



## Unité 7 : micro:bit avec Python

## Compétence 1 : L'affichage

Dans cette leçon, vous allez écrire votre premier programme Python pour contrôler l'affichage de la carte micro:bit de différentes manières.

Cette leçon est composée de deux parties :

Partie 1 : rencontre extraterrestre.

Partie 2 : affichage d'images.

### Objectifs :

- Contrôler l'affichage sur la carte micro:bit à l'aide `.show()`, `.scroll()` et `.show(image)`.
- Contrôler la vitesse de l'écran à l'aide de `sleep(ms)`

1. Avant de commencer, assurez-vous que :

- votre TI-83 Premium CE dispose de l'OS **5.6.1** ou supérieur
- vous êtes à l'aise avec la programmation en Python et/ou vous avez terminé les unités 1 à 5.
- votre **micro:bit** est connectée à votre TI-83 Premium CE
- vous avez suivi les instructions de configuration et les transferts de fichiers à l'aide du guide de mise en route micro:bit :

<https://education.ti.com/fr/enseignants/microbit>



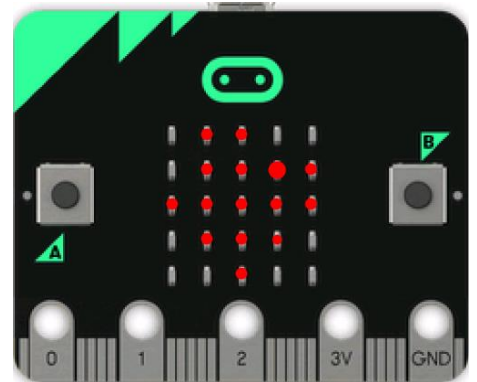
Rappel : les Modules sont à télécharger à l'adresse :

<https://education.ti.com/fr/enseignants/microbit/microbit-83ce>

*Ce processus de configuration ne devrait être effectué qu'une seule fois, mais tenez-vous informé périodiquement des mises à jour.*

2. Si tout est correct et que la configuration a été réalisée avec succès, votre micro:bit devrait afficher l'image ci-contre lorsqu'elle est connectée à la TI-83 Premium CE.

L'affichage sur la carte **micro:bit** montre le logo TI, une icône de l'État du Texas avec un point lumineux près de Dallas, le siège de **Texas Instruments, Inc.**



3. Et enfin...

Les modules micro:bit sont installés dans votre calculatrice. Dans un éditeur Python, appuyer sur **2<sup>nde</sup>** **catalog** puis sur la lettre **F**. Utiliser ensuite la touche de direction vers le bas afin de trouver l'instruction « from **SCRIPT** import \* ».

Compléter l'instruction en écrivant « **from microbit import \*** ».



EDITEUR : SB1  
LIGNE DU SCRIPT 0001  
from \_import \*

Fns... a A # Outils Exéc Script

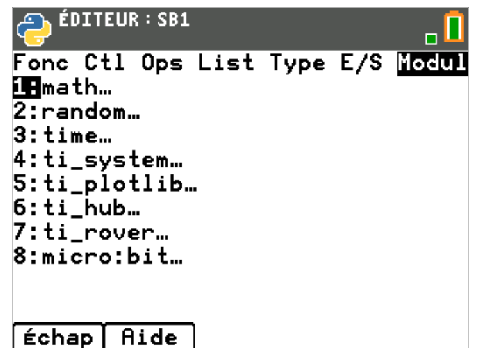
Appuyer sur la touche **F1** (Fns) puis ← afin de constater la présence du module **micro:bit** au côté des autres modules.

Les différentes librairies (modules : capteurs, boutons, affichage, radio...) du micro:bit seront accessibles par l'intermédiaire du menu **8 : micro:bit**. Une fois chargées, les instructions relatives à chaque module seront disponibles dans leur menu dédié.



EDITEUR : SB1  
LIGNE DU SCRIPT 0002  
from microbit import \*

Fns... a A # Outils Exéc Script



ÉDITEUR : SB1

Fonc	Ctl	Ops	List	Type	E/S	Modul
1:						math...
2:						random...
3:						time...
4:						ti_system...
5:						ti_plotlib...
6:						ti_hub...
7:						ti_rover...
8:						micro:bit...

Échap Aide

**Conseil de l'enseignant : Mise en route avec micro:bit**

Cette unité suppose que le micro:bit et la calculatrice TI-83 Premium CE EDITION PYTHON soient « prêts à l'emploi ». Assurez-vous que vous (ou vos élèves) avez terminé les étapes de configuration sur leurs calculatrices individuelles et micro:bits. Assurez-vous de lire et d'exécuter les instructions de configuration à l'aide du guide « Mise en route » qui est inclus dans le dossier de téléchargement à partir du site Web TI Education. <https://education.ti.com/fr/enseignants/microbit>.

Comme pour tous nouveaux logiciels et matériels, veuillez rester informé des mises à jour.

Il y a aussi un document d'introduction TI-83 Premium CE, **PROG1.8xv**, qui s'assure que la carte micro:bit est connectée et fonctionne correctement.





Le module **BBC micro:bit**, et les modules associés, sont des modules spéciaux fournis par Texas Instruments qui sont installés dans chaque calculatrice comme des variables d'application. Pour constater leur présence dans la calculatrice de vos élèves, Appuyez sur **2<sup>nde</sup> mém** puis choisir **2 : Gest. Mémoire/Suppr...** et enfin **B : Var App** (ou utilisez les touches de direction vers le bas afin d'atteindre cette instruction).

```
NORMAL FLOTT AUTO RÉEL RAD MP
MÉM RAM LIB 143271
MÉM ARCH LIB 1226K
6↑Var Y...
7:Prgm...
8:Pic et image...
9:BDG...
0:Chaîne...
A:Apps...
B:Var App...
C:Grouper...
```

```
NORMAL FLOTT AUTO RÉEL RAD MP
MÉM RAM LIB 143271
MÉM ARCH LIB 1226K
*MB_BUTNS 1626
*MB_DISP 4670
*MB_GROVE 3936
*MB_MUSIC 2493
*MB_NEOPX 1979
*MB_PINS 2134
*MB_RADIO 2632
▶*MB_SENSR 4507
```

**Les cinq premières unités du programme Python : 10 Minutes de Code doivent être réalisées avant cette unité.** Le codage présenté ici n'est pas compliqué, mais vous devez posséder une bonne maîtrise du codage Python sur la calculatrice TI-83 Premium CE EDITION PYTHON.

Cette unité est conçue pour fonctionner avec les versions micro:bit 1 et 2, mais ne traite pas des fonctionnalités uniques à la version 2. Le module BBC micro:bit prend en charge ces fonctionnalités.

**Tous les projets micro:bit trouvés en ligne (y compris les projets micro:bit version 2 ) peuvent être développés sur la TI-83 Premium CE en utilisant Python et un micro:bit attaché. Mais il y a deux distinctions importantes :**

- Vous ne pouvez pas déconnecter le micro:bit de la calculatrice pendant que le programme est en cours d'exécution, même si une batterie est connectée au micro:bit. Le runtime micro:bit (fichier .hex) est configuré pour communiquer avec la calculatrice pour ses instructions. La calculatrice alimente et contrôle le micro:bit.
- La plupart des programmes de démonstration trouvés en ligne utilisent l'instruction **While True:** pour créer une boucle infinie. Cette boucle s'exécute directement sur la carte micro:bit jusqu'à ce que le programme soit remplacé par un autre à l'aide d'un fichier .hex téléchargé. Lors de l'utilisation de la TI-83 Premium CE, la calculatrice contrôle le micro:bit de sorte que la boucle la plus couramment utilisée est « **While not escape( )** »: cela permet à l'utilisateur d'appuyer sur la touche **[annul]** pour mettre fin au programme.

Ce document est mis à disposition sous licence Creative Commons



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

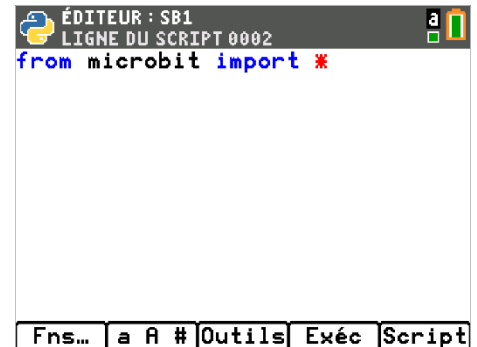


4. **Partie 1 - rencontre extraterrestre** : Bien sûr, comme avec toute autre première expérience de programmation, vous commencerez par afficher un message sur l'écran de la carte micro:bit.

Démarrer un nouveau script Python ; le nommer « **SB1** ». Dans l'éditeur Python appuyer **2<sup>de</sup>** **catalog** puis **F** pour sélectionner l'instruction « **from SCRIPT import \*** ». Valider en appuyant sur la touche **[entrer]** et compléter manuellement l'instruction :

**from microbit import \***

*Conseil : Si le message « micro:bit not connected » apparaît, débranchez simplement le micro:bit et rebranchez-le (reset)*



**Conseil de l'enseignant** : Si un micro:bit n'est pas connecté, l'instruction **import** elle-même ne signalera pas une erreur sur la TI-83 lors de l'exécution du programme. Étant donné que Python est un langage « Modulaire », des fonctionnalités supplémentaires doivent être ajoutées si nécessaire. Cette instruction d'importation charge des fonctions/méthodes spécifiques requises pour faire fonctionner le micro:bit à l'aide de la TI-83 Premium CE. Mais attention, certaines méthodes utiles à partir d'autres modules standards doivent être chargés manuellement. Par exemple, pour faire fonctionner des capteurs ou avoir accès aux fonction de l'affichage des DEL de la matrice de la carte BBC micro:bit. Pour charger un module spécifique appuyer sur :

**Fns** puis ← **Modul** et enfin **8 : micro:bit..**



5. Pour afficher un texte message sur l'affichage micro:bit, utilisez l'instruction « **display.show(image ou texte)** », après avoir chargé le module **affichage**, que l'on trouve dans le module général **micro:bit**.

Celle-ci se trouve sur :

**[Fns] < Modul 9 : affichage > 2 : .show(val)**

L'instruction est insérée en tant que **display.show(image ou texte)**, mais (image ou texte) est juste un espace réservé qui doit être remplacé par quelque chose. À l'intérieur des parenthèses, remplacez l'image ou le texte en tapant la chaîne de votre message entre guillemets :

Ce document est mis à disposition sous licence Creative Commons

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>





### « salutations terriens »

Lorsque vous exécutez ce programme en appuyant sur **F4 [Exec]**, vous verrez les lettres de votre message apparaître, une lettre à la fois, sur l'écran. Si votre message est composée d'une répétition successive de la même lettre, vous ne pouvez pas en distinguer deux.

*Si vous faites une erreur... Editez à nouveau le script pour le modifier après avoir interrompu son fonctionnement [annul], puis exécutez à nouveau le programme. Avez-vous oublié d'ajouter des guillemets autour du texte que vous vouliez afficher ?*

*Remarque : l'instruction « display.show(val, del=400, wait=True) » comporte des paramètres optionnels (delay=400 en ms, et wait) dont l'utilisation n'est pas utile dans cette leçon. Ils peuvent donc être effacés en utilisant la touche [suppr].*

6. Une autre méthode pour afficher les messages est :

#### **display.scroll(« salutations terriens »)**

qui se trouve également sur [Fns] < Modul 9 : affichage > 2 : .show(val).

Pour commenter la ligne précédente, placer le caractère # devant la ligne à commenter.

Faites de l'instruction .show( ) précédente un #commentaire (placez le curseur devant la ligne à commenter et appuyer sur 2<sup>nde</sup> 3, puis exécutez à nouveau le programme).

*Vous pouvez également simplement remplacer .show par .scroll en tapant l'instruction.*

```

ÉDITEUR : SB1
LIGNE DU SCRIPT 0005
from microbit import *
from mb_disp import *
#display.show("salutations terriens")
display.scroll("salutations terriens")

```

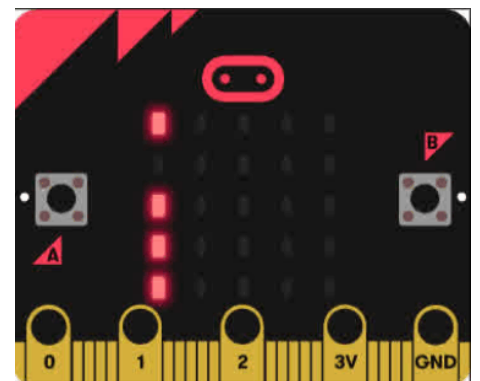
Fns... | a A # | Outils | Exéc | Script

7. La méthode .scroll( ) fait défiler le message de droite à gauche comme une bannière à travers l'affichage, ce qui le rend plus facile à lire.

Vous pouvez contrôler la vitesse du défilement en ajoutant le paramètre **delay = :**

#### **display.scroll(« salutations terriens », delay = 200)**

Cette instruction utilise un délai de 200 millisecondes (0,2 seconde) dans le défilement. Essayez également d'autres valeurs de délai.



« salutations.gif »

Ce document est mis à disposition sous licence Creative Commons

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>





## 8. Partie 2 - Battements du cœur

Appuyer sur **[F5] [script]** puis **[F3] [Nouv]** pour créer un nouveau script et le nommer « **COEUR** ».

Dans l'éditeur Python, utiliser **2<sup>nde</sup> catalog** puis **from SCRIPT import** et compléter l'instruction :

```
from microbit import *
```



9. Charger le Module d'affichage **[Fns] < Modul 8 : micro:bit > 1 : affichage**.

10. Pour afficher le symbole HEART sur l'affichage micro:bit, utilisez l'instruction :

```
display.show( )
```

Cette instruction se trouve sur :

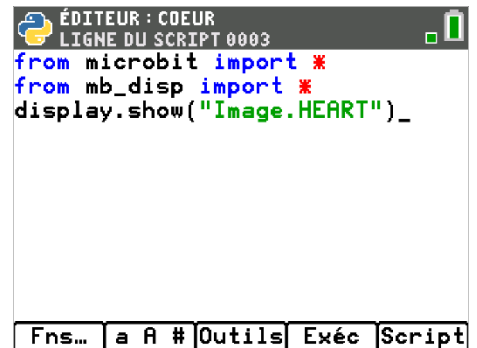
```
[Fns] < Modul 9 : affichage > 2 : .show(val)
```

Entre parenthèses, remplacez l'invite en sélectionnant :

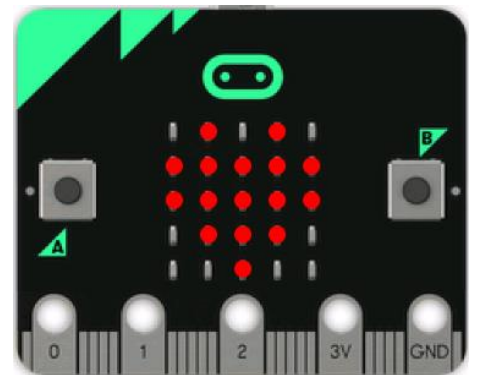
```
Image.HEART
```

depuis

```
[Fns] < Modul 9 : affichage > Image > 2 : HEART
```



11. Exécuter le programme (appuyez sur **[F4] [Exec]**) pour voir l'icône HEART affichée sur la grille LED 5x5 de la carte micro:bit. Cet affichage restera jusqu'à ce qu'un autre prenne sa place, même après la fin de l'exécution du programme.





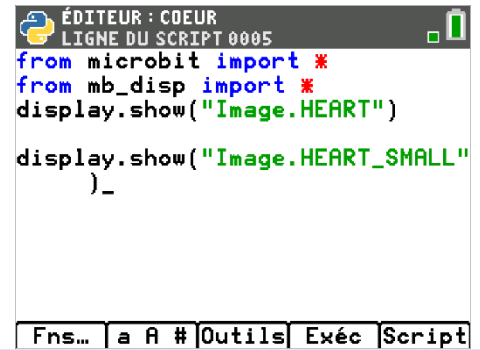
12. Revenir à l'éditeur de programmes sur la page précédente, et ajouter une autre instruction d'affichage pour montrer le petit cœur :

```
display.show(Image.HEART_SMALL)
```

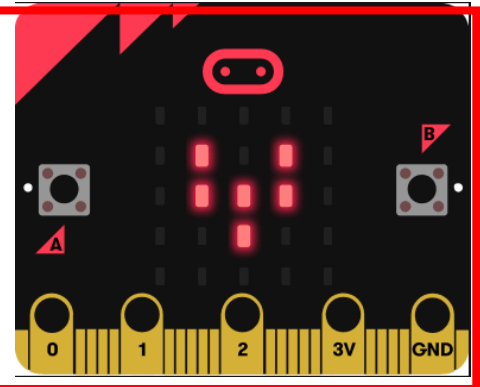
Vous pouvez trouver cette image dans le même menu :

```
[Fns] < Modul 9 : affichage > Image > 3 : HEART SMALL
```

*Astuce : vous pouvez également copier / coller la première instruction d'affichage et la modifier (insérez le **\_SMALL**).*



13. Exécuter à nouveau le programme. Il affiche rapidement le grand cœur, puis affiche le petit cœur qui ressemble à l'image ci-contre.



14. **Faire une boucle** : Pour que les deux cœurs clignotent à plusieurs reprises (« battements »), placez les deux instructions d'affichage dans une boucle. *Avant* les deux instructions d'affichage, insérez :

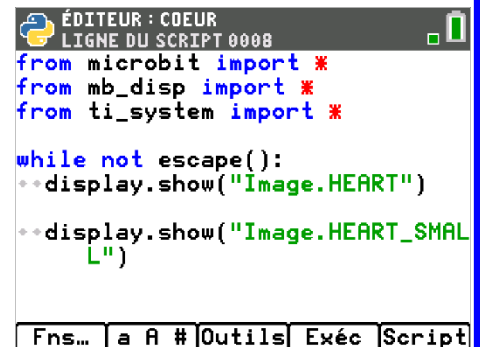
```
while not escape() :
```

trouvé dans

```
[Fns] < Modul > 4 : ti_system > 5 : while not escape() :
```

N'oubliez pas de charger préalablement le module **ti\_system** et mettez en retrait (indentation) les deux instructions d'affichage, afin qu'elles forment le corps de la boucle.

*Conseil important : L'indentation est essentielle dans les programmes Python. C'est ainsi que Python interprète les blocs de boucle et les blocs if. Si les deux instructions d'affichage ne sont pas mises en retrait la même nombre d'espaces, vous verrez une erreur de syntaxe. Utilisez la touche [F3] [Outils] puis 1 : Indent > pour mettre en retrait les deux lignes de la même quantité. Les espaces d'indentation sont indiqués dans l'éditeur par un symboles de diamant gris clair (◆◆) afin de faciliter une indentation correcte.*



Ce document est mis à disposition sous licence Creative Commons

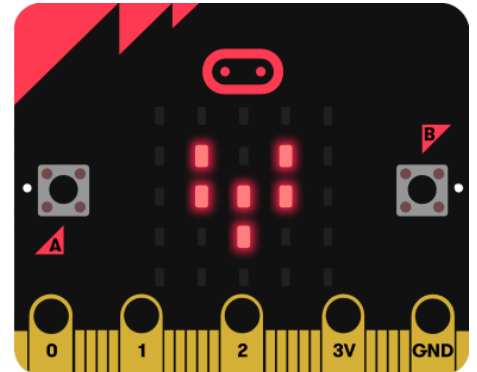


<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>



15. Exécuter à nouveau le programme et regarder le cœur battre ! Appuyer sur la touche **[esc]** pour terminer le programme.

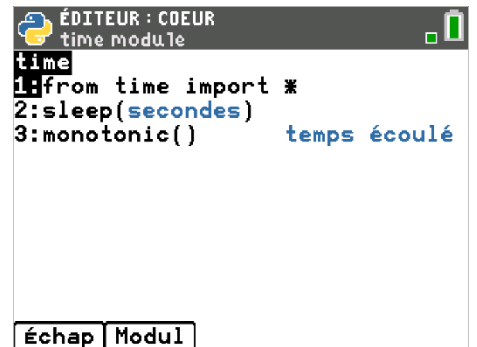
*Astuce : si jamais vous pensez que votre programme est bloqué dans une boucle infinie, appuyez et maintenez la touche **[on]** sur votre TI-83 Premium CE pour « casser » le programme. Cela peut se produire si vous l'utilisez alors que l'instruction **True**: est incorrectement placée dans le script. Ces leçons évitent ce type de structure.*



<cœur .gif>

16. Le menu **time** micro:bit contient des instructions Python utiles que l'on trouve également dans d'autres menus. Le module micro:bit importe ces commandes Python pour vous.

Vous pouvez trouver ces commandes et toutes les autres commandes Python à partir d'autres menus. Vous n'êtes pas limité à utiliser simplement le menu BBC micro:bit, mais vous devrez peut-être fournir les commandes d'importation appropriées.

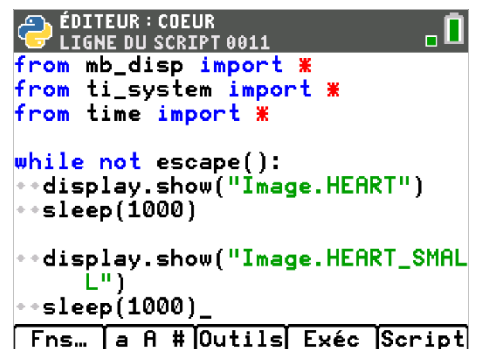


17. Pour contrôler la fréquence des battements cardiaques, ajoutez deux instructions **sleep()**, une après chaque instruction **d'affichage** : **sleep(1000)** signifie 1000 milli secondes ou un délai de 1 seconde.

Aussi trouvé sur

**[Fns] < Modul > 3 : time.. > 2 : sleep()**

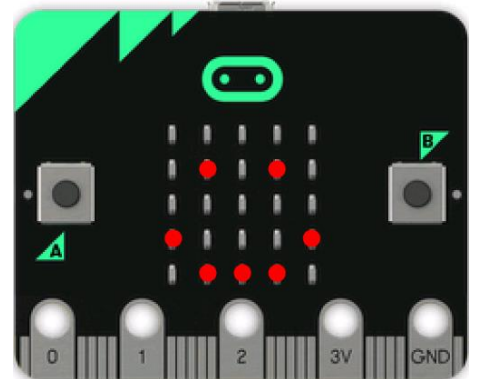
*Astuce : observez l'indentation !*







18. **Extension** : Essayer de faire une « série de visages » ou smiley. Utiliser une structure de programme similaire à celle du script « **Battements du coeur** », mais sélectionner plutôt les images de visages trouvées dans l'ensemble des images.



**Conseil de l'enseignant** : Les leçons micro:bit en ligne utilisent généralement une boucle **While True: en Python et** une boucle **forever** : dans MakeCode. Ces « boucles infinies » s'exécutent directement sur le micro:bit jusqu'à ce qu'il soit désactivé ou remplacé par un autre programme (fichier .hex). Lorsque vous utilisez la TI-83 Premium CE pour contrôler le micro:bit, la boucle infinie n'est pas nécessaire puisque la calculatrice est en plein contrôle. **While True: se trouve dans le menu Fns puis Ctl, mais doit être utilisé conjointement avec if (Expression booléenne) : break** pour sortir de la boucle infinie.

**Si un élève est bloqué dans une boucle infinie sur la calculatrice, pour « casser » le programme, appuyez et maintenez la touche [on].** Consultez le guide « Mise en route » (PDF) fourni avec le logiciel micro:bit de TI, pour davantage d'informations sur la rupture des programmes Python sur les ordinateurs et calculatrices.

Les menus contiennent un ensemble de commandes Python courantes et fréquemment utilisées qui sont utiles lors du codage pour micro:bit. Les modules micro:bit importent ces commandes et doivent être chargés au début du script après le module micro :bit. Mais toutes les commandes Python sont disponibles dans d'autres menus et peuvent être utilisées dans des projets micro:bit. L'utilisation des instructions d'importation appropriées est importante.

*Facultatif* : Pour récupérer l'affichage de son état « TI » (le logo TI du Texas) à la fin d'un programme, ajoutez l'instruction

```
display.show( ti )
```

après la fin de la boucle While. Cette instruction doit être désindentée.

*Ce n'est pas nécessaire, mais cela permet d'indiquer que le programme est terminé sur l'écran micro:bit ainsi que sur l'écran.*

*Au début d'un programme micro:bit (avant le début de la boucle while), il peut être utile d'imprimer quelque chose sur l'écran de la calculatrice, tel que **print(« running... »)**.*





À propos de `sleep()` : La seule fonction Python qui est considérablement modifiée pour le micro:bit est l'instruction `sleep()` trouvée dans le module `time`. Normalement, la valeur de l'argument représente des « secondes », mais le module `micro:bit` modifie l'instruction `sleep` afin que l'argument représente un temps en millisecondes (*millièmes* de seconde). Après l'importation du module `micro:bit`, `sleep(1000)` représente un délai d'une seconde. Notez que `sleep()` contrôle la vitesse du programme de la calculatrice, pas le micro:bit lui-même. Assurez-vous que l'importation de `micro:bit` se produit après le moment de l'importation (ou tout module qui pourrait importer du temps).

La méthode `.show()` a deux paramètres facultatifs : `delay=` et `wait=`  
`display.show(valeur, delay =xxx , wait = False/True)`

- la valeur du `delay` est en millisecondes et ralentit l'affichage ;
- `wait = True` bloquera le traitement jusqu'à ce que l'animation soit terminée, sinon l'animation se produira en arrière-plan.

Voici la liste complète des images :

Un projet similaire est « Série de visages » en utilisant la même structure de programme que « Battements du cœur » mais utilisez les icônes de visage à la place des cœurs.

```
ÉDITEUR : DE
affichage Image ver
1: var=Image(chaine)
2: HEART
3: HEART_SMALL
4: HAPPY
5: SMILE
6: SAD
7: CONFUSED
8: ANGRY
9: ASLEEP
0: SURPRISED
Échap Modul
```

```
ÉDITEUR : DE
affichage Image ver
A: SILLY
B: FABULOUS
C: MEH
D: YES
E: NO
F: TRIANGLE
G: TRIANGLE_LEFT
H: CHESSBOARD
I: DIAMOND
J: DIAMOND_SMALL
Échap Modul
```

```
ÉDITEUR : DE
affichage Image ver
K: SQUARE
L: SQUARE_SMALL
M: RABBIT
N: COW
O: MUSIC_CROCHET
P: MUSIC_QUAVER
Q: MUSIC_QUAVERS
R: PITCHFORK
S: XMAS
T: PACMAN
Échap Modul
```

```
ÉDITEUR : DE
affichage Image ver
Q: MUSIC_QUAVERS
R: PITCHFORK
S: XMAS
T: PACMAN
U: TARGET
V: TSHIRT
W: ROLLERSKATE
X: DUCK
Y: HOUSE
Z: TORTOISE
Échap Modul
```

Ce document est mis à disposition sous licence Creative Commons



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>