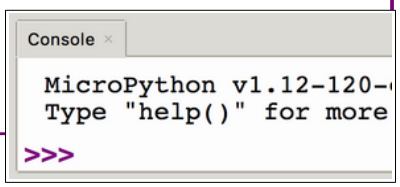
Objectif : Faire le lien entre période / fréquence et hauteur d'un son.

**Rappel : Il faut établir la connexion avec la carte microcontrôleur ESP32 – Soprolab -**
MENU : => Exécuter
=> sélectionner l'interpréteur … → microPython ESP32
=> Port → Silicon Lab CP210x ...




## 1ere étape : Produire un son

Nous allons utiliser le buzzer connecté à la broche n°25.

Comme on a pu le voir, un son ne peut être produit que par un dispositif vibrant.
On peut donc faire vibrer la membrane du buzzer :
- si la broche est à `on()` , la membrane se soulève,
- si la broche est à `off()` , la membrane reprend sa position de repos.

On va donc alterner les commandes `on()` et `off()` pour obtenir une vibration.

**Important : la somme** des deux durées d'attente : en position on() puis en position off() **détermine le temps total pour une vibration : c'est la période du son.**
La période se note T, l'unité est la seconde : **T(s)**. Le son est un phénomène périodique

```
# Charger la configuration des broches en mémoire
# Charger la fonction d'attente en microsecondes

# Déclarer une broche 25 en sortie

# Faire 200 fois :
    # Mettre l'état de la broche à on ( )
    # Attendre 2000 µs
    # Mettre la broche à off ( )
    # Attendre 2000 µs
```

```
from machine import Pin
from time import sleep_us

buzzer = Pin ( 25, Pin.OUT )

for i in range ( 200 ):
    buzzer.on ( )
    sleep_us ( 2000 )
    buzzer.off ( )
    sleep_us ( 2000 )
```

Les instructions ci-dessus produise un son.

Quelle est la période du son produit ?

**Important : l'inverse** de la période s'appelle la fréquence d'un son.
**La fréquence se note F et son unité est le Hertz ( Hz )**
La fréquence correspond au nombre de périodes par seconde.

$$F_{(Hz)} = \frac{1}{T_{(s)}}$$

Calculer la fréquence du son produit :

**<u>Sans faire d'essais</u>**, uniquement à partir de la définition de la fréquence, **si la période diminue, en déduire l'évolution de la fréquence : va-t-elle augmenter ou diminuer ?** *(justifier)*

## 2eme étape : Modifier la période / fréquence / durée d'un son

Quelle devra être la durée d'attente en micro secondes pour chaque temporisation `sleep_us( ? )` pour que la période d'une vibration soit divisée par 4 ?

Quelle sera alors la fréquence du son ?

**Faire vérifier vos calculs avant de passer aux essais ...**

```
# Charger la configuration des broches en mémoire
# Charger la fonction d'attente en microsecondes

# Déclarer une broche 25 en sortie

# Faire 200 fois :
    # Mettre l'état de la broche à on ( )
    # Attendre ??? µs
    # Mettre la broche à off ( )
    # Attendre ??? µs
```

```
from machine import Pin
from time import sleep_us

buzzer = Pin ( 25, Pin.OUT )

for i in range ( 200 ):
    buzzer.on ( )
    sleep_us ( ??? )
    buzzer.off ( )
    sleep_us ( ??? )
```

Le son produit est-il plus aigüe ou plus grave ?

**La caractéristique d'un son : aigüe # grave s'appelle la hauteur du son.**
=> Quelle relation peut-on établir entre la fréquence et la hauteur d'un son ?

Dans les deux cas, le buzzer produit 200 vibrations.
Expliquer pourquoi le son aigüe dure moins longtemps que le son grave. (justifier)

On souhaite produire un son de 2000 Hz. **Quelle doit être la période T** du son ?
En déduire **la durée de temporisation** `sleep_us( ? )` entre `on( )` et `off( )`.
**Combien de vibrations** doit-on programmer pour que la durée du son soit de 0,5 s ?

On modifie le code avec `sleep_us( 125 )` et 600 vibrations.
**Quelle sera la fréquence et la durée du son produit** ?

***Quel effet sonore va produire le code suivant ?***

***Indiquer dans les cadres ce que va produire chaque bloc d'instruction***

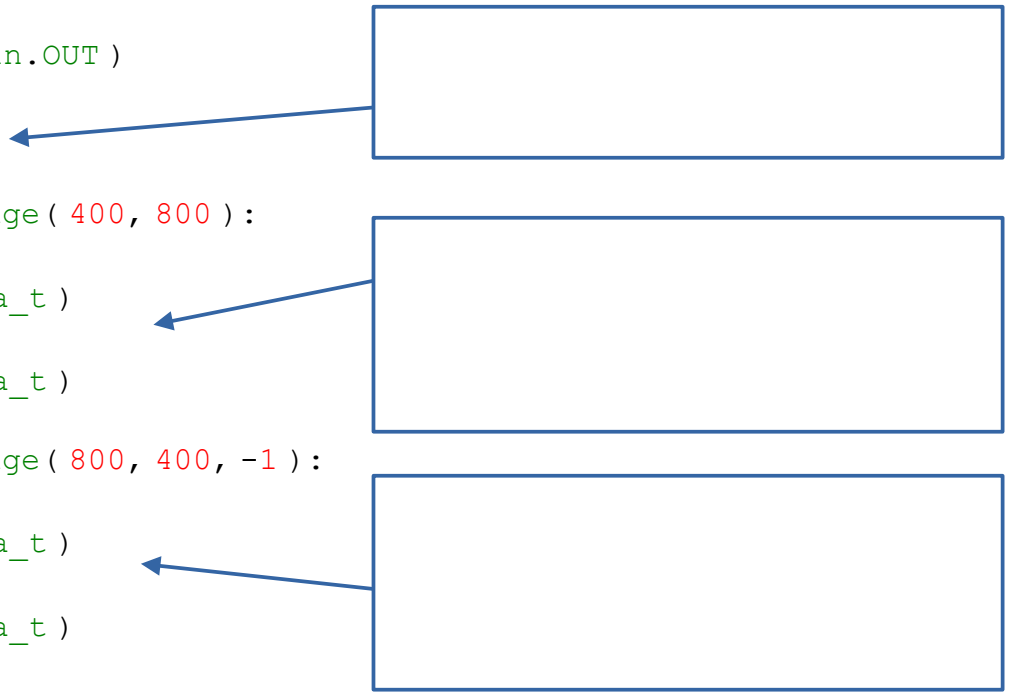
```
from machine import Pin
from time import sleep_us

buzzer = Pin ( 25, Pin.OUT )

for n in range(3) :

  for delta_t in range( 400, 800 ):
    buzzer.on()
    sleep_us ( delta_t )
    buzzer.off()
    sleep_us ( delta_t )

  for delta_t in range( 800, 400, -1 ):
    buzzer.on()
    sleep_us ( delta_t )
    buzzer.off()
    sleep_us ( delta_t )
```



***Après avoir fait vérifier vos hypothèses, vous pouvez-coder les instructions pour les vérifier.***