

# Jeu du pendu

Dans ce TP, on veut programmer un jeu de pendu à un joueur. L'ordinateur choisira un mot aléatoire, et le joueur devra trouver ce mot avant d'être pendu.

On commencera par récupérer le fichier Python sur le site <https://www.quirin.science>.

Ce fichier contient une liste de 200 mots français, choisis aléatoirement, ainsi qu'une liste de dessins correspondants aux étapes du jeu.

Dans les questions qui suivent, on utilisera les variables suivantes :

- `mot` sera le mot choisi par l'ordinateur
  - `lettres` sera la liste des lettres proposées par le joueur
  - `k` sera le nombre de mauvaises lettres choisies par le joueur
1. Commencer par écrire une fonction qui prend en paramètre un entier `k`, et qui affiche l'étape du pendu correspondante.
  2. Écrire une fonction qui prend en paramètre une liste de caractères `lettres` et un mot `mot`, et qui fait la chose suivante : pour chaque lettre du mot, si elle est dans la liste `lettres`, on l'affiche ; sinon, on affiche un caractère "`_`".  
On pourra utiliser l'option `end=" "` de `print` pour éviter de revenir à la ligne.
  3. Écrire une fonction qui prend en paramètre une liste `lettres` et qui renvoie une lettre demandée au joueur ; on s'assurera que le caractère choisi n'est pas déjà dans `lettres`, dans le cas contraire, on demande une autre lettre.
  4. Écrire des fonctions `win(lettres,mot)` et `lose(k)`, qui renvoient des booléens. `win` renverra `True` si toutes les lettres du mot sont dans `lettres`, et `False` sinon ; `lose` renvoie `true` si `k` est supérieur ou égal à 9.
  5. En utilisant les fonctions précédentes, écrire une fonction `jeu()` qui simule le jeu :
    - l'ordinateur choisit un mot au hasard dans la liste fournie
    - tant que les booléens `win(lettres,mot)` ou `lose(k)` ne sont pas vérifiés, on demande des lettres au joueur
    - si la lettre est dans le mot, on continue ; sinon, on affiche une étape de pendu supplémentaire, et on continue
    - à la fin de la partie, afficher un message indiquant si le joueur a gagné ou non.
  6. Éventuellement, améliorer le code en affichant les caractères `é`, `è`, `ê` si le joueur propose `e`, *etc.*