

Etude de la popularité des personnages de Nintendo dans la presse



Objectif pédagogique : Mettre en œuvre le corpus élémentaire en Python (affectations, conditions, boucles) pour analyser une chaîne de caractère.

Vous disposez dans une base de données contenant tous les textes des articles du magazine « Nintendo Magazine » paru de 2002 à 2007.

Consulter les ressources disponibles sur [picassciences](#).

Programme 1 : Déterminer la popularité des personnages mario, peach, wario. Plus un personnage est cité, plus il est populaire. Vous devez donc parcourir chaque ligne et décompter le nombre fois où le nom des personnages apparaît dans l'ensemble des numéros du magazine.

Vous utiliserez trois variables mario, peach et wario initialisées à 0, pour décompter le nombre d'apparition. Vous n'utiliserez pas de liste.

Programme 2 : Le but de l'activité est de trouver le premier article dans la base qui parle du personnage ou du jeu de votre choix et d'afficher l'URL de l'image de la page du magazine correspondante. Il suffit ensuite de la copier/coller dans un navigateur.

Programme 3 : Même consigne que l'étape 1 mais en initialisant la liste de personnages suivants :

```
Liste = ["mario", "peach", "wario", "luigi", "kirby", "link",  
"bowser", "yoshi", "ganon", "toad", "donkey kong", "diddy kong",  
"knuckles", "layton", "samus", "waluigi", "falcon", "baby mario"]
```

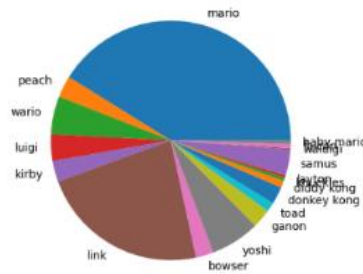
Cette liste est disponible sur [picassciences](#), inutile de la retaper à la main.

Vérifier que vos résultats sont cohérents avec ceux du programme 1 pour les trois premiers personnages.

Programme 4 : Compléter le programme 3 pour grapher les résultats selon un diagramme en camembert en utilisant matplotlib. Vous utiliserez une recherche sur Internet pour trouver comment réaliser un graphique avec matplotlib.

Voici le résultat attendu :

Popularité des personnages Nintendo dans Nintendo Magazine de 2002 à 2007*



« pie chart »

Programme 5 : Pour les élèves ultra rapides : Liens entre les personnages.

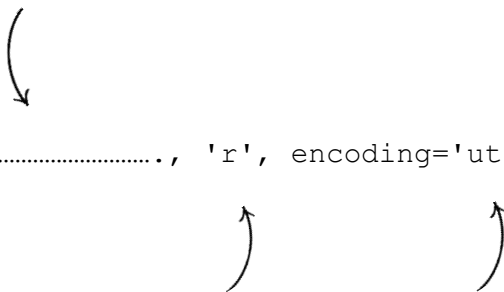
Vous avez surement remarqué que Knuckles est un personnage Sega et non Nintendo, remplacez-le dans la Liste par Zelda à la main.

L'idée est de voir ici quels sont les liens entre les personnages. On prend le personnage Ganon. On parcourt le texte d'une page, et on regarde quels personnages sont cités (ceux de la liste) dans une même page avec Ganon. Il est probable de trouver Link ou Zelda comme personnage cités avec Ganon, et peu probable de trouver Donkey Kong. (*même si il peut y avoir plusieurs articles dans une même page sur des sujets différents*)

On passe à la page suivante, on décompte au fur et à mesure le nombre d'apparition de chaque personnage de la liste. Une fois que tout fonctionne pour Ganon, on teste avec d'autres personnages.

Document 1 : Lire un fichier texte :

Ici, vous devez placer el chemin complet vers le fichier à charger. Je vous conseille de placer votre fichier dans **C:\Temp** (à créer s'il n'existe pas) à l'aide de l'explorateur de fichier de Windows, le chemin vers le fichier sera alors 'C:\\Temp\\Nintendo.txt'



```
fichier = open(....., 'r', encoding='utf8')
```

r pour « read » pour les caractères spéciaux

```
lignes = fichier.readlines()
```

Pour lire les premières lignes du fichier :

```
print("Ligne 1 : " + lignes[0])
print("Ligne 2 : " + lignes[1])
print("Ligne 3 : " + lignes[2])
print("Ligne 4 : " + lignes[3])
print("Ligne 5 : " + lignes[4])
print("Ligne 6 : " + lignes[5])
```

Et savoir combien il y a de lignes dans le fichier :

```
print(len(lignes))
```

Document 2 : Structure de la base de données.

Le texte des articles est organisé de la manière suivante :

Texte de la Page 1 ligne 1

Texte de la Page 1 ligne 2

Texte de la Page 1 ligne 3

Etc...

url:#le lien vers l'image de la page 1 du magazine n°xxx

Texte de la Page 2 ligne 1

Texte de la Page 2 ligne 2

Texte de la Page 2 ligne 3

Etc...

url:#le lien vers l'image de la page 2 du magazine n°xxx

Texte de la Page 3 ligne 1

Texte de la Page 3 ligne 2

Texte de la Page 3 ligne 3

Etc...

url:#le lien vers l'image de la page 3 du magazine n°xxx

Toutes les pages de tous les magazines sont ensuite empilées à la suite de la même manière pour chaque numéro de magazine. Ouvrez le fichier nintendo.txt (clic droit > ouvrir avec Notepad++) pour observer la structure.

Document 3 : Détecter la présence d'une chaîne de caractère dans une ligne.

```
Ligne1 = 'mario est un plombier'
Ligne2 = 'peach est une princesse'

if 'mario' in Ligne1:
    print('le mot présent dans la ligne')
else:
    print('le mot pas présent dans la ligne')

if 'peach' in Ligne2:
    print('le mot présent dans la ligne')
else:
    print('le mot pas présent dans la ligne')
```

Si votre condition se trouve dans une boucle, vous pouvez utiliser **break** pour stopper la boucle.

Programme 1 : Indice 1 : Vous devez créer une boucle for pour accéder à chacune des lignes. Il faut transformer les lignes suivantes avec une boucle, pour afficher par exemple toutes les lignes du fichier

```
print("Ligne 1 : " + lignes[0])
print("Ligne 2 : " + lignes[1])
print("Ligne 3 : " + lignes[2])
print("Ligne 4 : " + lignes[3])
print("Ligne 5 : " + lignes[4])
print("Ligne 6 : " + lignes[5])
```

```
for i in range (.....
```

Programme 1 : Indice 2 : Il faut utiliser une variable **compteur** avant la boucle, affectée à 0. Cette variable compteur sera augmentée (on dit incrémentée) d'une unité à chaque fois quand le mot 'mario' est contenu dans la ligne étudiée. (voir document)

Programme 2 : Indice 1 :

Vous devez créer

- Une variable Drapeau initialisée à 0.
- Une variable Recherche contenant le terme à rechercher
- Une boucle for pour accéder à chacune des lignes

Programme 2 : Indice 2 :

Dans la boucle, si la ligne étudiée contient le terme de recherche, alors, on passe la variable Drapeau à 1.

Exemple :

ligne 4967 :voici la carte complétée de super **mario** world! tous les niveaux

ligne 4968 : y apparaissent. ceux représentés par un point rouge ...

etc...

etc...

ligne 5025 :

```
url:http://download.abandonware.org/magazines/Nintendo%20Le%20Magazine%20Of
ficiel/nintendomagzineofficiel_numero01/Le Magazine Officiel Nintendo N°001
- Page 045 (Mai 2002).jpg
```

En ligne 4967, le drapeau est levé et passe à 1 car le terme de recherche « mario » est trouvé dans la ligne étudiée. On sait maintenant qu'il va falloir montrer cette page à l'utilisateur. On doit donc continuer de parcourir en recherchant la ligne commençant par « url : »

On continue de parcourir les lignes suivantes 4968, 4969, 4970, ..., mais si le drapeau est levé on **ajoute une condition** supplémentaire dans la boucle. Si la ligne étudiée contient « url : » alors on s'arrête avec un break et on affiche la ligne en question, c'est le cas en ligne 5025.

Programme 3 : Indice 1 :

Pour pouvoir décompter le nombre de chaque personnage, vous devez commencer par créer une liste `Résultat`, ne contenant que des 0.

Votre algorithme aura pour but **d'incrémenter d'une unité** ces valeurs, à chaque fois qu'un élément de `Liste` est découvert dans la ligne étudiée.

A la fin, voici comment la liste `Résultat` est remplie :

```
Liste : ['mario', 'peach', 'wario', 'luigi', 'kirby', 'link', 'bowser',  
'yoshi', 'ganon', 'toad', 'donkey kong', 'diddy kong', 'knuckles',  
'layton', 'samus', 'waluigi', 'falcon', 'baby mario']
```

```
Resultat : [980, 69, 122, 86, 67, 543, 57, 158, 61, 30, 56, 22, 11, 8, 83, 4, 14,  
11]
```

Chaque position de `Résultat` correspond aux positions de `Liste`.

Programme 3 : Indice 2 :

Au final, il faut utiliser une double boucle imbriquée :

- Une boucle pour parcourir les lignes come dans l'exercice 1
- Une autre boucle pour parcourir la liste des personnages

```
for i in range(0,len(lignes)):  
    for j in range(0,len(Liste)):
```

Programme 5 : Les étapes :

- Faire un algorithme qui découpe le texte des pages pour les mettre dans une liste (une case d'une liste contient tout le texte d'une page). On initialise une chaîne de caractère vide et on parcourt les lignes une à une en mettant chaque ligne dans la chaîne de caractère (concaténation). Quand on trouve le marqueur de fin de page « url : », on place le contenu de la chaîne de caractère dans une liste avec la méthode `append()`. On vide la chaîne de caractère et on continue. Pour vérifier que vous avez bien découpé le bon nombre de page, il y a 2129 pages (c'est donc la longueur `len()` de la liste produite)
- Créer l'algorithme de recherche qui doit retourner la liste des résultats

Personnages proches de ganon

```
Resultat : [7, 2, 2, 0, 2, 28, 1, 2, 34, 1, 2, 0, 24, 0, 1, 0, 1, 0]
```

- Même si on peut lire les résultats à la main, créer un algorithme pour afficher les résultats de manière synthétique comme ceci. Indices : On recherche la valeur maximale de la liste avec la fonction intégrée de Python `max(Resultat)` et on parcourt à partir de cette valeur maximale en décroissant vers 0 les valeurs de résultats pour les afficher dans l'ordre du plus grand au plus petit

Personnages proches de ganon

```
Resultat : [7, 2, 2, 0, 2, 28, 1, 2, 34, 1, 2, 0, 24, 0, 1, 0, 1, 0]
```

```
ganon - link - zelda - mario - peach - wario - kirby - yoshi - donkey kong  
- bowser - toad - samus - falcon -
```