

Commencer à travailler avec l'Astro Pi

Utiliser le Raspberry Pi pour comprendre le langage de programmation

Bienvenue dans le Challenge Astro Pi européen !

Astro Pi est le nom d'un petit ordinateur développé par la Raspberry Pi Foundation, en collaboration avec l'Agence spatiale britannique et l'Agence spatiale européenne (ESA).

Il existe deux Astro Pi vraiment spéciaux. Baptisés Ed et Izzy, ils se sont qualifiés pour s'envoler dans l'espace. Ils sont maintenant à bord de la Station spatiale internationale (ISS) pour que des élèves comme vous les utilisent.

Dans le cadre du premier ensemble d'activités, vous allez monter pour la première fois votre Raspberry Pi et apprendre le langage de codage essentiel dont vous avez besoin pour programmer votre Astro Pi. De la sorte, vous aurez la possibilité de voir vos code et expériences scientifiques exécutés sur l'ISS.



Équipement

- Kit Astro Pi
- Moniteur
- Clavier USB
- Souris USB

Activité 1 – Prise en main du Raspberry Pi

Le Raspberry Pi est un ordinateur, qui ressemble beaucoup aux ordinateurs que vous avez l'habitude de voir, si ce n'est qu'il est petit et n'a pas de boîtier. Vous pouvez utiliser votre Raspberry Pi pour naviguer sur Internet ou envoyer des e-mails, ou encore le transformer en lecteur MP3 et aider les astronautes de l'ISS à se détendre pendant leur temps libre. Et, bien d'autres choses encore ! Vous pouvez effectuer des expériences scientifiques en utilisant cet outil informatique et analyser les données collectées, comme le font les scientifiques dans l'espace !

Dans cette activité, vous allez utiliser votre Raspberry Pi et découvrir ce que vous pouvez faire avec.

Exercice

1. Pour commencer votre mission Astro Pi, vous avez besoin d'un équipement spécial que vous trouverez dans votre kit Astro Pi. Vous aurez aussi besoin d'un moniteur, d'un clavier et d'une souris. La Figure 1 illustre l'équipement dont vous aurez besoin. Ces composants constituent le matériel. Étiquetez les différents composants de l'ordinateur Raspberry Pi illustrés à la Figure 1 (carte Micro SD, adaptateur pour carte SD, Raspberry Pi 3, alimentation avec broches pour prise UE, câble HDMI).

1. _____
2. _____
3. _____
4. _____
5. _____

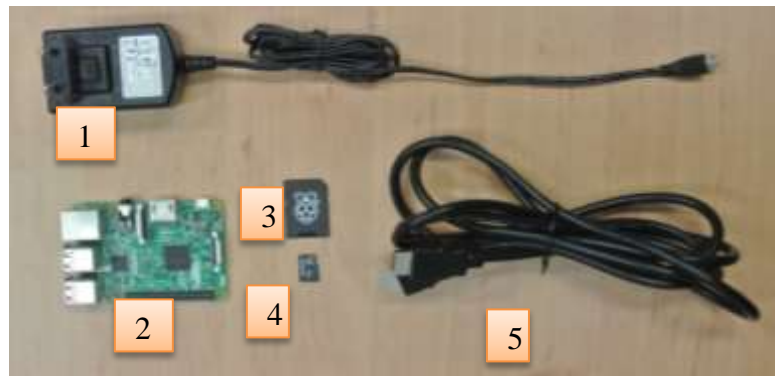


Figure 1 – Les composants du Raspberry Pi

2. Vous allez maintenant monter votre Raspberry Pi ! Pour cela, suivez les instructions I à IV et reportez-vous à la Figure 2 ci-dessous.

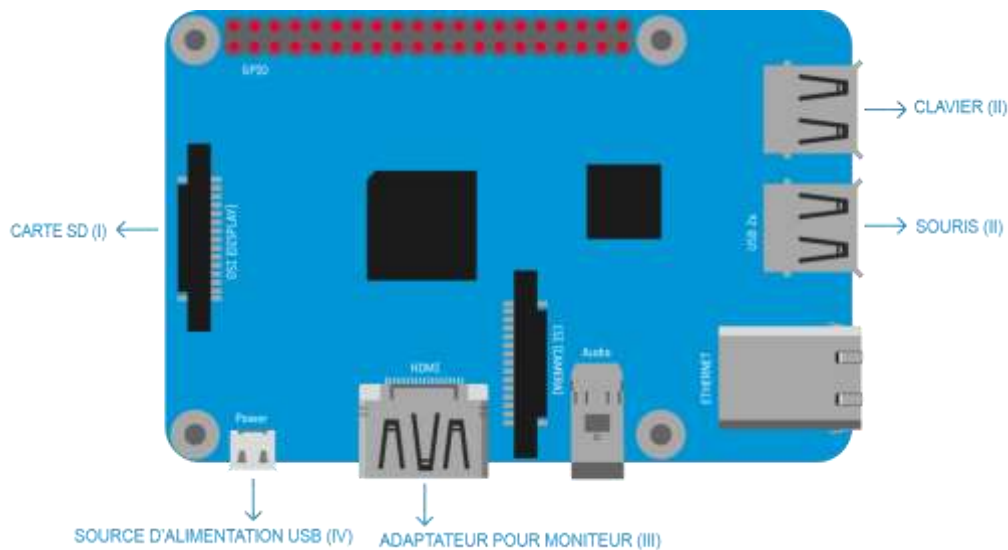


Figure 2 – Comment monter votre Raspberry Pi

- I. Retirez la carte SD de l'adaptateur de carte SD. Placez-la dans le logement de carte SD du Raspberry Pi. Elle ne s'insère que dans un sens.
- II. Branchez votre clavier et votre souris aux ports USB.
- III. Branchez le câble HDMI du moniteur (ou de la TV) au port HDMI du Raspberry Pi et allumez le moniteur.

CONSEIL : ceci est sans doute l'étape la plus délicate du montage de votre Raspberry Pi. Vérifiez que vous avez sélectionné la bonne entrée (par ex., HDMI, DVI, etc.). Dans le cas où le moniteur n'aurait qu'un connecteur VGA (voir figure 3), vous devez utiliser l'adaptateur HDMI-VGA (dans le KIT).



Figure 3 – Ports VGA et HDMI

- IV. Branchez l'alimentation micro-USB. Cette opération met sous tension et amorce votre Raspberry Pi. Cela peut prendre jusqu'à 5 minutes. Vous devriez voir la page de garde du Raspberry Pi et du texte devrait s'afficher sur l'écran. Une fois le démarrage du Raspberry Pi terminé et que le texte a arrêté de défiler à toute vitesse sur l'écran, vous devriez voir le visuel illustré à la Figure 4.



Figure 4 – L'environnement de bureau ou Interface utilisateur graphique

3. Vous pouvez maintenant commencer à utiliser le logiciel du Raspberry Pi. Vous pouvez, par exemple, ouvrir un navigateur Web et surfer sur Internet. Pour cela, cliquez sur le menu principal dans le haut de l'écran (le logo Raspberry) et choisissez *Internet > Chromium Web Browser* (vous devez avoir le wifi pour pouvoir utiliser Internet). Utilisez le Web pour essayer de trouver l'emplacement courant de l'ISS. Notez-le.

4. Explorez le logiciel du Raspberry en choisissant différentes options dans le menu de programmation. Cela fait, pouvez-vous expliquer les différences entre matériel et logiciel ?

Remarque : pour éteindre votre Raspberry Pi, ne vous limitez pas à le débrancher. À chaque fois que vous avez fini de travailler avec, arrêtez-le en cliquant sur le bouton Shutdown (Arrêter) dans le menu principal.

Extension

L'habit ne fait pas le moine ! Faites la liste des appareils numériques que vous considérez être des ordinateurs et que vous utilisez dans votre vie quotidienne. Comparez votre liste à celle de vos camarades et réfléchissez à ce qui fait d'un ordinateur un ordinateur !

Activité 2 – Le Python en bref

Pour que vous puissiez communiquer avec les astronautes de l'ISS en utilisant votre Raspberry Pi, vous devez lui donner les instructions nécessaires. Comme n'importe quel autre ordinateur, il ne fera que ce que vous lui enseignez ou ce pour quoi vous le programmez. Pour cela, vous devez apprendre un langage de programmation (de la même façon que vous avez dû apprendre la langue que vous parlez pour communiquer avec le monde qui vous entoure !). Au cours de cette activité, vous allez apprendre le vocabulaire de base du langage Python et écrire votre premier programme.

Le saviez-vous ?

Les astronautes doivent eux aussi apprendre plusieurs langues avant de pouvoir s'envoler pour l'espace. Ils doivent tous, par exemple, apprendre le russe qui est la langue des commandes du vaisseau Soyouz. À l'heure actuelle, Soyouz est le seul vaisseau disponible pour transporter les équipages vers l'ISS et les ramener. Soyouz signifie « union » en russe. Bien qu'il ait été conçu par l'Union soviétique au début de l'ère de l'exploration spatiale, le vaisseau Soyouz est encore utilisé aujourd'hui, avec, certes, des modifications importantes.



Exercice

1. Pour commencer à utiliser Python sur votre Raspberry Pi, vous devez cliquer sur le menu principal dans le haut de l'écran (le logo Raspberry) et sélectionner *Programming > Open Python 3(IDLE)* (Programmation > Ouvrir Python 3(IDLE)). Une fenêtre qui contient le shell de Python s'ouvre (Figure 5, à gauche). Vous pouvez saisir votre programme directement dans ce shell, mais il est plus facile de travailler dans une fenêtre d'éditeur de texte. Pour ouvrir une nouvelle fenêtre, sélectionnez *File> New File* (Fichier > Nouveau fichier). Deux fenêtres devraient maintenant être ouvertes. La première vous permet d'écrire votre programme (la fenêtre d'éditeur de texte) et la seconde d'afficher la sortie (le shell de Python), c'est-à-dire les résultats, les messages d'erreur, etc. Vérifiez que les deux fenêtres sont visibles.

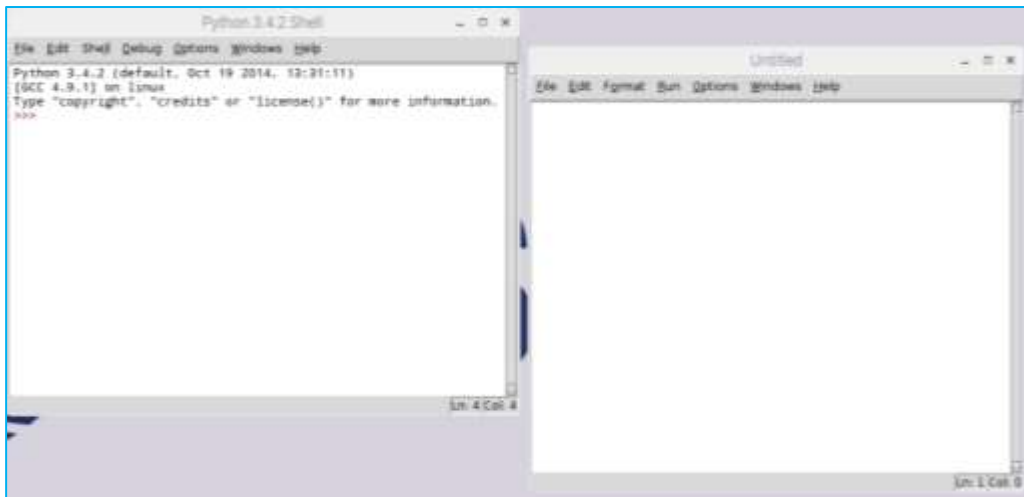


Figure 5 – Le shell de Python (à gauche) et la fenêtre d'éditeur de texte (à droite)

2. Vous êtes maintenant prêt à commencer à coder en Python. Saisissez le code suivant dans la fenêtre d'éditeur de texte.

```
File Edit Format Run Options Windows Help
print("Bonjour l'ISS")
```

3. Pour voir les résultats de votre code, vous devez choisir *Run > Run Module* (Exécuter > Exécuter le module). Le programme vous demande d'enregistrer votre code. Saisissez un nom pour le fichier et exécutez-le. Qu'a-t-il fait ?

Remarque : n'oubliez pas d'enregistrer régulièrement votre travail ! Pour enregistrer le fichier sur lequel vous travaillez, choisissez File -> Save (Fichier -> Enregistrer).

4. Essayez de vous présenter aux astronautes qui vivent sur l'ISS. Copiez le code ci-dessous dans votre fichier et complétez-le avec le message que vous aimeriez transmettre aux astronautes.

Félicitations ! Vous venez de créer votre premier programme en Python.

```
File Edit Format Run Options Windows Help
# Mon premier programme :)
# imprimer un texte
print("Bonjour l'ISS")
print("Comment ça va?")
print("Je m'appelle.... je suis...et je travaille sur la mission Astro Pi.")
```

Remarque : l'ordinateur va essayer de comprendre tout ce que vous écrivez dans la fenêtre. Si vous voulez écrire un bref commentaire juste pour vous, utilisez le signe # devant votre note. C'est ce que l'on appelle « commenter ». Les commentaires ne sont pas traduits dans le langage de la machine. L'ordinateur ignore tout simplement le texte qui suit le signe #.

Activité 3 – Faire un quiz en Python

Pour le moment, vous avez donné des instructions simples à l'ordinateur. Nous allons maintenant faire en sorte que l'ordinateur vous pose des questions et réagisse à vos réponses.

Exercice

1. Ouvrez une nouvelle fenêtre Python 3 et saisissez le code ci-dessous.

```
File Edit Format Run Options Windows Help
#un mot ou un nombre qui suit le symbole '=', s'appelle une variable.
#les variables sont utilisées pour stocker des informations dont l'ordinateur doit se rappeler

orbites = input("Combien de temps faut-il à l'ISS pour effectuer un orbite autour de la Terre?")
```

2. Sélectionnez *Run>Run Module* (Exécuter > Exécuter le module) pour exécuter votre code. Que s'est-il passé ?

3. Python utilise la fonction **input ()** pour vous poser des questions. Savez-vous combien il faut de temps à l'ISS pour effectuer une orbite autour de la Terre ? Saisissez la réponse dans le shell de Python.

4. Vous pouvez aussi utiliser le shell de Python comme une calculatrice. Examinez les exemples suivants.

```
File Edit Shell Debug Options Windows Help
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>> 12+12
24
>>> 12*2
24
>>> 12/2
6.0
>>> 12*2-12
12
```

Essayez de répondre à la question ci-dessous en utilisant le shell de Python pour effectuer les calculs nécessaires. Expliquez les calculs utilisés.

À combien de couchers de soleil les astronautes assistent-ils chaque jour ?

5. Nous allons maintenant faire un quiz scientifique en utilisant le Raspberry Pi. Pour cela, vous devrez sélectionner les bonnes et les mauvaises réponses. En langage Python, vous pouvez faire cela en utilisant des **instructions de sélection**, que l'on appelle aussi parfois des « **conditions if** » (si). Ouvrez un nouveau fichier et saisissez le code ci-dessous. Vous devez compléter le code en écrivant la réponse. À votre avis, que va faire le programme ? Exécutez le module pour voir.

Remarque : le code « **int** » est devant « **input** » (saisir) pour indiquer à Python que votre réponse sera un nombre entier. Un nombre entier est un chiffre rond comme, par exemple, 5 ou 10. En Python, si vous voulez traiter des informations comme un nombre, vous devez donner cette instruction à l'ordinateur. En plus des nombres entiers, vous pouvez aussi avoir des nombres à virgule flottante. Un nombre à virgule flottante est un nombre avec des chiffres après la virgule, comme 5,1 ou 10,99.

```
File Edit Format Run Options Windows Help
CoucherDeSoleil=int(input("A combien de couchers du soleil les astronautes assistent-ils chaque jour?"))
Reponse= input("Indiquez ici la réponse que vous avez donnée à la question 4")

if CoucherDeSoleil==Reponse:
    print("Bravo! C'est la bonne réponse")
else:
    print("Réponse erronée. Réessayez")

elif CoucherDeSoleil>13 and CoucherDeSoleil<16:
    print("Vous y êtes presque! Réessayez")
```

6. Vous pouvez aussi donner des indices pendant le quiz. Juste avant la ligne « **else** » (sinon), ajoutez la commande ci-dessous et exécutez de nouveau votre module. Écrivez un commentaire pour expliquer ce que fait le mot-clé « **elif** » (compression de « **else if** », signifie sinon si en français).

#

```
elif CoucherDeSoleil>13 and CoucherDeSoleil<16:
    print("Vous y êtes presque! Réessayez")
```

Extension

Les conditions « **if** » peuvent contrôler une liste d'éléments au lieu d'un unique élément. Pour cela, il vous suffit d'utiliser le nombre de conditions « **elif** » dont vous avez besoin. Écrivez un programme qui demandera à l'utilisateur de saisir un nombre entre 45 et 60. Donnez une réponse qui indique à l'utilisateur si le nombre est « dans la plage », « trop élevé » ou « trop bas ».

Activité 4 – Dessiner des formes géométriques avec Python

Il y a dans Python de nombreux modules que vous pouvez utiliser dans vos programmes. Ces modules s'apparentent à des manuels d'instructions qui expliquent comment faire quelque chose. Ils sont stockés par Python dans une bibliothèque. Au cours de cette activité, vous apprendrez à utiliser le « module Turtle » de Python pour dessiner des formes géométriques tout en apprenant encore un peu de Python, ce qui sera utile pour votre mission.

Exercice

1. Ouvrez une nouvelle fenêtre Python 3 et saisissez le code ci-dessous. Enregistrez-le et exécutez-le.

```
File Edit Format Run Options Windows Help
# pour commencer à utiliser Turtle vous devez importer Turtle de la bibliothèque de Python
from turtle import Turtle, Screen

screen = Screen() # crée la fenêtre qui affichera les dessins de Turtle
turtle = Turtle() # crée une tortue. La variable (ou nom) turtle est créée pour y faire référence.
turtle.shape("turtle") # donne la forme de la tortue
```

2. Vous venez de créer la tortue qui va vous aider à réaliser des dessins et des formes sympas ! Il faut maintenant que vous donniez des instructions à votre tortue.

2.1 Ajoutez la ligne suivante à votre code :

turtle.forward(100)

Qu'est-il arrivé à la tortue ?

2.2 Ajoutez une autre instruction :

turtle.right(90)

Changez les nombres entre parenthèses. Essayez de remplacer « right » (droite) par « left » (gauche). Que fait la tortue avec ces instructions ?

Remarque : la valeur 90 entre parenthèses dans *turtle.right(90)* (tortue.droite [90]) est en degrés.

2.3 Ajoutez une autre instruction dessous pour déplacer de nouveau votre tortue :

turtle.forward(100)

Qu'est-il arrivé à la tortue ?

2.4 Que devez-vous ajouter dans votre code pour faire un carré ? Dans le cadre ci-dessous, écrivez le code final nécessaire pour que votre tortue trace un carré.

3. Vous pouvez donner de nombreuses instructions différentes à votre tortue. Examinez le code suivant.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

3.1 Quelle est, à votre avis, la forme géométrique que va dessiner la tortue ? Notez votre réponse par écrit.

3.2 Enregistrez et exécutez votre code. Aviez-vous vu juste ?

3.3 Dans le code ci-dessus, vous avez répété quelques lignes de code. En informatique, ce n'est pas la manière la plus efficace de procéder. Au lieu de saisir de nombreuses lignes de code, vous pouvez les écrire une fois et ajouter une instruction pour les répéter. En Python, il y a deux façons de répéter une instruction : la *boucle while* (tant que) et la *boucle for* (pour).

3.3.1 Ouvrez une nouvelle fenêtre Python 3 et saisissez le code ci-dessous. Le code qui est dans la boucle (c'est-à-dire le code qui est indenté ou décalé sur la droite) est le jeu d'instructions que le programme va répéter. Enregistrez et exécutez votre programme. Qu'a-t-il fait ?

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

while True:
    turtle.forward(100)
    turtle.right(120)
```

3.3.2 Dans cet exemple, une *boucle for* a été utilisée. Copiez le code ci-dessous et exécutez-le pour voir ce qui se passe.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

for i in range(3):
    turtle.forward(100)
    turtle.right(120)
```

Remarque : avez-vous remarqué l'imbrication des instructions sous les boucles while/for ? C'est ce que l'on appelle l'indentation. Python utilise l'indentation pour délimiter les blocs de code quand des boucles ou des conditions « if » sont utilisées. Chaque niveau d'indentation compte quatre espaces, mais vous pouvez appuyer sur Tab qui insèrera les espaces pour vous. Pensez à l'indentation quand vous copiez et collez du code.

3.3.3 Quelle boucle choisiriez-vous pour réaliser le dessin ? Expliquez votre choix et expliquez les différences entre les deux types de boucles.

Extension

Vous savez désormais avancer, aller à droite et à gauche, et faire des répétitions en utilisant des boucles. C'est le moment d'être créatif ! Seriez-vous capable de dessiner Paxi, la mascotte de l'ESA ? Quelles formes géométriques reconnaissez-vous dans l'image de Paxi ?



ÉTAPES SUIVANTES

Vous venez de monter votre Raspberry Pi et maîtrisez maintenant les bases de la structure de programmation du Python ! Passez à l'Activité 2 pour apprendre à utiliser la matrice de LED Sense HAT pour communiquer avec les astronautes sur l'ISS.