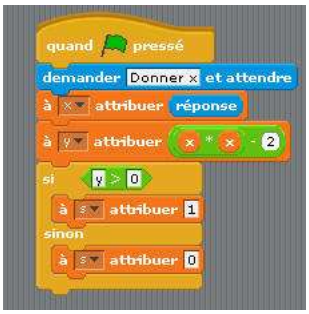
Découverte du langage Python

Introduction



Tu as déjà découvert que pour programmer une application tu avais besoin d'une interface de développement (IDE). Tu as déjà utilisé la programmation par blocs (type blockly). Cette programmation était assez simple puisqu'elle limitait les erreurs. En effet, il suffisait de faire des glissés déplacés.

Tu vas découvrir une autre façon d'écrire un code de programmation dans un nouveau langage.

Ce langage de programmation est le Python.

Contrairement à ce que tu as déjà vu, tu devras cette fois ci taper des commandes. Il faudra alors être très rigoureux sur la syntaxe d'écriture de tes commandes !

Pour programmer en Blockly, tu as eu besoin d'un IDE comme Snap ou Scratch.

Pour programmer en Python il existe plusieurs IDE. Tu vas utiliser un IDE en ligne :

http://pythontutor.com/visualize.html#mode=edit

Premier essai sur un IDE

Tu vas commencer par voir comment programmer des actions simples.

Par exemple, tu vas programmer l'affichage d'informations sur ton écran d'ordinateur en affichant « Programme d'affichage des capteurs de l'Astropi »

Pour cela utilise la commande print(" …… ") ; (attention de ne pas oublier le point-virgule à la fin !)

Programmation avec l'IDE astropi (simulateur)

Tu vas aussi tester de la même façon une action de sortie sur le système Astropi en mode simulation.

Tu as vu que le système Astropi proposait en sortie le pilotage de 5x5 led de couleurs. Il sera alors possible grâce à ces led d'allumer une Led voulue à la couleur voulue et d'afficher aussi un simple texte qui défilera.

Affiche alors le même texte : « Programme d'affichage des capteurs de l'Astropi »

https://astro-pi.org/updates/sense-hat-emulator/

ou ici

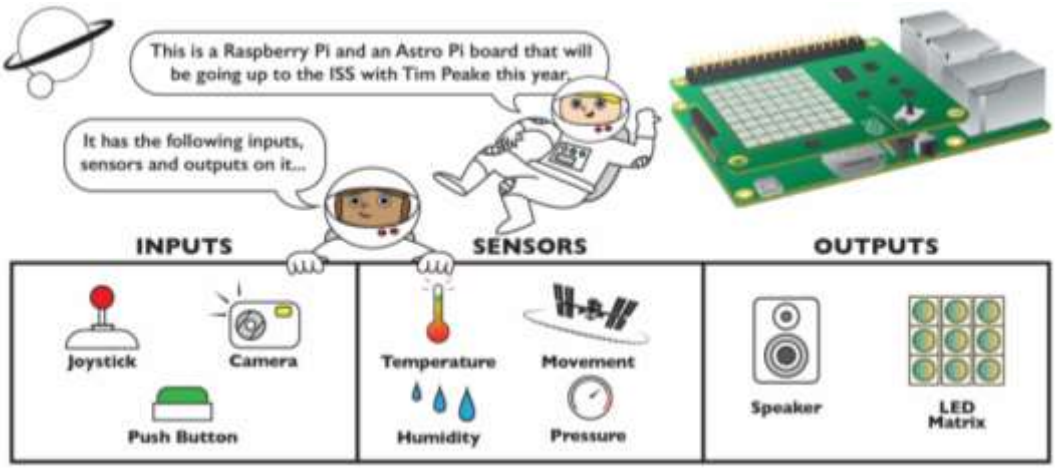
https://trinket.io/sense-hat

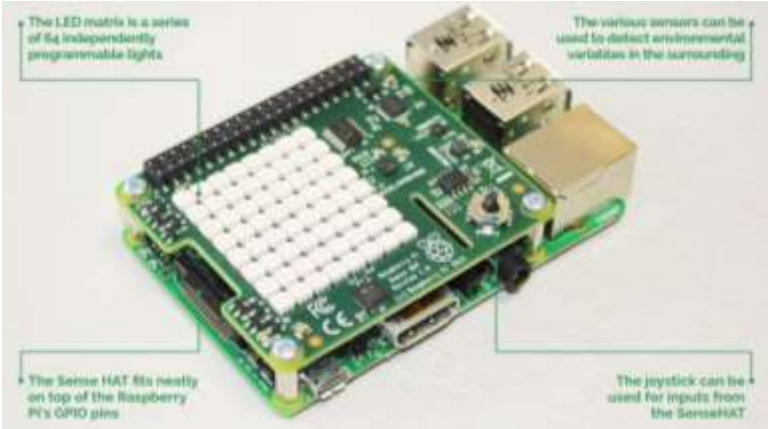
Tu souhaites maintenant afficher les informations issues de quelques capteurs.

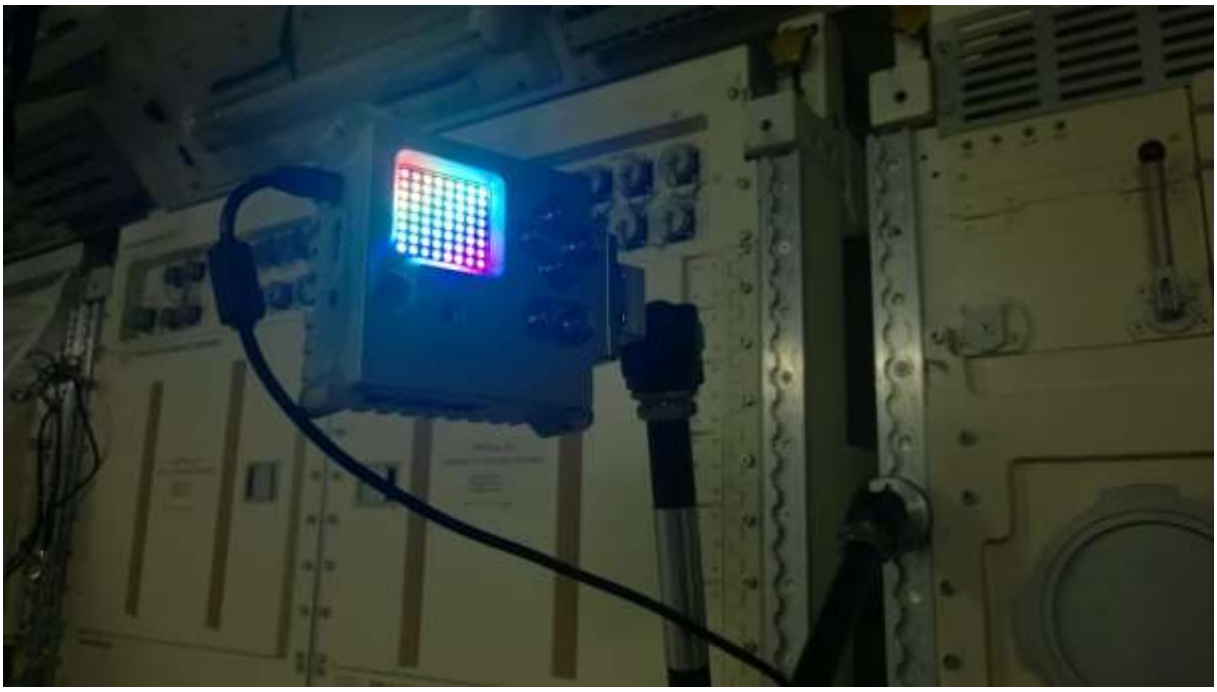
La liste des capteurs de l'Astropi est disponible ici :

http://esamultimedia.esa.int/docs/edu/Sensors_intro_sheet.pdf

Selon toi quels sont les capteurs à utiliser pour savoir si les astronautes de l'ISS sont en activité physique ?







Doc sur l'Astropi :

https://www.raspberrypi.org/magpi-issues/Essentials_SenseHAT_v1.pdf

Afficher un capteur

Pour afficher la valeur d'un capteur

```
from sense_hat import SenseHat

sense = SenseHat()
sense.clear()

temp = sense.get_temperature()
print(temp)
```

Tu peux alors voir l'affichage de ton information en bas à droite de l'IDE

Est-ce que l'affichage de la valeur de ton capteur te convient ?

Il est alors possible de modifier le nombre de chiffres significatifs après la virgules en utilisant la fonction round().

Tu peux modifier les paramètres de l'affichage : (attention à taper sur une seule ligne !)

```
sense.show_message("Astro Pi is awesome!!", scroll_speed=0.05,
text_colour=[255,255,0], back_colour=[0,0,255])
```

Afficher avec des chiffres significatifs

A l'aide de la documentation ci-après, recherche pour afficher correctement ton information avec un nombre de chiffres significatifs corrects.

https://docs.python.org/3/library/functions.html

```
from sense_hat import SenseHat

sense = SenseHat()



temperature = sense.temperature

sense.show_message("Temperature is %d" % temperature)
```

Afficher en degrés Kelvin

Modifier votre programme pour afficher la température en °C et °K. Les 2 informations seront affichées les unes en dessous des autres

Afficher en permanence

Tu vas maintenant réaliser un programme qui permet de faire une boucle infinie pour afficher en permanence un capteur.

Pour cela tu vas utiliser une boucle conditionnelle. La condition de la boucle sera toujours validée (vraie). True=Vrai et Fasle=Faux

while True :

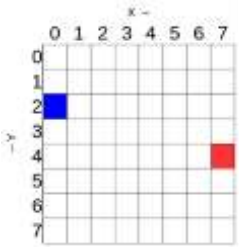
    Commandes….

Utilisation du pavé à LED

La visualisation de l'affichage des données capteurs n'est pas très pratique sur les leds, car les informations défilent.

Il y a une solution en utilisant les LED comme « bar_graphe ». Il suffit d'allumer plus ou moins de led en fonction de la température. On pourra même changer de couleur en fonction de la température.

Pour cela tu vas utiliser la méthode set_pixel()



Exemple de code pour afficher le pixel bleu :

sense.set_pixel(0, 2, [0, 0, 255])

Pour choisir une couleur il te faudra fixer 3 valeurs (R,G,B). Il est possible de les trouver à partir de la couleur que tu veux grâce au simulateur suivant :

http://www.colorspire.com/rgb-color-wheel/

Il est aussi possible de positionner l'ensemble des 8x8 pixels à partir d'un fichier image.

Tu peux créer rapidement ton fichier image à partir d'une application de gestion d'image comme :

https://pixlr.com/editor/

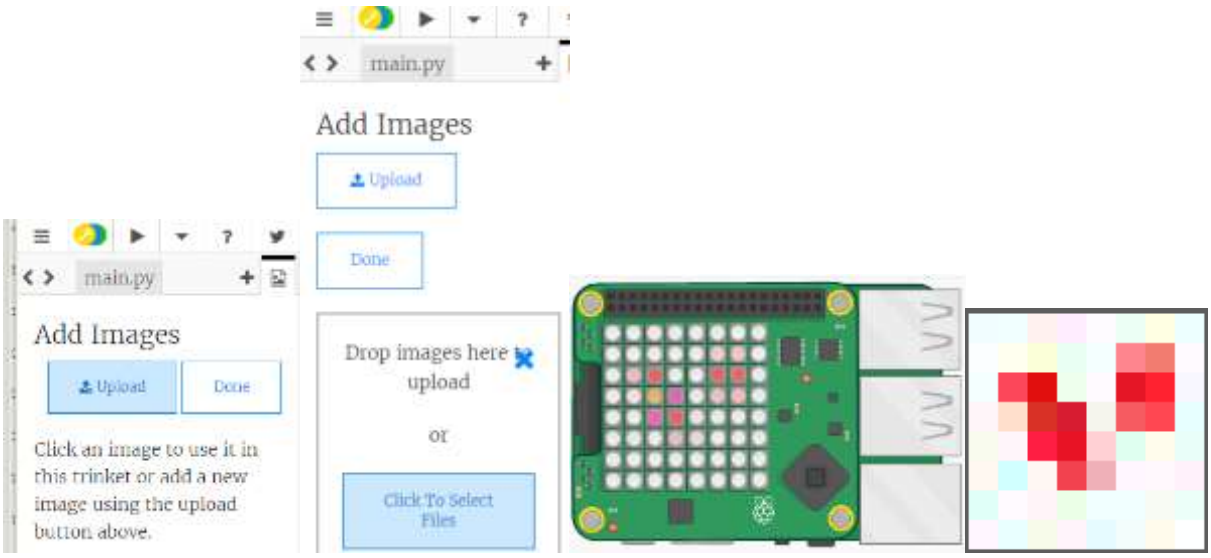
Sauvegarde ton fichier avec l'extension png ou jpeg

Puis utilise la méthode

```
sense.load_image(repertoire_acces/nom_fichier_image)
```

Dans le simulateur trinket.io

Puis téléverse ton fichier image



https://wiki.mchobby.be/index.php?title=RASP-SENSE-HAT-ASTRO-PI-Matrice

Affichage d'une ligne sur l'écran de l'Astropi

Réaliser une application qui permet d'allumer une ligne sur les led de l'astropi.

Pour cela tu utiliseras une boucle itérative.

Réalise ton algorithme et développe ton code associé en Python.

Tu utiliseras le document d'aide sur les structures algorithmiques pour Python.

Tu feras la même application, mais cette fois-ci avec une variable et une boucle conditionnelle. Il suffira d'incrémenter (variable ← variable + incrément) une variable.

Test conditionnel

Réalise une application qui permet d'afficher un pixel de couleur en fonction de la température.

Tu utiliseras les commandes de test conditionnel. Voir la documentation.

Faire rebondir un pixel

Fais un petit programme qui permet de faire déplacer un pixel latéralement, puis le faire rebondir sur un des bords. On démarrera le mouvement du déplacement latéral du pixel dès que l'on appuie sur le flèche vers le haut.

Pour cela il suffit d'allumer un pixel à x+1 et d'éteindre le pixel allumé à x.

Pour réaliser une pause de x seconde, tu importeras la bibliothèque time.

Puis tu utiliseras la méthode time.sleep(x), avec x en seconde.

Gestion de projet

Choisis un projet parmi la liste suivante : (travail en binôme)

Projet n°1 : Alerter l'équipage si la température n'est pas correcte ou s'il y a une dépressurisation ou si l'humidité n'est pas correcte. Attention de bien analyser le problème.

Rechercher les capteurs à utiliser.

Projet n°2 : compter le nombre de tours de la terre à l'aide du bon capteur.

Projet n°3 : détection d'un changement d'orbite. On rappelle que l'ISS est attirée par la terre. L'ISS descend donc inexorablement. On pousse alors l'ISS pour la remettre sur son orbite. Tu as vu que si la vitesse v de l'ISS augmente alors h va augmenter. Vérifie-le sur un exemple avec $\frac{G \cdot M}{(R+h)^2} = \frac{v^2}{(R+h)}$. En effet, M = constante, R = constante.

Projet n°4 : faire un jeu de pong avec une raquette

Projet n°5 : faire un jeu de pac-man

Projet personnel : à définir…

Enregistrer des valeurs dans un fichier

L'objectif est de sauvegarder des données dans un fichier à intervalles réguliers. Cela permettra ensuite de traiter des informations une fois l'expérience terminée.

Tu feras un premier essai pour sauvegarder 20 mesures de la température du sens hat. Tu afficheras alors les mesures au fur et à mesure. Rédige un petit algorithme.

On sauvegardera alors les données dans un fichier sur la carte mémoire Flash de l'Astropi.

Il faudra créer un fichier dans un répertoire précis dans la carte mémoire.

Il faudra permettre d'écrire des données dans ce fichier.

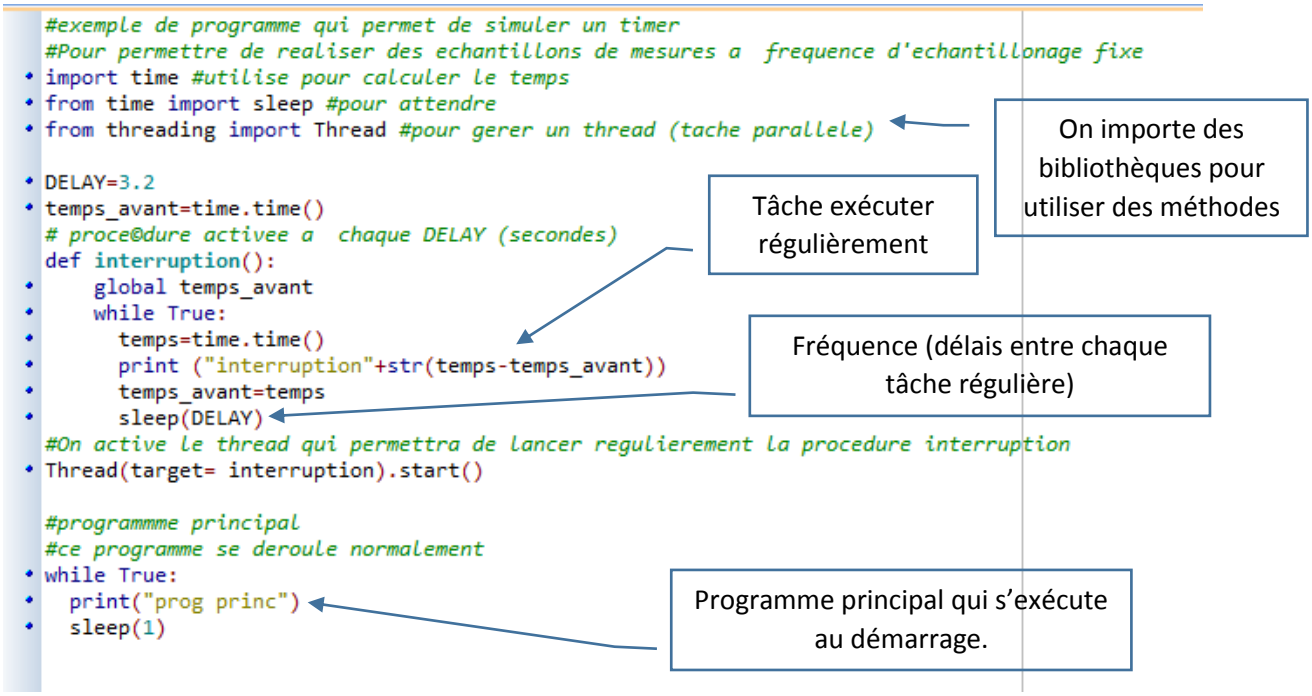
Puis il suffira d'utiliser une commande pour écrire une donnée.

Utilise pour cela la documentation sur le python fournie.

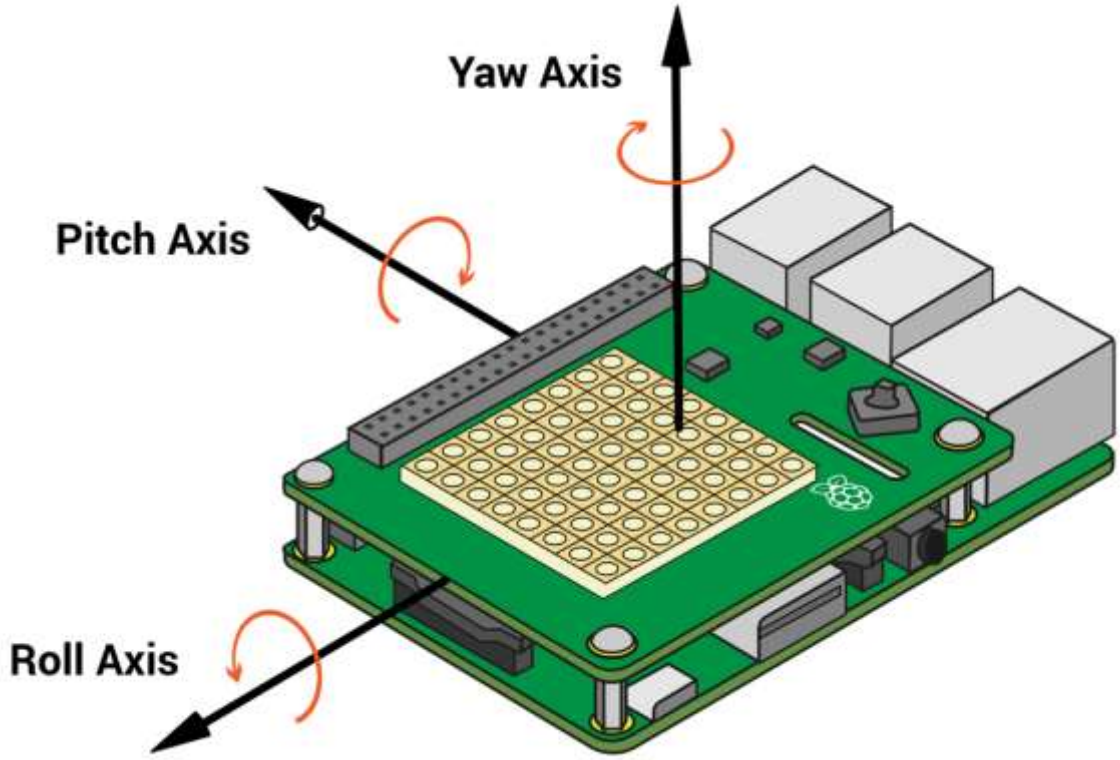
Tu liras ton fichier à l'aide d'un tableur, tu vérifieras les données de mesures.

Pour faire une attente de x secondes on importera la bibliothèque time et on utilisera la méthode time.sleep(x) (x en seconde).

Réaliser une tâche à intervalles réguliers



https://www.raspberrypi.org/learning/getting-started-with-the-sense-hat/worksheet/



Impression 3D

https://www.raspberrypi.org/learning/3d-printed-astro-pi-flight-case/worksheet/