



Tello Drone

Activity Background and Learning Outcomes:

Students will master Python programming basics and fly the Tello drone using the [TI-Nspire CX II with the BBC micro:bit](#). Each challenge presents a new flight skill and promotes safety while flying the drone in the classroom. The objective of each challenge is to fly a square using different Tello methods and Python programming techniques.

Learning outcomes of this activity include:

Python:

- editing a program
- running a program in shell
- types of variables
- defining and recalling variables
- basic input and output
- definite loops

Tello:

- tello.takeoff()
- tello.land()
- tello.forward()
- tello.left()
- tello.right()
- tello.backward()
- tello.turn_right()
- tello.turn_left()
- tello.goto()



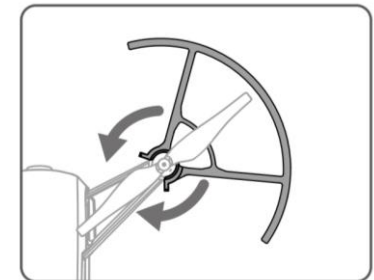
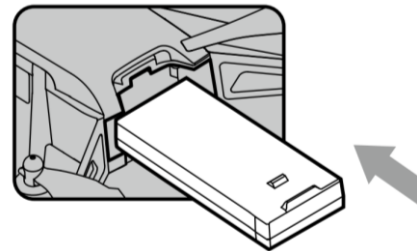
Scan QR code or click [here](#) to watch the associated videos for this activity

Are you flight-rated for this activity?

Before proceeding with the activity, students should have completed the "Getting Your Wings" activity, and the Tello - micro:bit communication system is assembled and paired. If not, please complete the ["Getting Your Wings"](#) activity to prepare for this activity.

Preflight Check: The Walk Around

1. Securely insert a fully charged battery into the drone. Be sure to slide the battery in with the correct orientation. **Never force the battery!**
2. Check that the propellers do not have nicks, and gently but firmly press each propeller onto the motor's shaft.
3. Confirm the propellers rotate freely and do not hit the safety guards.
4. Inspect the airframe for cracks or other damage.
5. Ensure the flight airspace is clear of people and objects.



Reference Guide

Python Syntax

takeoff()	tello.takeoff()
land()	tello.land()
forward(distance)	tello.forward(100)
backward(distance)	tello.backward(100)
fly_right(distance)	tello.fly_right(100)
fly_left(distance)	tello.fly_left(100)
turn_right(angle)	tello.turn_right(90)
turn_left(angle)	tello.turn_left(90)
enable_mission_pad(number)	tello.enable_mission_pad(6)
 goto(x,y,z)	 tello.goto(50,75,100)
 print("string", variable)	 print("Side =",n)
 var = input("prompt")	 length = input("side length")
 var = float(var)	 length = float(length)
 for index in range(size):	 for side in range(4):
♦♦ block	♦♦tello.forward(100)
	♦♦tello.right(90)

Notes

Takeoff must be the first flight command in the program.

Land must be the last flight command in the program.

Fly forward at a distance from 10cm to 500cm.

Fly backward a distance from 10cm to 500cm.

Fly right, with no turn, from 10cm to 500cm.

Fly left, with no turn, from 10cm to 500cm.

Rotate right at an angle from 0° to 360°.

Rotate left at an angle from 0° to 360°.

(EDU only) Enable the visual positioning system to recognize the numbered mission pad under the drone at takeoff. The rocket printed on the pad points in the positive X-axis direction. This command must be issued after takeoff().

(EDU only) Fly to the x, y, and z coordinates over the enabled mission pad.

Coordinates must be within -500cm to 500cm. Note that x,y, and z coordinates can't be set simultaneously between -20cm and 20cm.

The print() function will display a string and/or variable in the calculator's Python shell.

The input() function requires the prompt displayed in the shell and returns a string to the assigned variable.

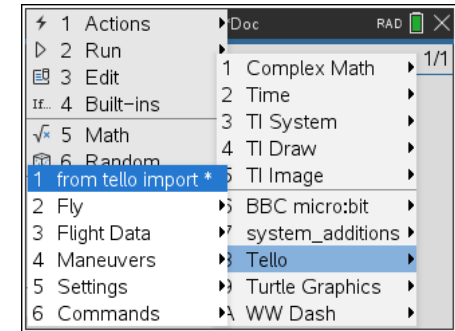
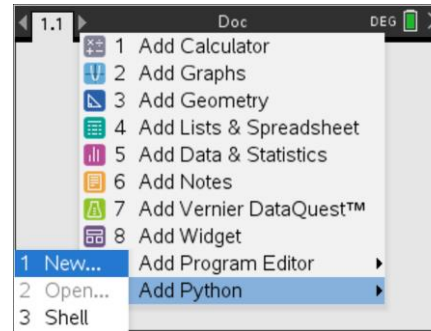
If the string returned from the input() function must be a number, for example, for a Tello flight method, then the string must be converted to a number type. The float() function returns the argument variable as a floating point number type.

The for control structure repeats a block of code the number of times set in the size of the range() function. Notice the code block is indented by two spaces ♦♦ in the TI Nspire Python editor. This indentation is essential to group all of the commands that are controlled with the for statement. If a command comes after the for statement and is not indented, it will not be repeated in the control structure.

Challenge 1: Square one

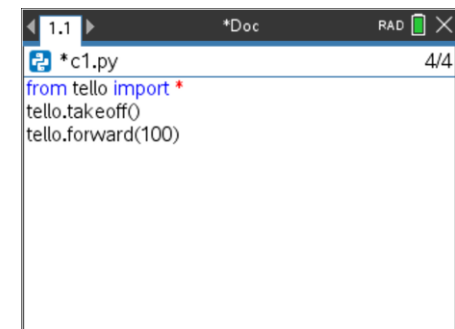
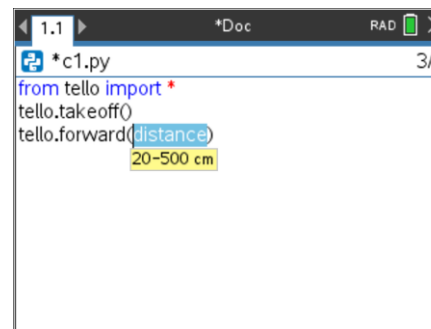
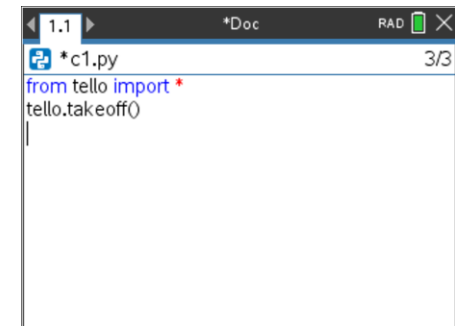
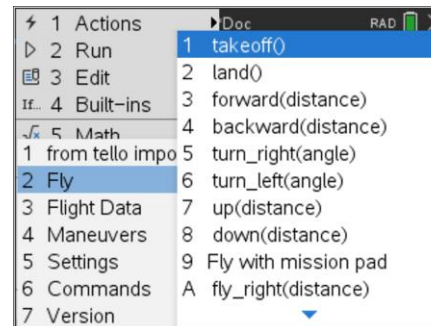
The challenge: Write a program named 'c1' to fly a square using only forward() and turn_right() or turn_left()

1. Select **[on]** key and start a **New document**.
2. All Python methods and modules are accessed by pressing the **[menu]** key.
3. Add a new Python editor, select: **Add Python → New → Name it "c1"**.
4. Import the tello module, select: **More Modules → Tello → from tello import ***.
5. Select **More Modules → Tello → Fly → takeoff()**



Note – to ensure your cursor goes to the following line after entering a parameter, press [ctrl] + [enter]. This prevents the closed parenthesis from being dropped into the next line.

6. Select: **More Modules → Tello → Fly → .forward(100)**



TI-Nspire CX II Four Square



Tello Drone

7. Select: **More Modules** → **Tello** → **Fly** → **turn_right(90)**

```
1.1 *Doc RAD 4/4
from tello import *
tello.takeoff()
tello.forward(100)
tello.turn_right(angle)
degrees
```

```
1.1 *Doc RAD 5/5
from tello import *
tello.takeoff()
tello.forward(100)
tello.turn_right(90)
```

8. Repeat steps 6 and 7 to complete flying a square returning to the original position.

9. When finished, land the drone - **More Modules** → **Tello** → **Fly** → **land()**.

```
1.1 *Doc RAD 5/5
from tello import *
tello.takeoff()
tello.forward(100)
tello.turn_right(90)
tello.forward(distance)
20-500 cm
```

```
1.1 *Doc RAD 6/6
from tello import *
tello.takeoff()
tello.forward(100)
tello.turn_right(90)
tello.forward(100)
tello.turn_right(angle)
degrees
```

10. Set your drone on the floor on a Tello mission pad, clear from people and obstacles. Think about the program and where the drone will fly before launch.

11. Press the power button on the drone's side to turn it on.

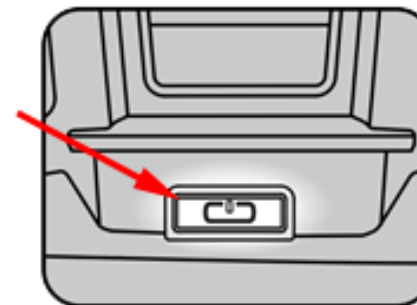
12. Wait until the status LED on the front of the drone blinks steadily.

Caution: Check that the flight path above and around the drone is clear of people and obstacles before running your program.

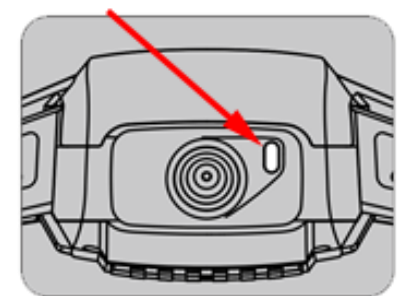
13. Select [menu] → **Run** → **Run (Ctrl+R)** to run the program.

14. After the drone has landed, turn off the drone by pressing the power button again.

15. Congratulations! You have written the first fly-a-square program in Python!



Power Button



Status LED

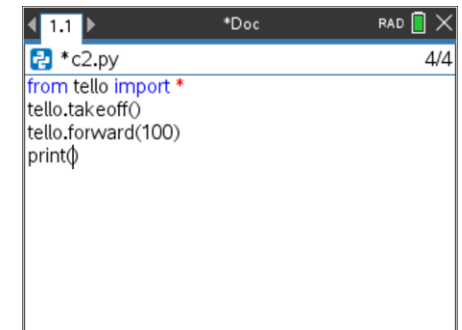
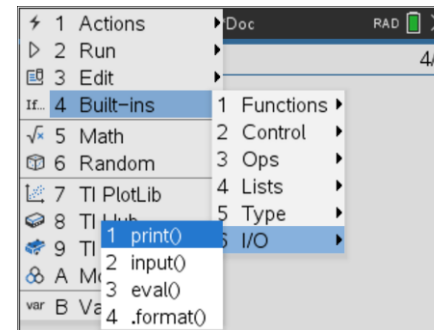
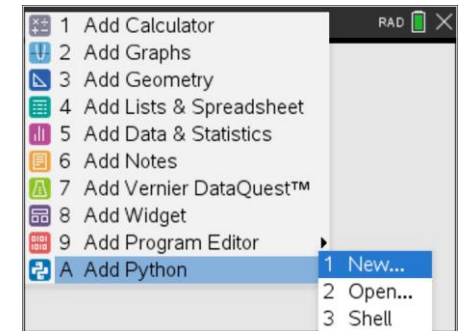
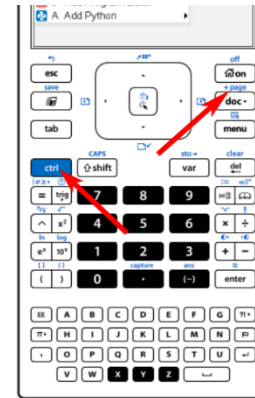


Tello Drone

Challenge 2: Square two

The challenge: Write a program named "c2" to fly a square with an edge length of 100 cm using only forward(), backward(), fly_right(), and fly_left() methods. Add a print statement after each side is flown with an appropriate message indicating which side of the square has been completed.

1. Add a new page to the document [ctrl] and +page ([doc]).
2. Add a new Python editor, select: **Add Python** → **New** → **Name it "c2"**.
3. Import the tello module, select: **More Modules** → **Tello** → **from tello import ***.
4. Select **More Modules** → **Tello** → **Fly** → **.takeoff()**
5. Select **More Modules** → **Tello** → **Fly** → **.forward(100)**
6. Select **Built-ins** → **I/O** → **print()**



TI-Nspire CX II Four Square



Tello Drone

7. Select the [ctrl] and then the "□" key to enter a single quote.
8. Next, use the alpha keypad to enter "side 1 has completed".
9. End the line with one more quote (the [ctrl] and then the "□").
10. Select [ctrl] + [enter] to go to the following line and avoid splitting the parentheses.
11. Complete the challenge by selecting the necessary flight methods and printing the corresponding side number. Don't forget to land!
12. Prepare the drone for takeoff as done in challenge 1.

Caution: Check that the flight path above and around the drone is clear of people and obstacles before running your program.

13. Select: **Run → Run (Ctrl+R)**
14. After landing, turn off the drone power by pressing the power button.



```
1.1 *Doc RAD 5/8
from tello import *
tello.takeoff()
tello.forward(100)
print("side 1 has completed")
```

Challenge 3: Square three

The Challenge: Write a program named 'c3' that uses a *for index in range(size):* control loop with fly forward, turn, and a print statement displaying the side in the code block to fly a square with an edge length of 100 cm. When the square is completed, and the drone has landed, print a message indicating the drone has landed.

1. Add a new page to the document [ctrl] and +page ([doc]).
2. Add a new Python editor, select: **Add Python → New → Name it "c3".**
3. Import the tello module, select: **More Modules → Tello → from tello import *.**
4. Select **More Modules → Tello → Fly → .takeoff()**
5. Select **Built-ins → Control → for index in range(size):**

```
1.1 *Doc RAD 3/6
from tello import *
tello.takeoff()
```

```
1 Actions Doc RAD 3/7
2 Run
3 if..
4 if..else..
5 if..elif..else..
6 for index in range(size):
7 for index in range(start, stop):
8 for index in range(start, stop, step):
9 for index in list:
10 while..
11 elif:
12 else:
```

TI-Nspire CX II Four Square



Tello Drone

6. The index is a named variable that contains what step of the loop is running—type in the name "side" to describe the index variable used in this example.
7. Press the [tab] key to advance the cursor to the range() function argument.
8. The size parameter for the range() function is the number of times the loop will run. Since we are flying a square with four sides, the range is four.
9. Press the [tab] key to advance the cursor to the following prompt in the method.
10. The cursor is now indented; notice the two gray diamonds. Any code written in the indented block will be repeated in the loop. To fly a square, fly forward 100cm and then turn 90 degrees. Since these two methods will be repeated four times, the loop will fly a square.

```
1.1 *Doc RAD 3/11
from tello import *
tello.takeoff()
for index in range(size):
    **block
```

```
1.1 *Doc RAD 3/11
from tello import *
tello.takeoff()
for side in range(size):
    **block
```

```
1.1 *Doc RAD 4/11
from tello import *
tello.takeoff()
for side in range(4):
    **block
```

```
1.1 *Doc RAD 6/13
from tello import *
tello.takeoff()
for side in range(4):
    **tello.forward(100)
    **tello.turn_right(90)
    **|
```

Note: The *for index in range(size):* structure makes flying a square much easier than in Challenge 1 since the instructions for flying a side are repeated four times! Loops are a powerful coding tool.

11. Add a print statement to the block of code.
12. Type in "side " with a space after the "e"; be sure to close the quote and then enter a comma outside of the second quote to add a second parameter to the print() statement. Notice the quote is green, and the comma is black. This color coding helps avoid syntax errors.

```
1.1 *Doc RAD 6/13
from tello import *
tello.takeoff()
for side in range(4):
    **tello.forward(100)
    **tello.turn_right(90)
    **print()
```

```
1.1 *Doc RAD 6/13
from tello import *
tello.takeoff()
for side in range(4):
    **tello.forward(100)
    **tello.turn_right(90)
    **print("side",)
```




Tello Drone

13. The index variable named "side" keeps track of which step the loop is running. This variable can be printed to indicate the side of the square being flown. However, In Python, the first step of a loop is zero. Since numbering the first side of a square as "side 0" is awkward, a value of 1 should be added to the index variable. This expression, side+1, will add one to the index and display one when the index is zero.
 14. After the expression, type a comma to add a third parameter.
 15. Use quotes to add the string "completed" to the print statement. Add the space before the "c" to format the display correctly.
 16. Select [ctrl] + [enter] to go to the following line and avoid splitting the parentheses.
 17. Notice the cursor remains indented with two diamonds, indicating this code line is within the loop. Since the drone should not land after the first step of the loop, the cursor needs to be dedented to the left to exit the loop before landing.
 18. Select [shift] + [tab] to dedent the cursor to the left and exit the loop. Alternatively, the [del] key can delete the gray diamonds.
 19. When all four steps of the loop have been completed, the drone should land. Be sure the tello.land() method is not indented and is part of the loop.
 20. Lastly, print a message reporting the drone has landed.
- Caution: Check that the flight path above and around the drone is clear of people and obstacles before running your program.**
21. Select Run → Run (Ctrl+R).
 22. After landing, turn off the drone power by pressing the power button.

```
1.1 *Doc RAD 6/13
from tello import *
tello.takeoff()
for side in range(4):
    tello.forward(100)
    tello.turn_right(90)
    print("side",side+1,)
```

```
1.1 *Doc RAD 7/14
from tello import *
tello.takeoff()
for side in range(4):
    tello.forward(100)
    tello.turn_right(90)
    print("side",side+1," completed")
|
```

```
1.1 *Doc RAD 7/15
from tello import *
tello.takeoff()
for side in range(4):
    tello.forward(100)
    tello.turn_right(90)
    print("side",side+1," completed")
|
```

```
1.1 *Doc RAD 6/13
from tello import *
tello.takeoff()
for side in range(4):
    tello.forward(100)
    tello.turn_right(90)
    print("side",side+1," completed")
```

```
1 Action
2 Run
3 Edit
4 Built-in
5 Math
6 Rand
7 TI Plots
8 TI Histogram
9 TI Regression
A More
B Variables
```

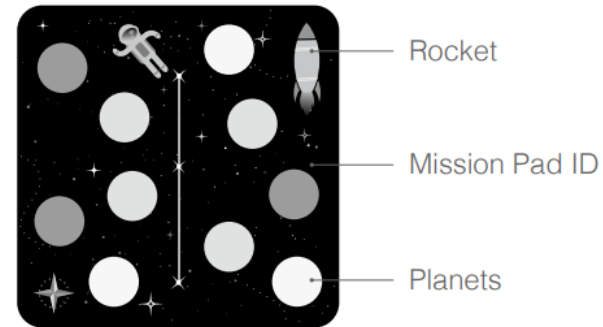
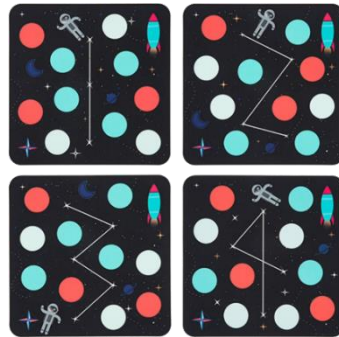
```
1.1 *Doc RAD 9/18
from tello import *
tello.takeoff()
for side in range(4):
    tello.forward(100)
    tello.turn_right(90)
    print("side",side+1," completed")
tello.land()
print("Houston, the drone has landed")
|
```

Challenge 4: Square four

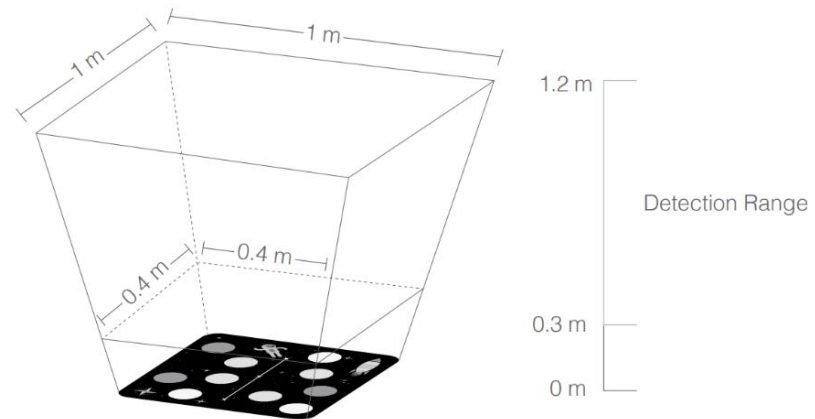
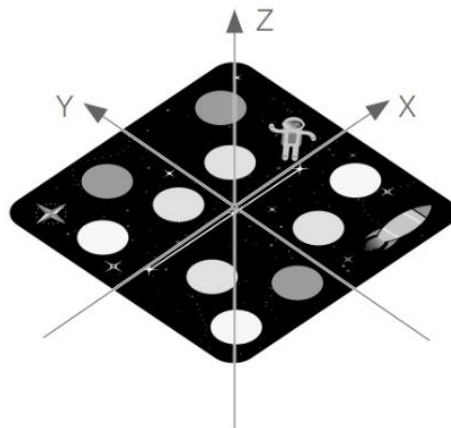
The Challenge: Write a program named 'c4' that uses a mission pad to fly a square, with the size set by user input, within Tello's 3-D grid system. Use the tello.goto(x,y,z) method and var = input("prompt") function to complete the challenge.

Background: There are four Tello mission pads; each has an ID, rocket, and planets printed on one side of the pad that is recognized by the drone's vision system. The pads 1, 2, 3, and 4 are shown below. The pads establish a 3-dimensional coordinate system the drone can fly within.

TI-Nspire CX II Four Square



The rocket in the corner of the pad points in the positive X-axis direction and defines the 3-D coordinate system for the drone. The center of the pad is the origin (0,0,0) of the system. The pad has a zero height on the Z-axis; however, the drone will not fly lower than $z = 30\text{cm}$ to prevent crashes into the floor. Limiting the goto(x,y,z) values within this recommended volume, see diagram below, is best.

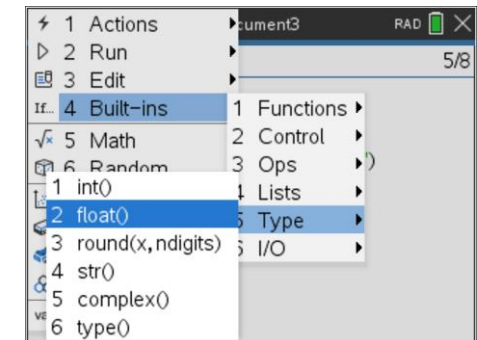
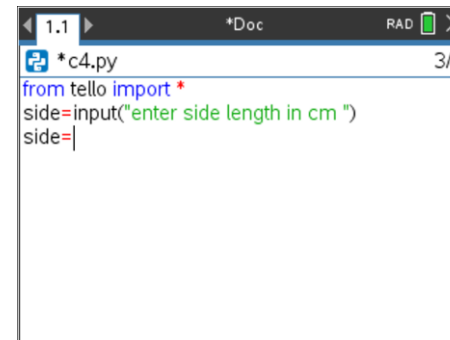
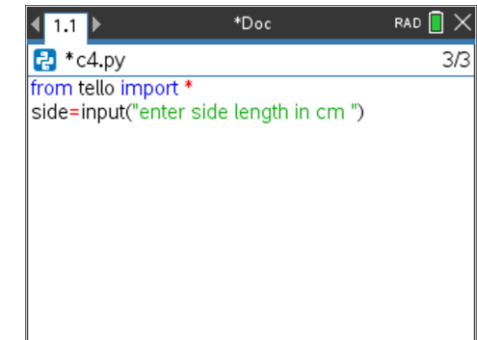
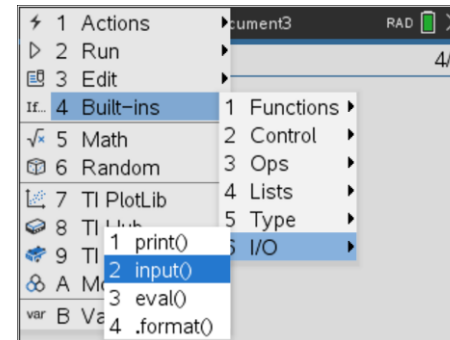
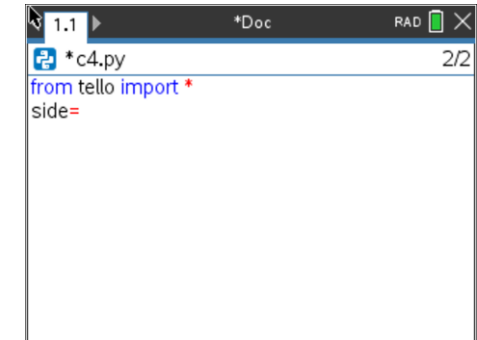


TI-Nspire CX II Four Square



Tello Drone

1. Add a new page to the document [ctrl] and +page ([doc]).
2. Add a new Python editor, select: **Add Python → New → Name it "c4"**.
3. Import the tello module, select: **More Modules → Tello → from tello import ***.
4. The input function prompts the user to enter something, in this program, the length of the square's edge in cm. The function returns the user's keyboard entry as a string. The returned string must be stored in a variable for later use. In the example code, the variable is named "side". Use the keypad to type in the variable name, "side", and then type in the equals, "=", symbol.
5. Select **Built-ins → I/O → input()**
6. The input function requires a prompt string as an argument; this string will be displayed on the calculator and will tell the user to enter. Within the parentheses, select the [ctrl] and then the "[]" key to enter a single quote. Use the keypad to type in "enter side length in cm" with a space after the "m" be sure to close the quotation.
7. The input() function returns a string into the variable named "side"; forcing the "side" variable to be a string type. String type variables may contain any keyboard character, including numbers. However, string types differ from number types. It must be a number type to do any mathematical operations on a variable. There are two number types; integers such as 1,2, or 3, and floating point such as 4.5, .0023, or 10.0. The Tello fly method arguments require a floating point type, and a string type must be converted to a float type. The float () function can redefine the "side " variable.
8. Use the keypad to type "side =".
9. Select **Built-ins → Type → float()**



TI-Nspire CX II Four Square



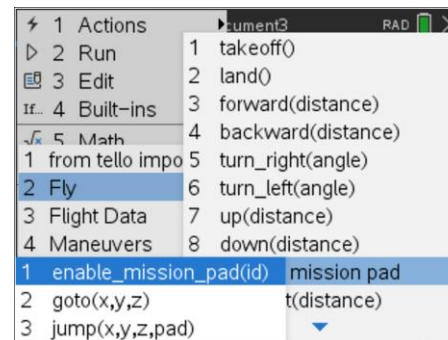
Tello Drone

10. Use the keypad to type in the name of the variable "side". This line of code redefines the "side" variable from a string type to a floating point type.
11. Select **More Modules** → **Tello** → **Fly** → **.takeoff()**

```
1.1 *Doc RAD 4/7
from tello import *
side=input("enter side length in cm ")
side=float(side)
```

```
1.1 *Doc RAD 5/9
from tello import *
side=input("enter side length in cm ")
side=float(side)
tello.takeoff()
```

12. Select **More Modules** → **Tello** → **Fly** → **Fly with mission pad** → **.enable_mission_pad(id)**



```
1.1 *Doc RAD 4/7
from tello import *
side=input("enter side length in cm ")
tello.takeoff()
tello.enable_mission_pad(pad ID)
```

13. Choose a Tello mission pad to use for your flight. In the example code, mission pad 3 was used. Be sure to enter the id of the mission pad you will use on your flight.

Note: The tello.enable_mission_pad() method must be entered after the tello.takeoff() method; this allows the drone to "see" the pad from above.

```
1.1 *Doc RAD 6/10
from tello import *
side=input("enter side length in cm ")
side=float(side)
tello.takeoff()
tello.enable_mission_pad(3)
```



Tello Drone

14. Imagine the drone flying the square on the left with a side = 75 cm.

Make a list of the coordinates of each vertex of the square.

X1, Y1 = (75 , 0)

X2, Y2 = (,)

X3, Y3 = (,)

X4, Y4 = (,)

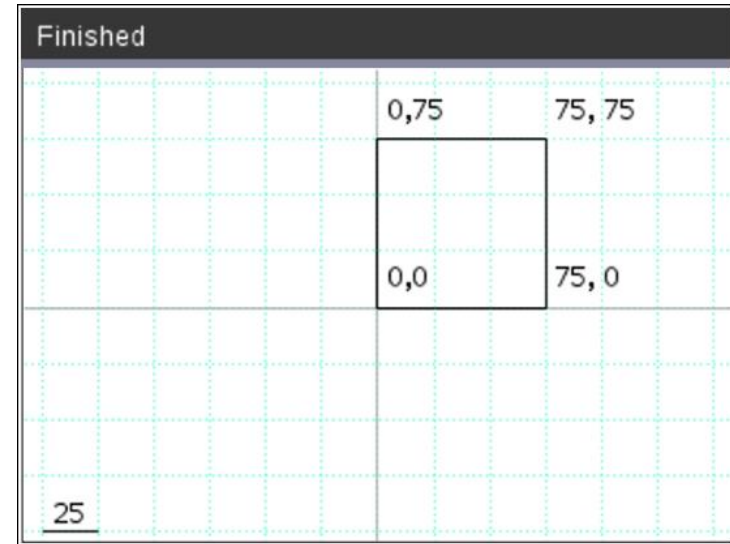
Make a similar list of the coordinates of each vertex of the square using the variable side instead of the number.

X1, Y1 = (side , 0)

X2, Y2 = (,)

X3, Y3 = (,)

X4, Y4 = (,)



TI Nspire CX II turtle program showing the flight path

15. Select **More Modules** → **Tello** → **Fly** → **Fly with mission pad** → **.goto(x,y,z)**

16. Use the number pad to type in the x,y, and z coordinates for the first vertex of the square. The example code uses a height of 50 cm, z=50.

Note: The drone can't fly below 30 cm. All (x,y,z) triplets must have a z value greater than 30.

17. Repeat steps 14-16 for the other three vertices of the square.

18. When completed, land the drone.

19. Center your drone on the mission pad and test your program!

Caution: Check that the flight path above and around the drone is clear of people and obstacles before running your program.

20. Select **Run** → **Run (Ctrl+R)**. After landing, turn off the drone power by pressing the power button.

