev	Avläser x-koordinat, y-koordinat, tid, riktning, förflyttning, antal hjulvarv, kommandoantal för tidigare position.
waypoint_eta	Returnerar den beräknade tiden för att köra till en vägpunkt.
path_done()	Returnerar värdet 0 eller 1 beroende på om Rovern rör sig (0) eller står still (1).
pathlist_x()	Svarar med en lista över X-värden från början och inkluderande den aktuella vägpunktens X-värde.
pathlist_y()	Svarar med en lista över Y-värden från början och inkluderande den aktuella vägpunktens Y-värde.
pathlist_time()	Svarar med en lista över den tid i sekunder från början och inkluderande nuvarande tidsvärde för vägpunkt.
pathlist_heading()	Svarar med en lista av riktningar från början inkluderande aktuellt värde på vägpunktsriktning.
pathlist_distance()	Svarar med en lista på de avstånd som har tillryggalagts från början inkluderande aktuellt värde på vägpunktsavstånd.
pathlist_revs()	Svarar med en lista på det antal varv som har avverkats från början inkluderande aktuellt värde på vägpunktsvarv.
pathlist_cmdnum()	Svarar med en lista av kommandonummer för vägen.
waypoint_x()	Svarar med x-koordinat för aktuell vägpunkt.
waypoint_y()	Svarar med y-koordinat för aktuell vägpunkt.
waypoint_time()	Svarar med använd tid för förflyttning från tidigare till nuvarande vägpunkt.
waypoint_heading()	Returnerar absolut rubrik för aktuell vägpunkt.
waypoint_distance()	Svarar med förflyttningsavståndet mellan tidigare och aktuell position.
waypoint_revs()	Svarar med antalet varv som krävs för förflyttning mellan tidigare och nuvarande position.

## Settings

Post	matris
units/s	Alternativ för hastighet i enheter per sekund.
m/s	Alternativ för hastighet i meter per sekund.
revs/s	Alternativ för hastighet i hjulvarv per sekund.
Enheter	Alternativ för avstånd i rutnätsenheter.
m	Alternativ för avstånd i meter.
revs	Alternativ för avstånd i hjulvarv.
grader	Alternativ för att vända i grader.
radianer	Alternativ för att vända i radianer.
gradians	Alternativ för att vända i nygrader.
clockwise	Alternativ för att specificera hjulriktningen.
counter-clockwise (moturs)	Alternativ för att specificera hjulriktningen.

## Kommandon

Dessa kommandon är en funktionssamling från andra moduler samt från TI Rover-modulen.

Post	matris
sleep(seconds)	Pausar programkörningen under ett visst antal sekunder. Importerad från tidsmodulen.
text_at(row,"text","align")	Visar "text" i plottningsområdet med specificerad "align" (justering). Importerad från ti_plotlib-modulen.
cls()	Rensar Shell-skärmen för plottning. Importerad från ti_plotlib-modulen.
medan get_key() != "esc":	Kör kommandona i "while"-slingan tills "esc"-tangenter trycks in.
wait_until_done()	Pausar programmet tills Rovern avslutar det aktuella kommandot. Detta är ett användbart sätt att synkronisera icke-Rover-kommandon med Rovern.
while not path_done()	Kör kommandona i "while"-slingan tills Rovern är klar med all rörelse. Funktionen path_done() returnerar värdet 0 eller 1

Post	matris
	beroende på om Rovern rör sig (0) eller står still (1).
position(x,y)	Ställer in positionen för Rovern på det virtuella rutnätet till den specificerade x,y-koordinaten.
position(x,y,heading,"unit")	Ställer in positionen för Rovern på det virtuella rutnätet till den specificerade x,y-koordinaten och den virtuella rubriken i förhållande till den virtuella x-axeln ställs in om en rubrik tillhandahålls (i enheterna för specificerade vinklar). Positiva vinklar från 0 till 360 antas vara moturs från den positiva x-axeln. Negativa vinklar från 0 till 360 antas vara medurs från den positiva x-axeln.
grid_origin()	Väljer RV som aktuell nollpunkt (0,0) i koordinatsystemet.
grid_m_unit(scale_value)	Ställer in det virtuella rutnätsavståndet i meter per enhet (m/enhet) till det specificerade värdet. 0,1 är standardmeter/enhet och motsvarar 1 enhet = 100 mm, 10 cm, 1 dm eller 0,1 m. Intervallet för giltigt scale_value (skalvärde) är från 0,01 till 10,0.
path_clear()	Rensar tidigare befintlig information om vägar eller vägpunkter.
zero_gyro()	Återställer Roverns gyro till 0,0-vinkel och rensar vänster och höger hjulkodare.



## Menyn Komplex matematik

Denna undermeny finns under **Fler moduler**.

Post	matris
from cmath import *	Importerar alla metoder från modulen cmath.
complex(real,imag)	Ger ett komplext tal.
rect(modulus,argument)	Omvandlar polära koordinater till rektangulär form av ett komplext tal.
.real	Returnerar den realdelen av det komplexa talet.
.imag	Returnerar imaginärdelen av ett komplext tal.
polar()	Omvandlar rektangulär form till polära koordinater av ett komplext tal.
phase()	Returnerar argumentet av ett komplext tal.
exp()	Returnerar $e^{**}x$ .
cos()	Returnerar cosinus för ett komplext tal.
sin()	Returnerar sinus för ett komplext tal.
log()	Ger den naturliga logaritmen för ett komplext tal.
log10()	Ger bas 10-logaritmen för ett komplext tal.
sqrt()	Returnerar kvadratroten för ett komplext tal.

## Menyn Tid

Denna undermeny finns under **Fler moduler**.

Post	matris
from time import *	Importerar alla metoder från modulen time.
sleep(seconds)	Pausar programkörningen under ett visst antal sekunder.
clock()	Returnerar den aktuella processortiden som ett flytande tal uttryckt i sekunder.
localtime()	Omvandlar en tid som uttrycks på sekunder sedan 1 januari 2000 i en nio-tupel som innehåller en flagga för år, månad, dag, timme, minut, sekund, veckodag, år och sommardag. Om det valfria argumentet (seconds) inte tillhandahålls används realtidsklockan.
ticks_cpu()	Returnerar en processorspecifik, ökande millisekunderäknare med godtycklig referenspunkt. För att mäta tid konsekvent över olika system, använd ticks_ms().
ticks_diff()	Mäter perioden mellan på varandra följande anrop till ticks_cpu () eller ticks_ms(). Denna funktion ska inte användas för att mäta godtyckligt långa tidsperioder.

## Tl systemmeny

Denna undermeny finns under **Fler moduler**.

**Obs:** När du skapar ett nytt program som använder den här modulen rekommenderas användning av program med **datadelning**. Detta säkerställer att alla relevanta moduler importeras.

Post	matris
<code>from ti_system import *</code>	Importerar alla metoder (funktioner) från den <code>ti_system</code> -modulen.
<code>recall_value("name")</code>	Återkallar en fördefinierad OS-variabel (värde) med namnet "namn".
<code>store_value("name",value)</code>	Lagrar en Python-variabel (värde) som en OS-variabel med namnet "namn".
<code>recall_list("name")</code>	Återkallar en fördefinierad OS-lista med namnet "namn".
<code>store_list("name",list)</code>	Lagrar en Python-lista (lista) till en OS-listvariabel med namnet "namn".
<code>eval_function("name",value)</code>	Utvärderar en fördefinierad OS-funktion med det specificerade värdet.
<code>get_platform()</code>	Returnerar "hh" för handenhet och "dt" för dator.
<code>get_key()</code>	Returnerar en sträng som representerar den tangent som trycks ned. Knappen "1" returnerar "1", "esc" returnerar "esc" och så vidare. När den anropas utan några parametrar, <code>get_key()</code> , så svarar den omedelbart. När den anropas med en parameter, <code>get_key(1)</code> , så väntar den tills en tangent trycks ned.
<code>get_mouse()</code>	Returnerar muskoordinater som en tupel med två element, antingen arbetsytans-pixelposition eller (-1, -1) om utanför arbetsytan.
<code>while get_key() != "esc":</code>	Kör kommandona i "while"-loopen tills "esc"-tangents trycks in.
<code>clear_history()</code>	Rensar historiken i Shell.
<code>get_time_ms()</code>	Svarar med tid i millisekunder med millisekundprecision. Denna funktion kan användas för att beräkna en varaktighet snarare än att bestämma den faktiska klocktiden.

## Tl Draw-meny

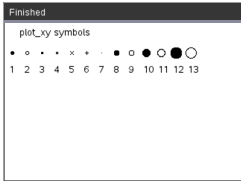
Denna undermeny finns under **Fler moduler**.

**Obs:** När du skapar ett nytt program som använder den här modulen rekommenderas användning av program med både **geometri och grafer**. Detta säkerställer att alla relevanta moduler importeras.

Post	matris
from ti_draw import *	Importerar alla metoder från modulen ti_draw.

### Form

Post	matris
draw_line()	Ritar en linje från den specificerade x1,y1-koordinaten till x2,y2.
draw_rect()	Ritar en rektangel med början i den specificerade x,y-koordinaten med specificerad bredd och höjd.
fill_rect()	Ritar en rektangel med början i den specificerade x,y-koordinaten med specificerad bredd, höjd och fyllning med den angivna färgen (med hjälp av set_color eller svart om ingen färg har angivits).
draw_circle()	Ritar en cirkel med början vid den specificerade x,y-mittkoordinaten med den specificerade radien.
fill_circle()	Ritar en cirkel med början vid angiven x,y-mittkoordinaten med den specificerade radien som fylls med specificerad färg (med hjälp av set_color eller svart om ingen färg har angivits).
draw_text()	Ritar en textsträng med början vid den specificerade x,y-koordinaten.
draw_arc()	Ritar en båge med början i den specificerade x,y-koordinaten med specificerad bredd, höjd och vinklar.
fill_arc()	Ritar en båge med början i den specificerade x,y-koordinaten med specificerad bredd, höjd och vinklar fyllda med den angivna färgen (med hjälp av set_color eller svart om ingen färg har angivits).
draw_poly()	Ritar en polygon med hjälp av de specificerade värdena i x-/y-listan.
fill_poly()	Ritar en polygon med hjälp av den specificerade x-/y-listvärdena fyllda med angiven färg (med hjälp av set_color eller svart om den inte har angivits).
plot_xy()	Ritar en form med specificerad x,y-koordinat och specificerat tal från 1-13 som representerar olika former och symboler (se nedan).

Post	matris
	

## Kontroller

Post	matris
<code>clear()</code>	Rensar hela skärmen. Kan användas med parametrarna <code>x,y,bredd,höjd</code> för att rensa en befintlig rektangel.
<code>clear_rect()</code>	Tar bort rektangeln med början i den specificerade <code>x,y</code> -koordinaten med specificerad bredd och höjd.
<code>set_color()</code>	Ställer in färgen på de former som följer i programmet tills en annan färg har ställts in.
<code>set_pen()</code>	Ställer in den angivna tjockleken och stilen på kanten när du ritar former (inte tillämpligt när du använder fyllningskommandon).
<code>set_window()</code>	Ställer in storleken på fönstret där alla former ska ritas. Den här funktionen är användbar för att ändra storlek på fönstret så att det matchar data eller för att ändra den ursprungliga (0,0) i arbetsytan för ritningen.
<code>get_screen_dim()</code>	Returnerar <code>xmax</code> och <code>ymax</code> av skärmens dimensioner.
<code>use_buffer()</code>	Aktiverar en buffert utanför skärmen för att påskynda ritningsprocessen.
<code>paint_buffer()</code>	Visar det buffrade ritningsresultatet. Funktionerna <code>use_buffer()</code> och <code>paint_buffer()</code> är användbara i fall där visning av flera objekt på skärmen kan orsaka fördröjningar.

## Anmärkningar

- Standardkonfigurationen har (0,0) i skärmens övre vänstra hörn. De positiva x-axelpunkterna till höger och de positiva y-axelpunkterna längst ner. Detta kan ändras med funktionen `set_window()`.
- Funktionerna i `ti_draw` är bara tillgängliga på handenheten och i handenhetens `vy` på skrivbordet.

## Ti bildmeny

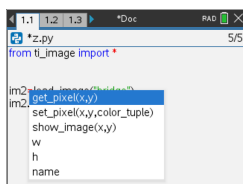
Denna undermeny finns under **Fler moduler**.

**Obs:** När du skapar ett nytt program som använder den här modulen rekommenderas användning av program med **bildbehandling**. Detta säkerställer att alla relevanta moduler importerats.

Post	matris
<code>from ti_image import *</code>	Importerar alla metoder från <code>ti_image</code> -modulen.
<code>new_image(width,height,(r,g,b))</code>	Skapar en ny bild med specificerad bredd och höjd för användning i Python-programmet. Färgen på den nya bilden definieras av värdena <code>(r,g,b)</code> .
<code>load_image("name")</code>	Läser in bilden som specificerats av "name" för användning i Python-programmet. Bilden måste vara en del av TNS-dokumentet antingen i anteckningar eller i grafapplikationen. Prompten "name" visar bildnamnen (om de har namngivits tidigare) eller ett nummer som anger deras infogningsorder.
<code>copy_image(image)</code>	Skapar en kopia av bilden som specificerats av variabeln "image".

### Methods of Image objects (metoder för bildobjektet)

Ytterligare funktioner som är relaterade till bildobjekten finns tillgängliga i Editor och Shell genom att skriva in variabelnamnet följt av en `.` (punkt).



- **get\_pixel(x,y):** Får värdet `(r,g,b)` för pixeln på plats definierad av koordinatparet `(x,y)`.

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

- **set\_pixel(x,y,color\_tuple):** Ställer in pixeln på plats `(x,y)` till färgen som anges i `color_tuple`.

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

Ställer in pixeln på `(100,100)` på färg `(0,0,255)`.

- **show\_image(x,y):** Visar bilden med det övre vänstra hörnet för plats `(x,y)`.

- **w, h, name:** Ger bildens bredd, höjd och namnparametrar.

### Till exempel

```
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)
```

## Menyn Variabler

**Obs:** Dessa listor innehåller inte variabler som definieras i några andra TI-Nspire™-applikationer.

Post	matris
Vars: Current Program	(Endast Editor) Visar en lista över globala funktioner och variabler som definierats i det aktuella programmet
Variabler: Last Run Program (kör om senaste programmet)	(Endast Shell) Visar en lista över globala funktioner och variabler som definierats i det senast körda programmet
Variabler: Alla	(Endast Shell) Visar en lista över globala funktioner och variabler från både i det senast körda programmet och importerade moduler



## Bilaga

---

Python-nyckelord .....	47
Python tangentmappning .....	48
Python-exempelprogram .....	50













## **Python-nyckelord**





















Följande nyckelord är inbyggda 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 tangentmappning

När du skriver in koden i redigeraren eller i Shell är knappsatsen utformad för att klistra in lämplig Python-aktivitet eller öppna menyer för enkel inmatning av funktioner, nyckelord, metoder, operatörer osv.

Knapp	Mapping
	Öppnar variabelmenyn
	Klistrar in = tecken
	Tar bort tecknet till vänster om markören
	Ingen åtgärd
	Klistrar in = tecken
	Klistrar in de valda symbolerna: <ul style="list-style-type: none"><li>• &gt;</li><li>• &lt;</li><li>• !=</li><li>• &gt;=</li><li>• &lt;=</li><li>• ==</li><li>• respektive</li><li>• eller</li><li>• icke</li><li>•  </li><li>• &amp;</li><li>• ~</li></ul>
	Klistrar in den valda funktionen: <ul style="list-style-type: none"><li>• 0</li><li>• cos</li><li>• 1.619775191</li><li>• atan2</li><li>• asin</li><li>• acos</li><li>• atan</li></ul>
	Visar tips
	Klistrar in :=
	Klistrar in **
	Ingen åtgärd
	Klistrar in **2

Knapp	Mapping
	Klistrar in sqrt()
	Klistrar in multiplikationstecken (*)
	Klistrar in ett dubbelt citationstecken (")
	Klistrar in divisionstecken (/)
	Ingen åtgärd
	Klistrar in exp()
	Klistrar in log()
	Klistrar in 10**
	Klistrar in logg(value,base)
	Klistrar in (
	Klistrar in )
	Klistrar in [ ]
	Klistrar in { }
	Klistrar in subtraktionstecken (-)
	Lägger till en ny rad efter den aktuella raden
	Klistrar in E
	Klistrar in de valda symbolerna: <ul style="list-style-type: none"> <li>• ?</li> <li>• !</li> <li>• \$</li> <li>• °</li> <li>• ' <ul style="list-style-type: none"> <li>• %</li> <li>• "</li> <li>• :</li> <li>• ;</li> <li>• -</li> <li>• \</li> <li>• #</li> </ul> </li> </ul>
	Klistrar in "pi"
	Befintligt flaggbeteende
	Lägger till en ny rad efter den aktuella raden

## Python-exempelprogram

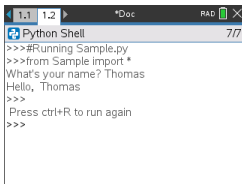
Använd följande exempelprogram för att bekanta dig med Pythons metoder. De finns också i filen **Getting Started Python.tns** i mappen **Exempel**.

**Obs:** Om du kopierar och klistrar in exempelkod som innehåller tab-indenteringsmarkeringar ( \* \* ) i TI-Nspire™ software, måste du ersätta dem med riktiga tab-indenteringar.

### 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")
```

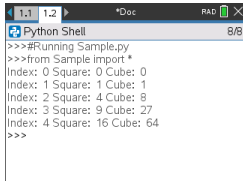


```
Python Shell 7/7
>>>#Running Sample.py
>>>from Sample import *
What's your name? Thomas
Hello, Thomas
>>>
Press ctrl+R to run again
>>>
```

## Ladda exempel

```
# 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 8/8
>>>#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
>>>
```

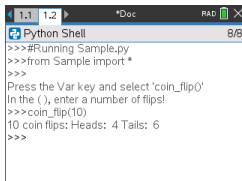
## Krona eller klave

```
# 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: RAD 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

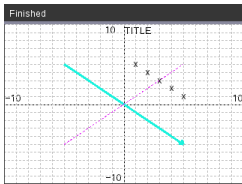
```
# Plotting example
import ti_plotlib as plt

# 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")
```





## Ritning

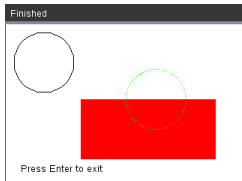
```
from ti_draw import *

# (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")
```



## Bild

```
# 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

Detta program använder Python för att styra TI-Innovator™ Hub, en programmerbar mikrostyrenhet. Om programmet körs utan att en TI-Innovator™ Hub ansluts visas ett felmeddelande.

För mer information om TI-Innovator™ Hub, besök [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()
```

## **Allmän information**

### ***Hjälp-funktion online***

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

Välj ditt land för ytterligare produktinformation.

### ***Kontakta TI support***

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

Välj ditt land för teknisk och andra supportresurser.

### ***Service- och garanti-information***

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

Välj ditt land för information om garantins längd och villkor eller om produkttjänsten.

Begränsad garanti. Denna garanti påverkar inte dina lagstadgade rättigheter.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243