

PARTIE 1 : BOUTONS

 1. RAPPEL IMPORTANT

Il y a **deux boutons A et B** sur la face avant de la carte Micro:bit.

On peut détecter quand ces boutons sont pressés, ce qui permet de déclencher des instructions sur l'appareil.

`button_a.is_pressed()` : renvoie **True** si le bouton A est **actuellement** enfoncé et **False** sinon

`button_a.was_pressed()` : renvoie **True ou False** pour indiquer si le bouton A a été appuyé depuis le démarrage de l'appareil ou la dernière fois que cette méthode a été appelée

`button_a.get_presses()` : renvoie le nombre de fois où on a appuyé sur le bouton A

Avant de commencer, ouvrir le logiciel *Mu* et connecter votre carte Micro:bit sur un port USB.

 2. EXERCICE 1 : PIXEL EN BALADE

```

1 from microbit import *
2
3 x = 0
4 y = 2
5 while True:
6     display.set_pixel(x,y,0)
7     if button_a.was_pressed():
8         x = x-1
9     if button_b.was_pressed():
10        x = x + 1
11    x = max(0, min(x, 4))
12    display.set_pixel(x,y,9)
13    sleep(20)

```

1. Flasher ce programme sur la carte puis expliquer ce qu'il fait.
2. Quel est le rôle de ligne 11 ? ma réponse :
3. Modifier ce programme pour que le pixel se balade cette fois-ci verticalement sur la colonne centrale
4. Sauver le programme sous le nom **balade.py**

DEPÔT 1

déposer balade.py sur <https://entraide-ella.fr> 3. EXERCICE 2 : CLIGNOTANTS

```

1 from microbit import *
2
3 while True:
4     if button_a.is_pressed():
5         display.show(...)
6         sleep(...)
7
8     if button_b.is_pressed():
9         ...

```

1. Compléter ce programme afin qu'il affiche une flèche clignotante vers la gauche lorsque le bouton A est enfoncé et une flèche clignotante vers la droite lorsque le bouton B est enfoncé.
2. Flasher le programme sur la carte et vérifier qu'il fonctionne.
3. Sauver le programme sous le nom **fleche.py**

DEPÔT 2

déposer fleche.py sur <https://entraide-ella.fr>

? 1. CAPTEUR DE LUMIÈRE

En inversant les LEDs d'un écran pour devenir un point d'entrée, l'écran LED devient un capteur de lumière basique, permettant de détecter la luminosité ambiante.

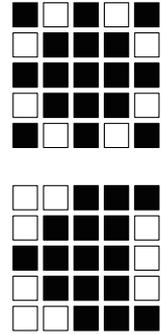
La commande `display.read_light_level()` retourne un entier compris entre 0 et 255 représentant le niveau de lumière que reçoit la carte

E 2. EXERCICE 3 : LE SOLEIL A RENDEZ-VOUS AVEC LA LUNE

```

1 from microbit import *
2
3 soleil = Image("90909:"
4               "09990:"
5               "99999:"
6               "09990:"
7               "90909")
8 lune = ....
9
10 while True:
11     if display.read_light_level()>... : # <--- valeur entre 0 et 255
12         display.show(soleil)
13     else:
14         display.show(...)
15         sleep(100)

```



1. Compléter le programme ci-dessus pour afficher une image de lune si on baisse la luminosité (en recouvrant la carte avec sa main par exemple) et un soleil sinon.
2. Modifier le programme pour qu'il affiche en défilement le niveau de luminosité ambiant après le symbole obtenu (soleil/lune).
3. Sauver le programme sous le nom `soleil.py`

DEPÔT 3

déposer `soleil.py` sur <https://entraide-ella.fr>

? 3. CAPTEUR DE TEMPÉRATURE

La carte Micro:bit na pas un capteur de température dédié. Au lieu de cela, la température fournie est en fait la température de la **puce de silicium du processeur principal**. Comme le processeur chauffe peu en fonctionnement (c'est un processeur ARM à grande efficacité), sa température est une bonne **approximation** de la température ambiante.

L'instruction `temperature()` renvoie la température de la carte Micro:bit en degrés Celsius.

E 4. EXERCICE 4 : THERMOMÈTRE DE SECOURS

1. Compléter le programme suivant afin qu'il affiche la température en défilement.

```

1 from microbit import *
2
3 while True:
4     t= temperature()
5     ...
6     sleep(1000)

```

2. Sauver le programme sous le nom `temperature.py`

DEPÔT 4

déposer `temperature.py` sur <https://entraide-ella.fr>

5. CAPTEUR DE MOUVEMENT

Un **accéléromètre** mesure l'accélération de la carte Micro:bit, ce composant détecte quand la carte est en mouvement. Il peut aussi détecter d'autres actions (gestes), par exemple quand elle est secouée, inclinée ou qu'elle tombe.

Des accéléromètres sont présents dans les smartphones et les manettes de jeux afin de pouvoir afficher l'image dans le bon sens, ou bien de changer de direction dans un jeu.

L'accéléromètre mesure le mouvement selon trois axes :

X - linclinaison de gauche à droite

Y - linclinaison devant en arrière

Z - le mouvement haut et bas

L'instruction `accelerometer.get_x()` permet de détecter un mouvement de gauche à droite en renvoyant un nombre compris entre -1023 et 1023 (0 étant la position "d'équilibre").

L'instruction `accelerometer.is_gesture('shake')` renvoie **True** ou **False** selon si la carte est secouée.

6. EXERCICE 5 : EN HAUT, EN BAS, À GAUCHE, À DROITE

```
1 from microbit import *
2
3 while True:
4     abscisse = accelerometer.get_x()
5     if abscisse > 500:
6         display.show(Image.ARROW_E)
7     elif abscisse < -500:
8         display.show(Image.ARROW_W)
9     else:
10        display.show("-")
```

1. Flasher ce programme sur la carte. Expliquer ce qu'il fait.

.....

.....

.....

2. Compléter le programme pour qu'il affiche une flèche vers le haut si on incline la carte en avant et une flèche vers le bas si on l'incline en arrière.
3. Sauver ce programme sous le nom **mouvement.py**

DEPÔT 5

déposer mouvement.py sur <https://entraide-ella.fr>

7. EXERCICE 6 : LANCER DE DÉ

1. Écrire un programme qui simule un dé en affichant une face au hasard lorsque la carte Micro:bit est secouée. On rappelle que l'instruction **random.randint(1,6)** renvoie un entier au hasard entre 1 et 6

```
1 from microbit import *
2 import random
3
4 while True:
5     ...
```

2. Sauver ce programme sous le nom **de.py**

DEPÔT 6

déposer de.py sur <https://entraide-ella.fr>