

I Mise en place

1.1 Winpython

Si ce n'est déjà fait, téléchargez **Winpython**. Pour l'installation, laissez-vous guider, en notant tout de même l'endroit où vous pourrez trouver **Spyder.exe**.

1.2 Espace de travail

Créez un dossier nommé “python” quelque part dans vos documents, puis copiez le dossier DM1 à l'intérieur.

Pour plus de facilité, nous allons créer un espace de travail dans Spyder. Si vous n'avez pas d'onglet appelé “Explorateur de projet”, remédiez à cela en passant par le menu Affichage -> Fenêtres et barres d'outils, et en cochant la case idoïne.

Il ne reste qu'à cliquer sur le dossier en haut à droite de cet onglet et naviguer jusqu'à votre dossier python. Confirmer la création de l'espace de travail. Ensuite Clic droit -> import -> Répertoire existant, et choisissez le dossier DM1.

1.3 Rendu

Pour rendre ce DM, une seule et unique procédure :

- vous connecter à [mon site](#)
- téléverser votre fichier de script en cliquant sur “rendre le devoir”.

Une remarque : si vous avez une deuxième version à envoyer, la même procédure s'applique et seule la version la plus récente est conservée.

Assurez-vous que votre script peut-être exécuté avec F5 avant de l'envoyer.

1.4 But du DM

Le but de ce DM est de créer un petit script qui permet de choisir un élève au hasard dans la classe, de manière plus ou moins élaborée. On peut tout à fait imaginer que ce script servira à désigner des volontaires...

Pour cela le fichier *elevés.py* vous est fourni, et il contient une liste nommée *elevés*. Pour y accéder dans un script ou dans la console :

```
import élevés
elevés.elevés
```

ou plus simplement, en utilisant une autre syntaxe de la commande d'import

```
from élevés import élevés
elevés
```

Dans les deux cas, python doit être capable de trouver le fichier *elevés.py*. Cela ne posera pas de problème si vous utilisez F5 pour envoyer directement le script que vous créez dans la console, et si ce fichier créé est dans le même répertoire que *elevés.py*.

1.5 Tester le code produit

Il est très important de toujours tester les fonctions que l'on vient de créer. Pour cela, une aide est disponible dans ce DM

1.5.1 Docstrings

Dans le fichier que vous allez devoir compléter (*tirage.py*), les fonctions à créer ont déjà une docstring, c'est à dire une description du comportement attendu, avec en plus des exemples d'exécution. Ces descriptions font partie intégrante de l'énoncé.

```
def ma_super_fonction(x):
    """
    Je suis une docstring. Il est très malpoli de m'effacer
    """
    return x
```

Les 3 dernières lignes du fichier permettent de vérifier que les exemples d'exécutions se déroulent comme prévu lorsque que vous exécutez le script via F5. Pour le moment vous obtenez des messages d'erreur. Le but est de les faire tous disparaître.

II Code

2.1 Approche naïve

La première approche est de choisir au hasard (chaque élève a autant de chances de passer à chaque fois). A vous de trouver l'outil dans [la documentation de python](#) pour compléter la fonction **tire_au_hasard** du fichier *tirage.py*.

2.2 Compter les passages

Une méthode peut-être plus juste consiste à compter les passages au tableau de chaque élève. On décide de créer une liste **passages** qui contient le nombre de passage de chaque élève. Plus précisément, le nombre d'indice *i* dans **passages** est le nombre de fois où l'élèves d'indice *i* dans **elevés** a déjà été choisi.

Complétez la fonction **choisi_eleve**.

Pour tester votre fonction, il est important d'utiliser la même liste **passages** créée une fois pour toute dans la console.

2.3 Tenir compte des passages

On décide dans un premier temps d'équilibrer le nombre de passages :

- Quand passage est rempli de 1, on remet tout ses éléments à 0.
- On ne peut choisir qu'un élève dont le nombre de passage courant est 0.

Complétez les fonction **remise_a_zero** (qui doit fonctionner sur toute liste passée en argument, pas seulement sur une liste de longueur 40) et **choix_sans_remise**.

Une fonction qui pourrait vous servir :

```
L = [2,7,9,10]
L.index(7) #retourne 1, c'est à dire l'indice de l'élément 7 dans la liste L
```

Comme pour tout programme, il existe plusieurs manière d'écrire la fonction **choix_sans_remise** et vous pouvez tout à fait vous passer d'utiliser la méthode **index**

2.4 Pour aller plus loin

On peut décider de raffiner le processus de choix. En particulier pour éviter la trop grande prévisibilité du système précédent. Voici une manière parmi d'autres.

- On ne remet plus passages à zéro.
- La probabilité d'être choisi dépend du nombre de passage déjà effectué : on calcule la moyenne de passages, et les élèves ayant un nombre de passage supérieur à cette moyenne ont moins de chances d'être de nouveau envoyés au tableau.
- Plus précisément, on décide qu'un élève étant passé au tableau en moyenne une fois de plus que les autres aura 2 fois moins de chances d'être re-choisi, et réciproquement qu'un élève étant passé 1 fois de moins en moyenne aura 2 fois plus de chances d'être choisi.

2.4.1 Lemmes

Pour cela il nous faut des outils techniques : complétez les fonction **moyenne** et **sommes_cumulees**.

2.4.2 On se lance

Votre rôle est de traduire en Python dans la fonction **choix_pondere** l'algorithme suivant.

1. Calculer la moyenne m de la liste passages.
2. Créer la liste **chances** des chances respectives d'être choisi (en proportion) : c'est la liste des 2^{m-p} où p prend toutes les valeurs de passages. Evidemment, **chances[i]** correspond à **eleves[i]**.

Remarque : on voit que si $p = m$, alors la proportion de chance d'être choisi est 1, c'est à dire qu'un élève étant passé au tableau exactement autant de fois que la moyenne est notre élève de référence.

Si $p = m + 1$ (un passage de plus déjà effectué), alors $2^{m-p} = \frac{1}{2}$. Un élève étant passé une fois de plus que la moyenne à 2 fois moins de chances d'être re-choisi.

3. La liste des probabilités d'être choisi (**probas** par exemple) s'obtient maintenant en divisant chaque élément de **chances** par la somme des éléments de **chances**. La somme des éléments de la liste **probas** doit être 1 aux erreurs d'arrondis près.
4. Créer la liste **cumul** des sommes cumulées de **probas**. Son dernier élément est 1 d'après la remarque précédente.
5. Tirer un nombre au hasard entre 0 et 1. On le note p .
6. On considère **cumul** comme une liste d'intervalle : Si p est entre 0 et **cumul[0]**, c'est l'élève 0 qui a été choisi, si p est entre **cumul[0]** et **cumul[1]**, c'est l'élève 1, etc. On convient d'exclure la borne supérieure de chaque intervalle ainsi considéré, sauf la dernière.
Trouver quel élève a été choisi.
7. Mettre à jour **passages**
8. Retourner le nom de l'élève choisi.