



# Jeu du Pendu

**MGA802 Introduction à la programmation avec Python**

# JEU DU PENDU

Pour ce mini-projet, vous allez écrire un code qui permet de jouer au jeu du pendu contre votre programme!

Votre programme doit être **composé de fonctions seulement**:

- Sélectionner un mot aléatoire (`import random`) à partir d'un fichier texte contenant un grand nombre de mots qui sera fourni sur Moodle
- Faire tourner une boucle tant que le nombre de chances (fixé initialement à 6) de l'utilisateur est supérieur à zéro
- À chaque boucle, le programme doit:
  1. Afficher l'état actuel des mots en utilisant des `_` pour les lettres non devinées
  2. Demander à l'utilisateur d'entrer une lettre
  3. Indiquer à l'utilisateur si la lettre faisait partie du mot ou pas
  4. Mettre à jour les chances si nécessaires et, si c'est le cas:
    - a) Indiquer à l'utilisateur s'il a gagné
    - b) Indiquer à l'utilisateur s'il a perdu
- Boucler.



# RANDOM

Le module (inclus avec python) **random** permet de générer des nombres aléatoires ou manipuler des listes pour sélectionner des éléments aléatoires.

```
import random

mots = ["Mot1", "Mot2", "Mot3", "Mot4", "Mot5"]

# Fonction pour choisir un mot au hasard
def choisir_mot(mots):
    return random.choice(mots)

mot_aleatoire = choisir_mot(mots)
```

# GITHUB & CONDITIONS

Le code doit être réposé sur un dépôt Github avec un README.md

Il faudra avoir utiliser Github pour la construction graduelle du code (possible de regarder les commits et leurs messages du dépôt sur Github).

- L'utilisateur doit être capable de fournir son propre fichier « mots\_pendu.txt », sinon le fichier fourni par doit être sélectionné par défaut
- Les mots contenant des lettres avec des accents (é è ê à â û ...) doivent être reliés à leurs équivalent sans accents (e a u ...)
- Si l'utilisateur perd ou gagne, le programme doit proposer à l'utilisateur de recommencer ou quitter

**BONUS:** S'il reste une chance à l'utilisateur, le programme peut lui donner un indice (donner une lettre qui n'est pas dans le mot)

# POINTS D'ÉVALUATION

- Code fonctionnel? 1/10
  - Tâches accomplies? 2/10
  - Commentaires/README? 2/10
  - Style de code? 1/10
  - Interaction-utilisateur? 2/10
  - Utilisation de Github? 2/10
  - Bonus 1/10
- 
- Détails pour chaque point fournis sur le Moodle.
  - Comme pour le premier exercice, il faudra évaluer deux projets de deux autres membres du groupe, à télécharger à partir du Github cette fois.