

Un orchestre symphonique dans mon ordinateur

I – Notre oreille peut-elle mesurer une fréquence de façon précise ?

- **Document 1 : Oreille absolue et oreille relative**

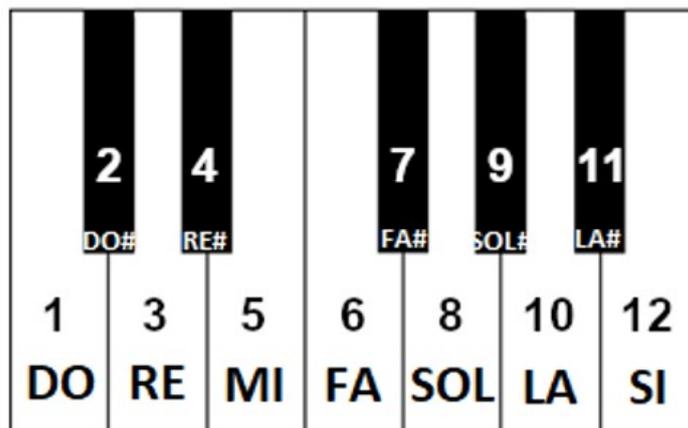
L'oreille absolue est la capacité d'une personne à identifier la hauteur d'une note musicale sans référence sonore. Une personne dotée d'une telle faculté peut déterminer le nom d'une note sans écoute préalable. L'oreille absolue a suscité et suscite toujours des passions et des polémiques non seulement dans le milieu des musiciens mais aussi dans le milieu scientifique. Ce qui caractérise l'oreille absolue, c'est en fait une mémoire auditive exceptionnelle. Elle est considérée comme rare.

Elle ne doit pas être confondue avec l'oreille relative qui est l'aptitude à identifier des intervalles musicaux. Une oreille relative est capable de nommer une note à partir d'une référence initiale donnée au préalable.

D'après <https://www.oreillemudry.ch/oreille-absolue/>

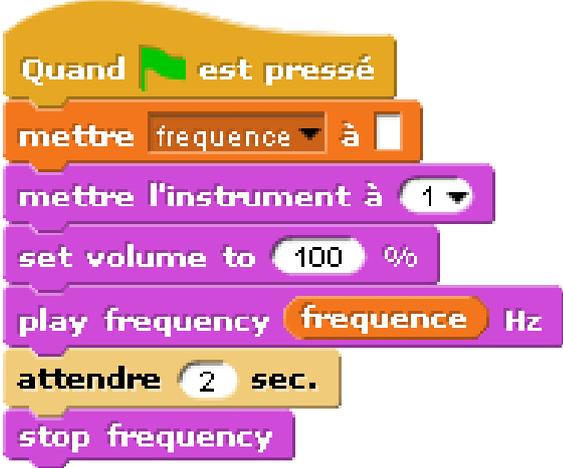
- **Document 2 : hauteur d'une note et gamme tempérée**

La hauteur d'une note correspond à la fréquence du son joué. Plus la fréquence d'une note est élevée, plus le son est aigu, et inversement un son grave correspond à une fréquence basse. Dans la gamme tempérée, l'octave est séparée en douze intervalles (en comptant les dièses). La fréquence d'une note s'obtient en multipliant celle de la note précédente (touche du piano à gauche) par $2^{1/12}$ ($2^{1/12} = 1,05946$)



1- Quelle grandeur physique
différencie deux notes distinctes ?

- 2- A partir d'un son de référence mis à votre disposition (*son_reference1.wav* et/ou *son_reference2.wav*) et d'un script snap /python, évaluer la valeur de la fréquence à laquelle il est émis.

Script snap!	Script python (plusieurs possibilités)
 <pre> Quand le drapeau vert est cliqué mettre fréquence à [] mettre l'instrument à 1 set volume to 100 % play frequency fréquence Hz attendre 2 sec. stop frequency </pre>	<pre> import winsound def note(f): # fréquence f sans décimale winsound.Beep(f,2000) # winsound.Beep(fréquence, durée de la note en ms) import math import sounddevice as sd # impossible sur Edupython audio = [] sample_rate = 44100.0 def append_sinewave(freq=440.0, duration_milliseconds=500, volume=1.0): global audio num_samples = duration_milliseconds * (sample_rate / 1000.0) for x in range(int(num_samples)): audio.append(volume * math.sin(2 * math.pi * freq * (x / sample_rate))) sd.play(audio, sample_rate) return ### Compléter le script en rajoutant la valeur de la fréquence append_sinewave(freq =, duration_milliseconds = 2000, volume=1) import math import wave import struct audio = [] sample_rate = 44100.0 def append_sinewave(freq=440.0, duration_milliseconds=500, volume=1.0): num_samples = duration_milliseconds * (sample_rate / 1000.0) for x in range(int(num_samples)): audio.append(volume * math.sin(2 * math.pi * freq * (x / sample_rate))) return def save_wav(file_name): wav_file=wave.open(file_name,"w") nchannels = 1 </pre>

```

sampwidth = 2
nframes = len(audio)
comptype = "NONE"
compname = "not compressed"
wav_file.setparams((nchannels,
sampwidth, sample_rate, nframes, comptype,
compname))
for sample in audio:
    wav_file.writeframes(struct.pack('h',
int( sample * 32767.0 )))
    wav_file.close()
return
### Indiquer la valeur de la fréquence
append_sinewave(freq = .....,
duration_milliseconds = 2000, volume=1)
save_wav("Fichier.wav")

```

3- Comparer la valeur de la fréquence trouvée avec la valeur théorique donnée par l'enseignant.e en calculant un écart relatif

$$\text{Rappel écart relatif} = \left| \frac{\text{fréquence trouvée} - \text{fréquence réelle}}{\text{fréquence réelle}} \right|$$

Conclure. Votre oreille peut-elle être un instrument de précision?

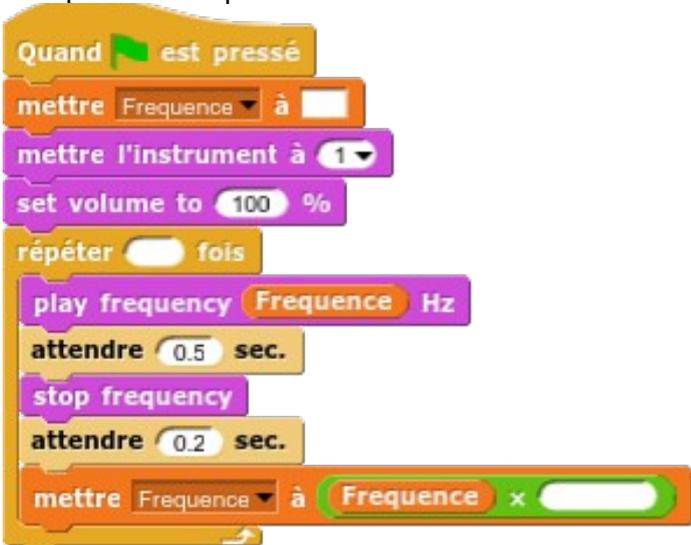
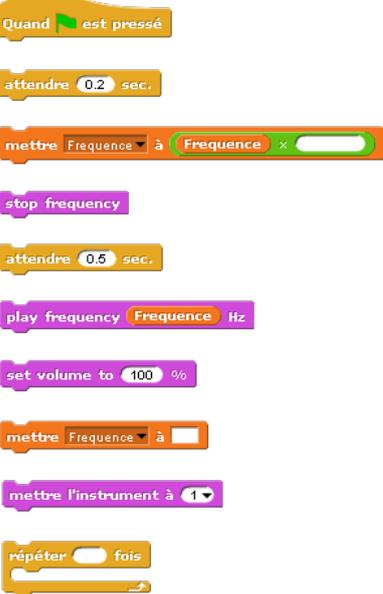
II - Gamme tempérée

1- La hauteur du La₃ est fixée à 440 Hz. Compléter le tableau ci-dessous en y ajoutant la fréquence de chacune des notes. Arrondir les valeurs à l'unité.

Do	Ré	Mi	Fa	Sol	La	Si	Do
					440 Hz		

2- Quelle distinction peut-on faire entre votre oreille et une oreille absolue ?

3- Réaliser un script permettant de reproduire la gamme tempérée précédente.

Explorateur	Initié
<p>Compléter le script suivant :</p> 	<p>Compléter le script suivant :</p> <pre data-bbox="842 394 1481 629"> import winsound def gamme(): f=... #fréquence de départ for i in range(...): winsound.Beep(int(f),400) # int car nombre entier obligatoire f=f*..... </pre> <p>Compléter et remettre dans l'ordre le script suivant.</p> 
Expert	
<p>En utilisant la bibliothèque winsound et la fonction <code>winsound.Beep(fréquence ,durée)</code>, réaliser le script complet.</p>	

4. Tester le programme pour le valider. Commenter.

III- Production d'une mélodie

- **Document 3 : valeurs des notes**

La musique possède un langage qui lui est propre, et il est alors possible de comparer les notes aux lettres de l'alphabet. La succession de ces sons avec des notes s'appelle le « rythme ».

Les valeurs des notes, qui créent le rythme de la musique, sont même plus importantes que leur hauteur car, si vous changez la durée des notes dans une partition, la musique en sera complètement différente.

La valeur des notes indique donc leur durée :

La **noire** vaut 1 temps.

La **blanche** vaut 2 temps.

La **ronde** vaut 4 temps.

On souhaiterait réaliser la mélodie « Au clair de la lune » à l'aide d'un programme.

On dispose de la partition simplifiée suivante :

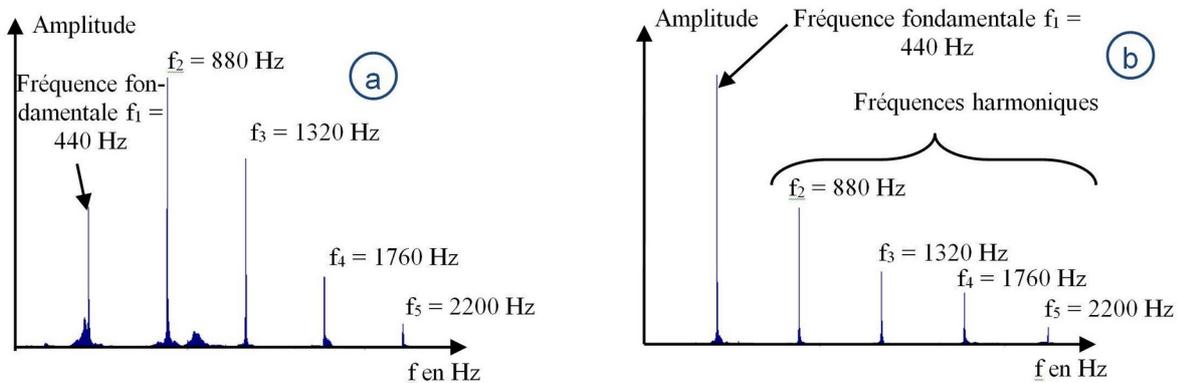


Explorateur	Initié
<p>1. Compléter le programme suivant en respectant les notes et leurs valeurs.</p> <pre> Quand est pressé mettre fréquence à 262 mettre listeFrequences à liste mettre Notes à liste 1 mettre Durée à liste 1 mettre l'instrument à 1 set volume to 100 % pour i allant de 1 à 12 insérer fréquence en position i de listeFrequences mettre fréquence à fréquence × 1.05946 pour i allant de 1 à longueur de Notes play frequency élément élément i de de listeFrequences attendre élément i de Durée × 0.4 sec. stop frequency </pre> <p>2. Tester le programme pour le valider. 3. Comparer votre production avec la mélodie jouée avec un violon (fichier <i>Au_Clair_de_lune_violon_reel.wav</i>).</p>	<p>Compléter le programme suivant.</p> <pre> import winsound def auclairdelalune(): L=[0] # début de la liste L f=262 # première fréquence de la gamme noire=400 # temps d'une noire = 400 ms blanche=..... # 1 blanche = 2 temps ronde=..... # 1 ronde = 4 temps for i in range(.....): L.append(f) # insère la gamme dans L f=int(f*1.05946) for i in range(.....): winsound.Beep(L[1],noire) # DO noire winsound.Beep(L[3],noire) winsound.Beep(L[.....],blanche) winsound.Beep(L[.....],.....) winsound.Beep(L[.....],.....) winsound.Beep(L[.....],.....) winsound.Beep(L[.....],.....) for i in range(.....): winsound.Beep(L[.....],.....) winsound.Beep(L[.....],.....) </pre> <p>Tester le programme pour le valider. (réaliser-valider)</p> <p>Comparer votre production avec la mélodie jouée avec un violon (fichier <i>Au_Clair_de_lune_violon_reel.wav</i>).</p>

Expert

• Document 4 : timbre d'une note

Lorsqu'une note de musique est jouée, on entend la fréquence de la note mais également celles des harmoniques.



Fréquences composant une note /a jouée à la guitare (a) et au piano (b)

Lorsqu'une note de musique est jouée, on entend la fréquence de la note mais également celles des harmoniques. L'ensemble du spectre en fréquence caractérise le timbre d'un instrument. Le timbre permet d'expliquer la perception différente de chaque instrument.

Différents instruments peuvent jouer la même note, mais ils seront perçus différemment car leurs timbres sont différents. Plus la forme du signal est complexe, plus le timbre est riche.

La qualité d'un instrument va dépendre de la richesse du timbre qu'il est capable de produire.

1- Vérifier à partir de l'échantillon de musique et du script fournis que l'instrument produit des harmoniques.

2- Quelle relation simple obtient-on entre la hauteur de la note et les harmoniques sur les spectres?

3- Compléter le script `python_fft.py` de façon à produire la mélodie demandée

4- Comparer votre production avec la mélodie jouée avec un violon (fichier `Au_Clair_de_lune_violon_reel.wav`).

IV- Correction

Question I - 1 : Deux notes distinctes ont une fréquence différente (avec un rapport de $2^{1/12}$).

Question I- 3 : Élément de discussion : un musicien entraîné possède un écart relatif de moins de 1%. Les élèves devraient obtenir moins de 10%.

Question II-1 :

Do	Ré	Mi	Fa	Sol	La	Si	Do
262 Hz	294 Hz	330 Hz	349 Hz	392 Hz	440 Hz	494 Hz	523 Hz

Question II- 2 : L'oreille absolue est beaucoup plus rare et n'a pas besoin d'écouter une fréquence étalon pour retrouver la note. Les fréquences trouvées par les élèves doivent rester proches de la fréquence de référence. L'élève doit rester proche en fréquence cependant de la note recherchée.

Question II- 3 :

Explorateur et initié Snap!	Initié et Expert
	<pre>import winsound def gamme(): f=262 #fréquence de départ for i in range(12): winsound.Beep(int(f),400) f=f*1.05946</pre>

Question III

Explorateur	Initié
<p>Question III- 2</p>	<pre>import winsound def auclairdelalune(): L=[0] f=262 noire=400 blanche=800 (ou 2*noire) ronde=1600 (ou 4*noire) for i in range(12): L.append(f) f=int(f*1.05946) for i in range(3): winsound.Beep(L[1],noire) winsound.Beep(L[3],noire) winsound.Beep(L[5],blanche) winsound.Beep(L[3],blanche) winsound.Beep(L[1],noire) winsound.Beep(L[5],noire) for i in range(2):</pre>
<p>Question III-3 : Les sons produits par les</p>	

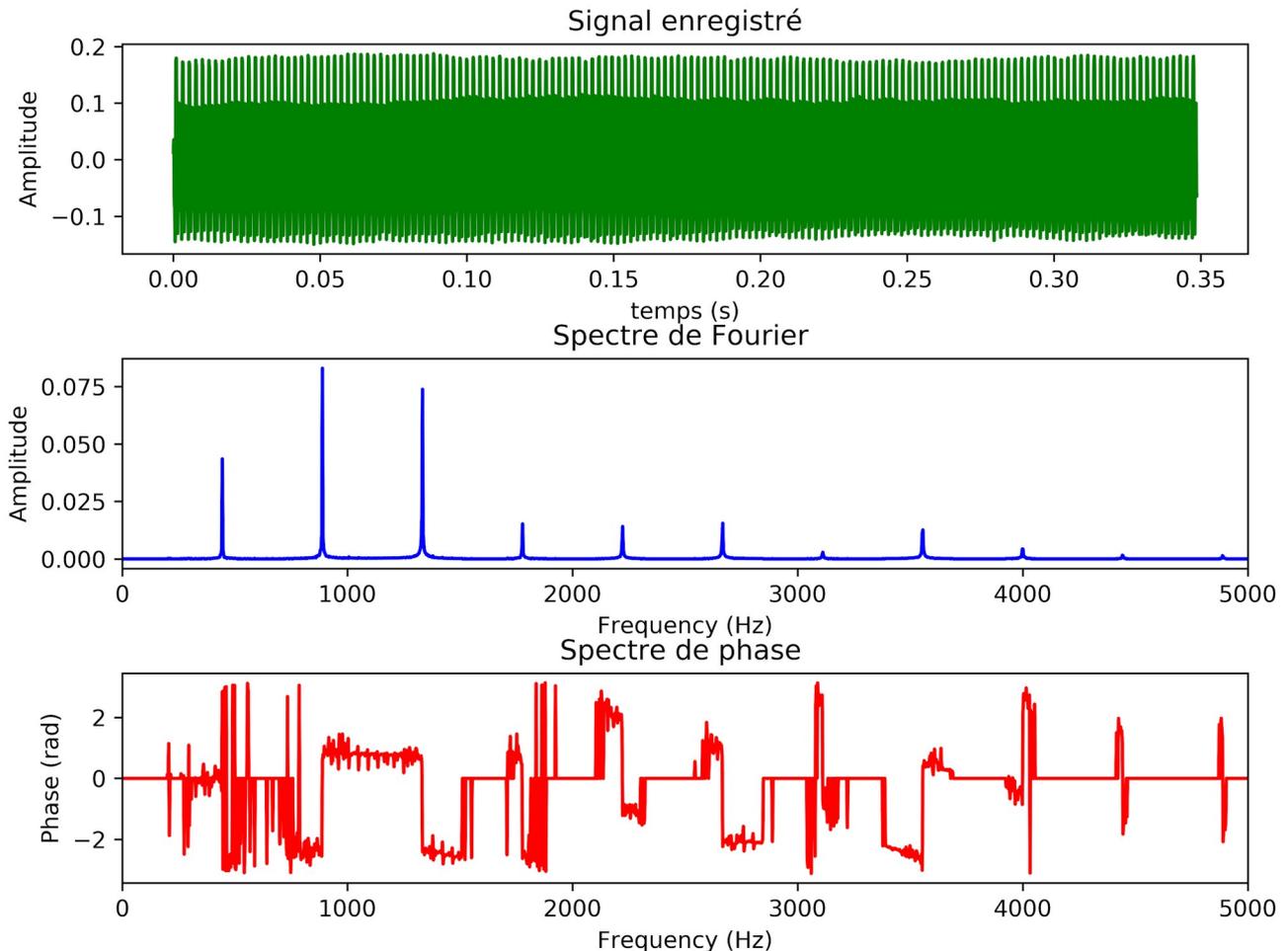
instruments restent complexes et difficiles à reproduire.

`winsound.Beep(L[3],noire)`
`winsound.Beep(L[1],ronde)`

Question III-3 : Les sons produits par les instruments restent complexes et difficiles à reproduire.

Expert

Question III-1



Question III-2 : Les fréquences des harmoniques sont des multiple entiers de la fréquence du fondamental

Question III-3 : voir script ci-dessous

```
print ('Do')
append_note(freq = 523, duration_milliseconds = 500, volume=1)
print ('Do')
append_note(freq = 523, duration_milliseconds = 500, volume=1)
print ('Do')
append_note(freq = 523, duration_milliseconds = 500, volume=1)
print ('Ré')
append_note(freq = 587, duration_milliseconds = 500, volume=1)
print ('Mi')
append_note(freq = 659, duration_milliseconds = 1000, volume=1)
print ('Ré')
append_note(freq = 587, duration_milliseconds = 1000, volume=1)
print ('Do')
```

```

append_note(freq = 523, duration_milliseconds = 500, volume=1)
print ('Mi')
append_note(freq = 659, duration_milliseconds = 500, volume=1)
print ('Ré')
append_note(freq = 587, duration_milliseconds = 500, volume=1)
print ('Ré')
append_note(freq = 587, duration_milliseconds = 500, volume=1)
print ('Do')
append_note(freq = 523, duration_milliseconds = 1500, volume=1)

```

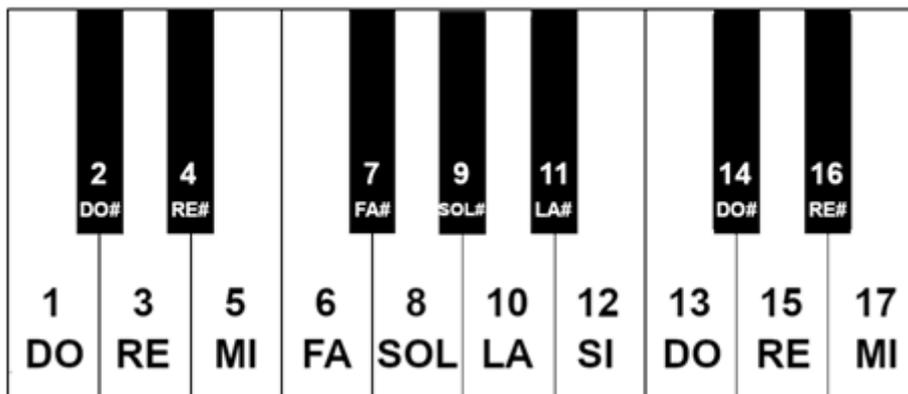
Question III-4 : Les sons produits par les instruments restent complexes et difficiles à reproduire.

V- Annexe 1 : Jouer une mélodie au choix

1- La lettre à Elise

On souhaiterait ne pas se contenter de jouer « au clair de la lune » et jouer n'importe quelle mélodie en créant un piano virtuel.

On souhaite donc créer une fonction qui permettrait de jouer la mélodie à partir de la liste des numéros des touches du piano (de 1 à 17, on entre 0 pour un silence) séparées par un espace.



<https://pixabay.com/fr/vectors/piano-touches-octave-musique-307653/>

Par exemple pour « Au clair de la lune », on entrerait dans la zone de saisie :

1 1 1 3 5 3 1 5 3 3 1

Voici le programme correspondant :

Explorateur	Initié
--------------------	---------------

	<pre>import winsound def piano(*a): note=list(a) L=[0] f=262 for i in range(17): L.append(f) f=int(f*1.05946) for i in range(len(note)): winsound.Beep(L[note[i]],400)</pre>
--	--

On donne :

Lettre à Élise (S pour silence)

mi (17) - ré# (16) - mi - ré# - mi - si - ré (15) - do (13) - la - S - do (1) - mi - la - si - S - mi - sol# - si - do (12) - S
mi (17) - ré# (16) - mi - ré# - mi - si - ré (15) - do (13) - la - S - do (1) - mi - la - si - S - mi - do (12) - si - la

Explorateur	Initié
<p>1. Quel est l'inconvénient de ce programme quand on joue « Au clair de la lune » ?</p> <p>Pour vous aider, vous pouvez tester le programme JouerSaMusique2.xml. (raisonner-valider)</p> <p>2. Quelle est la liste de nombres à saisir dans la zone de saisie pour jouer la « Lettre à Elise » ? (s'approprier)</p> <p>3. Tester le programme pour valider.</p>	<p>1- Que faut-il saisir dans la console pour jouer la « Lettre à Élise » ? (s'approprier)</p> <p>Piano(.....)</p> <p>2- Tester le programme. Que remarque-t-on ? (valider)</p> <p>3- Quelle modification pourrait-on effectuer à la ligne 4 (L=[0]) ? (raisonner-communicuer).</p> <p>4- Tester votre proposition et valider votre modification. (valider)</p>

Correction :

Explorateur	Initié
<p>1. Toutes les notes ont la même durée dans ce programme.</p> <p>2. 17 16 17 16 17 12 15 13 10 0 1 5 10 12 0 5 9 12 13 0 17 16 17 16 17 12 15 13 10 0 1 5 10 12 0 5 13 12 10</p>	<p>1- piano(17,16,17,16,17,12,15,13,10,0,1,5,10,12,0,5,9,12,13,0,17,16,17,16,17,12,15,13,10,0,1,5,10,12,0,5,13,12,10)</p> <p>2- On obtient un message d'erreur au moment du silence (0). En Python, la fréquence doit être supérieure à 37 Hz.</p> <p>3- L=[0] pourrait devenir L=[37].</p>

VI - Annexe 2 : Consonance et dissonance

Deux notes jouées sont dites consonantes quand leur écoute est appréciée par l'oreille. Nous estimons inconsciemment les rapports de fréquence (ce qu'on appelle intervalles) entre les notes successives (dites « mélodiques »).



<https://pxhere.com/fr/photo/1452207>

Pour être consonantes, les fréquences des notes jouées doivent être séparées d'un intervalle qui repose sur un rapport fractionnaire simple ($2/1$; $3/2$; $5/4$; $4/3$; ...). Les intervalles utilisés sont :

- l'octave de rapport 2 ;
- la quinte de rapport $3/2$;
- la quarte de rapport $4/3$;
- la tierce majeure $5/4$;
- la tierce mineure $6/5$.

Problématique : Comment construire aléatoirement une mélodie avec des notes consonantes ?

On décide de réaliser la mélodie avec des notes consonantes à l'aide d'un programme en langage Python.

Explorateur	Initié
-------------	--------

