Devine !

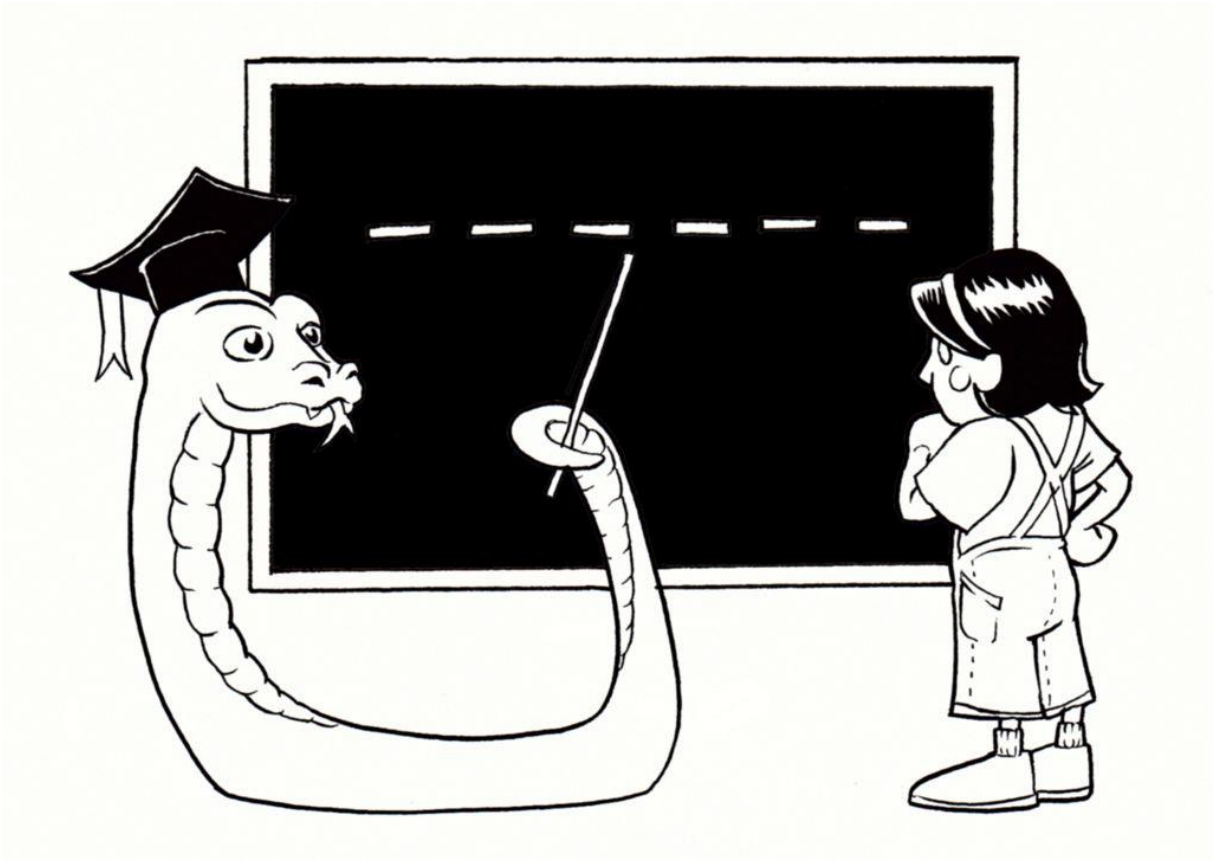
version 1.1.0

DIVERSITY by EPITECH

I. Introduction

Audrey est une jeune élève du secondaire 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.



Audrey a choisi Maître Python pour lui apprendre le jeu

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. Les bases du Python

Le python est un langage de programmation qui est apparu au début des années 90, il se caractérise par sa structure orientée objet comme le Java mais il est beaucoup plus abordable dans sa syntaxe et il est plus facile d'utilisation pour faire de l'algorithmie !

Audrey n'a jamais fait de python, elle se renseigne ici. Pour débuter, elle crée un fichier qu'elle nomme **pendu.py.** Dans ce fichier, elle décide de commencer son premier programme pour afficher le contenu d'une variable dans le terminal.

IV. Création des variables du jeu

Pour débuter, Audrey va devoir choisir le mot qui devra être deviné et va le mettre dans une variable appelé « solution » pour ne pas le perdre. Il pourra être changé plus tard. Il faudra ensuite qu'elle implémente une seconde variable appelée « tours » afin de définir le nombre possible d'essais pour le joueur. Cette seconde variable définira quand la partie sera terminée.

Il faudra aussi créer deux variables de texte vide qui serviront à l'affichage du mot avec les lettres trouvées et la liste des lettres trouvées.

```
affichage = ""
lettres_trouvees = ""
```

Maintenant que les variables sont créées, Audrey va pouvoir commencer à coder son jeu !

V. Que peut être ce mot ?

Dans le jeu du pendu, à chaque fin d'un tour, on montre au joueur le mot avec les lettres qui ont été trouvées et des traits pour les lettres qui restent à deviner.

Dans un premier temps, il faut faire une boucle sur le nombre de lettres du mot à deviner et compléter la variable d'affichage avec des tirets pour chaque lettre. À la fin, on aura un résultat équivalent à celui-ci : « _ _ _ _ _ _ » pour un mot de six lettres.

VI. Jouons tous ensemble à un petit jeu

a. Et que ça tourne !

Tant qu'Audrey n'aura pas épuisé tous ses essais, la partie continue. Elle a donc besoin d'une boucle qui gérera les actions à effectuer à chaque tour.

```
while ...                        # La boucle à besoin d'une condition
    print(...)
    proposition = input(...).lower() #Le "lower" est pour transformer
                                     #la lettre en minuscule
```

Le principe est le suivant : Tant que la partie n'est pas terminée, le programme affiche le mot qu'il faut deviner avec les lettres déjà trouvées puis attend qu'Audrey en entre une.

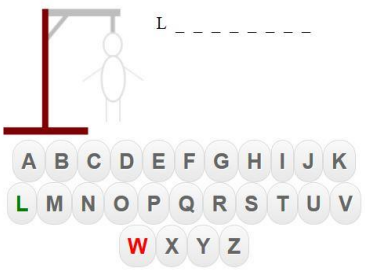
b. Audrey a-t-elle trouvée la bonne lettre ?

Si la proposition d'Audrey est dans le mot à trouver, elle va devoir être ajoutée dans la liste des lettres trouvées et indiquer à Audrey que la lettre trouvée est dans le mot cherché.

```
    if ...           # Condition "Si"
        lettres_trouvees = lettres_trouvees + ...
        ...
```

Et si la lettre n'est pas dans le mot, il faut enlever un essais à Audrey et lui indiquer le nombre qu'il lui reste.

```
    else             # Condition "Ou"
        ...
```



Example d'un Pendu

c. Est-ce une victoire ?

Il faut maintenant redéfinir la variable d'affichage, car des lettres ont pu être trouvées dans le tour. Pour cela, il faut regarder que pour chaque lettre de la solution, si elle se trouve dans la liste des lettres trouvées, l'insérer dans la variable d'affichage ou sinon afficher "_ " si la lettre n'a pas été trouvée.

Pour terminer, dans la variable d'affichage s'il ne reste aucun "_" alors il faudrait l'indiquer par un message de victoire à Audrey puis sortir de la boucle de jeu.

```
    if "_" not in ...
        ...
```

Si la partie est terminée, il faudrait peut-être en informer Audrey !

VII. Conclusion

Félicitations, vous avez aidé Audrey à créer son jeu !

Pour aller plus loin, voici quelques idées :

* Prendre un mot aléatoire dans une liste définie
* Avoir plusieurs mots séparés par un espace
* Dessiner le pendu en fonction du nombre de tours restantes
* Afficher les lettres déjà testées