

TI-Nspire™ Python Ohjelmointiopas

Tärkeää

Texas Instruments ei myönnä minkäänlaista nimenomaista tai oletettua takuuta mukaan lukien rajoituksetta oletetut takuut soveltuvuudesta kaupankäynnin kohteeksi tai sopivuudesta tiettyyn tarkoitukseen koskien ohjelmia tai kirjallista aineistoa, jotka annetaan saataville "sellaisina kuin ne ovat". Texas Instruments ei ole missään tapauksessa vastuussa näiden aineistojen hankinnasta tai käytöstä aiheutuvista erityisistä, rinnakkaisista, satunnaisista tai välillisistä vahingoista. Kannemuodosta riippumatta Texas Instrumentsin vastuu rajoittuu vain ja ainoastaan kyseisen tuotteen hankintahintaan. Texas Instruments ei myöskään ole velvoitettu vastaamaan minkäänlaisiin vaatimuksiin johtuen näiden materiaalien käytöstä muiden osapuolten toimesta.

© 2024 Texas Instruments Incorporated

"Python" ja Python-logot ovat Python Software Foundation -säätiön tavaramerkkejä tai rekisteröityjä tavaramerkkejä, joita Texas Instruments Incorporated käyttää säätiön luvalla.

Todelliset tuotteet saattavat erota hieman mukana tulevista kuvista.

Python-ohjelmoinnin aloittaminen	1
Python-moduulit	1
Python-ohjelman asentaminen moduulina	2
Python-työtilat	4
Python-editori	4
Python-komentotulkki	8
Python-valikkokartta	12
Toiminnot-valikko	12
Ajovalikko	13
Työkalut-valikko	13
Muokkausvalikko	14
Sisäänrakennettujen kohteiden valikko	16
Matematiikka-valikko	18
Satunnaislukovalikko	19
TI PlotLib -valikko	20
TI Hub -valikko	22
TI Rover -valikko	30
Monimutkaisen matematiikan valikko	37
Aikavalikko	38
TI-järjestelmävalikko	38
TI-kuvaajavalikko	39
TI Image -valikko	41
Muuttujat-valikko	43
Liite	44
Python-avainsanat	44
Python-näppäintapahtumat	44
Python-malliohjelmat	46
Yleistä	54

Python-ohjelmoinnin aloittaminen

Käyttämällä Pythonia TI-Nspire™-tuotteiden kanssa voit:

- lisätä Python-ohjelmia TNS-tiedostoihin
- luoda Python-ohjelmia mallipohjien avulla
- olla vuorovaikutuksessa ja jakaa tietoja muiden TI-Nspire™-sovellusten kanssa
- olla vuorovaikutuksessa TI-Innovator™ Hubin ja TI-Innovator™ Roverin kanssa

TI-Nspire™ Python -toteutus pohjautuu MicroPythoniin, joka on pieni, mikrokontrollereilla toimimaan suunnitellun Python 3 -vakiokirjaston alajoukko. Alkuperäinen MicroPython-toteutus on mukautettu TI-käyttöön soveltuvaksi.

Huomaa: Jotkin numeeriset vastaukset saattavat erota laskimen tuloksista taustalla olevien matematiikkatoteutusten eroista johtuen.

Python on saatavilla näissä TI-Nspire™-tuotteissa:

Kämmenlaitteet	Työpöytäohjelmisto
TI-Nspire™ CX II	Opettajan TI-Nspire™ CX Premium Teacher Software -ohjelmisto
TI-Nspire™ CX II CAS	Opettajan TI-Nspire™ CX CAS Premium Teacher Software -ohjelmisto
TI-Nspire™ CX II-T	TI-Nspire™ CX Student -ohjelmisto
TI-Nspire™ CX II-T CAS	TI-Nspire™ CX CAS Student Software -ohjelmisto
TI-Nspire™ CX II-C	
TI-Nspire™ CX II-C CAS	

Huomaa: Useimmissa tapauksissa toiminnallisuus on täysin samanlainen sekä kämmenlaite- että ohjelmistonäkymässä, mutta joitakin eroja saattaa esiintyä. Tässä oppaassa sinun oletetaan käyttävän kämmenlaitetta tai ohjelmiston kämmenlaitenäkymää.

Python-moduulit

TI-Nspire™ Python sisältää seuraavat moduulit:

Vakiomoduliit	TI-moduulit
Matematiikka (math)	TI PlotLib (ti_plotlib)
Satunnaisluvut (random)	TI Hub (ti_hub)
Monimutkainen matematiikka (cmath)	TI Rover (ti_rover)
Aika (time)	TI-järjestelmä (ti_system)
	TI-piirto (ti_draw)
	TI Image (ti_image)

Huomaa: Jos sinulla on muissa Python-kehitysympäristöissä luotuja Python-ohjelmia, saattaa olla tarpeen muokata niitä toimimaan TI-Nspire™ Python -ratkaisussa. Moduulit saattavat käyttää erilaisia ohjelman sisäisiä menetelmiä, argumentteja sekä menetelmien järjestystä TI-moduuleihin verrattuna. Yleisesti ottaen yhteensopivuus kannattaa ottaa huomioon käytettäessä mitä tahansa Python- ja Python-moduulien versiota.

Siirtäessäsi Python-ohjelmia jostain muusta kuin TI-alustasta TI-alustalle TAI jostain TI-tuotteesta toiseen, muista:

- Ydinkieliominaisuuksia ja vakiokirjastoja (matematiikka, satunnainen jne.) käyttävät ohjelmat voidaan siirtää ilman muutoksia.
- Alustakohtaisia kirjastoja, kuten PC:lle tai TI-moduuleille suunniteltua matplotlib-kirjastoa käyttävät ohjelmat edellyttävät muokkauksia, jotta ne toimivat muilla alustoilla. Tämä saattaa päteä myös eri TI-alustojen kesken.

Kaikkien Python-versioiden tavoin on tuonit sisällytettävä, jotta jonkin tietyn moduulin sisältämiä toimintoja, menetelmiä tai vakioita voidaan käyttää. Esimerkiksi `cos()`-funktion suorittaminen matematiikkamoduulista edellyttää seuraavien komentojen käyttämistä:

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

Luettelo valikoista niiden sisältämine kohteineen sekä kuvauksineen löytyvät [Valikkokartta](#)-osiosta.

Python-ohjelman asentaminen moduulina

Python-ohjelman tallentaminen moduulina:

- Valitse editorissa **Toiminnot > Asenna Python-moduulina**.
- Valitse komentotulkista **Työkalut > Asenna Python-moduulina**.

Valinnan jälkeen tapahtuu seuraavaa:

- Python-syntaksi tarkistetaan.
- Tiedosto tallennetaan ja siirretään PyLib-kansioon.
- Näyttöön tulee valintaikkuna, jossa vahvistetaan, että tiedosto on asennettu moduulina.
- Tiedosto suljetaan ja moduuli on käyttövalmis.
- Moduulin nimi lisätään **Lisää moduuleja** -valikkoon, jossa on **<module> -tuonnista *** -valikkokohde.

Jos aiot jakaa tämän moduulin muiden kanssa, on suositeltavaa noudattaa näitä ohjeita:

- Tallenna vain yksi moduuli kutakin TNS-tiedostoa kohden.
- Moduulin nimi vastaa TNS-tiedoston nimeä (esim. "my_program"-moduuli on "my_program.tns"-tiedostossa).

- Lisää Muistiinpanot-sivun ennen Python-editoria; tämä sivu selittää moduulin tarkoituksen, version ja toiminnot.
- Käytä ver()-toimintoa moduulin versionumeron näyttämiseen.
- (Valinnainen) Lisää ohjetoiminto funktion menetelmäluettelon näyttämiseksi.

Python-työtilat

Python-ohjelmointiin on käytettävissä kaksi työtilaa: Python-editori ja Python-komentotulkki.

Python-editori	Python-komentotulkki
• Python-ohjelmien luominen, muokkaaminen ja tallentaminen • syntaksin korostus ja automaattinen sisennys • tekstiin sidotut kehoitteet toimintoargumenttien apuna • työkaluvihjeet kelvollisten arvojen välin näyttämiseksi • <code>var</code> avain näyttää senhetkisessä ohjelmassa määritetyt yleiset käyttäjämuuttujat ja -funktiot • näppäimistön pikavalinnat	• Python-ohjelmien suorittaminen • kätevä pienten koodifragmenttien testaamiseen • vuorovaikutus komentotulkin aikaisempien tapahtumien kanssa aiempien syöttöjen ja tuotosten valitsemiseksi uudelleenkäyttöä varten • <code>var</code> avain luettelee yleisiä, jossain tiettyssä ongelmassa suoritetun viimeisen ohjelman määritettyjä käyttäjämuuttujia

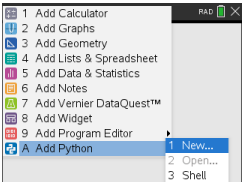
Huomaa: ongelmaan voidaan lisätä useita Python-ohjelmia ja -komentotulkkeja.

Python-editori

Python-editorissa voidaan luoda, muokata ja tallentaa Python-ohjelmia.

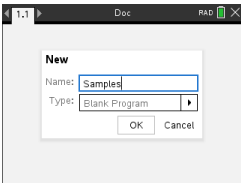
Python-editorisivun lisääminen

Jos haluat lisätä uuden Python-editorisivun nykyiseen ongelmaan, paina `menu` ja valitse **Add Python > New** (Lisää Python > Uusi).

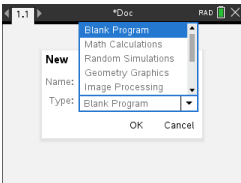


Voit luoda tyhjän ohjelman tai valita mallipohjan.

Tyhjä ohjelma



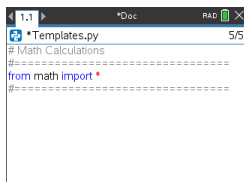
Mallipohja



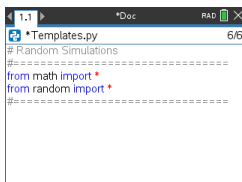
Python-editori tulee näkyviin ohjelman luonnin jälkeen. Jos valitsit mallipohjan, tarvittavat tuontilausekkeet lisätään automaattisesti (ks. alempana).

Huomaa: Yhdessä TNS-tiedostossa voi olla useita ohjelmia muiden sovellusten tavoin. Jos Python-ohjelma on tarkoitettu käytettäväksi moduulina, TNS-tiedosto voidaan tallentaa PyLib-kansioon. Tätä moduulia voidaan sitten käyttää muissa ohjelmissa ja asiakirjoissa.

Matemaattiset laskutoimitukset Satunnaislukusimulaatiot

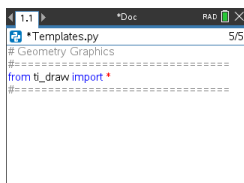


```
1.1 > *Doc RAD 5/5
*Templates.py
# Math Calculations
=====
from math import *
=====
```



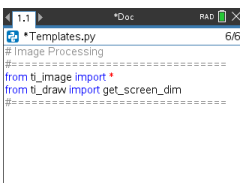
```
1.1 > *Doc RAD 6/6
*Templates.py
# Random Simulations
=====
from math import *
from random import *
=====
```

Geometrinen grafiikka



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

Kuvankäsittely



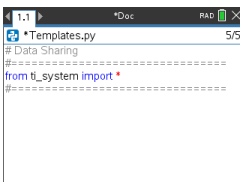
```
1.1 > *Doc RAD 6/6
*Templates.py
# Image Processing
=====
from ti_image import *
from ti_draw import get_screen_dim
=====
```

Piirtäminen (x,y) & teksti



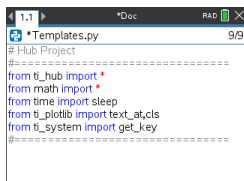
```
1.1 > *Doc RAD 5/5
*Templates.py
# Plotting (x,y) & Text
=====
import ti_plotlib as plt
=====
```

Tietojen jakaminen



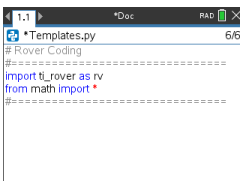
```
1.1 > *Doc RAD 5/5
*Templates.py
# Data Sharing
=====
from ti_system import *
=====
```

TI-Innovator Hub -projekti



```
1.1 > *Doc RAD 9/9
*Templates.py
# 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-koodaus



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


Python-ohjelman avaaminen

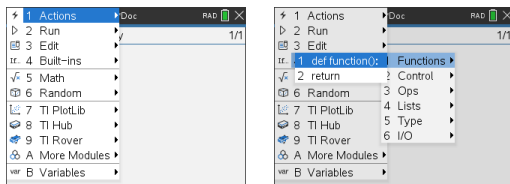
Voit avata aiemmin luodun Python-ohjelman painamalla `[doc]` ja valitsemalla **Insert > Add Python > Open** (Lisää > Lisää Python > Avaa). Näyttöön tulee luettelo TNS-tiedostoon tallennetuista ohjelmista.

Vaikka ohjelman luontiin käytetty editorisivu olisikin poistettu, ohjelma on edelleen käytettävissä TNS-tiedostossa.

Python-editorissa työskentely

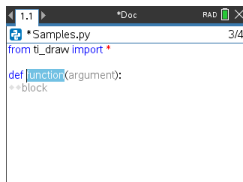
Document Tools (Asiakirjatyökalut) -valikon saa esiin painamalla `[menu]`. Näiden valikkovaihtoehtojen avulla voit lisätä, siirtää ja kopioida ohjelmasi koodilohkoja.

Asiakirjojen työkaluvalikko

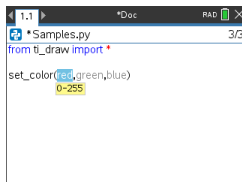


Moduulivalikoista valitut kohteet lisäävät automaattisesti koodimallin editoriin, jossa on tekstiin sidottuja kehotteita funktion jokaiselle osalle. Voit siirtyä yhdestä argumentista toiseen painamalla `[tab]` (eteenpäin) tai `[shift]+[tab]` (taaksepäin). Oikeiden arvojen valinnan avuksi tulee näkyviin työkaluvinkejä tai ponnahdusluetteloita, kun sellaisia on käytettävissä.

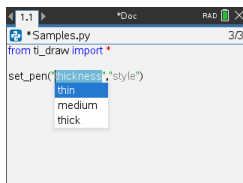
Tekstiin sidotut kehotteet



Työkaluvinkit



Ponnahdusluettelot



Ohjelman nimen oikealla puolella olevat numerot vastaavat kohdistimen nykyisen rivin rivinumeroa ja ohjelman rivien kokonaismäärää.

```

1.1 ▶ Samples RAD 7/12
Samples.py
from math import *

x=2
y=4.5
z=pi

def f1(n):
    return n**2

def f2(n):
    return n**3

```

Nykyisen kohdistimen sijainnin yläpuolella olevilla riveillä määritetyt yleiset funktiot ja muuttujat voidaan lisätä painamalla `var` ja valitsemalla ne luettelosta.

```

1.1 ▶ Samples RAD 11/11
Samples.py
from math import *

x=2
y=4.5
z=pi

def f1(n):
    return n**2

def f2(n):
    return n**3

```

Kun lisäät koodia ohjelmaan, editori näyttää avainsanoja, operaattoreita, kommentteja, merkijonoja ja sisennyksiä eri väreissä, jotta eri elementit on helpompi tunnistaa.

```

1.1 ▶ *Pythagoras.py RAD 9/9
*Pythagoras.py
def f(a,b,c):
    return a**2+b**2-c**2

def trirec(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")

```

Liitetyt valikkokohteet asetetaan nykyiselle riville tai seuraavalle riville kontekstin perusteella.

Versiota 6.2.0 edeltävät versiot *Versio 6.2.0 ja uudemmat*

```

1.1 ▶ *Doc RAD 2/2
*p1.py
import rv as rv
rv.forward(3)
rv.left(90)

```

```

1.1 ▶ *Doc RAD 4/4
*p2.py
import rv as rv
rv.forward(3)
rv.left(90)
rv.forward(3)

```

Päävalikko näyttää moduulivalikon muokattavan ohjelman perusteella, jotta valinnat ovat nopeammin käytettävissä. Vain yksi moduuli näytetään kerrallaan.

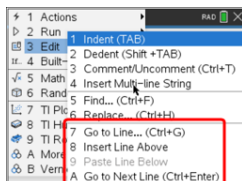
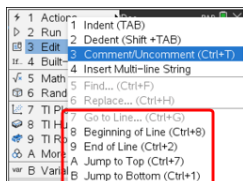
```

1.1 ▶ *Doc RAD 3/3
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 Vernier Go Direct

```

Muokkausvalikko näyttää yleisesti käytetyt valikkokohteet ja pikanäppäimet.

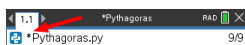
Versiota 6.2.0 edeltävät versiot *Versio 6.2.0 ja uudemmat*



Ohjelmien tallentaminen ja suorittaminen

Kun ohjelma on valmis, paina **[menu]** ja valitse **Run > Check Syntax & Save** (Suorita > Tarkista syntaksi > Tallenna). Tämä tarkistaa Python-ohjelman syntaksin ja tallentaa sen TNS-tiedostoon.

Huomaa: Jos ohjelmassa on tallentamattomia muutoksia, ohjelman nimen vieressä näkyy tähti.



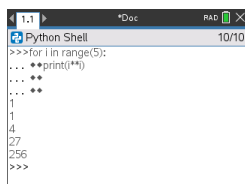
Voit suorittaa ohjelman painamalla **[menu]** ja valitsemalla **Run > Run** (Suorita > Suorita). Tämä suorittaa nykyisen ohjelman seuraavalla Python-komentotulkkisivulla tai uudella sivulla, jos seuraava sivu ei ole komentotulkki.

Huomaa: Ohjelman suorittaminen tarkistaa syntaksin automaattisesti ja tallentaa ohjelman.

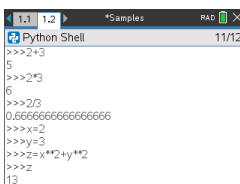
Python-komentotulkki

Python-komentotulkki on tulkki, joka suorittaa Python-ohjelmia, Python-koodin muita osia tai yksinkertaisia komentoja.

Python-koodi

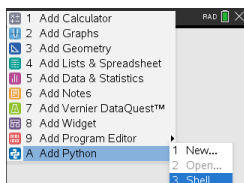


Yksinkertaiset komennot

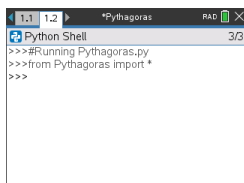


Python-komentotulkkisivun lisääminen

Jos haluat lisätä uuden Python-komentotulkkisivun senhetkiseen ongelmaan, paina **[menu]** ja valitse **Lisää Python > Komentotulkki**.



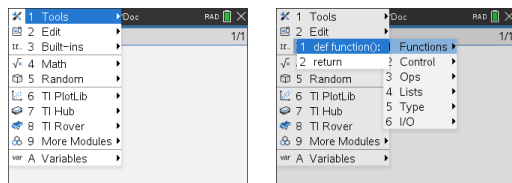
Python-komentotulkki voidaan käynnistää myös Python-editorista suorittaen ohjelman painamalla **menu** ja valitsemalla **Suorita > Suorita**.



Python-komentotulkin käyttäminen

Asiakirjatyökalut-valikon saa esiin painamalla **menu**. Näiden valikkovaihtoehtojen avulla voit lisätä, siirtää ja kopioida koodilohkoja.

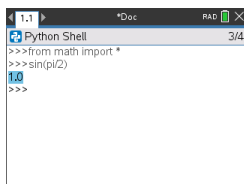
Asiakirjojen työkaluvalikko



Huomaa: Jos käytät jotakin käytettävissä olevien moduulien menetelmää, muista suorittaa tuontimoduulin lauseke ensin, kuten missä tahansa Python-koodausympäristössä.

Vuorovaikutus komentotulkin tuotoksen kanssa on samanlainen kuin Laskin-sovelluksen kanssa, jossa voit valita ja kopioida aiempia syötteitä ja tuotoksia käytettäväksi muualla komentotulkissa, editorissa tai muissa sovelluksissa.

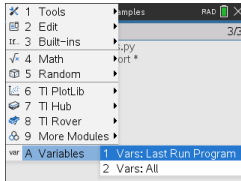
Valitse ylöspäin osoittavalla nuolella ja kopioi ja liitä sitten haluamaasi paikkaan



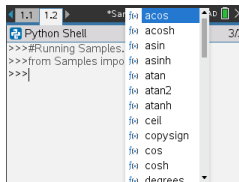
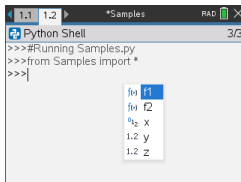
Viimeksi suoritetun ohjelman yleiset funktiot ja muuttujat voidaan lisätä painamalla **var** tai **ctrl** + **L** ja valitsemalla luettelosta tai painamalla **menu** ja valitsemalla **Muuttujat > Vars: Viimeksi suoritettu ohjelma**.

Jos haluat valita sekä viimeksi suoritetun ohjelman ja minkä tahansa tuotujen moduulien leisten funktioiden ja muuttujien luettelosta, paina **menu** ja valitse **Muuttujat > Vars: Kaikki**.

Muuttujat-valikko

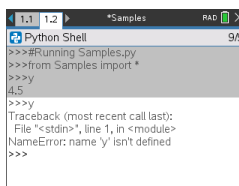
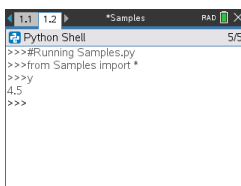


Viimeksi suoritetun ohjelman muuttujat *Kaikki muuttujat*



Kaikilla saman ohjelman Python-komentotulkisivuilla on sama tila (käyttäjän määrittämät ja tuodut muuttujamäärytykset). Kun tallennat tai suoritat Python-ohjelmaa kyseisessä ongelmassa tai painat **menu** ja valitset **Työkalut > Alusta komentotulkki uudelleen**, aikaisemmissa komentotulkin tapahtumissa on harmaa tausta. Tämä tarkoittaa, että edellinen tila ei ole enää kelvollinen.

Ennen tallentamista tai alustamista *Tallentamisen tai alustamisen jälkeen*



Huomaa: Valinta **menu** **Työkalut > Tyhjennä aikaisemmat tapahtumat** tyhjentää näkyvästä komentotulkin kaiken aikaisemman toiminnan, mutta muuttujat ovat edelleen käytettävissä.

Ilmoitukset

Python-istunnossa ollessasi esiin saattaa tulla virhe- ja muita tietoviestejä. Jos komentotulkkiin tulee virheilmoitus ohjelmaa suoritettaessa, näkyviin tulee ohjelman rivinumero. Paina **ctrl** + **menu** ja valitse **Siirry Python-editoriin**. Paina Editorissa **menu** ja

valitse sitten **Muokkaa > Siirry riville**. Anna rivin numero ja paina . Kohdistin näkyy sen rivin ensimmäisellä merkillä, jossa virhe ilmaantui.

Käynnissä olevan ohjelman keskeyttäminen

Kun ohjelma tai toiminto on käynnissä, näytetään ☹ varattu osoitin.

► Jos haluat pysäyttää ohjelman tai toiminnon,

- Windows®: Paina **F12**-painiketta.
- Mac®: Paina **F5**-painiketta.
- Kämmenlaite: Paina  -painiketta.

Python-valikkokartta

Tämä osio sisältää kaikki Python-editorin ja -komentotulkin valikot ja valikkokohteet sekä lyhyen kuvauksen kullekin.

Huomaa: Jos valikkokohteella on pikanäppäimiä, Mac®-käyttäjien tulee käyttää **⌘** (Cmd)-painiketta aina, kun **Ctrl**-painiketta käytetään. Täydellinen luettelo TI-Nspire™ kämmenlaitteiden ja -ohjelmistojen pikanäppäimistä löytyy TI-Nspire™ Technology -oppaasta.

Toiminnot-valikko	12
Ajovalikko	13
Työkalut-valikko	13
Muokkausvalikko	14
Sisäänrakennettujen kohteiden valikko	16
Matematiikka-valikko	18
Satunnaislukuvälikko	19
TI PlotLib -valikko	20
TI Hub -valikko	22
TI Rover -valikko	30
Monimutkaisen matematiikan valikko	37
Aikavälikko	38
TI-järjestelmävalikko	38
TI-kuvaajavälikko	39
TI Image -valikko	41
Muuttujat-valikko	43

Toiminnot-valikko

Huomaa: Tämä koskee vain editoria.

objekti	Kuvaus
Uusi	Avaa Uusi -valintaikkunan, johon kirjoitetaan nimi ja jossa valitaan uuden ohjelman tyyppi.
Avaa	Avaa senhetkisessä asiakirjassa käytettävissä olevien ohjelmien luettelon.
Luo kopio	Avaa Luo kopio -valintaikkunan, jossa senhetkinen ohjelma voidaan

objekti	Kuvaus
	tallentaa jollain toisella nimellä.
Nimeä uudelleen	Avaa Nimeä uudelleen -valintaikkunan, jossa senhetkinen ohjelma voidaan nimetä uudelleen.
Sulje	Sulkee senhetkisen ohjelman.
Asetukset	Avaa Asetukset -valintaikkunan, jossa voidaan muuttaa sekä editorin että komentotulkin kirjasinkokoa.
Asenna Python-moduulina	Tarkistaa senhetkisen TNS-tiedoston Python-syntaksin ja siirtää sen PyLib-kansioon.

Ajovalikko

Huomaa: Tämä koskee vain editoria.

objekti	Pikanäppäin	Kuvaus
Suorita	Ctrl+R	Tarkistaa syntaksin, tallentaa ohjelman ja suorittaa Python-komentotulkissa.
Tarkista syntaksi ja tallenna	Ctrl+B	Tarkistaa syntaksin ja tallentaa ohjelman.
Siirry komentotulkkiin	-	Siirtää painopisteen senhetkiseen ohjelmaan liittyvään komentotulkkiin tai avaa editorin viereen uuden komentotulkkisivun.

Työkalut-valikko

Huomaa: Tämä koskee vain komentotulkkia.

objekti	Pikanäppäin	Kuvaus
Suorita viimeinen ohjelma uudelleen	Ctrl+R	Suorittaa viimeisen senhetkiseen komentotulkkiin liittyvän ohjelman uudelleen.
Siirry Python-editoriin	-	Avaa senhetkiseen komentotulkkiin liittyvän editorisivun.
Suorita	-	Avaa senhetkisessä asiakirjassa käytettävissä olevien ohjelmien luettelon. Valittu ohjelma suoritetaan valinnan jälkeen.
Clear History (Tyhjennä historiatiedot)	-	Tyhjentää senhetkisen komentotulkin aikaisemmat tapahtumat, mutta ei

objekti	Pikanäppäin	Kuvaus
		alusta komentotulkkia uudelleen.
Alusta komentotulkki uudelleen	-	Palauttaa kaikkien avoinna olevien komentotulkkisivujen tilan senhetkisessä ongelmassa. Määritetyt muuttujat ja tuodut funktiot eivät ole enää käytettävissä.
dir()	-	Näyttää määritetyn moduulin toimintoluettelon, kun sitä käytetään tuontilausekkeen jälkeen.
From PROGRAM import *	-	Avaa senhetkisessä asiakirjassa käytettävissä olevien ohjelmien luettelon. Valinnan jälkeen tuontilauseke liitetään komentotulkkiin.
Asenna Python-moduulina	-	Käytössä vain binäärimuodossa oleville moduuleille. Siirtää nykyisen TNS-tiedoston PyLib-kansioon.

Muokkausvalikko

Huomaa: Ctrl+A valitsee kaikki koodi- tai tuotosrivit leikkaamista tai poistamista varten (vain editori) tai kopioimista ja liittämistä varten (editori ja komentotulkki).

Objekti	Pikanäppäin	Kuvaus
Sisennys	SARKAIN*	Sisentää tällä hetkellä valitun rivin tai valittujen rivien tekstin. * Vajaiden tekstiin sidottujen kehotteiden tapauksessa SARKAIN siirtyy seuraavaan kehotteeseen.
Ulonnus	Vaihtonäppäin+SARKAIN**	Uloittaa tällä hetkellä valitun rivin tai valittujen rivien tekstin. ** Vajaiden tekstiin sidottujen kehotteiden tapauksessa vaihtonäppäin+SARKAIN siirtyy edelliseen kehotteeseen.
Kommentoi / poista kommentti	Ctrl+T	Lisää kommenttisymbolin

Objekti	Pikanäppäin	Kuvaus
		tällä hetkellä valitun rivin alkuun tai poistaa sen rivin alusta.
Lisää monirivinen merkkijono	–	(Vain editori) Lisää monirivisen merkkijonomallin.
Etsi	Ctrl+F	(Vain editori) Avaa Find (Etsi) -valintaikkunan ja hakee syötettyä merkkijonoa nykyisestä ohjelmasta.
Korvaa	Ctrl+H	(Vain editori) Avaa Replace (Korvaa) -valintaikkunan ja hakee syötettyä merkkijonoa nykyisestä ohjelmasta.
Siirry riville	Ctrl+G	(Vain editori) Avaa Go to Line (Siirry riville) -valintaikkunan ja siirtyy määrätyle nykyisen ohjelman riville.
Rivin alku	Ctrl+8	Siirtää osoittimen nykyisen rivin alkuun.
Rivin loppu	Ctrl+2	Siirtää osoittimen nykyisen rivin loppuun.
Siirry yläreunaan	Ctrl+7	Siirtää kohdistimen ohjelman ensimmäisen rivin alkuun.
Siirry alareunaan	Ctrl+1	Siirtää kohdistimen ohjelman viimeisen rivin loppuun.
Tyhjennä rivi	Ctrl+Del	Tyhjentää sen rivin, jolla kohdistin on.

Sisäänrakennettujen kohteiden valikko

funktiot

objekti	Kuvaus
def function():	Määrittää määritetyistä muuttujista riippuvaisen funktion.
return	Määrittää funktion tuottaman arvon.

Control (Kontrolli)

objekti	Kuvaus
if..	Ehdollinen lauseke.
if..else..	Ehdollinen lauseke.
if..elif..else..	Ehdollinen lauseke.
for index in range(size):	Iteroi tietyllä vaihteluvälillä.
for index in range(start,stop):	Iteroi tietyllä vaihteluvälillä.
for index in range(start,stop,step):	Iteroi tietyllä vaihteluvälillä.
for index in list:	Iteroi luetteloelementeissä.
while..	Suorittaa koodilohkon sisältämät lausekkeet, kunnes ehto on epätosi.
elif:	Ehdollinen lauseke.
else:	Ehdollinen lauseke.

Operaattorit

objekti	Kuvaus
x=y	Määrittää muuttujan arvon.
x==y	Liittää 'yhtä kuin' (==) -vertailuoperaattorin.
x!=y	Liittää 'ei ole yhtä kuin' (!=) -vertailuoperaattorin.
x>y	Liittää 'suurempi kuin' (>) -vertailuoperaattorin.
x>=y	Liittää 'suurempi kuin' tai 'yhtä suuri kuin' (>=) -vertailuoperaattorin.
x<y	Liittää 'pienempi kuin' (<) -vertailuoperaattorin.
x<=y	Liittää 'pienempi kuin' tai 'yhtä suuri kuin' (<=) -vertailuoperaattorin.

objekti	Kuvaus
ja	Liittää loogisen operaattorin 'ja' (and).
tai	Liittää loogisen operaattorin 'tai' (or).
Ei	Liittää loogisen operaattorin 'ei' (no).
Tosi	Liittää boolean-arvon true.
Epätosi	Liittää boolean-arvon false.

Joukot

objekti	Kuvaus
[]	Liittää hakasulkeet ([]).
list()	Muuntaa sekvenssin "list"-tyypiksi (luettelo).
len()	Antaa luettelon elementtien määrän.
max()	Antaa luettelossa olevan enimmäisarvon.
min()	Antaa luettelossa olevan vähimmäisarvon.
.append()	Menetelmä liittää elementin luetteloon.
.remove()	Menetelmä poistaa luettelosta elementin ensimmäisen ilmentymän.
range(start,stop,step)	Antaa lukujoukon.
for index in range(start,stop,step)	Käytetään iterointiin tietyllä alueella.
.insert()	Menetelmä lisää elementin määritettyyn kohtaan.
.split()	Menetelmä antaa luettelon, jossa elementit on erotettu määritetyllä erottimella.
sum()	Antaa luettelon elementtien summan.
sorted()	Antaa lajitellun luettelon.
.sort()	Menetelmä lajittelee paikallaan olevan luettelon.

Tyyppi

objekti	Kuvaus
int()	Palauttaa kokonaisosan.

objekti	Kuvaus
float()	Antaa liukulukuarvon.
round(x,ndigits)	Antaa liukuluvun pyöristettynä määritettyyn numeroiden määrään.
str()	Antaa merkkijonon.
complex()	Antaa vastauksen kompleksilukuna.
type()	Antaa objektin tyyppin.

I/O

objekti	Kuvaus
print()	Näyttää argumentin merkkijonona.
input()	Pyytää käyttäjää antamaan syötteen.
eval()	Laskee merkkijonona esitetyn lausekkeen.
.format()	Menetelmä muotoilee annetun merkkijonon.

Matematiikka-valikko

Huomaa: Uutta tätä moduulia käyttävää ohjelmaa luotaessa on suositeltavaa käyttää **Matemaattiset laskutoimitukset** -ohjelmatyyppejä. Tällä varmistetaan kaikkien asiaankuuluvien moduulien tuonti.

objekti	Kuvaus
from math import *	Tuo kaikki menetelmät (funktiot) matematiikkamoduulista.
fabs()	Palauttaa reaaliluvun absoluuttisen arvon.
sqrt()	Antaa reaaliluvun neliöjuuren.
exp()	Antaa tulokseksi e^{**x} .
pow(x,y)	Antaa x:n korotettuna potenssiin y.
log(x,base)	Antaa arvon $\log_{\text{kantaluku}}(x)$. log(x) ilman kantalukua antaa luonnollisen logaritmin x.
fmod(x,y)	Palauttaa x:n ja y:n moduuliarvon. Käytä, jos x ja y ovat liukulukuja.
ceil()	Palauttaa pienimmän kokonaisluvun, joka on suurempi tai yhtä suuri kuin reaaliluku.
floor()	Palauttaa suurimman kokonaisluvun, joka on pienempi tai yhtä suuri kuin reaaliluku.

objekti	Kuvaus
trunc()	Typistää reaaliluvun kokonaisluvuksi.
frexp()	Antaa parin (y,n), jossa $x == y * 2^{**}n$.

Vakiot

objekti	Kuvaus
e	Returns value for the constant e.
pii	Returns value for the constant pi.

Trigonometriset funktiot

objekti	Kuvaus
radians()	Muuntaa asteina annetun kulman radiaaneiksi.
degrees()	Muuntaa radiaaneina annetun kulman asteiksi.
sin()	Antaa argumentin sinin radiaaneina.
cos()	Antaa argumentin kosinin radiaaneina.
tan()	Antaa argumentin tangentin radiaaneina.
asin()	Antaa argumentin arkussinin radiaaneina.
acos()	Antaa argumentin arkuskosinin radiaaneina.
atan()	Antaa argumentin arkustangentin radiaaneina.
atan2(y,x)	Antaa y/x:n arkustangentin radiaaneina.

Satunnaislukuvallikko

Huomaa: Uutta tätä moduulia käyttävää ohjelmaa luotaessa on suositeltavaa käyttää **Satunnaislukujen simuloinnit** -ohjelmatyyppeä. Tällä varmistetaan kaikkien asiaankuuluvien moduulien tuonti.

objekti	Kuvaus
from random import *	Tuo kaikki menetelmät satunnaislukumoduulista.
random()	Antaa liukulukuvun 0:sta 1,0:aan.
uniform(min,max)	Antaa satunnaisluvun x (liukuluku) niin, että $min \leq x \leq max$.

objekti	Kuvaus
randint(min,max)	Antaa satunnaisen kokonaisluvun vähimmäis- ja enimmäisarvon välillä.
choice(sequence)	Antaa satunnaisen elementin ei-tyhjistä sekvenssistä.
randrange(start,stop,step)	Antaa satunnaisluvun alusta loppuun vaihteittain.
seed()	Alustaa satunnaislukugeneraattorin.

TI PlotLib -valikko

Huomaa: Uutta tätä moduulia käyttävää ohjelmaa luotaessa on suositeltavaa käyttää **Piirtäminen (x,y) ja teksti** -ohjelmatyyppiä. Tällä varmistetaan kaikkien asiaankuuluvien moduulien tuonti.

objekti	Kuvaus
import ti_plotlib as plt	Tuo kaikki menetelmät (funktiot) ti_plotlib-moduulista plt-nimitilassa. Tämän seurauksena kaikkia valikoista liitettyjä funktioiden nimiä edeltää "plt."

Asetukset

objekti	Kuvaus
cls()	Tyhjentää piirtoalustan.
grid(x-scale,y-scale,"style")	Näyttää ruudukon käyttäen määritettyä asteikkoa x- ja y-akseleille.
window(xmin,xmax,ymin,ymax)	Määrittää piirtoikkunan yhdistämällä annetun vaakavälin (xmin, xmax) ja pystyvälin (ymin, ymax) määrättyyn piirtoalueeseen (pikselit).
auto_window(x-list,y-list)	Skaalaa piirtoikkunan automaattisesti niin, että se sopii ohjelmassa ennen auto_window()-muuttujaa määritettyihin x-list- ja y-list-dataväleihin.
axes("mode")	Näyttää annetun ikkunan akselit piirtoalueella.
labels("x-label","y-label",x,y)	Näyttää piirtoakselien "x-label"- ja "y-label"-merkinnät riviasemissa x ja y.
title("title")	Näyttää "otsikon" keskitettynä ikkunan ylimmälle riville.
show_plot()	Näyttää puskuroidun piirustuksen tuloksen. use_buffer()- ja show_plot()-funktiot ovat hyödyllisiä tapauksissa, joissa useiden objektien esittäminen näkymässä voisi aiheuttaa viiveitä (ei

objekti	Kuvaus
	tarpeen useimmissa tapauksissa).
use_buffer()	Mahdollistaa näkymän ulkopuolisen puskurin piirtämisen nopeuttamiseksi.

Piirrä

objekti	Kuvaus
color(red,green,blue)	Määrittää värin kaikille seuraaville grafiikoille/piirtämiselle.
cls()	Tyhjentää piirtoalustan.
show_plot()	Näyttää kuvaajan ohjelmassa määritetyllä tavalla.
scatter(x-list,y-list,"mark")	Piirtää järjestykseen asetetun parin sekvenssin kohteesta (x-list,y-list) annetulla merkityyllä.
plot(x-list,y-list,"mark")	Piirtää viivan käyttämällä järjestettyjä pareja annetuista x-luettelosta ja y-luettelosta.
plot(x,y,"mark")	Piirtää pisteen käyttämällä koordinaatteja x ja y annetulla merkityyllä.
line(x1,y1,x2,y2,"mode")	Piirtää viivasegmentin, jonka alku on (x1,y1) ja loppu (x2,y2).
lin_reg(x-list,y-list,"display")	Laskee ja piirtää x-list,y-list-kohteen lineaarisen regressiomallin $ax+b$.
pen("size","style")	Asettaa kaikkien seuraavien rivien ulkoasun, kunnes seuraava pen() suoritetaan.
text_at(row,"text","align")	Näyttää "tekstin" piirtoalueella määritetyllä "tasauksella".

Ominaisuudet

objekti	Kuvaus
xmin	Määritetty muuttuja ikkuna-argumenteille, joiden määrittäminen on plt.xmin.
xmax	Määritetty muuttuja ikkuna-argumenteille, joiden määrittäminen on plt.xmax.
ymin	Määritetty muuttuja ikkuna-argumenteille, joiden määrittäminen on plt.ymin.
ymax	Määritetty muuttuja ikkuna-argumenteille, joiden määrittäminen on plt.ymax.
m	Kun plt.linreg() suoritetaan ohjelmassa, lasketut kaltevuuden (m) ja

objekti	Kuvaus
	leikkauspisteen (b) arvot tallennetaan kohteisiin plt.m ja plt.b.
b	Kun plt.linreg() suoritetaan ohjelmassa, lasketut kaltevuuden (a) ja leikkauspisteen (b) arvot tallennetaan kohteisiin plt.a ja plt.b.

TI Hub -valikko

Huomaa: Uutta tätä moduulia käyttävää ohjelmaa luotaessa on suositeltavaa käyttää **Hub-projekti**-ohjelmatyyppejä. Tällä varmistetaan kaikkien asiaankuuluvien moduulien tuonti.

objekti	Kuvaus
from ti_hub import *	Tuo kaikki menetelmät ti_hub-moduulista.

Hubin sisäiset laitteet > Värituotos

objekti	Kuvaus
rgb(red,green,blue)	Määrittää RGB-merkkivalon värin.
blink(frequency,time)	Määrittää valitun värin vilkkumisen tiheyden ja keston.
off()	Sammuttaa RGB-merkkivalon.

Hubin sisäiset laitteet > Valotuotos

objekti	Kuvaus
on()	Sytyttää merkkivalon.
off()	Sammuttaa merkkivalon.
blink(frequency,time)	Määrittää merkkivalon vilkkumisen tiheyden ja keston.

Hubin sisäiset laitteet > Äänituotos

objekti	Kuvaus
tone(frequency,time)	Toistaa äänen määritetyllä taajuudella määritetyn ajan.
note("note",time)	Toistaa määritetyn nuotin määritetyn ajan. Nuotti määritetään käyttämällä nuotin nimeä ja

objekti	Kuvaus
	oktaavia. Esimerkiksi : A4, C5. Nuottien nimet ovat C, CS, D, DS, E, F, FS, G, GS, A, AS, and B. Oktaavinumerot vaihtelevat välillä 1–9 (ääriarvot mukaan lukien).
tone(frequency,time,tempo)	Toistaa äänen määritetyllä frekvenssillä määritetyn ajan ja tempon. Tempo määrittää piippausten määrän sekunnissa välillä 0–10 (mukaan lukien).
note("note",time,tempo)	Toistaa määritetyn nuotin määritetyn ajan ja tempon. Nuotti määritetään käyttämällä nuotin nimeä ja oktaavia. Esimerkiksi : A4, C5. Nuottien nimet ovat C, CS, D, DS, E, F, FS, G, GS, A, AS, and B. Oktaavinumerot vaihtelevat välillä 1–9 (ääriarvot mukaan lukien). Temponumerot vaihtelevat välillä 0–10 (ääriarvot mukaan lukien).

Hubin sisäiset laitteet > Kirkkauden syöttö

objekti	Kuvaus
measurement()	Lukee kiinteää KIRKKAUS-anturia (valon määrä) ja antaa lukeman. Oletusalue on 0–100. Tätä voidaan muuttaa range()-toiminnon avulla.
range(min,max)	Määrittää vaihteluvälin valon määrää mittaavan anturin lukemien kartoittamiselle. Jos molemmat puuttuvat tai arvoksi on määritetty "Ei mitään", oletuskirkkausalueena on 0–100.

Lisää syöttölaite

Tässä valikossa on luettelo ti_hub-moduulin tukemista antureista (syöttölaitteista). Kaikki valikkovaihtoehdot liittyvät objektin nimen ja odottavat muuttujaa sekä anturin kanssa käytettävää porttia. Jokaisella anturilla on measurement()-menetelmä, joka antaa anturin arvon.

objekti	Kuvaus
DHT (digitaalinen kosteus ja lämpötila)	Antaa senhetkisestä lämpötilasta, kosteudesta, anturin tyypistä ja viimeisestä välimuistissa olevasta lukemasta koostuvan luettelon.
Etäisyysmittari	Antaa senhetkisen etäisyyden annetusta ultraäänietäisyysmittarista. measurement_time() – Palauttaa ajan, joka ultraäänisignaali kestää tavoittaa esine ("lentoaika").
Valotaso	Antaa kirkkauden määrän ulkoisen valon määrän (kirkkauden) anturista.
Lämpötila	Antaa lämpötilalukeman ulkoisesta lämpötila-anturista. Oletusmäärityksenä on tukea Sseed-lämpötila-anturia IN 1-, IN 2- tai IN 3 -portissa. Jos haluat käyttää TI-Innovator™ -hubin koekytentälevypaketin TI LM19 -lämpötila-anturia, tulee porttia muokata käytössä olevaan BB-nastan ja käyttää valinnaista argumenttia "TIANALOG". Esimerkki: mylm19=temperature("BB 5","TIANALOG")
Kosteus	Antaa kosteusanturin lukeman.
Magneettinen	Havaitsee mahdollisen magneettikentän. Kentän havaitsemisen raja-arvo määritetään trigger()-funktioilla. Oletusraja-arvo on 150.
Vernier	Lukee komentoon määritetyn analogisen Vernier-anturin arvon. Komento tukee seuraavia Vernier-antureita: temperature – lämpötila-anturi ruostumattomasta teräksestä. lightlevel – TI-anturi valon määrälle. pressure – alkuperäinen paineanturi kaasulle pressure – uudempi paineanturi kaasulle. pH – pH-anturi.

objekti	Kuvaus
	force10 - ± 10 N -asetus, kaksoisvoimantunnistin. force50 - ± 50 N -asetus, kaksoisvoimantunnistin. accelerometer – pienen G-voiman kiihtyvyyssmittari. generic – mahdollista asentaa muita antureita, joita ei tueta suoraan yläpuolella, ja calibrate()-API:n käytön yhtälökertoimien asettamiseen.
Analoginen tulo	Tukee analogiseen syöttöön käytettäviä yleisiä laitteita.
Digitaalinen tulo	Palauttaa DIGITAL-objektiin liitetyn digitaalinaan sen hetkisen tilan tai objektiin viimeksi ASETETUN digitaalisen lähtöarvon välimuistissa olevan tilan.
Potentiometri	Tukee potentiometrianturia. Anturin aluetta voidaan muuttaa range()-funktioilla.
Termistori	Lukee termistoriantureita. Oletuskertoimet on suunniteltu vastaamaan TI-Innovator™ Hubin koeyhteyntälevypakettiin sisältyvää termistoria, kun sitä käytetään 10 K Ω :n kiinteän vastuksen kanssa. Termistorille voidaan määrittää uusi kalibroitakerroinjoukko ja vertailuvastus käyttämällä calibrate()-funktioita.
Äänenvoimakkuus	Tukee äänenvoimakkuusantureita.
Väritulo	Antaa rajapintoja I2C-liitettyyn värituloanturiin. bb_port-nastaa käytetään I2C-portin lisäksi värianturin LED-merkkivalon ohjaamiseen. color_number(): Antaa välillä 1–9 olevan, anturin havaitsemaa väriä edustavan arvon. Numerot edustavat seuraavan erittelyn mukaisia värejä: 1: Punainen 2: Vihreä

objekti	Kuvaus
	3: Sininen 4: Syaani 5: Magenta 6: Keltainen 7: Musta 8: Valkoinen 9: Harmaa • red(): Antaa välillä 0–255 olevan arvon, joka edustaa havaitun PUNAISEN värin voimakkuutta. • green(): Antaa välillä 0–255 olevan arvon, joka edustaa havaitun VIHREÄN värin voimakkuutta. • blue(): Antaa välillä 0–255 olevan arvon, joka edustaa havaitun SINISEN värin voimakkuutta. • gray(): Antaa välillä 0–255 olevan, havaittua harmaan määrää edustavan arvon, jossa 0 on musta ja 255 valkoinen.
BB-portti	Varmistaa tuen kaikkien kymmenen BB-porttinastan käyttöön yhdistettynä digitaalisena tulo-/lähtöporttina. Alustustoiminnoissa on valinnainen "mask"-parametri, joka mahdollistaa kymmenen nastan alijoukon käytön. • read_port(): Lukee BB-portin tulonastojen senhetkiset arvot. • write_port(value): Asettaa lähtönastan arvot määritettyyn arvoon, jossa arvo on välillä 0–1023. Huomaa, että arvoa säädetään myös var=bbport(mask)-toiminnon maskiarvon mukaan, jos maski on annettu.
Hub-aika	Mahdollistaa sisäisen millisekuntiajastimen käytön.
TI-RGB-matriisi	Sisältää funktioita TI-RGB-matriisin ohjelmointiin.

objekti	Kuvaus
	Alustustoiminto hyväksyy valinnaisen LAMP-parametrin, jolla voidaan ottaa käyttöön erittäin kirkas tila ulkoisen virtalähteen vaativalle TI-RGB-matriisille. • set(led_position, r,g,b): Asettaa tietyn led_position-aseman (0–15) määritettyyn r,g,b-arvoon, jossa r,g,b-arvot välillä 0–255. • set(led_list,red,green,blue): Asettaa "led_list"-listassa määritetyt LED-valot "punaiseksi", "vihreäksi", "siniseksi". "led_list" on Python-luettelo, joka sisältää LED-valojen indeksit 0–15. Esimerkiksi sarja([0,2,4,6,15], 0, 0, 255) asettaa LED-valot 0, 2, 4, 6 ja 15 siniseksi. • set_all(r,g,b): Asettaa matriisiin kaikki RGB-merkkivalot samaan r,g,b-arvoon. • all_off(): Sammuttaa matriisiin kaikki RGB:t. • measurement(): Antaa likimääräisen virrankulutuksen, jota RGB-matriisi käyttää TI-Innovator™-järjestelmästä milliampeereina. • pattern(pattern): Ottaa käyttöön pikselit, joissa 1-arvo esityksessä olisi käyttäen argumentin arvoa binaariarvona välillä 0–65535. Merkkivalot syttyvät punaisina, ja pwm-tason arvo on 255. • pattern(value,red,green,blue): Asettaa "pattern"-kuvion määrittämät LED-valot "punaiseksi", "vihreäksi", "siniseksi".

Lisää lähtölaite

Tässä valikossa on luettelo ti_hub-moduulin tukemista lähtölaitteista. Kaikki valikkovaihtoehdot liittävät objektin nimen ja odottavat muuttujaa ja laitteen kanssa käytettävää porttia.

objekti	Kuvaus
LED-valo	Funktiot ulkoisesti liitettyjen LED-valojen ohjaamiseen.

objekti	Kuvaus
RGB	Tuki ulkoisten RGB-merkkivalojen ohjaamiseen.
TI-RGB-matriisi	Sisältää funktioita TI-RGB-matriisin ohjelmointiin.
Kaiutin	Funktiot ulkoisten kaiuttimien tukemiseen TI-Innovator™ Hub:ia käytettäessä. Funktiot ovat samat kuin edellä olevassa ”ääni”-vaihtoehdossa.
Teho	Funktiot ulkoisen virran ohjaamiseksi TI-Innovator™ Hubilla. • set(value): Asettaa virran tason määritettyyn arvoon välillä 0–100. • on(): Asettaa virran tasoksi 100. • off(): Asettaa virran tasoksi 0.
Jatkuva servo	Funktiot jatkuvien servomoottorien ohjaamiseksi. • set_cw(speed,time): Servo pyörii myötöpäivään annetulla nopeudella (0–255) ja tietyn ajan sekunteina. • set_ccw(speed,time): Servo pyörii vastapäivään annetulla nopeudella (0–255) ja tietyn ajan sekunteina. • stop(): Pysäyttää jatkuvan servon.
Analoginen lähtö	Funktiot analogiseen syöttöön käytettäviä yleisiä laitteita varten.
Väriinämoottori	Funktiot väriinämoottoreiden ohjaamiseen. • set(val): Asettaa väriinämoottorin voimakkuuden arvoon val (0–255). • off(): Sammuttaa väriinämoottorin. • on(): Kytkee väriinämoottorin päälle suurimmalla teholla.
Rele	Säätää ohjausreleiden rajapintoja. • on(): Laittaa releen PÄÄLLE. • off(): Laittaa releen POIS PÄÄLTÄ.
Servo	Funktiot servomoottorien ohjaamiseen. • set_position(pos): Asettaa pyyhkäisy servoasennon välille -90 – +90. • zero(): Asettaa pyyhkäisy servon nolla-asentoon.
Neliöaalto	Funktiot neliöaallon luontiin. • set(frequency,duty,time): Määrittää lähtevän neliöaallon oletuskäyttökajaksolle 50 % (jos käyttöä ei ole määritetty) ja frequency-parametrin määrittämällä lähtötaajuudella. Taajuus voi olla välillä 1–500 Hz. Jos käyttökajako on annettu, se voi olla välillä 0–100 %. • off(): Laittaa neliöaallon pois päältä.
Digitaalinen lähtö	Rajapinnat digitaalisen lähdon ohjaamiseen.

objekti	Kuvaus
	• set(val): Asettaa digitaalisen lähdön val-parametrin määrittämään arvoon (0 tai 1). • on(): Asettaa digitaalisen lähdön tilaksi korkea (1). • off(): Asettaa digitaalisen lähdön tilaksi matala (0).
BB-portti	Sisältää funktioita TI-RGB-matriisin ohjelmointiin. Ks. tarkemmat tiedot edellä.

Komennot

objekti	Kuvaus
sleep(seconds)	Keskeyttää ohjelman annetun sekuntimäärän ajaksi. Tuotu time-moduulista.
text_at(row,"text","align")	Näyttää määritetyn "tekstin" piirtoalueella määritetyllä "tasauksella". Osa ti_plotlib-moduulia.
cls()	Tyhjentää komentotulkinäytön piirtämistä varten. Osa ti_plotlib-moduulia.
while get_key() != "esc":	Suorittaa komennot while-silmukassa, kunnes esc-näppäintä painetaan.
get_key()	Antaa painettua näppäintä edustavan merkkijonon. Näppäimellä 1 saadaan 1, näppäimellä esc saadaan esc ja niin edelleen. Kun kutsutaan ilman parametreja – get_key() – se antaa tuloksen välittömästi. Kun kutsutaan parametrilla – get_key(1) – se odottaa näppäimen painamista. Osa ti_system-moduulia.

Portit

Nämä ovat TI-Innovator™ Hubissa käytettävissä olevat tulo- ja lähtöportit.

objekti
LÄHTÖ 1
LÄHTÖ 2

objekti
LÄHTÖ 3
TULO 1
TULO 2
TULO 3
BB 1
BB 2
BB 3
BB 4
BB 5
BB 6
BB 7
BB 8
BB 9
BB 10
I2C

TI Rover -valikko

Huomaa: Uutta tätä moduulia käyttävää ohjelmaa luotaessa on suositeltavaa käyttää **Rover-koodaus** -ohjelmatyyppiä. Tällä varmistetaan kaikkien asiaankuuluvien moduulien tuonti.

objekti	Kuvaus
import ti_rover as rv	Tuo kaikki menetelmät (funktiot) ti_rover-moduulista "rv"-nimitilassa. Tämän seurauksena kaikkia valikoista liitettyjä funktioiden nimiä edeltää "rv.".

Ajo

objekti	Kuvaus
forward(distance)	Siirtää Roveria eteenpäin määrätyn etäisyyden verran ruudun yksiköissä.
backward(distance)	Siirtää Roveria taaksepäin määrätyn etäisyyden verran ruudun yksiköissä.
left(angle_degrees)	Kääntää Roveria vasemmalle määrättyssä kulmassa

objekti	Kuvaus
	asteina.
right(angle_degrees)	Kääntää Roveria oikealle määrättyssä kulmassa asteina.
stop()	Pysäyttää senhetkisen liikkeen välittömästi.
stop_clear()	Pysäyttää senhetkisen liikkeen välittömästi ja tyhjentää kaikki odottavat komennot.
resume()	Jatkaa komentojen käsittelyä.
stay(time)	Rover pysyy paikallaan annetun ajan verran sekunteina (valinnainen). Jos aikaa ei ole määritetty, Rover pysyy paikallaan 30 sekunnin ajan.
to_xy(x,y)	Siirtää Roverin koordinaattisijaintiin (x,y) virtuaaliruudukossa.
to_polar(r,theta_degrees)	Siirtää Roverin napakoordinaattisijaintiin (r, theta) virtuaaliruudukossa. Kulma annetaan asteina.
to_angle(angle,"unit")	Pyörittää Roveria annettuun kulmaan virtuaaliruudukossa. Kulma on suhteessa nollakulmaan, joka osoittaa virtuaaliruudukon x-akselilla alaspäin.

Ajo > Ajo vaihtoehtoilla

objekti	Kuvaus
forward_time(time)	Siirtää Roveria eteenpäin määritetyn ajan verran.
backward_time(time)	Siirtää Roveria taaksepäin määritetyn ajan verran.
forward(distance,"unit")	Siirtää Roveria eteenpäin oletusnopeudella määritetyn etäisyyden verran. Etäisyys voidaan määrittää ruudukon yksiköissä, metreissä tai pyörän kierroksissa.
backward(distance,"unit")	Siirtää Roveria taaksepäin oletusnopeudella määritetyn etäisyyden verran. Etäisyys voidaan määrittää ruudukon yksiköissä, metreissä tai pyörän kierroksissa.

objekti	Kuvaus
left(angle,"unit")	Kääntää Roveria vasemmalle määrättyssä kulmassa. Kulma voi olla asteina, radiaaneina tai gooneina.
right(angle,"unit")	Kääntää Roveria oikealle määrättyssä kulmassa. Kulma voi olla asteina, radiaaneina tai gooneina.
forward_time(time,speed,"rate")	Siirtää Roveria eteenpäin määrätyn ajan verran määrätyllä nopeudella. Nopeus voidaan määrittää ruudukon yksiköissä, metreissä tai pyörän kierroksissa.
backward_time(time,speed,"rate")	Siirtää Roveria taaksepäin määrätyn ajan verran määrätyllä nopeudella. Nopeus voidaan määrittää ruudukon yksiköissä, metreissä tai pyörän kierroksissa.
forward(distance,"unit",speed,"rate")	Siirtää Roveria eteenpäin määrätyn etäisyyden verran määrätyllä nopeudella. Etäisyys voidaan määrittää ruudukon yksiköissä, metreissä tai pyörän kierroksissa. Nopeus voidaan määrittää ruudukon yksiköissä, metreissä tai pyörän kierroksissa.
backward(distance,"unit",speed,"rate")	Siirtää Roveria taaksepäin määrätyn etäisyyden verran määrätyllä nopeudella. Etäisyys voidaan määrittää ruudukon yksiköissä, metreissä tai pyörän kierroksissa. Nopeus voidaan määrittää ruudukon yksiköissä, metreissä tai pyörän kierroksissa.

Syötteet

objekti	Kuvaus
ranger_measurement()	Lukee Roverin etupuolella olevan ultraäänietäisyysanturin ja antaa senhetkisen etäisyyden metreinä.

objekti	Kuvaus
color_measurement()	Antaa välillä 1–9 olevan arvon, joka ilmaisee Rover-värituloanturin ”näkemän” hallitsevan värin. 1 = punainen 2 = vihreä 3 = sininen 4 = syaani 5 = magenta 6 = keltainen 7 = musta 8 = harmaa 9 = valkoinen
red_measurement()	Antaa välillä 0–255 olevan arvon, joka ilmaisee värituloanturin havaitseman punaisen värin määrän.
green_measurement()	Antaa välillä 0–255 olevan arvon, joka ilmaisee värituloanturin havaitseman vihreän värin määrän.
blue_measurement()	Antaa välillä 0–255 olevan arvon, joka ilmaisee värituloanturin havaitseman sinisen värin määrän.
gray_measurement()	Antaa välillä 0–255 olevan arvon, joka ilmaisee värituloanturin havaitseman harmaan värin määrän.
encoders_gyro_measurement()	Antaa luettelon arvoista, jotka sisältävät vasemman ja oikean pyörän enkooderin lukeman sekä senhetkisen gyroskoopin suunnan.
gyro_measurement()	Antaa arvon, joka edustaa asteissa senhetkistä gyro-lukemaa, mukaan lukien siirtymä.
ranger_time()	Palauttaa ajan, joka TI-Rover-etäisyysmittarin ultraäänisignaaliilta kestää tavoittaa kohde (”lentoaika”).

Tulokset

objekti	Kuvaus
color_rgb(r,g,b)	Määrittää Roverin RGB-merkkivalon värin tiettyihin punaisen, vihreän ja sinisen arvoihin.
color_blink(frequency,time)	Määrittää valitun värin vilkkumisen tiheyden ja keston.

objekti	Kuvaus
color_off()	Sammuttaa Roverin RGB-merkkivalon.
motor_left(speed,time)	Asettaa vasemman moottorin tehon määritettyyn arvoon määrätyn ajaksi. Nopeus on välillä -255 – 255 ja 0 tarkoittaa pysähtymistä. Positiiviset nopeusarvot ovat vastapäivään pyörimistä ja negatiiviset nopeusarvot myötäpäivään pyörimistä. Valinnaisen aikaparametrin – mikäli sellainen määritetään – kelvollinen vaihteluväli on 0,05–655,35 sekuntia. Jos sitä ei määritetä, käytetään oletusarvoista 5 sekunnin aikaa.
motor_right(speed,time)	Asettaa vasemman moottorin tehon määritettyyn arvoon määrätyn ajaksi. Nopeus on välillä -255 – 255 ja 0 tarkoittaa pysähtymistä. Positiiviset nopeusarvot ovat vastapäivään pyörimistä ja negatiiviset nopeusarvot myötäpäivään pyörimistä. Valinnaisen aikaparametrin – mikäli sellainen määritetään – kelvollinen vaihteluväli on 0,05–655,35 sekuntia. Jos sitä ei määritetä, käytetään oletusarvoista 5 sekunnin aikaa.
motors("ldir",left_val,"rdir",right_val,time)	Asettaa vasemman ja oikean pyörän määrättyihin nopeuksiin valinnaiseksi ajaksi sekunteina. Nopeuden (left_val, right_val) arvot ovat välillä 0–255, ja 0 tarkoittaa pysähtymistä. Parametrit ldir ja rdir määrittävät vastaavien pyörien myötäpäivään tai vastapäivään suuntautuvan pyörimissuunnan. Valinnaisen aikaparametrin – mikäli sellainen määritetään – kelvollinen vaihteluväli on 0,05–655,35 sekuntia. Jos sitä ei määritetä, käytetään oletusarvoista 5 sekunnin aikaa.

objekti	Kuvaus
waypoint_xythdrn()	Lukee x-koordinaatin, y-koordinaatin, ajan, suunnan, kuljetun etäisyyden, pyörien kierrosten määrän, senhetkisen reittipisteen komentonumeron. Antaa luettelon kaikista näistä arvoista elementteinä.
waypoint_prev	Lukee x-koordinaatin, y-koordinaatin, ajan, suunnan, kuljetun etäisyyden, pyörien kierrosten määrän, edellisen reittipisteen komentonumeron.
waypoint_eta	Antaa arvioidun ajoajan reittipisteeseen.
path_done()	Antaa arvon 0 tai 1 sen mukaan, liikkuuko (0) Rover vai onko se lopettanut kaiken liikkumisen (1).
pathlist_x()	Antaa luettelon X-arvoista alusta senhetkisen reittipisteen X-arvoon ja sen mukaan lukien.
pathlist_y()	Antaa luettelon Y-arvoista alusta senhetkisen reittipisteen Y-arvoon ja sen mukaan lukien.
pathlist_time()	Antaa luettelon ajasta sekunteina alusta senhetkisen reittipisteen aika-arvoon ja sen mukaan lukien.
pathlist_heading()	Antaa luettelon suunnista alusta senhetkisen reittipisteen suunta-arvoon ja sen mukaan lukien.
pathlist_distance()	Antaa luettelon kuljetuista etäisyyksistä alusta senhetkisen reittipisteen etäisyysarvoon ja sen mukaan lukien.
pathlist_revs()	Antaa luettelon kuljetuista kierroksista alusta senhetkisen reittipisteen kierrosarvoon ja sen mukaan lukien.
pathlist_cmdnum()	Antaa polun komentonumeroiden luettelon.
waypoint_x()	Antaa senhetkisen reittipisteen x-koordinaatin.
waypoint_y()	Antaa senhetkisen reittipisteen y-koordinaatin.
waypoint_time()	Antaa edellisestä reittipisteestä nykyiseen reittipisteeseen kulkemiseen käytetyn ajan.
waypoint_heading()	Antaa senhetkisen reittipisteen absoluuttisen suunnan.
waypoint_distance()	Antaa edellisen ja senhetkisen reittipisteen välillä kuljetun etäisyyden.
waypoint_revs()	Antaa edellisen ja senhetkisen reittipisteen välillä kulkemiseen tarvittavien kierrosten määrän.

Asetukset

objekti	Kuvaus
units/s	Asetus nopeudelle ruudukon yksikköä sekunnissa.
m/s	Asetus nopeudelle metreinä sekunnissa.
revs/s	Asetus nopeudelle pyörien kierroksia sekunnissa.
Yksiköt	Asetus etäisyydelle ruudukon yksiköissä.
m	Asetus etäisyydelle metreissä.
revs	Asetus etäisyydelle pyörien kierroksissa.
asteet	Asetus kääntymiselle asteissa.
radiaania	Asetus kääntymiselle radiaaneina.
gradians	Asetus kääntymiselle gooneina.
clockwise	Asetus pyörän suunnan määrittämiseen.
counter-clockwise	Asetus pyörän suunnan määrittämiseen.

Komennot

Nämä komennot ovat kokoelma muista moduuleista sekä TI Rover -moduulista peräisin olevia funktioita.

objekti	Kuvaus
sleep(seconds)	Keskeyttää ohjelman annetun sekuntimäärän ajaksi. Tuotu time-moduulista.
text_at(row,"text","align")	Näyttää ”tekstin” piirtoalueella määritetyllä ”tasauksella”. Tuotu ti_plotlib-moduulista.
cls()	Tyhjentää komentotulkkinäytön piirtämistä varten. Tuotu ti_plotlib-moduulista.
while get_key() != "esc":	Suorittaa komennot while-silmukassa, kunnes esc-näppäintä painetaan.
wait_until_done()	Keskeyttää ohjelman, kunnes Rover päättää senhetkisen komennon. Tämä on kätevä tapa synkronoida muita kuin Rover-komentoja Roverin liikkeen kanssa.
while not path_done()	Suorittaa komennot while-silmukassa, kunnes Rover on lopettanut kaiken liikkumisen.

objekti	Kuvaus
	Funktio <code>path_done()</code> antaa arvon 0 tai 1 sen mukaan, liikkeuko (0) Rover vai onko se lopettanut kaiken liikkumisen (1).
<code>position(x,y)</code>	Sijoittaa Roverin virtuaaliruudukossa annettuun x,y-koordinaattiin.
<code>position(x,y,heading,"unit")</code>	Sijoittaa Roverin virtuaaliruudukossa annettuun x,y-koordinaattiin, ja virtuaaliseen x-akseliin suhteessa oleva virtuaalisuunta määritetään, jos suunta on annettu (valituissa kulmayksiköissä). Positiivisten kulmien 0–360 oletetaan olevan vastapäivään positiivisesta x-akselista. Negatiivisten kulmien 0 – -360 oletetaan olevan myötäpäivään positiivisesta x-akselista.
<code>grid_origin()</code>	Asettaa RV:n sijaitsemaan senhetkisen ruudukon alkuperäispisteessä (0,0).
<code>grid_m_unit(scale_value)</code>	Asettaa virtuaalisen ruudukon välit metreinä yksikköä kohden (m/yksikkö) annettuun arvoon. Oletusarvoinen m/yksikkö on 0,1 ja tarkoittaa, että 1 yksikkö = 100 mm tai 10 cm tai 1 dm tai 0,1 m. Kelvollinen <code>scale_value</code> -väli on 0,01–10,0.
<code>path_clear()</code>	Poistaa kaikki aiemmin luodut polku- tai reittipistetiedot.
<code>zero_gyro()</code>	Nollaa Rover-gyroarvon 0,0 asteen kulmaan ja tyhjentää vasemman ja oikean pyörän enkooderilaskurit.

Monimutkaisen matematiikan valikko

Tämä alivalikko sijaitsee kohdassa **Lisää moduuleja**.

objekti	Kuvaus
<code>from cmath import *</code>	Tuo kaikki menetelmät <code>cmath</code> -moduulista.
<code>complex(real,imag)</code>	Antaa vastauksen kompleksilukuna.
<code>rect(modulus,argument)</code>	Muuntaa napakoordinaatit kompleksiluvun suorakaiteeksi.
<code>.real</code>	Antaa kompleksiluvun reaaliosan.
<code>.imag</code>	Antaa kompleksiluvun imaginaariosan.
<code>polar()</code>	Muuntaa suorakaiteen kompleksiluvun napakoordinaateiksi.

objekti	Kuvaus
phase()	Antaa kompleksiluvun vaiheen.
exp()	Antaa tulokseksi e^{**x} .
cos()	Antaa tulokseksi kompleksiluvun kosinin.
sin()	Antaa tulokseksi kompleksiluvun sinin.
log()	Antaa kompleksiluvun luonnollisen logaritmin.
log10()	Antaa kompleksiluvun kymmenkantaisen logaritmin.
sqrt()	Antaa kompleksiluvun neliöjuuren.

Aikavalikko

Tämä alivalikko sijaitsee kohdassa **Lisää moduuleja**.

objekti	Kuvaus
from time import *	Tuo kaikki menetelmät aikamoduulista.
sleep(seconds)	Keskeyttää ohjelman annetun sekuntimäärän ajaksi.
clock()	Palauttaa senhetkisen suoritinajan sekunteina ilmaistuna liukulukuna.
localtime()	Muuntaa sekunneissa ilmaistun ajan päivämäärästä 1. tammikuuta 2000 lähtien yhdeksänkertaiseksi sisältäen vuoden, kuukauden, päivän, tunnin, minuutin, sekunnin, viikonpäivän, vuoden päivän ja kesäajan merkinnän. Jos valinnaista (sekunnit) argumenttia ei anneta, käytetään reaaliaikaista kelloa.
ticks_cpu()	Antaa suoritinkohtaisen kasvavan millisekuntilaskurin sattumanvaraisella viitepisteellä. Jos aikaa halutaan mitata johdonmukaisesti eri järjestelmissä, käytä kohdetta ticks_ms().
ticks_diff()	Mittaa jakson peräkkäisten ticks_cpu()- tai ticks_ms()-kutsujen välillä. Tätä toimintoa ei tule käyttää pitkien ajanjaksojen sattumanvaraiseen mittaamiseen.

TI-järjestelmävalikko

Tämä alivalikko sijaitsee kohdassa **Lisää moduuleja**.

Huomaa: Uutta tätä moduulia käyttävää ohjelmaa luotaessa on suositeltavaa käyttää **Tiedonjako**-ohjelmatyyppejä. Tällä varmistetaan kaikkien asiaankuuluvien moduulien tuonti.

objekti	Kuvaus
<code>from ti_system import *</code>	Tuo kaikki menetelmät (funktiot) <code>ti_system-</code> moduulista.
<code>recall_value("name")</code>	Antaa ennalta määritetyn OS-muuttujan (value), jonka nimi on name.
<code>store_value("name",value)</code>	Tallentaa Python-muuttujan (value) OS-muuttujaan nimeltä name.
<code>recall_list("name")</code>	Palauttaa ennalta määritetyn OS-luettelon nimeltä name.
<code>store_list("name",list)</code>	Tallentaa Python-luettelon (list) OS-luettelomuuttujaan nimeltä name.
<code>eval_function("name",value)</code>	Arvioi ennalta määritetyn OS-funktion annetulla arvolla.
<code>get_platform()</code>	Antaa arvon "hh" kämmenlaitteelle ja arvon "dt" pöytätietokoneelle.
<code>get_key()</code>	Antaa painettua näppäintä edustavan merkkijonon. Näppäimellä 1 saadaan 1, näppäimellä esc saadaan esc ja niin edelleen. Kun kutsutaan ilman parametreja – <code>get_key()</code> – se antaa tuloksen välittömästi. Kun kutsutaan parametrilla – <code>get_key(1)</code> – se odottaa näppäimen painamista.
<code>get_mouse()</code>	Antaa hiiren koordinaatit kaksiosaisena monikkona, joko piirtoalueen pikselisijaintina tai (-1,-1), jos se on alueen ulkopuolella.
<code>while get_key() != "esc":</code>	Suorittaa komennot <code>while</code> -silmukassa, kunnes <code>esc</code> -näppäintä painetaan.
<code>clear_history()</code>	Tyhjentää komentotulkin aikaisemmat tapahtumat.
<code>get_time_ms()</code>	Antaa ajan millisekuntein millisekunnin tarkkuudella. Tätä funktiota voidaan käyttää keston laskemiseen ennemmin kuin todellisen kelloajan määrittämiseen.

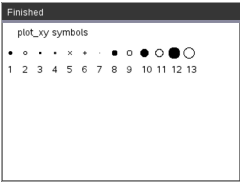
Ti-kuvaajavalikko

Tämä alivalikko sijaitsee kohdassa **Lisää moduuleja**.

Huomaa: Uutta tätä moduulia käyttävää ohjelmaa luotaessa on suositeltavaa käyttää **Geometriakuvaajat**-ohjelmatyyppejä. Tällä varmistetaan kaikkien asiaankuuluvien moduulien tuonti.

objekti	Kuvaus
from ti_draw import *	Tuo kaikki menetelmät ti_draw-moduulista.

Kuvio

objekti	Kuvaus
draw_line()	Piirtää viivan alkaen määritetystä x1,y1-koordinaatista x2,y2-koordinaattiin.
draw_rect()	Piirtää suorakulmion aloittaen annetusta x,y-koordinaatista annetuilla leveydellä ja korkeudella.
fill_rect()	Piirtää suorakulmion aloittaen annetusta x,y-koordinaatista annetuilla leveydellä ja korkeudella ja täytettynä annetulla värillä (asetus on set_color tai musta, jos ei määritetty).
draw_circle()	Piirtää ympyrän aloittaen annetusta x,y-keskikoordinaatista annetulla säteellä.
fill_circle()	Piirtää ympyrän aloittaen annetusta keskikoordinaatista annetulla säteellä ja täytettynä annetulla värillä (asetus on set_color tai musta, jos ei määritetty).
draw_text()	Piirtää merkkijonon aloittaen annetusta x,y-koordinaatista.
draw_arc()	Piirtää kaaren aloittaen annetusta x,y-koordinaatista annetuilla leveydellä, korkeudella ja kulmilla.
fill_arc()	Piirtää kaaren aloittaen annetusta x,y-koordinaatista annetuilla leveydellä, korkeudella ja kulmilla ja täytettynä annetulla värillä (asetus on set_color tai musta, jos ei määritetty).
draw_poly()	Piirtää monikulmion käyttäen annettuja x-list-,y-list-arvoja.
fill_poly()	Piirtää monikulmion käyttäen annettuja x-list,y-list-arvoja ja täytettynä määritetyllä värillä (asetus on set_color tai musta, jos ei määritetty).
plot_xy()	Piirtää muodon käyttäen annettua x,y-koordinaattia ja annettua numeroa väliltä 1–13 edustaen eri muotoja ja symboleja (ks. alempana). 

Control (Kontrolli)

objekti	Kuvaus
<code>clear()</code>	Tyhjentää koko näkymän. Voidaan käyttää jo luodun suorakulmion poistamiseen <code>x,y,width,height</code> -parametreilla.
<code>clear_rect()</code>	Piirtää suorakulmion annettuun <code>x,y</code> -koordinaattiin annetuilla leveydellä ja korkeudella.
<code>set_color()</code>	Määrittää ohjelmassa seuraavien muotojen värin, kunnes joku toinen väri määritetään.
<code>set_pen()</code>	Määrittää reunaviivan annetun paksuuden ja tyylin muotoja piirrettäessä (ei sovellettavissa täyttökomentoja käytettäessä).
<code>set_window()</code>	Määrittää sen ikkunan koon, johon muodot piirretään. Tämä toiminto on hyödyllinen ikkunan koon muuttamiseksi tietoja vastaavaksi tai piirtoalustan alkuperän (0,0) muuttamiseksi.
<code>get_screen_dim()</code>	Antaa näkymän mittojen <code>xmax</code> - ja <code>ymax</code> -arvot.
<code>use_buffer()</code>	Mahdollistaa näkymän ulkopuolisen puskurin piirtämisen nopeuttamiseksi.
<code>paint_buffer()</code>	Näyttää puskuroidun piirustuksen tuloksen. <code>use_buffer()</code> ja <code>paint_buffer()</code> -toiminnot ovat hyödyllisiä tapauksissa, joissa useiden objektien esittäminen näkymässä voisi aiheuttaa viiveitä.

Muistiinpanot

- Oletusmäärittäyksessä on (0,0) näkymän vasemmassa yläkulmassa. Positiivinen `x`-akseli osoittaa oikealle ja positiivinen `y`-akseli osoittaa alareunaa kohti. Tätä voidaan muokata `set_window()`-toiminnolla.
- Toiminnot `ti_draw`-moduulissa ovat käytettävissä vain kämmenlaitteella ja pöytäkoneen kämmenlaitenäkymässä.

Ti Image -valikko

Tämä alivalikko sijaitsee kohdassa **Lisää moduuleja**.

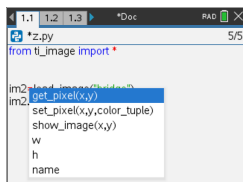
Huomaa: Uutta tätä moduulia käyttävää ohjelmaa luotaessa on suositeltavaa käyttää **Image-käsittely**-ohjelmatyyppejä. Tällä varmistetaan kaikkien asiaankuuluvien moduulien tuonti.

objekti	Kuvaus
<code>from ti_image import *</code>	Tuo kaikki menetelmät <code>ti_image</code> -moduulista.

objekti	Kuvaus
<code>new_image(width,height,(r,g,b))</code>	Luo uuden kuvan Python-ohjelmassa käytettäväksi tarkoitetulla tietyllä leveydellä ja korkeudella. Uuden kuvan väri määräytyy (r,g,b)-arvojen mukaan.
<code>load_image("name")</code>	Lataa name-parametrin määrittämän kuvan Python-ohjelmassa käytettäväksi. Kuvan on oltava osa TNS-asiakirjaa joko Muistiinpanot- tai Kuvaajat-sovelluksessa. Kehotteessa name näkyvät kuvien nimet (jos ne on nimetty aiemmin) tai niiden lisäysjärjestyksen ilmaiseva numero.
<code>copy_image(image)</code>	Luo kopion image-muuttujan määrittämästä kuvasta.

Image-objektin menetelmät

Image-objekteihin liittyviä lisätoimintoja saa käyttöön editorissa ja komentotulkissa kirjoittamalla muuttujan nimen ja sen jälkeen merkin `'.'` (piste).



- **get_pixel(x,y):** Hakee pikselin (r,g,b)-arvon (x,y)-koordinaattiparin määrittämässä sijainnissa.

```
px_val = get_pixel(100,100)
print(px_val)
```
- **set_pixel(x,y,color_tuple):** Asettaa pikselin sijainnissa (x,y) color_tuple-muuttujassa määritetyn väriseksi.

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

Asettaa pikselin (100,100)-muuttujassa väriin (0,0,255).

- **show_image(x,y):** Näyttää kuvan siten, että vasen yläkulma on kohdassa (x,y).
- **w, h, name:** Hakee kuvan leveys-, korkeus- ja nimiparametrit.

Esimerkki

```
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)
```

```
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)
```

Muuttujat-valikko

Huomaa: Nämä luettelot eivät sisällä muissa TI-Nspire™-sovelluksissa määritettyjä muuttujia.

objekti	Kuvaus
Vars: Current Program	(Vain editori) Näyttää luettelon senhetkisessä ohjelmassa määritetyistä yleisistä funktioista ja muuttujista
Vars: Last Run Program	(Vain komentotulkki) Näyttää luettelon viimeksi suoritettussa ohjelmassa määritetyistä yleisistä funktioista ja muuttujista
Vars: Kaikki	(Vain komentotulkki) Näyttää luettelon sekä viimeksi suoritettun ohjelman että mahdollisten tuotujen moduulien yleisistä funktioista ja muuttujista

Python-avainsanat 44

Python-näppäintapahtumat 44

Python-malliohjelmat 46

Python-avainsanat

Seuraavat avainsanat sisältyvät TI-Nspire™ Python -toteutukseen.

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-näppäintapahtumat

Näppäimistö on suunniteltu liittämään asiaankuuluvia Python-toimintoja tai avaamaan valikkoja koodia syötettäessä editoriin tai komentotulkkiin. Tämän ansiosta toimintoihin, avainsanoihin, menetelmiin, operaattoreihin jne. pääsee siirtymään helposti.

Näppäin	Tapahtumat
	Avaa Muuttujat-valikon
	Liittää merkin =
	Poistaa kohdistimen vasemmalla puolella olevan merkin
	Ei toimintoa
	Liittää merkin =
	Liittää valitun/valitut symbolin/symbolit: > <

Näppäin	Tapahtumat
	• != • >= • <= • == • ja • tai • Ei • • & • ~
	Liittää valitun toiminnon: • sin • cos • tan • atan2 • asin • acos • atan
	Näyttää vihjeet
	Liittää :=
	Liittää **
	Ei toimintoa
	Liittää **2
	Liittää sqrt()
	Liittää kertolaskumerkin (*)
	Liittää yhden lainausmerkin (")
	Liittää jakomerkin (/)
	Ei toimintoa
	Liittää exp()
	Liittää log()
	Liittää 10**
	Liittää log(value,base)
	Liittää (

Näppäin	Tapahtumat
	Liittää)
	Liittää []
	Liittää { }
	Liittää vähennysmerkin (-)
	Lisää uuden rivin senhetkisen rivin jälkeen
	Liittää E
	Liittää valitun/valitut symbolin/symbolit: • ? • ! • \$ • ° • ' • % • " • : • ; • _ • \ • #
	Liittää "pi"
	Jo luodun merkinnän toiminta
	Lisää uuden rivin senhetkisen rivin jälkeen

Python-malliohjelmat

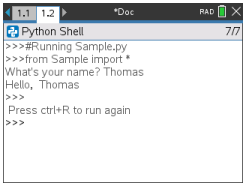
Seuraavia malliohjelmia voidaan käyttää Python-menetelmiin tutustumiseen. Ne ovat saatavilla myös **Esimerkkejä**-kansiossa löytyvässä **Getting Started Python.tns** -tiedostossa.

Huomaa: Jos kopioit ja liität TI-Nspire™-ohjelmistoon sarkainsisennysindikaattoreita (••) sisältävän mallikoodin, nämä ilmentymät tulee korvata todellisilla sarkainsisennyksillä.

Hei

```
# 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")
```

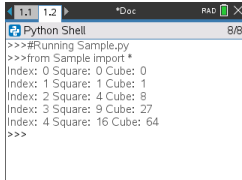


```
>>>#Running Sample.py
>>>from Sample import *
What's your name? Thomas
Hello, Thomas
>>>
Press ctrl+R to run again
>>>
```

Silmukkaesimerkki

```
# 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)
```



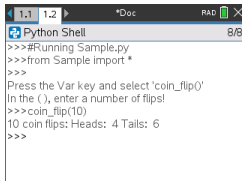
Kruuna vai klaava

```
# 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!")
```



Piirtäminen

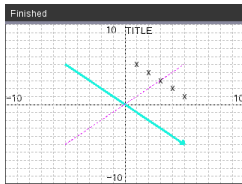
```
# Plotting example
import ti_plotlib as plt

# 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")
```



Kuvaaja

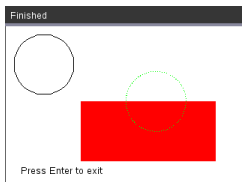
```
from ti_draw import *

# (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)

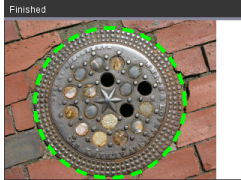
set_color(0,0,0)
draw_text(20,200,"Press Enter to exit")
```



Kuva

```
# 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)
```



Hubi

Tämä ohjelma käyttää Pythonia ohjaamaan TI-Innovator™ Hubia, ohjelmoitavaa mikrokontrolleria. Ohjelman suorittaminen kiinnittämättä TI-Innovator™ Hubia tuo näkyviin virheilmoituksen.

Lisätietoja TI-Innovator™ Hubista löytyy osoitteesta 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()
```


Yleistä

Online-tuki

education.ti.com/eguide

Valitse maasi, niin näet lisää tuotetietoja.

Ota yhteyttä TI-tukeen

education.ti.com/ti-cares

Valitse maasi, niin näet teknisiä tietoja ja muita tukiresursseja.

Huolto- ja takuutiedot

education.ti.com/warranty

Valitse maasi, niin saat tietoa takuun kestosta ja ehdoista tai tuotepalvelusta.

Rajoitettu takuu. Tämä takuu ei vaikuta lainmukaisiin oikeuksiisi.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243