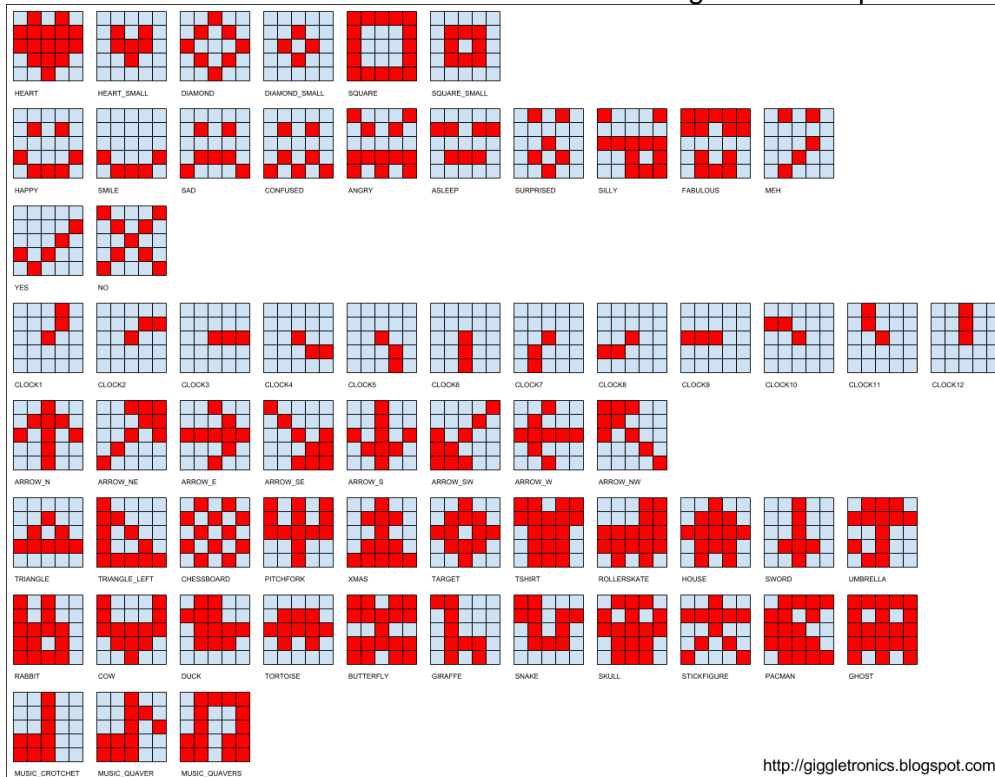


TP SUR LES LISTES

I. Liste d'images

```
from microbit import *  
  
display.show(Image.SAD)
```

1. Dans le logiciel *Mu*, écrire le programme ci-dessus puis flasher le programme sur la carte. Observer. Il s'agit d'une des images intégrées contenues dans la bibliothèque MicroPython. On constate que la carte est un peu triste . . .
2. En s'aidant de la liste des images intégrées suivantes, modifier le programme précédent pour lui redonner de la joie !
3. Créer un défilement sans fin d'une liste d'images choisies parmi les images prédéfinies.



II. Moyenne et médiane

Voici quelques fonctions de la bibliothèque « random » de Python utilisable en microPython.

```
random.random()
```

Return the next random floating point number in the range [0.0, 1.0)

```
random.uniform(a, b)
```

Return a random floating point number N such that $a \leq N \leq b$ for $a \leq b$ and $b \leq N \leq a$ for $b < a$.

<https://microbit-micropython.readthedocs.io/fr/stable/random.html>

Pour trier une liste dans l'ordre croissant en Python, vous pouvez utiliser :

- soit la fonction `sorted` qui renvoie une nouvelle liste triée : `sorted(ma_liste)`
- soit la méthode `sort` qui modifie la liste elle-même : `ma_liste.sort()`

<https://docs.python.org/fr/3/howto/sorting.html>

- 1) Créer un programme qui génère une liste de 5 flottants compris entre 1 et 10, les affiche puis affiche la moyenne de ces nombres.
Enregistrer ce programme sous le nom « moyenne ».
- 2) Créer un programme qui génère une liste de 5 ou 6 flottants compris entre 1 et 10, les affiche puis affiche la médiane de ces nombres.
Enregistrer ce programme sous le nom « médiane ».

Avec l'affichage défilant, il n'est pas facile de comprendre les nombres utilisés.

On peut demander l'affichage de la console micro :bit en cliquant sur le bouton « REPL » abréviation de « *Read-Evaluate-Print-Loop* »

Mais cliquer sur REPL crée une interruption clavier . Pour y remédier, il faut faire Ctrl+D (soft reboot) pour redémarrer le programme.

La commande « print » dans un programme créera un affichage dans la console.

- 3) Modifier les programmes précédents pour obtenir en plus un affichage de liste puis de la moyenne ou de la médiane dans la console. Enregistrer ces programmes.

III. Capteur de distance

A. Installation

- 1) Brancher le capteur ultrason sur la carte moteur du robot avec :
 - la broche 3 sera sur la sortie TRIG
 - la broche 4 sur la sortie ECHO
 - la sortie GND sur GND
 - la sortie VCC sur 3V3
- 2) Vérifier que le commutateur de la carte est sur 3.3V et non 5V.
- 3) Brancher votre carte microbit sur la carte du robot.

B. Premier programme

Charger le programme « BN_05_mesure_distance.py » sur Mu.

Analyser le code de ce programme : que permet-il de faire ?

C. Tests

- 1) Tester le programme et le capteur.
- 2) Modifier le programme afin de créer une liste de 5 valeurs puis d'afficher la moyenne de ces 5 valeurs sur la console. Enregistrer ce programme sous « mesure_distance_liste_moyenne ».
- 3) Modifier le programme afin de créer une liste de 5 valeurs puis d'afficher la médiane de ces 5 valeurs sur la console. Enregistrer ce programme sous « mesure_distance_liste_mediane ».
- 4) Créer un programme qui enregistre la distance chaque seconde et qui affiche, dans la console, la moyenne dès qu'il y a eu 5 mesures. Enregistrer ce programme sous « mesure_distance_auto_liste_moyenne ».
- 5) Créer un programme qui enregistre la distance chaque seconde et qui affiche, dans la console, la médiane dès qu'il y a eu 5 mesures. Enregistrer ce programme sous « mesure_distance_auto_liste_mediane ».