Tello Drone Python Reference Guide

Tello Drone

Overview		
The Tello Reference Guide is a companion document to the Tello Getting Started Guide and Videos. This reference lists all Tello module python methods that can be used to program the drone. Please see the Getting Started Guide to learn how to set up the micro:bit, pair the drone, and write your first python program. The methods supported only on the TI Nspire CX II or the TI-84 CE Python are noted. Additionally, the methods supported only on the EDU (black) model are noted. These commands depend on the drone's firmware; use the Tello app on your phone to ensure your drone has the latest firmware version loaded.		
Tello Module		
from tello import *	from tello import *	The import statement must be the first line of the program.
Fly		
Python Method	**Example Code**	**Notes**
takeoff()	tello.takeoff()	Takeoff must be the first flight command in the program.
land()	tello.land()	Land must be the last flight command in the program.
forward(distance)	tello.forward(100)	Fly forward at a distance from 10cm to 500cm.
backward(distance)	tello.backward(100)	Fly backward a distance from 10cm to 500cm.
turn_right(angle)	tello.turn_right(90)	Rotate right an angle from 0◦ to 360◦.
turn_left(angle)	tello.turn_left(90)	Rotate left at an angle from 0◦ to 360◦.
up(distance)	tello.up(50)	Ascend a distance from 10cm to 500cm.
down(distance)	tello.down(50)	Descend a distance from 10cm to 500cm.
fly_right(distance)	tello.fly_right(100)	Fly right, with no turn, from 10cm to 500cm.
fly_left(distance)	tello.fly_left(100)	Fly left, with no turn, from 10cm to 500cm.
forward_time(duration)	tello.forward_time(5)	Fly forward for the given seconds at the set speed. The calculated distance must be between 10cm to 500cm.
backward_time(duration)	tello.backward_time(5)	Fly backward for the given seconds at the set speed. The calculated distance must be between 10cm to 500cm.
enable_mission_pad(number)	tello.enable_mission_pad(6)	**(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().
goto(x,y,z)	tello.goto(50,75,100)	**(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.

jump(x,y,z,heading,pad)	tello.jump(100,100,100,45,3)	**(CX II only) and (EDU only)** Use this method to change coordinate systems among mission pads. When executed, the drone goes to the coordinates x, y, and z, heading over the present mission pad, and visually recognizes the given mission pad. Any following goto(x,y,z) commands will be within the new pad's coordinate system.

Flight Data

Python Method	Example Code	Notes
var=battery()	charge= tello.battery()	This method returns the percent charge of the drone battery.
var=height()	height= tello.height()	This method returns the relative height of the drone, in cm, using the time-of-flight sensor.
var=altitude()	altitude= tello.altitude()	This method returns the absolute altitude, in cm, using the barometer.
var=flight_heading()	heading = flight_feading()	Zero heading is established at takeoff, and this method reports the relative heading of the drone in degrees.
var=flight_time()	time= tello.flight_time()	This method returns the time, in seconds, of the time the drone has been flying since takeoff.
var=temperature()	temperature= tello.temperature()	This method returns the internal temperature of the drone in ◦C.
var=barometer()	pressure= tello.barometer()	**(TI-Nsprie CX II only)** This method returns the absolute barometric pressure surrounding the drone.

Maneuvers

Python Method	Example Code	Notes
hover(time)	tello.hover(10)	The drone hovers in place for the given time in seconds.
flip_forward()	tello.flip_forward()	Performs a forward flip. The drone must have at least 50% battery charge.
flip_backward()	tello.flip_backward()	Performs a backward flip. The drone must have at least 50% battery charge.
flip_left()	tello.flip_left()	Performs a left flip. The drone must have at least 50% battery charge.
flip_right()	tello.flip_right()	Performs a right flip. The drone must have at least 50% battery charge.
throw_and_fly()	tello.throw_and_fly()	**(EDU only)** The method is an alternative to takeoff(); when used as the first flight command, hold the drone in the palm, run the program, and gently heave the drone into the air when the propellers begin spinning, it will take flight.

Tello Drone Python Reference Guide

Tello Drone

Settings		
Python Method	**Example Code**	**Notes**
set_speed(speed)	tello.set_speed(40)	The method sets the speed of the drone in cm/s. The range is 10 to 60 cm/s. The default is 30 cm/s.
set_feedback(visibility)	tello.set_feedback("hide")	This method sets the visibility of the command feedback displayed in the shell. The command feedback from the drone is hidden when set to "hide."
motors_on()	tello.motors_on()	**(EDU only)** Turns on all motors to a slow speed to aid in cooling.
motors_off()	tello.motors_off()	**(EDU only)** Turns off the drone's motors.
Commands		
Python Method	**Example Code**	**Notes**
sleep(time)	sleep(2000)	This command pauses program execution for the given time in milliseconds. Note – Do not import the time module; doing so will interfere with the Tello module. Note - if a pause is desired while the drone is in flight, use the hover(time) method.
disp_clr()	disp_clr()	**(TI-84 Plus CE Python only)** Use this command to clear the history in the python shell.
clear_history()	clear_history()	**(TI-Nspire CX II only)** This command clears the history in the python shell.
while get_key() != "esc":	while get_key() != "esc":	**(TI-Nspire CX II only)** This structure is an indefinite loop that exits after the [esc] key is pressed. To enable on the 84 CE, import the ti_system module.
store_list("name",list)	store_list("y",the_heights)	**(TI-Nspire CX II only)** only. The method saves list data from a program for later use with the calculator. To enable on the 84 CE, import the ti_system module.