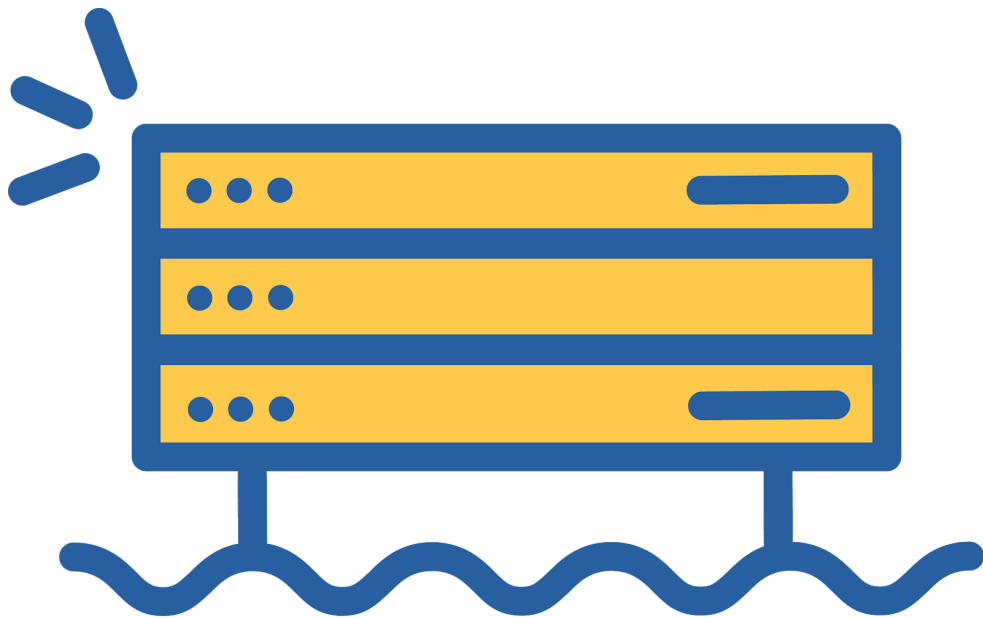

La Plateforme_

Python, des fichiers, des graphiques, de la récursivité et des regex ...

Python is powerful... and fast; and open; and ... many other things.



La Plateforme_

A decorative graphic at the bottom right of the slide, consisting of several blue geometric shapes, including a large triangle and a rectangle, arranged in a way that suggests a modern, architectural design.

Job 01

Créer un programme qui demande à l'utilisateur de renseigner une chaîne de caractères et qui écrit cette chaîne de caractères dans un fichier "output.txt"

Job 01

Créer un programme qui lit le contenu du fichier "output.txt" et qui l'affiche dans le terminal

Job 02

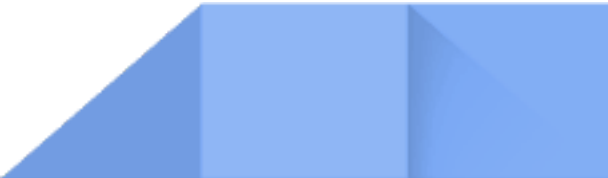
Créer un programme qui parcourt le contenu du fichier "data.txt" et qui compte le nombre de mots qui s'y trouvent.

Job 03

Créer un programme qui demande à l'utilisateur de renseigner un nombre entier. Le programme devra alors parcourir le contenu du fichier "data.txt" compter le nombre de mots de la taille renseignée qui s'y trouvent.

Job 05

Ecrire un programme qui parcourt le fichier "data.txt" et qui compte le nombre d'occurrence de chaque lettre (Minuscules et Capitales comptent pour la même lettre). A l'aide du module Matplotlib, générer un histogramme représentant le pourcentage d'apparition de chaque lettre.



Job 08

Ecrire un programme qui parcourt le fichier "data.txt" et qui compte le nombre de mots de chaque taille. A l'aide du module Matplotlib, générer un histogramme représentant le pourcentage d'apparition de chaque taille de mot.

Job 13

Ecrire un programme qui parcourt le fichier "data.txt" et qui compte le nombre d'occurrence de chaque lettre (Minuscules et Capitales comptent pour la même lettre) en début de mot. A l'aide du module Matplotlib, générer un histogramme représentant le pourcentage de présence de chaque lettre en début de mot.

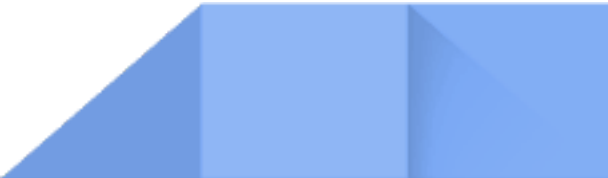
Job 21

Ecrire un programme qui parcourt le fichier "data.txt" et qui, pour chaque lettre, compte le nombre d'occurrence de la lettre suivante. Générer, ensuite, un graphique de courbes superposées, une courbe par lettre, montrant le pourcentage d'apparition de chaque lettre la suivant.

Par exemple, pour le a: a(2%), b(5%), c(2.3%) pour le b: a(3%), b(0%), c(1%), ...

Job 34

Créer un générateur de mots en vous appuyant sur les statistiques précédemment calculées (longueur des mots, première lettre des mots, enchaînement de lettres).



Job 55

En analysant à nouveau le fichier “Data.txt”, établissez des statistiques synthétisant le nombre de mots par phrase. Produisez un histogramme présentant ces statistiques, puis, à l’aide de votre générateur de mot, créez un générateur de phrases à consonance de “Lorem Ipsum”.

Job 89

Créer un programme demandant à l’utilisateur de renseigner un nombre entier. Votre programme devra calculer la factorielle de ce nombre, sans utiliser de fonction autre que les vôtres. Attention, vous ne devez utiliser ni while, ni for, ni foreach ni ... boucle. Seulement de la récursivité.

Job 144

Créer un programme qui modélise un plateau de jeu, carré, de $n \times n$ cases. Placez sur ce plateau n dames de manière à ce qu’aucune dame ne puisse se “prendre”, quand cela est possible. La valeur de n est renseignée par l’utilisateur. Quand cela est possible, le programme devra afficher dans le terminal le plateau de jeu avec le caractère ‘O’ pour les cases vides et le caractère ‘X’ pour représenter les dames.

Job 233

Un pokemon anglais se cache dans “data.txt” ... lequel est-il ?

