

# Jour 3 - Regex & Graphique

Python, des fichiers, des graphiques et des regex

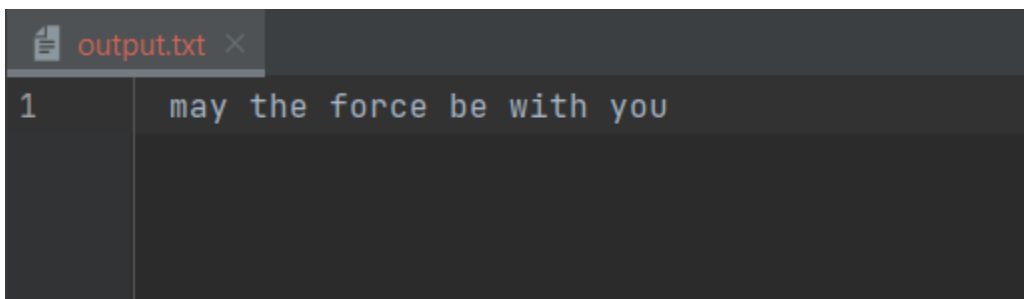


## Job 0

---

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

Entrer une chaîne de caractères *may the force be with you*



```
output.txt x
1 may the force be with you
```

## Job 0.1

---

Créer un programme qui lit le contenu du fichier “**output.txt**” et qui l’affiche dans le terminal.

```
may the force be with you  
  
Process finished with exit code 0
```

## Job 01

---

Créer un programme qui parcourt le contenu du fichier “[domains.xml](#)” et qui compte le nombre d’extension de domaines qui s’y trouvent (.com, .net, etc ...).

## Job 01

---

Créer un programme qui parcourt le contenu du fichier “[data.txt](#)” et qui compte le nombre de mots (sans caractère spéciaux) qui s’y trouvent.

## Job 02

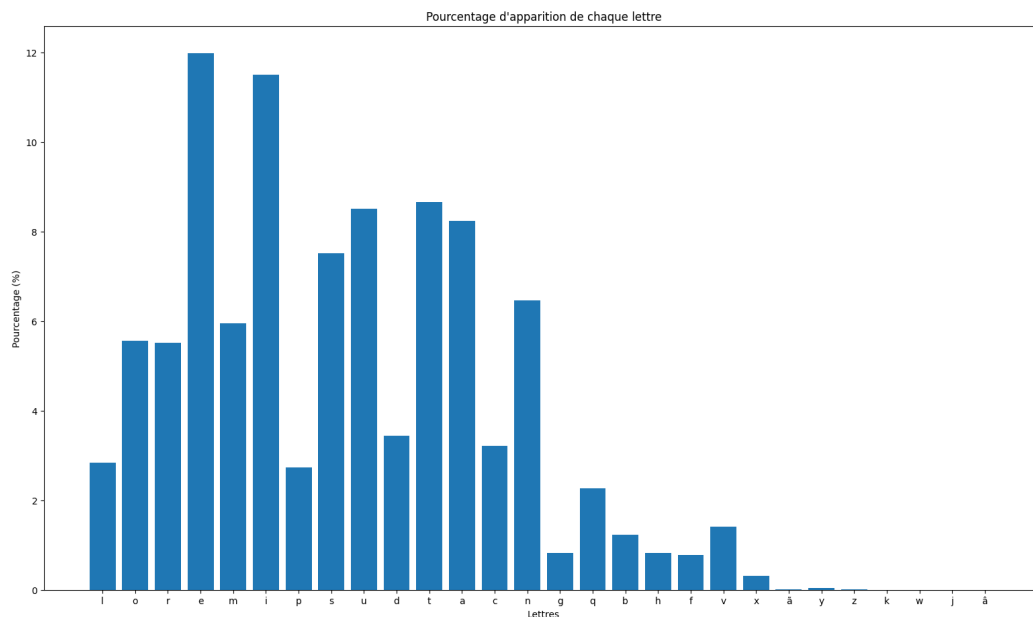
---

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 03

---

Écrire 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 05

---

Écrire 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 08

---

Écrire 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 13

---

Écrire 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 21

---

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

## Job 34

---

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 55

---

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

## Rendu

Dans votre répertoire github “**runtrack-python**”, créez un dossier “**jour03**” et dans ce dossier, pour chaque job, un dossier “**jobXX**” où XX est le numéro du job. N’oubliez pas d’envoyer vos modifications dès qu’une étape est avancée ou terminée et utilisez des commentaires explicites.

---

## Compétences visées

- Manipuler des fichiers en python
  - Maîtriser les Regex en python
  - Installer et utiliser la librairie graphique Matplotlib
- 

## Base de connaissances

- [Matplotlib](#)  
Site officiel matplotlib, documentation et téléchargement.
- [Python File Open](#)  
Manipuler des fichiers en python
- [Regex Python](#)  
Documentation d’utilisation de Regex en python