

# Etude de cas /\*Python\*\

## TOUS AU CASINO

---

### Notre sujet

---

Il s'agira d'un petit jeu de roulette très simplifié dans lequel vous pourrez miser une certaine somme et gagner ou perdre de l'argent. Quand vous n'avez plus d'argent, vous avez perdu.

#### Notre règle du jeu

Le joueur mise sur un numéro compris entre 0 et 49 (50 numéros en tout). En choisissant son numéro, il y dépose la somme qu'il souhaite miser.

La roulette est constituée de 50 cases allant naturellement de 0 à 49. Les numéros pairs sont de couleur noire, les numéros impairs sont de couleur rouge. Le croupier lance la roulette, lâche la bille et quand la roulette s'arrête, relève le numéro de la case dans laquelle la bille s'est arrêtée. Dans notre programme, nous ne reprendrons pas tous ces détails « matériels » mais ces explications sont aussi à l'intention de ceux qui ont eu la chance d'éviter les salles de casino jusqu'ici. Le numéro sur lequel s'est arrêtée la bille est, naturellement, le numéro gagnant.

**Si** le numéro gagnant est celui sur lequel le joueur a misé (probabilité de 1/50, plutôt faible), le croupier lui remet 3 fois la somme mise.

**Sinon**, le croupier regarde si le numéro misé par le joueur est de la même couleur que le numéro gagnant (s'ils sont tous les deux pairs ou tous les deux impairs). Si c'est le cas, le croupier lui remet 50 % de la somme mise. Si ce n'est pas le cas, le joueur perd sa mise.

Dans les deux scénarios gagnants vus ci-dessus (le numéro misé et le numéro gagnant sont identiques ou ont la même couleur), le croupier remet au joueur la somme initialement mise avant d'y ajouter ses gains. Cela veut dire que, dans ces deux scénarios, le joueur récupère de l'argent. Il n'y a que dans le troisième cas qu'il perd la somme mise. On utilisera pour devise le dollar \$ à la place de l'euro pour des raisons d'encodage sous la console Windows.

Pour ce projet, nous n'allons pas écrire de module. Nous allons utiliser ceux de Python, qui sont bien suffisants pour l'instant, notamment celui permettant générer de l'aléatoire, que je vais présenter plus bas. En attendant, ne vous privez quand même pas de créer un répertoire et d'y mettre le fichier `Casino.py`, tout va se jouer ici.

Vous êtes capables d'écrire le programme `Casino` tel qu'expliqué dans la première partie sans difficulté... sauf pour générer des nombres aléatoires. Python a dédié tout un module à la génération d'éléments pseudo-aléatoires, le module `random`.

### Le module `random`

Dans ce module, nous allons nous intéresser particulièrement à la fonction `randrange` qui peut s'utiliser de deux manières :

- en ne précisant qu'un paramètre (`randrange(6)` renvoie un nombre aléatoire compris entre 0 et 5) ;
- en précisant deux paramètres (`randrange(1, 7)` : renvoie un nombre aléatoire compris entre 1 et 6, ce qui est utile, par exemple, pour reproduire une expérience avec un dé à six faces).

Pour tirer un nombre aléatoire compris entre 0 et 49 et simuler ainsi l'expérience du jeu de la roulette, nous allons donc utiliser l'instruction `randrange(50)`.

Il existe d'autres façons d'utiliser `randrange` mais nous n'en aurons pas besoin ici et je dirais même que, pour ce programme, seule la première utilisation vous sera utile.

N'hésitez pas à faire des tests dans l'interpréteur de commandes et essayez plusieurs syntaxes de la fonction `randrange`. Je vous rappelle qu'elle se trouve dans le module `random`, n'oubliez pas de l'importer.

### Arrondir un nombre

Vous l'avez peut-être bien noté, dans l'explication des règles je spécifiais que si le joueur misait sur la bonne couleur, il obtenait 50% de sa mise. Oui mais... c'est quand même mieux de travailler avec des entiers. Si le joueur mise 3\$, par exemple, on lui rend 1,5\$. C'est encore acceptable mais, si cela se poursuit, on risque d'arriver à des nombres flottants avec beaucoup de chiffres après la virgule. Alors autant arrondir au nombre supérieur. Ainsi, si le joueur mise 3\$, on lui rend 2\$. Pour cela, on va utiliser une fonction du module `math` nommée `ceil`. Je vous laisse regarder ce qu'elle fait, il n'y a rien de compliqué.

### À vous de jouer

Voilà, vous avez toutes les clés en main pour coder ce programme. Prenez le temps qu'il faut pour y arriver, le but du TP est que vous appreniez à coder vous-mêmes un programme... et celui-ci n'est pas très difficile. Si vous avez du mal, morcelez le programme, ne codez pas tout d'un coup. Et n'hésitez pas à passer par l'interpréteur pour tester des fonctionnalités : c'est réellement une chance qui vous est donnée, ne la laissez pas passer.