

Le Guide de Référence Tello est un document complémentaire au Guide de Démarrage Tello et aux Vidéos. Ce document répertorie toutes les méthodes Python du module Tello qui peuvent être utilisées pour programmer le drone. Veuillez consulter le Guide de Démarrage pour apprendre à configurer le micro:bit, à coupler le drone et à écrire votre premier script Python sur la calculatrice TI-83 Premium CE Edition Python ou sur la TI-nspire CX II. Les méthodes **surlignées en jaune** ne fonctionnent qu'avec les modèles Tello EDU (noir) et TT (rouge).

Module Tello

`from tello import *`

`from tello import *`

L'instruction d'importation doit être la première ligne du programme.

Vol

Méthode Python	Exemple de Code	Notes
<code>takeoff()</code>	<code>tello.takeoff()</code>	<code>takeoff()</code> (décollage) doit être la première commande de vol du programme.
<code>land()</code>	<code>tello.land()</code>	<code>land()</code> (atterrissage) doit être la dernière commande de vol du programme.
<code>forward(distance)</code>	<code>tello.forward(100)</code>	Voler vers l'avant d'une distance de 10cm à 500cm.
<code>backward(distance)</code>	<code>tello.backward(100)</code>	Voler vers l'arrière d'une distance de 10cm à 500cm.
<code>turn_right(angle)</code>	<code>tello.turn_right(90)</code>	Tourner à droite d'un angle de 0° à 360°.
<code>turn_left(angle)</code>	<code>tello.turn_left(90)</code>	Tourner à gauche d'un angle de 0° à 360°.
<code>up(distance)</code>	<code>tello.up(50)</code>	Monter d'une distance de 10cm à 500cm.
<code>down(distance)</code>	<code>tello.down(50)</code>	Descendre d'une distance de 10cm à 500cm.
<code>fly_right(distance)</code>	<code>tello.fly_right(100)</code>	Voler à droite, sans tourner, de 10cm à 500cm.
<code>fly_left(distance)</code>	<code>tello.fly_left(100)</code>	Voler à gauche, sans tourner, de 10cm à 500cm.
<code>forward_time(temps)</code>	<code>tello.forward_time(5)</code>	Voler en avant pendant le temps donné en secondes à la vitesse fixée. La distance calculée doit être comprise entre 10 cm et 500 cm.
<code>backward_time(temps)</code>	<code>tello.backward_time(5)</code>	Voler en arrière pendant le temps donné en secondes à la vitesse fixée. La distance calculée doit être comprise entre 10 cm et 500 cm.
<code>enable_mission_pad(nombre)</code>	<code>tello.enable_mission_pad(6)</code>	Permet au système de positionnement visuel de reconnaître le Mission Pad numéroté sous le drone au décollage. La fusée imprimée sur le pad pointe dans la direction positive de l'axe des x. Cette commande doit être effectuée après <code>takeoff()</code> .
<code>goto(x,y,z)</code>	<code>tello.goto(50,75,100)</code>	Voler jusqu'au point de coordonnées (x, y, z) au-dessus du Mission Pad activé. Les coordonnées doivent être comprises entre -500cm et 500cm. Notez que les valeurs x, y et z ne peuvent pas être définies simultanément entre -20cm et 20cm.
<code>jump(x,y,z,cap,pad)</code>	<code>tello.jump(100,100,100,45,3)</code>	Utiliser cette méthode uniquement sur TI-nspire CX II pour changer de système de coordonnées entre deux Mission Pads. Une fois exécutée, le drone se rend au point de coordonnées (x, y, z), pivote vers le cap indiqué au-dessus du Mission Pad actuel et reconnaît visuellement le Mission Pad passé en paramètre. Toutes les commandes <code>goto(x,y,z)</code> suivantes feront référence au système de coordonnées du nouveau Mission Pad.

Données de vol (Data)

Méthode Python	Exemple de Code	Notes
<code>var=battery()</code> <code>var=height()</code>	<code>charge = tello.battery()</code> <code>hauteur = tello.height()</code>	Cette méthode renvoie le pourcentage de charge de la batterie du drone. Cette méthode renvoie la hauteur relative du drone, en cm, en utilisant le capteur de temps de vol.
<code>var=altitude()</code> <code>var=flight_time()</code>	<code>altitude = tello.altitude()</code> <code>temps = tello.flight_time()</code>	Cette méthode renvoie l'altitude absolue, en cm, en utilisant le baromètre. Cette méthode renvoie la durée, en secondes, du temps de vol du drone depuis le décollage.
<code>var=temperature()</code> <code>var=barometer()</code>	<code>température = tello.temperature()</code> <code>pression = tello.barometer()</code>	Cette méthode retourne la température interne du drone en °C. Cette méthode renvoie la pression atmosphérique absolue autour du drone.

Manceuvres

Méthode Python	Exemple de Code	Notes
<code>hover(temps)</code> <code>flip_forward()</code> <code>flip_backward()</code> <code>flip_left()</code> <code>flip_right()</code> <code>throw_and_fly()</code>	<code>tello.hover(10)</code> <code>tello.flip_forward()</code> <code>tello.flip_backward()</code> <code>tello.flip_left()</code> <code>tello.flip_right()</code> <code>tello.throw_and_fly()</code>	Le drone effectue un vol stationnaire d'une durée indiquée en secondes. Effectue un flip avant. Le drone doit avoir une batterie chargée à au moins 50 %. Effectue un flip arrière. Le drone doit avoir une batterie chargée à au moins 50 %. Effectue un flip gauche. Le drone doit avoir une batterie chargée à au moins 50 %. Effectue un flip droite. Le drone doit avoir une batterie chargée à au moins 50 %. Cette méthode est une alternative à <code>takeoff()</code> ; lorsqu'elle est utilisée comme première commande de vol, tenez le drone dans la paume de votre main, exécutez le programme et soulevez doucement le drone en l'air lorsque les hélices commencent à tourner, il prend alors son envol.

Réglages (Set)

Méthode Python	Exemple de Code	Notes
<code>set_speed(vitesse)</code>	<code>tello.set_speed(40)</code>	Cette méthode permet de définir la vitesse du drone en cm/s. La valeur par défaut est de 30 cm/s.
<code>motors_on()</code> <code>motors_off()</code>	<code>tello.motors_on()</code> <code>tello.motors_off()</code>	Met sous tension les moteurs à une vitesse lente pour faciliter le refroidissement. Arrête les moteurs du drone.
<code>set_feedback(aff.)</code>	<code>tello.set_feedback(False)</code>	Cette méthode gère l'affichage des commandes dans le Shell. Par défaut les commandes sont affichées (<code>True</code>).

Commandes

Méthode Python	Exemple de Code	Notes
<code>sleep(temps)</code>	<code>sleep(2)</code>	Cette commande interrompt l'exécution du programme pendant le temps donné en secondes. Remarque : si une pause est souhaitée pendant que le drone est en vol, utilisez la méthode <code>hover(temps)</code> .
<code>disp_clr()</code>	<code>disp_clr()</code>	Utilisez cette commande sur la TI-83 Premium CE Edition Python uniquement, pour effacer l'historique dans le Shell Python.
<code>clear_history()</code> <code>while get_key() != "esc":</code>	<code>clear_history()</code> <code>while get_key() != "esc":</code>	Nspire CX II seulement. Cette commande efface l'historique dans le shell python. Nspire CX II seulement. Cette structure est une boucle infinie qui s'interrompt en appuyant sur la touche [esc]. Pour l'activer sur la TI-83 Premium CE Edition Python, importez le module <code>ti_system</code> .
<code>store_list("nom",liste)</code>	<code>store_list("y",les_hauteurs)</code>	Nspire CX II seulement. La méthode permet d'enregistrer les données d'un programme dans une liste pour une utilisation ultérieure avec la calculatrice. Pour l'activer sur la TI-83 Premium CE Edition Python, importez le module <code>ti_system</code> .