

Les drones TELLO EDU peuvent recevoir des commandes par liaison Wi-Fi. Le programme est exécuté sur l'ordinateur et envoie des instructions au drone, qui les réalise en temps réel.

En classe de seconde, nous utilisons le langage Python. Pour simplifier l'installation et la configuration, il est recommandé d'installer une version récente de Python 3 depuis Microsoft Store.

Cette installation permet notamment d'utiliser l'outil **pip** (*Python Package Installer*), qui sert à installer des bibliothèques Python prêtes à l'emploi. Ces bibliothèques facilitent considérablement la programmation du drone.

### Installation des bibliothèques (à effectuer une seule fois)

#### a) Se connecter à Internet

Connectez l'ordinateur à un réseau Wi-Fi disposant d'un accès à Internet.

#### b) Ouvrir l'invite de commandes

Dans la barre de recherche de Windows, tapez :

cmd

Puis choisissez **Exécuter en tant qu'administrateur**.

#### c) Installer la bibliothèque du drone

Dans la fenêtre qui s'ouvre, saisissez :

pip install djitellopy

puis validez avec la touche Entrée.

#### d) Vérifier l'installation

Des lignes de téléchargement et d'installation doivent apparaître à l'écran.

À la fin de l'opération, un message indique que l'installation a réussi.

#### e) Fermer la fenêtre

La bibliothèque est maintenant installée et prête à être utilisée.

---

### Création et exécution d'un programme

#### a) Lancer IDLE

Dans la barre de recherche de Windows, tapez :

Idle puis validez.

Une première fenêtre s'ouvre : elle permet le dialogue entre Python et l'utilisateur.

#### b) Créer un nouveau fichier

Dans cette fenêtre, cliquez sur :

File → New File

Une seconde fenêtre s'ouvre. C'est dans celle-ci que vous écrirez votre programme.

Respectez scrupuleusement la syntaxe du modèle fourni.

#### c) Connecter le drone

- Allumez le drone.
- Activez le Wi-Fi de l'ordinateur.
- Repérez le réseau Wi-Fi du drone (son nom est indiqué sur l'étiquette située près de la batterie).
- Connectez l'ordinateur à ce réseau.

#### d) Préparer la zone de décollage

Placez le drone sur une surface plane et dégagée.

Assurez-vous qu'aucune personne ne se trouve à proximité immédiate des hélices et prévenez les personnes présentes qu'un décollage va avoir lieu.

#### e) Exécuter le programme

Dans la fenêtre contenant le programme, cliquez sur :

Run → Run Module

Python vous demandera éventuellement d'enregistrer le fichier avant son exécution.

#### f) Observer le vol

Le drone doit maintenant exécuter les instructions du programme.

Une fois l'exercice terminé, faites atterrir le drone puis éteignez-le.

**Attention** : l'autonomie d'une batterie est d'environ 10 minutes.

La bibliothèque djitellopy fournit beaucoup de méthodes, mais pour un usage en seconde ou en initiation, on peut les classer en quelques familles.

### 1. Connexion

```
tello.connect()
    Connexion au drone.
tello.get_battery()
    Affiche le pourcentage de batterie.
tello.get_temperature()
    Température du drone.
```

---

### 2. Décollage et atterrissage

```
tello.takeoff()
    Décollage.
tello.land()
    Atterrissage.
tello.emergency()
    Arrêt immédiat des moteurs (à utiliser
    seulement en cas d'urgence).
```

---

### 3. Déplacements

Distances en centimètres.

```
tello.move_forward(50)
    Avancer de 50 cm.
tello.move_back(50)
    Reculer.
tello.move_left(50)
    Aller à gauche.
tello.move_right(50)
    Aller à droite.
tello.move_up(50)
    Monter.
tello.move_down(50)
    Descendre.
```

---

### 4. Rotations

Angles en degrés.

```
tello.rotate_clockwise(90)
    Tourner à droite de 90°.
tello.rotate_counter_clockwise(90)
    Tourner à gauche de 90°.
```

---

### 5. Figures simples

Carré :

```
for i in range(4):
    tello.move_forward(50)
    tello.rotate_clockwise(90)
```

Triangle :

```
for i in range(3):
    tello.move_forward(50)
    tello.rotate_clockwise(120)
```

### 6. Contrôle continu

On peut contrôler simultanément :

```
tello.send_rc_control(
    gauche_droite,
    avant_arriere,
    haut_bas,
    rotation
)
```

Exemple :

```
tello.send_rc_control(0, 30, 0, 0)
avance continuellement.
```

Cette commande est surtout utile pour le pilotage en temps réel.

---

### 7. Caméra

```
tello.streamon()
    Activer le flux vidéo :
tello.streamoff()
    Couper le flux :
frame = tello.get_frame_read().frame
    Récupérer une image :
```

---

### 8. Informations de vol

```
tello.get_height()
    Altitude.
tello.get_speed_x()
tello.get_speed_y()
tello.get_speed_z()
    Vitesses.
tello.get_flight_time()
    Temps de vol.
tello.get_barometer()
    Altitude barométrique.
```

---

### 9. Commandes spéciales du Tello EDU

Le Tello EDU peut exécuter des trajectoires prédéfinies.

Par exemple :

```
tello.go_xyz_speed(100, 0, 50, 30)
```

Va au point (100, 0, 50) à la vitesse 30 cm/s.

Ou encore :

```
tello.curve_xyz_speed(
    50,0,50,
    100,50,50,
    30
)
```

Suit une trajectoire courbe.