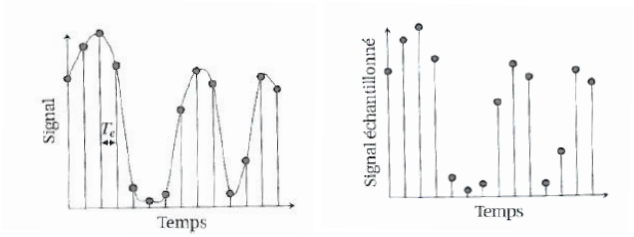
Compétences : liste, slicing (extraction de tranche), maximums d'une liste

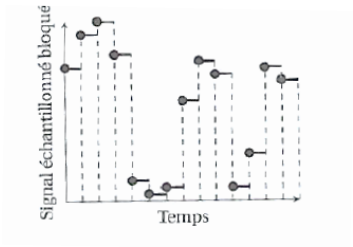
***Lien capytale :* 809c-3797608**

Fabriquer un effet audio : le delay

Le but de cet exercice est d'illustrer la numérisation d'un signal en utilisant comme exemple le son. L'échantillonnage d'un signal consiste à prendre la valeur du signal à différents instants et de manière périodique (avec une période T_e et donc une fréquence d'échantillonnage $f_e = \frac{1}{T_e}$) :



La seconde opération d'une conversion consiste à bloquer la valeur de l'échantillon pendant T_e :



On obtient une série de valeurs s_i du signal échantillonné. La dernière étape de numérisation consiste à convertir ces tensions sur n bits. Par exemple la conversion d'un signal variant entre -5V et 5V sur 16 bits consiste à convertir ces 10 V sur 65536 niveaux (de -32768 à 32767).

Nous allons utiliser la librairie scipy.io.wavfile qui permet de récupérer le niveau de chaque échantillon ainsi que les librairies numpy et matplotlib:

```
import scipy.io.wavfile
import matplotlib.pyplot as plt
import numpy as np
```

Pour ouvrir avec python le fichier audio « projet2h.wav » fourni, obtenir la fréquence fe d'échantillonnage et une liste data de données audios échantillonnées (au format int16), on utilise les commandes ci-dessous :

```
"""lecture"""
fe,data =scipy.io.wavfile.read('projet2h.wav')
data=data.tolist()
```

La commande ci-dessous permet de créer un fichier audio projet2h.wav à partir de son fichier data et de sa fréquence d'échantillonnage :

```
scipy.io.wavfile.write('projet2h_reponse.wav',fe,np.array(data,dtype="int16"))
```

1) Ecrire une fonction double(L) qui prend en argument une liste L et qui double toutes les valeurs de la liste L.
2) Appliquer la fonction *double* sur la liste data et écrire une fonction test, qui prend en argument la liste data modifiée par la fonction double et qui renvoie False si au moins un élément de data dépasse les valeurs autorisées par le codage sur 16 bits et True dans le cas contraire.
3) Définir une fonction echo qui prend comme argument la liste data ainsi qu'un entier positif a. Cette fonction ne renvoie rien mais crée un nouveau fichier.wav qui reprend le fichier initial en y ajoutant un écho décalé de a seconde (on a donc le signal initial sur lequel vient se rajouter une copie décallée de a secondes). Tester cette fonction sur le fichier projet2h_reponse.wav (par exemple avec a=1).