

# Séquence 3

*Algorithme et programmation*



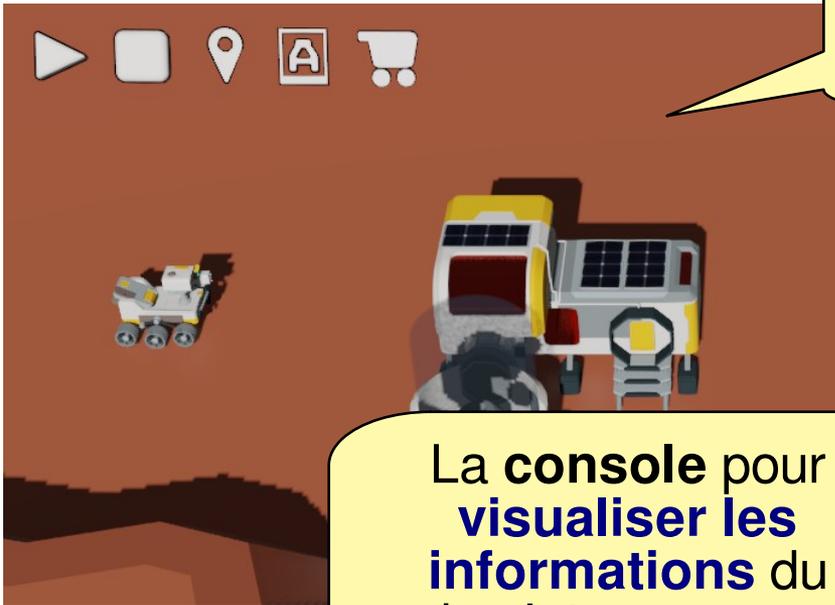
# Introduction à la programmation Python avec Ropy



# Présentation de Ropy et de son environnement de programmation

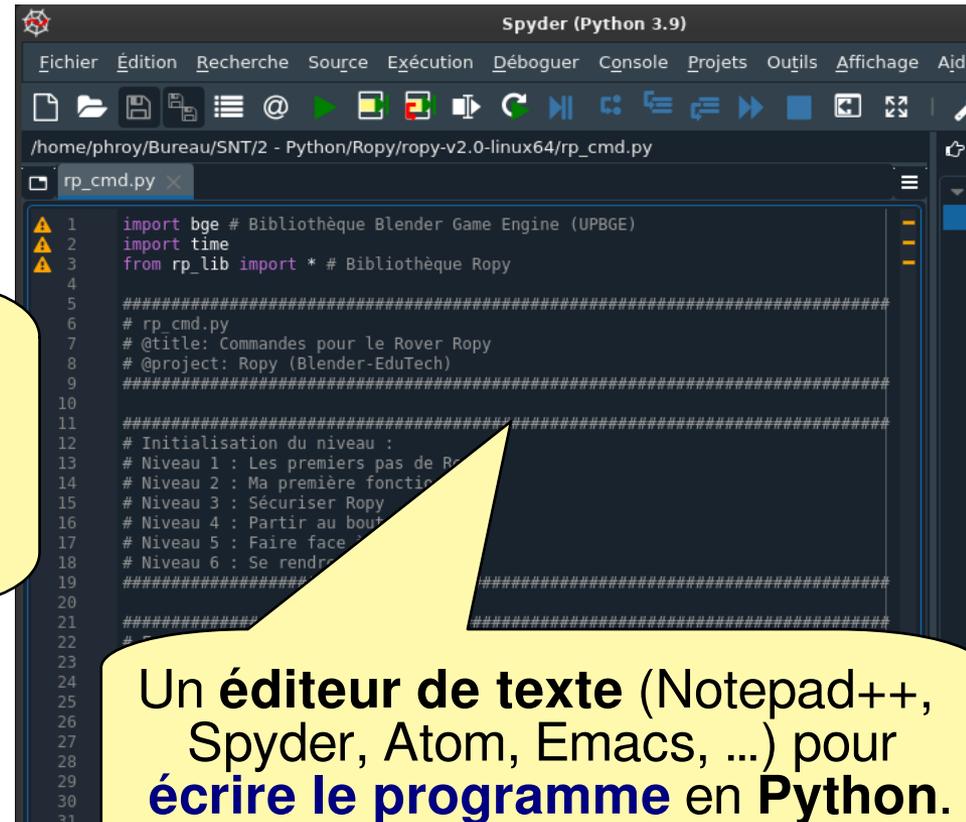
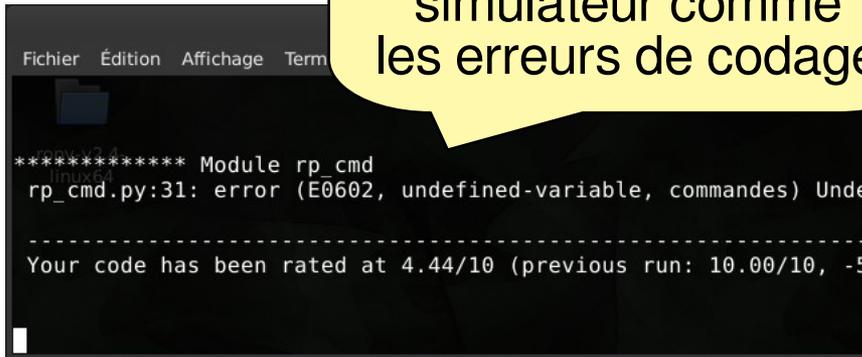


**Ropy** est un rover martien qui se commande grâce au langage **Python**. L'interface de programmation se décompose en **3 fenêtres** : un éditeur de texte, le simulateur et la console.



Le **simulateur** permet de **visualiser l'évolution du Rover**.

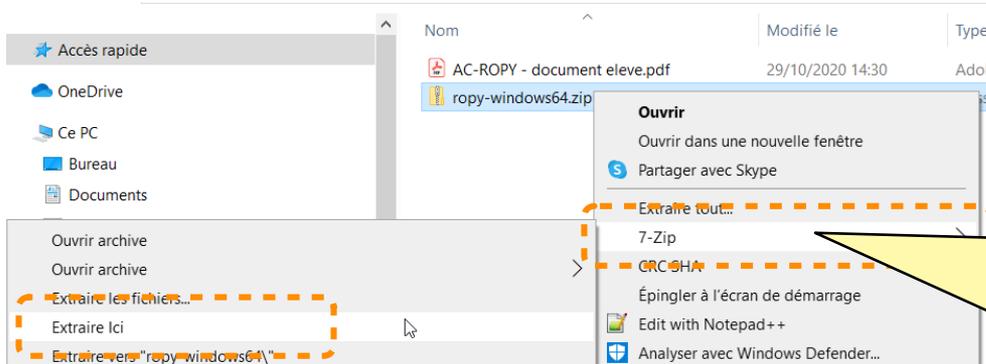
La **console** pour **visualiser les informations** du simulateur comme les erreurs de codage.



Un **éditeur de texte** (Notepad++, Spyder, Atom, Emacs, ...) pour **écrire le programme** en **Python**.

# Éditer le programme avec Spyder

## Ouvrir le fichier rp\_cmd.py

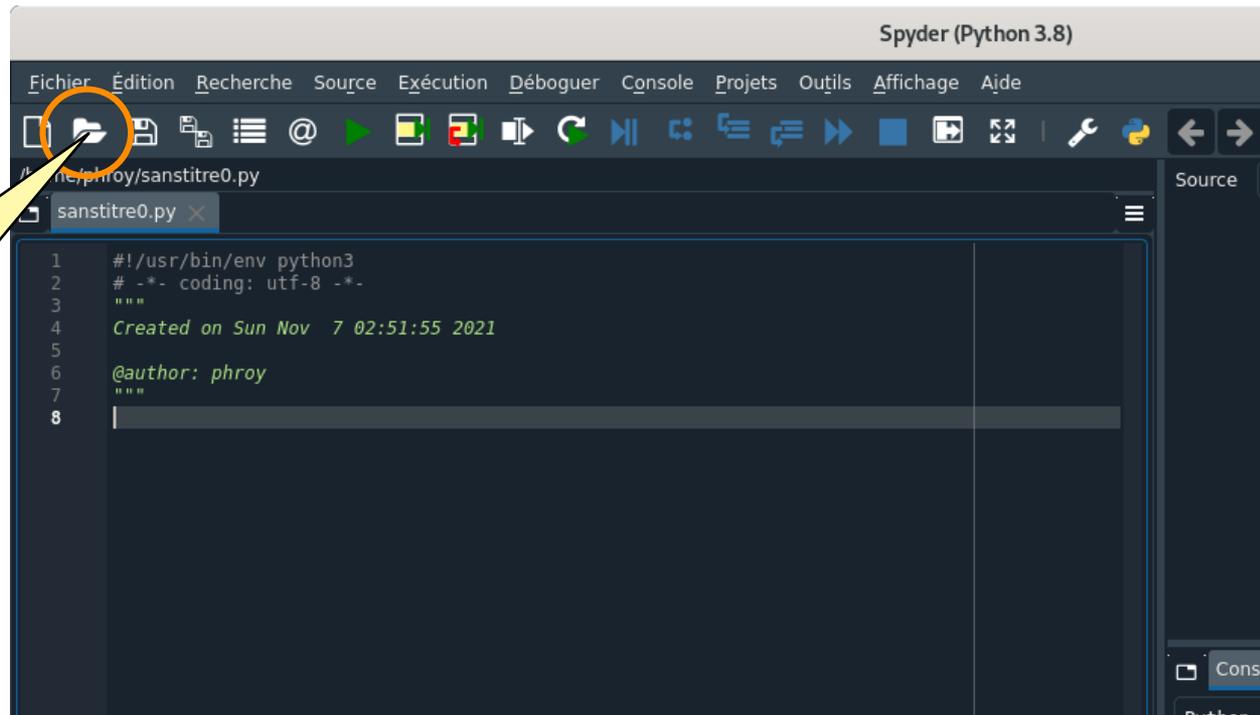


1 : Récupérer l'archive **ropy-windows64.zip** et la décompresser avec **7-Zip** dans votre répertoire. L'extraction va créer le répertoire **ropy**

2 : Lancer le Logiciel **Spyder**.



3: Ouvrir le fichier Python à éditer **rp\_cmd.py** (Ropy commandes) présent dans le répertoire **ropy**.



# Éditer le programme avec Spyder



## Exécution du programme

5 : **Sauvegarder** le fichier

**Attention !**

Toujours sauvegarder le fichier avant son exécution avec le simulateur.

Le **simulateur** et la **console** se lancent en même temps avec le programme **ropy.bat**

**Arrêter et réinitialiser**

**Afficher l'aide**

**Niveau actuel**

6 : **Exécuter** le programme

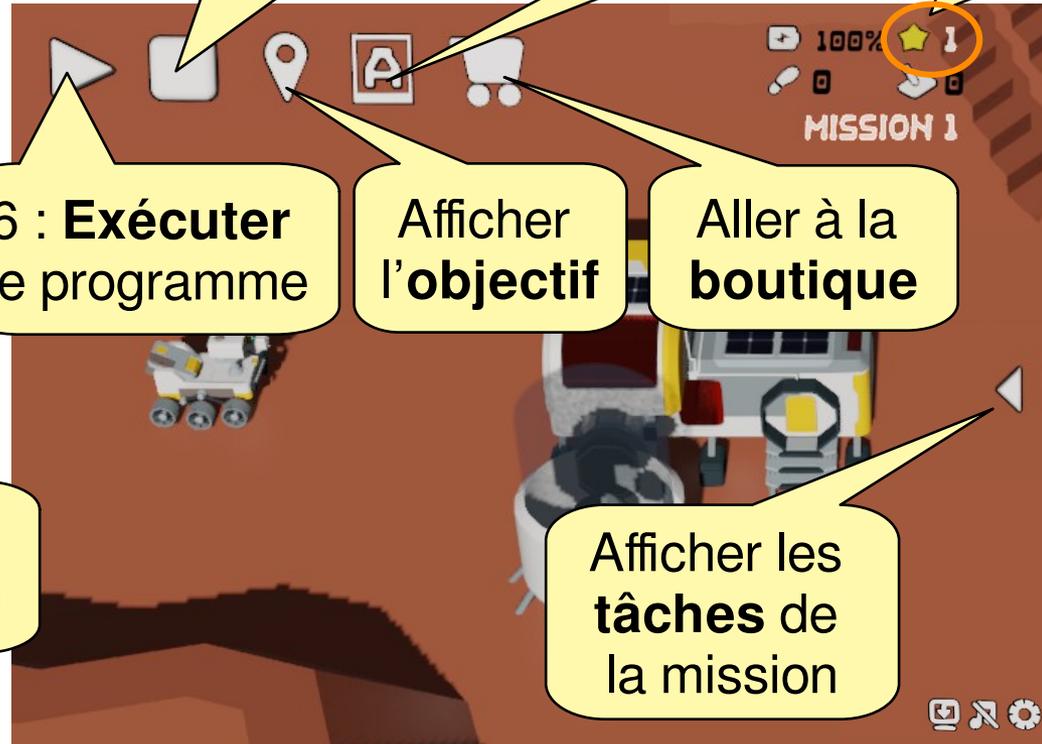
**Afficher l'objectif**

**Aller à la boutique**

4 : **Écrire** le code Python

**Afficher les tâches** de la mission

```
Fichier  Édition  Recherche  Source  Exécution  Déboguer  Console
/home/phroy/Bu
rp_cmd.py x
2 import time
3 from rpy import
4
5 #####
6 # rp_cmd.py
7 # @title: Commandes
8 # @project: Ropy
9 #####
10
11 #####
12 # Initialisation du niveau :
13 # Niveau 1 : Les premiers pas de Ropy
14 # Niveau 2 : Ma première fonction
15 # Niveau 3 : Sécuriser Ropy
16 # Niveau 4 : Partir au bout du monde
17 # Niveau 5 : Faire face à l'inconnu
18 # Niveau 6 : Se rendre utile
19 #####
20
21 #####
22 # Fonctions
23 #####
24
25 #####
26 # Commandes
27 #####
28
29 def commandes():
30
31     rp_gauche()
32     rp_avancer()
33     rp_avancer()
34     rp_avancer()
35     rp_avancer()
36
37
38     rp_fin() # A garder
39
```



# Contenu du fichier rp\_cmd.py



Le fichier `rp_cmd.py` comporte 4 sections.

```
import bge # Bibliothèque Blender Game Engine (UPBGE)
import time
from rp_lib import * # Bibliothèque Ropy

#####
# rp_cmd.py
# @title: Commandes pour le Rover Ropy
# @project: Ropy (Blender-EduTech)
#####

#####
# Initialisation du niveau :
# Niveau 1 : Les premiers pas de Ropy
# Niveau 2 : Ma première fonction
# Niveau 3 : Sécuriser Ropy
# Niveau 4 : Partir au bout du monde
# Niveau 5 : Faire face à l'inconnu
# Niveau 6 : Se rendre utile
#####

#####
# Fonctions
#####

#####
# Commandes
#####

def commandes():
    ➔ rp_gauche()
    rp_avancer()
    rp_avancer()
    rp_avancer()
    rp_avancer()

    rp_fin() # A garder

#####
# En: Externals calls << DONT CHANGE THIS SECTION >>
# Fr: Appels externes << NE PAS MODIFIER CETTE SECTION >>
#####

if __name__=='start':
    thread_cmd_start(commandes)
if __name__=='stop':
    thread_cmd_stop()
```

Le code doit être indenté  
(décalé sur la droite) avec  
la touche Tab

} **Import des bibliothèques**  
**Ne pas modifier cette section**

} **Fonctions** : section pour le  
codage de **vos fonctions**

} **Commandes** : section pour le  
codage des commandes du robot

} **La commande `rp_fin()`**  
**est à conserver.**

} **Appels du simulateur**  
(Blender Game Engine)  
**Ne pas modifier cette section**



# Mission 2 - Ma première fonction

## Création d'une fonction



**Objectif 2** : Créer la fonction `mrp_avancer()` regroupant `avancer` et `marquer`.

```
#####  
# Fonctions  
#####
```

---

---

---

---

---

---

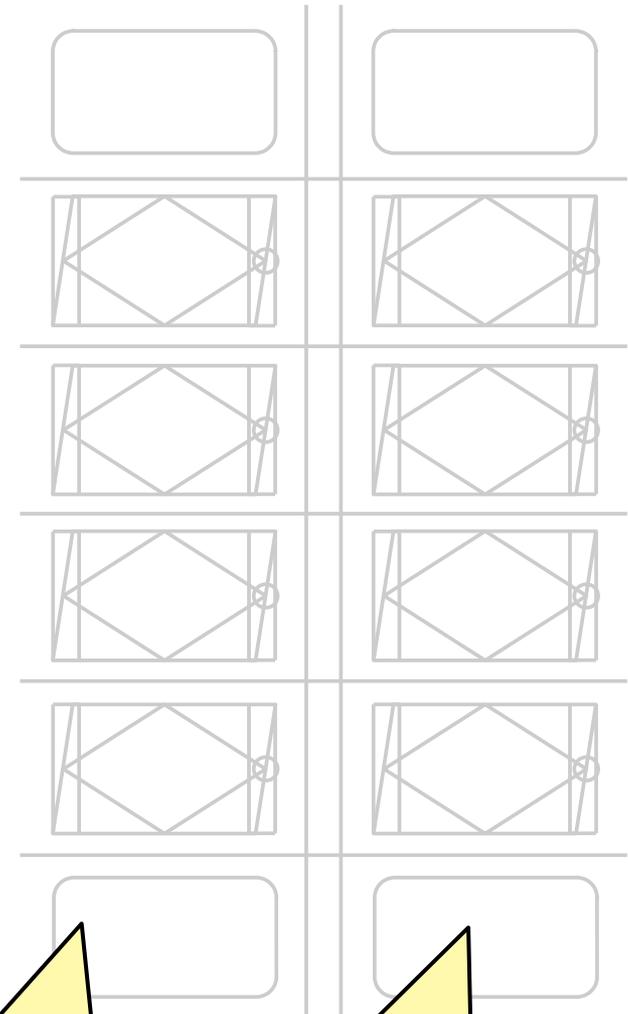
---

---

```
#####  
# Commandes  
#####
```

---

---



Représenter que le premier appel

Représenter la fonction

# Mission 3 – Apprendre le danger

## Structure conditionnelle (si, alors, sinon)



**Objectif 3.1** : Aller à la mission 3, provoquer une collision avec un obstacle en avançant et observer ce qui se passe. Il vous faut donc sécuriser l'avance du robot.

```
#####  
# Commandes  
#####
```

---

---

---

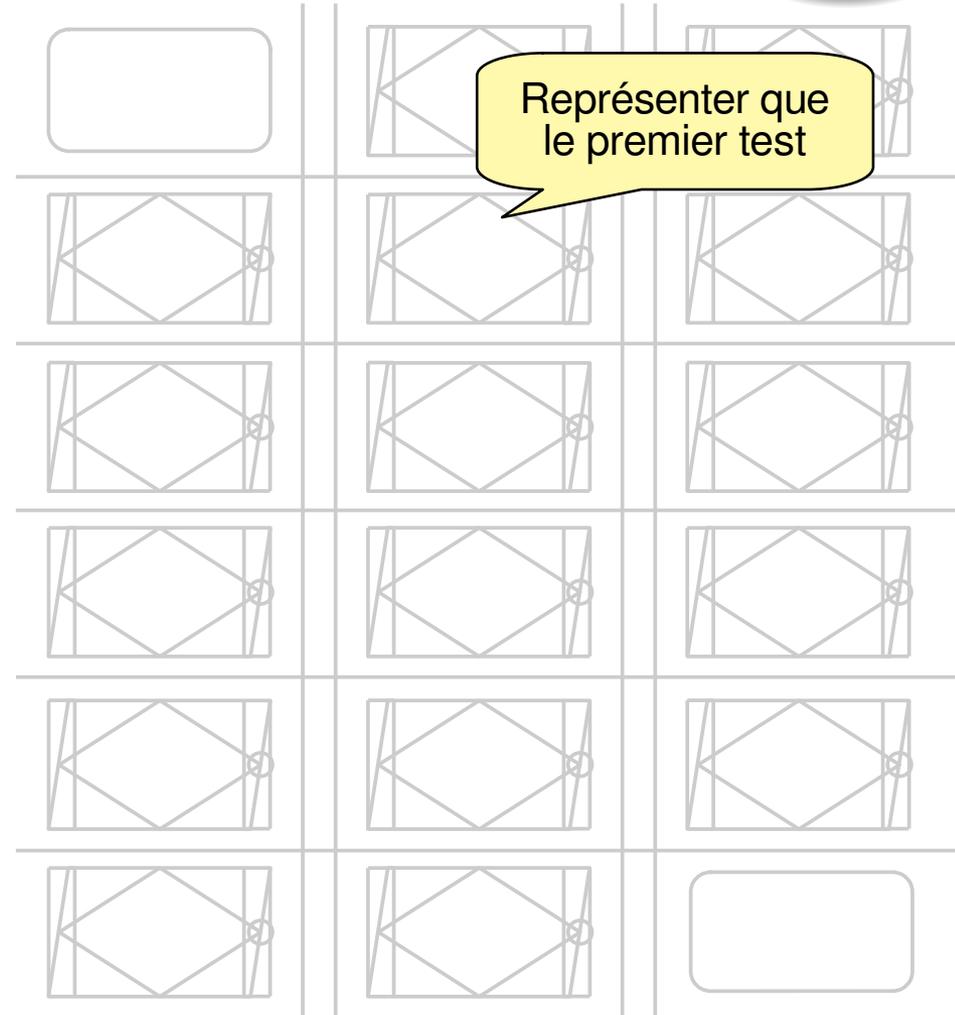
---

---

---

---

---



La fonction pour **détecter un mur** est : `rp_detect ()`. La fonction retourne **True** si il a un mur et **False** si il n'y a pas de mur.















