

# CODING CLUB

EPITECH LYON



## Pendu

---

VERSION 1.2.1

# I. Introduction

Audrey est une jeune collégienne qui voudrait jouer au jeu du pendu. Ce fameux jeu consiste à trouver un mot en devinant quelles sont les lettres qui le composent.

Habituellement ce jeu se joue à deux ou plus, une personne choisit un mot et le second doit le deviner. Mais Audrey est actuellement chez elle pour le confinement et n'a personne pour jouer avec elle.

Elle décide donc de se lancer dans la conception d'un programme pour choisir un mot qu'elle devra deviner.

## II. Consignes

- Pour l'installation, veuillez suivre le tutoriel « Installation Python et ses outils ».
- Pour ce projet, il vous sera demandé de choisir comme nom de repository : `cc_pendu`
- N'oubliez pas de push régulièrement.
- En cas de question, pensez à demander de l'aide à votre voisin de droite. Puis de gauche. Demandez enfin à un Cobra (ceux-là ne mordent pas) si vous êtes toujours bloqué(e).
- Vous avez tout à fait le droit d'utiliser internet pour trouver des réponses ou pour vous renseigner.
- N'hésitez pas à faire des bonus et à ajouter des fonctionnalités lorsque votre projet sera terminé et validé.

## III. Création des variables du jeu

La variable `solution` contient un mot choisi au hasard parmi la liste `"mots.txt"`

`pendu` contient un dessin du pendu, au début il est vide. En faisant `pendu.set_step(2)` on modifie ce dessin pour l'afficher avec 2 erreurs.

Il faudra ensuite qu'elle implémente une seconde variable appelée `erreurs` afin de pouvoir arrêter la partie si le joueur se trompe plus de 11 fois.

Il lui faudra aussi une variable `lettres_trouvees` pour se souvenir des lettres justes proposées par le joueur.

Et pour finir, une variable `mot` qui contient la solution avec des `"_"` à la place des lettres que le joueur n'a pas trouvés.

## IV. Déroulement du jeu

À chaque fois que nous afficherons une nouvelle étape de notre pendu, nous effacerons le contenu de l'invite de commandes grâce à `os.system("clear")`

L'affichage de votre jeu en cours de partie doit ressembler à ça :

```

      | |   //   |
      | |   //   |
      | |   //
      | |  //
      | |
      | |
      | |
      | |
=====
- - - - -

```

<message>

Proposez une lettre :

- Si la lettre était incorrecte, on affiche le message : "Lettre incorrecte" et on ajoute une erreur
- Si la lettre précédente était correcte, on affiche le message ! "Lettre correcte" et on affiche sa (ou ses) occurrences dans le mot
- Si la lettre est déjà dans le mot, on affiche le message : "Lettre déjà proposée" et on ne compte pas de faute
- Si le joueur a déjà proposé la lettre mais qu'elle était fausse, il peut la reproposer et ça lui comptera une erreur de plus
- En début de partie, on n'affiche aucun message

Si le joueur trouve toutes les lettres du mot, on affiche ">>> Gagné! <<<" et on quitte le jeu.

Sinon on affiche ">>> Perdu! <<<" et ">>> Le mot était : .... <<<"

Quoi qu'il arrive, avant de quitter le jeu, on affiche toujours " *Fin de la partie* "

## Exemples:

```

      | |   //   |
      | |   //   |
      | |   //   |
      | |//      ( _ )
      | |
      | |
      | |
      | |
=====
S O I F

>>> Gagné! <<<
* Fin de la partie *
```

```

      | |   //   |
      | |   //   |
      | |   //   |
      | |//      ( _ )
      | |        \|/
      | |        |
      | |        /\
      | |
=====
- - - - -

>>> Perdu! <<<
>>> Le mot était : guepe <<<
* Fin de la partie *
```