

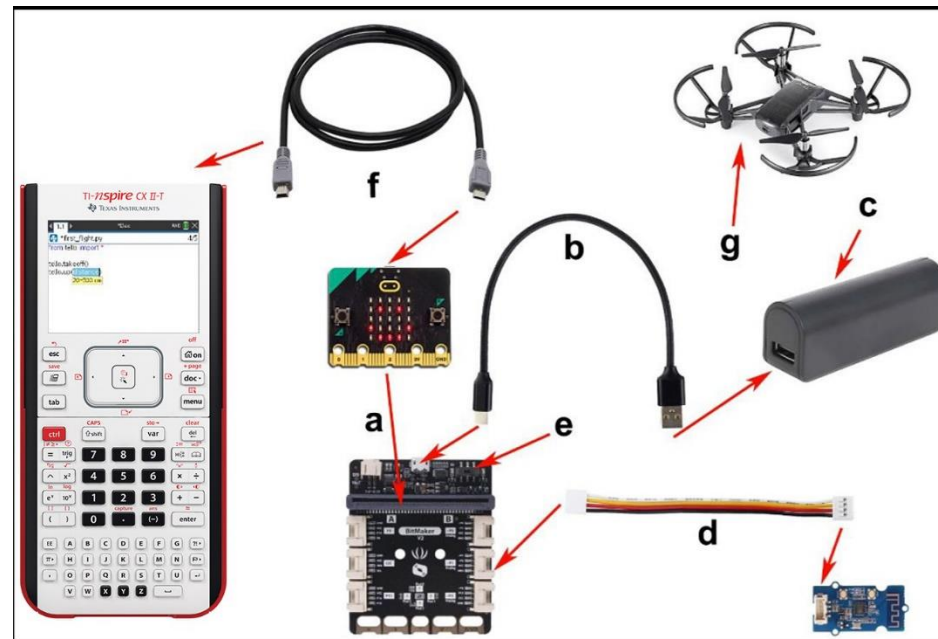
Achtergrond:

Bij deze activiteiten leren leerlingen de basis van programmeren in Python en het vliegen met de Tello drone. Daarbij maken ze gebruik van de [TI-Nspire CX II-T met de BBC micro:bit](#). Alle drie de uitdagingen vragen om een andere vaardigheid. Als de leerlingen die beheersen draagt dit bij aan het veilig vliegen met de drone in de klas. Als leerlingen de activiteiten afronden kunnen ze een certificaat krijgen

Let op – Het buiten vliegen met drones is op veel plekken toegestaan. Bekijk op de website van de Rijksoverheid waar dat niet is toegestaan. De Tello drone is niet waterdicht en de fabrikant waarschuwt om niet buiten te vliegen als het waait.

Ben je er klaar voor?

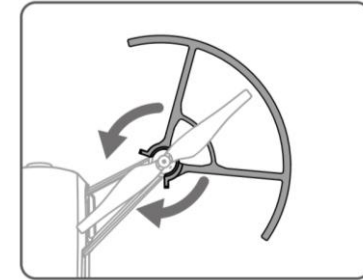
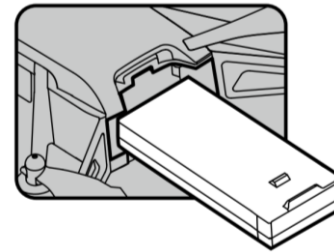
Voordat je kunt starten met de activiteit, moeten de Tello drone en het micro:bit communicatiesysteem in elkaar worden gezet en worden gekoppeld. Bekijk hieronder wat je ervoor nodig hebt.





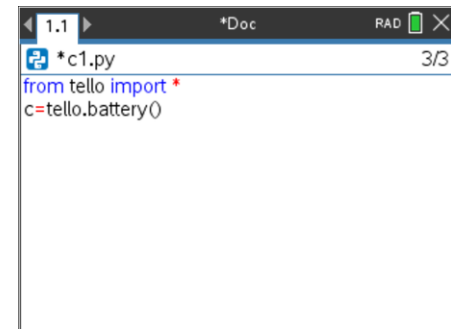
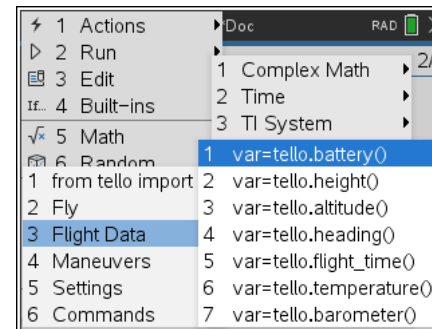
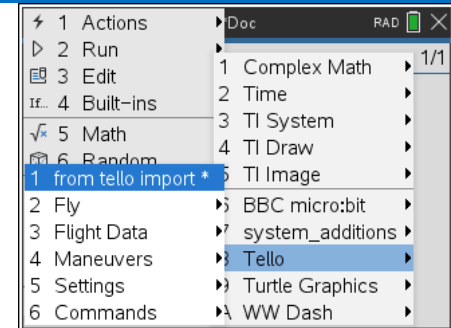
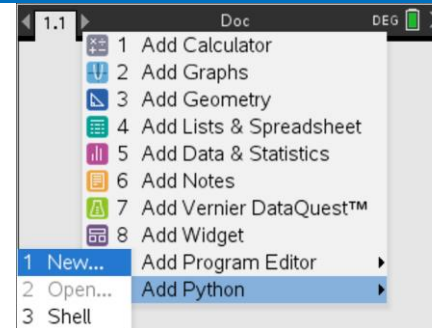
Controle vóór de vlucht: alles doorlopen

1. Plaats een volledig opgeladen batterij stevig in de drone. Zorg ervoor dat je de batterij in de juiste richting in de drone schuift. **Forceer de batterij nooit!**
2. Controleer of er geen beschadigingen zijn aan de propellers en druk elke propeller voorzichtig maar stevig op de as van de motor.
3. Controleer of de propellers vrij ronddraaien en de veiligheidsbeugels niet raken.
4. Controleer het casco op scheuren of andere schade.



Opdracht 1: Clear—Contact!

1. Druk op de **[on]** toets en start een **New document**.
2. Alle Python methoden en modules kun je vinden door op de **[menu]** toets te drukken.
3. Begin met een nieuw Python programma: **Add Python → New** en noem het **c1**
4. Importeer de tello module, selecteer: **More Modules → Tello → from tello import ***.
5. Selecteer **More Modules → Tello → Flight Data → var = tello.battery()** en noem de variabele **c** (voor charge).



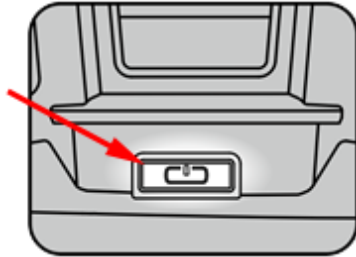
Let op – druk op [ctrl] + [enter] om te zorgen dat de cursor naar de volgende regel gaat na het invoeren van een parameter. Dit voorkomt dat de gesloten haakjes in de volgende regel vallen.

TI-Nspire CX II-T Vleugels krijgen!

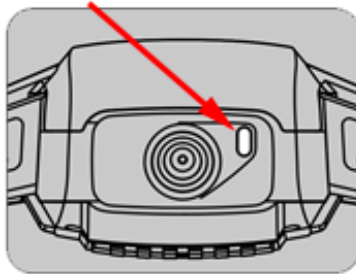


Tello Drone

6. Selecteer: **More Modules** → **Tello** → **Settings** → **.motors_on()**
7. Selecteer: **More Modules** → **Tello** → **Commands** → **sleep(5000)**; de tijd is in milliseconden. 1000 milliseconden = 1 seconde.
8. Selecteer: **More Modules** → **Tello** → **Settings** → **.motors_off()**
9. Plaats de drone op een vlakke ondergrond, en zorg dat er geen mensen in de buurt zijn.
10. Zet de drone aan met de knop die op de achterzijde zit.



11. Wacht tot het LED-lampje aan de voorkant van de drone constant geel knippert.



12. Selecteer: **Run** → **Run (Ctrl+R)**
13. De rekenmachine toont de batterijlading van de drone; de motoren draaien 5 seconden en gaan dan uit.
14. Let op de feedback van de drone op het scherm van de TI-Nspire CX II-T.
15. Schakel de drone uit door nog een keer op de knop op de achterzijde te drukken.

Gefeliciteerd! Je hebt je eerste Python drone programma geschreven!

Let op: Voor de methoden `motors_on()` en `motors_off()` is een Tello EDU met de laatste firmware-upgrade nodig.

```

1 Actions
2 Run
3 Edit
4 Built-ins
5 Math
6 Random
1 from tello import *
2 Fly
3 Flight Data
4 Maneuvers
5 Settings
6 Commands
1 Complex Math
2 Time
3 TI System
4 TI Draw
5 TI Image
6 BBC micro:bit
1 .set_speed(speed)
2 .set_feedback(visibility)
3 .motors_on()
4 .motors_off()
  
```

```

1.1 *Doc RAD 4/5
*c1.py
from tello import *
c=tello.battery()
tello.motors_on()
  
```

```

1 Actions
2 Run
3 Edit
4 Built-ins
5 Math
6 Random
1 from tello import *
2 Fly
3 Flight Data
4 Maneuvers
5 Settings
6 Commands
1 Complex Math
2 Time
3 TI System
4 TI Draw
5 TI Image
6 BBC micro:bit
1 sleep(milliseconds)
2 Fly 2 while get_key() != "esc":
3 Flig3 clear_history()
4 Ma 4 store_list("name",list)
5 Set 5 Tello module version 3.2.3 BETA
6 Cor 6 Requires TI_runtime.hex 3.1.3 BETA
  
```

```

1.1 *Doc RAD 5/6
*c1.py
from tello import *
c=tello.battery()
tello.motors_on()
sleep(5000)
  
```

```

1 Actions
2 Run
3 Edit
4 Built-ins
5 Math
6 Random
1 from tello import *
2 Fly
3 Flight Data
4 Maneuvers
5 Settings
6 Commands
1 Complex Math
2 Time
3 TI System
4 TI Draw
5 TI Image
6 BBC micro:bit
1 .set_speed(speed)
2 .set_feedback(visibility)
3 .motors_on()
4 .motors_off()
  
```

```

1.1 *Doc RAD 6/8
*c1.py
from tello import *
c=tello.battery()
tello.motors_on()
sleep(5000)
tello.motors_off()
  
```



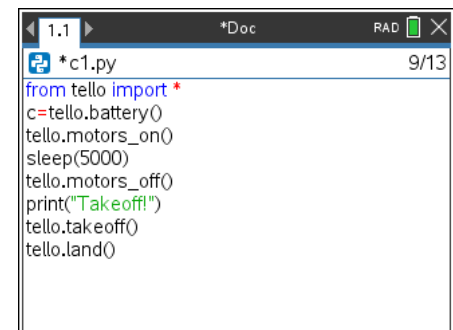
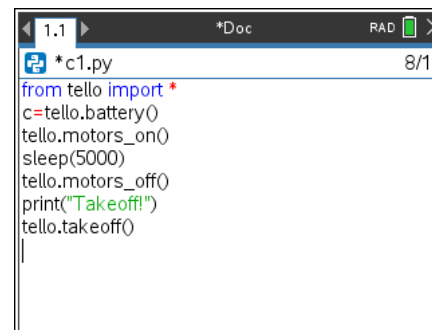
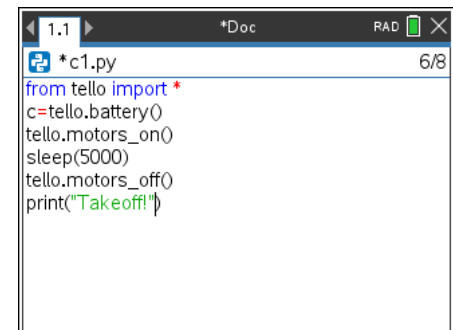
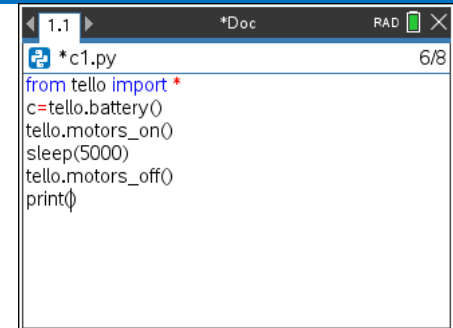
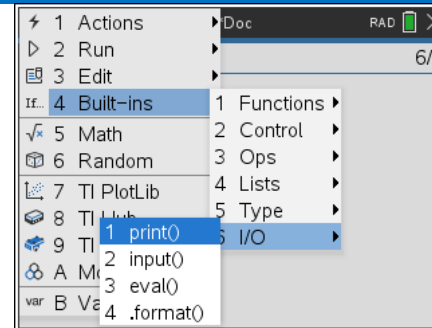
Tello Drone

Opdracht 2: Drie, Twee, Een... Takeoff!

1. Voeg een print-opdracht toe met de tekst "Takeoff!".
2. Selecteer: **Built-Ins** → **I/O** → **print()**
3. Typ binnen de haakjes de tekst Takeoff! tussen aanhalingstekens.
Deze kun je vinden met **ctrl + keerteken**
4. Selecteer: **More Modules** → **Tello** → **Fly** → **.takeoff()**
5. Selecteer: **More Modules** → **Tello** → **Fly** → **.land()**
6. Plaats de drone op een vlakke ondergrond, en zorg dat er geen mensen in de buurt zijn.
7. Zet de drone aan met de knop die op de achterzijde zit.
8. Wacht tot het LED-lampje aan de voorkant van de drone constant knippert.

Let op: Controleer of er geen mensen of spullen op de vliegroute van de drone zijn voordat je het programma uitvoert.

9. Selecteer: **Run** → **Run (Ctrl+R)**
10. Schakel de drone uit door nog een keer op de knop op de achterzijde te drukken.





Tello Drone

Opdracht 3: Ik snap het... laten we gaan vliegen

Gebruik je kennis van de Python menu's op zelf deze vluchten te proberen:

1. Vlieg uit en terug.
 - a. Stijg op
 - b. Vlieg 100cm vooruit
 - c. Draai om (180°)
 - d. Vlieg terug naar het startpunt
 - e. land.

2. Vlieg omhoog en maak een paar rondjes.
 - a. Stijg op
 - b. Vlieg 100cm omhoog (check de hoogte van het plafond)
 - c. Maak een salto (flip), vooruit dan achteruit, of links dan rechts.
 - d. Laat de drone landen.

*Let op: Na het invoeren van een parameter binnen de haakjes, moet de cursor buiten de rechter haakjes verplaatst worden voor je op Enter drukt. Nog makkelijker is het om **[ctrl]+[enter]** in te drukken binnen de haakjes om te voorkomen dat de regel wordt gesplitst*

```

1.1 | *Doc | RAD | 11/16
-----
*c1.py
from tello import *
c=tello.battery()
tello.motors_on()
sleep(5000)
tello.motors_off()
print("Takeoff!")
tello.takeoff()
tello.forward(100)
tello.turn_right(180)
tello.flip_forward(100)
tello.land()
  
```

Challenge 3.1

```

1.1 | *Doc | RAD | 11/16
-----
*c1.py
from tello import *
c=tello.battery()
tello.motors_on()
sleep(5000)
tello.motors_off()
print("Takeoff!")
tello.takeoff()
tello.up(100)
tello.flip_forward()
tello.flip_backward()
tello.land()
  
```

Challenge 3.2