Unité 2 : Débuter la programmation en Python | **Application : Boucles et tests**

Pour cette application de l'unité 2, on se propose de réinvestir les notions vues dans les leçons concernant les instructions conditionnelles ainsi que les boucles bornées et non bornées.

Objectifs :

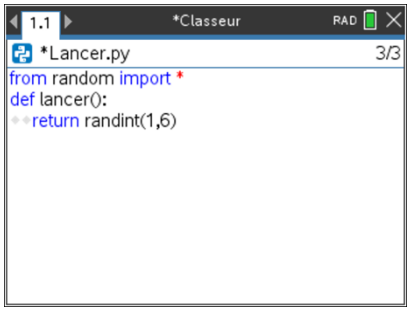
- Utiliser la boucle **While** et **For** pour mettre en œuvre un algorithme relatif à un problème de probabilités ou de statistiques.

Dans cette application, vous allez écrire un script permettant de :

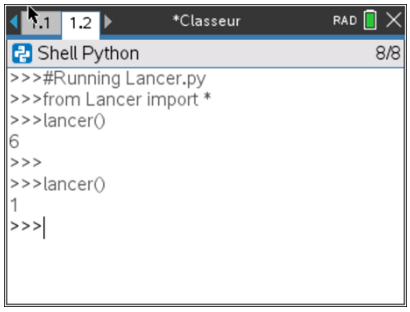
- Obtenir un nombre aléatoire en créant une fonction **lancer**.
- Utiliser cette fonction dans un autre script afin de déterminer le nombre de lancers nécessaires pour obtenir une somme de 12 lors du lancer de 2 dés à 6 faces parfaitement équilibrés.
- Lors du lancer d'un seul dé à 6 faces, obtenir le nombre de fois où chaque face apparait afin éventuellement de calculer une fréquence à comparer à la probabilité de sortie de chaque face.

Lancer un dé.

- Le travail sur les nombres aléatoires nécessite le chargement du module **random** avant la définition de la fonction.
- Créer un nouveau script et le nommer **Lancer**.
- Appuyer sur menu puis **6 Nombres aléatoires** et choisir le module **1 from random import***.
- Définir la fonction **lancer()** permettant d'obtenir un nombre aléatoire entier entre 1 et 6.



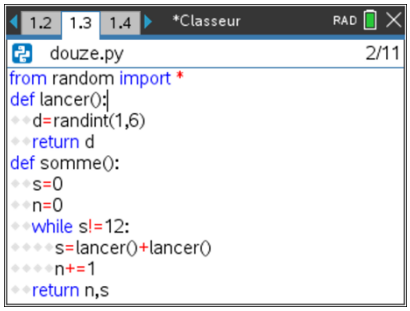
- Tester votre script après l'avoir vérifié.
- Utiliser les touches de direction vers le haut (▲) afin de relancer le script.



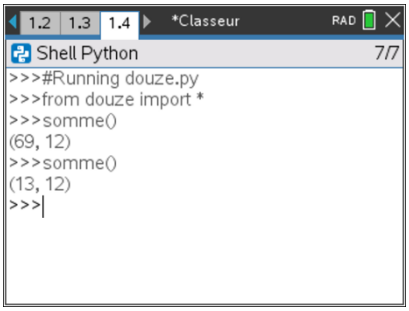
Nombre d'essais nécessaires.

On lance deux dés à 6 faces parfaitement équilibrés et on additionne les deux résultats obtenus. Vous allez écrire un autre script afin de modéliser les lancers de ces deux dés et rechercher le nombre *n* de lancers nécessaires afin d'obtenir la valeur 12.

De nombreuses solutions sont possibles, mais la première qui vient à l'esprit est de réutiliser la fonction précédente.



- La variable *s* donne la somme des lancers et *n* le nombre de lancers nécessaires avant d'atteindre 12.
- Toutes deux sont initialisées à 0.
- En langage Python, le symbole ≠ est représenté par **!=**. Ce symbole est obtenu en appuyant sur les touches menu puis **4 Intégrés** et enfin **3 Ops**.
- Compléter le script en veillant à respecter l'indentation puis l'exécuter.
- Le premier nombre donne le nombre de lancers nécessaires pour atteindre la somme 12 affichée par le second.



Échantillonnage et fréquence

Vous venez d'observer sur l'exemple précédent que le nombre d'essais nécessaires avant d'obtenir un 12 fluctue. On peut donc être conduit à calculer la fréquence d'échantillonnage, qui pour un grand nombre d'essais doit tendre vers la probabilité théorique.

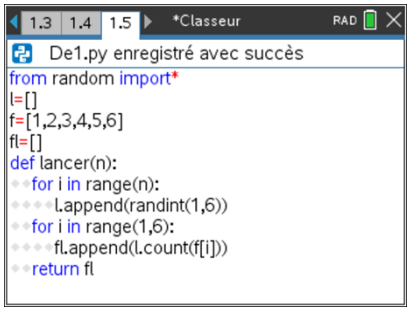
Vous allez dans ce dernier script :

Utiliser une boucle for pour répéter un seul lancer.

Calculer le nombre de fois où une face apparait.

Utiliser une liste, ce qui a l'avantage de rendre le script plus court.

- Créer un nouveau script et le nommer De1.
- Le résultat d'un lancer est stocké dans la liste l précédemment initialisée vide par l'instruction **l=[ ]**.
- La liste **f** contient les numéros des faces.
- Ensuite un test est effectué sur la valeur de la variable (1 à 6) et chaque fois qu'une condition est vérifiée, une nouvelle liste **fl** est créée dans laquelle est stocké, pour chaque numéro de face variant de 1 à 6, le nombre d'occurrence observées.



- Modifier éventuellement le script afin de calculer la fréquence d'apparition de chaque face.

- Appuyer sur la touche ctrl B pour enregistrer le script et vérifier sa syntaxe.
- Appuyer sur la touche ctrl R afin d'exécuter le script. Observer les effectifs de sortie de chaque face, pour 100 lancers : les fréquences sont simples à calculer.