
Exercice 1 Préparation

1. Copier le sous-dossier `Activite1` du dossier `ICN\Cartes` situé dans le partage `Donnees` de sa classe. Coller ce dossier dans son espace personnel.
2. `Activite1` contient un sous-dossier `ressources` avec les fichiers textes que nous allons manipuler. Ouvrir Pyzo, enregistrer dans `ressources` un fichier `Texte-Activite1.py`, puis exécuter ce fichier avec l'option `Run as script`.

1 Géomatique

Exercice 2

1. Ouvrir avec un navigateur la carte d'URL :

https://drive.google.com/open?id=1_AxFET51kqblnzU45RI0yhhwg8U&usp=sharing

Des icônes permettent de visualiser le candidat arrivé en tête lors du scrutin du premier tour de l'élection présidentielle 2017, pour chacun des 896 bureaux de vote de la commune de Paris.

L'objectif de ce parcours est de réaliser un calque au format KML qui peut être projeté sur une carte grâce à des logiciels de géomatique comme Gmap ou GoogleEarth. La démarche se décomposera en deux étapes :

- ☞ *Acquisition et traitement* de plusieurs jeux de données :
 - résultats d'élections par bureaux de votes à Paris ;
 - coordonnées géographiques (Latitude, Longitude) des bureaux de vote parisiens.
- ☞ *Intégration* des données traitées (résultats et coordonnées géographiques) dans un calque pour fonds de carte au format KML (Keyhole Markup Language) qu'on peut importer dans Gmap ou GoogleEarth.

2. Définir la *géomatique* et donner des exemples d'application.

2 Open Data

Exercice 3

1. L'expression *Open Data* désigne des données massives librement utilisables, réutilisables et pouvant être redistribuées par tous.
 - a. Quels usages peut-on faire de données libres ?
 - b. Quelles difficultés juridiques et techniques rencontre-t-on dans le mouvement *Open Data* ?
 - c. La métropole de Lyon propose plusieurs types de licences d'accès aux données, pourquoi ?

Ressources :

https://interstices.info/jcms/int_71721/l-open-data-l-ouverture-des-donnees-pour-de-nouveaux-

<http://owni.fr/dossiers/open-data-opendata/index.html>

<https://opendata.paris.fr/page/lademarche/>

<https://data.grandlyon.com/comprendre-la-demarche/>

2. Le jeu de données contenant les résultats du premier tour de l'élection présidentielle 2017 dans tous les bureaux de vote de Paris est accessible depuis l'URL :

https://opendata.paris.fr/explore/dataset/resultats_electoraux/

- Sous quelle licence est publié ce jeu de données et que contient-il ?
 - Quels sont les formats sous lesquels on peut l'exporter ?
 - Utiliser l'outil d'analyse pour afficher le nombre moyen de votes blancs par bureau de vote en ordonnée pour chaque arrondissement en abscisse, en ventilant la série des scrutins.
3. Dans le dossier Ressources se trouve le fichier `bureaux-de-votes.csv` avec l'export au format CSV (Comma Separated Values) du jeu de données.

Chaque ligne se compose de 17 champs d'information séparés par des points virgules.

On peut ouvrir ce fichier avec un éditeur de textes comme Notepad. `filtrageElection(source, election)` Les colonnes sont indexées à partir de 0 pour le libellé du scrutin) à 16 pour le nombre de procurations.

- Quels sont les index des colonnes contenant les informations suivantes : identifiant du bureau, , nombre de suffrages exprimés, identifiant du candidat (numéro de dépôt), nom du candidat, prénom du candidat, nombre de voix du candidat ?
- Que représente chaque ligne du fichier ? Comment les lignes sont-elles classées ?

3 Traitement des données

Exercice 4 Premier filtrage

```
def filtrageElection(source, sep, election):  
    f = open(source, 'r')  
    t = []  
    for ligne in f:  
        # à compléter  
    f.close()  
    return t
```

La fonction `filtrage1(source, sep, election)` prend en arguments un fichier texte `source` au format CSV, un caractère `sep` et une chaîne de caractères `election`. On l'utilisera ainsi :

```
In [4]: t = filtrage1('resultats_electoraux.csv', ';', 'Présidentielle 2017 - 1er tour')  
)
```

La fonction doit découper chaque ligne en une liste de champs selon le caractère `sep` et retourner une liste contenant ces listes de champs.

On rappelle la méthode `split` permettant de découper une chaîne de caractères.

```
In [5]: cid = 'viens, mon fils, viens, mon sang, viens réparer ma honte'  
  
In [6]: cid.split(',')  
Out[6]: ['viens', ' mon fils', ' viens', ' mon sang', ' viens réparer ma honte']
```

Recopier et compléter le code de la fonction `filtrage1`.

Exercice 5 Second filtrage

La fonction `filtrage2(t)` prend en argument une liste `t` retournée par la fonction `filtrage1`. Chaque élément de `t` est la liste des champs extraits d'une ligne du fichier `source` : ce sont les résultats dans l'un des 896 bureaux de vote parisiens de l'un des onze candidats au premier tour de l'élection présidentielle 2017. `filtrage2` remplace chaque élément de `t` par une liste comportant cinq informations :

- un couple constitué du numéro d'arrondissement et du numéro du bureau de vote ;
- le numéro de dépôt du candidat¹ ;
- son nom ;
- son prénom ;
- son score qui est le quotient entre son nombre de voix et le nombre de suffrages exprimés.

```
In [11]: data = filtrage2(t)

In [12]: data[0]
Out[12]: [(17, 20), 10, 'ASSELINEAU', 'François', 0.029978586723768737]
```

Recopier et compléter le code de la fonction `filtrage2`.

```
def filtrage2(t):
    data = []
    for ligne in t:
        arrondissement = int(ligne[2])
        bureau = int(ligne[3])
        id_bureau = (arrondissement, bureau)
        # à compléter
        data.append([id_bureau, id_candidat, nom_candidat, prenom_candidat, score])
    return data
```

Exercice 6 Tri final

Écrire une fonction `premierParBureau(data)` qui prend en argument une liste `data` retournée par la fonction `filtrage2` et qui retourne une liste avec pour chaque bureau de vote parisien la seule ligne de `data` correspondant au candidat arrivé en première position sur ce bureau. Par exemple pour le premier bureau du premier arrondissement, on obtient :

```
In [17]: lespremiers = premierParBureau(data)

In [18]: lespremiers[0]
Out[18]: [(1, 1), 3, 'MACRON', 'Emmanuel', 0.38074398249452956]
```

Quelques rappels :

- ☞ La fonction `sorted` permet de trier une liste. Le tri peut s'effectuer selon les valeurs d'une fonction passée comme argument. Par défaut il est croissant mais on peut l'inverser avec le paramètre optionnel `reverse`.

On donne ci-dessous une liste de notes d'élèves et leur tri par moyennes croissantes puis décroissantes.

```
In [19]: maths = [('Ismael', 18, 10, 15), ('Jean', 12, 8, 11), ('Isaac', 12, 13, 14)]

In [20]: sorted(maths, key = lambda t : (t[1] + t[2] + t[3])/3)
Out[20]: [('Jean', 12, 8, 11), ('Isaac', 12, 13, 14), ('Ismael', 18, 10, 15)]

In [21]: sorted(maths, key = lambda t : (t[1] + t[2] + t[3])/3, reverse = True)
Out[21]: [('Ismael', 18, 10, 15), ('Isaac', 12, 13, 14), ('Jean', 12, 8, 11)]
```

1. Voir <https://presidentielle2017.conseil-constitutionnel.fr/liste-officielle-des-candidats-a-lelection-presidentielle>.

☞ Pour parcourir une liste avec un pas différent de 1 on peut passer un troisième argument à range :

```
In [23]: for k in range(1, 23, 7):
...:     print(k, end=';')
```

1;8;15;22

4 Enjeux sociétaux et droit du numérique

Exercice 7 Sujets d'exposés

Par groupe de trois, traiter l'un des sujets suivants sous forme de présentation numérique (diaporama Impress, page web HTML, prezi ...) que vous devrez présenter en dix minutes devant la classe au cours du premier trimestre. Pour la réalisation de diaporama avec Impress, la ressource suivante est très bien : <https://dane.ac-lyon.fr/spip/Video-Prise-en-main-d-Impress>.

- Traitement informatique de données personnelles, que dit la *loi informatique et liberté* et dans quels cas s'applique-t-elle?

Ressources :

<https://www.cnil.fr/fr/informatique-et-libertes-suis-je-concerne>

<https://www.cnil.fr/comprendre-vos-droits>

<http://eduscol.education.fr/internet-responsable/ressources/legamedia/donnees-personnelles-suhtml>

- *Big data*, un nouvel eldorado économique?

Ressources :

https://interstices.info/jcms/p_84069/les-donnees-en-question

<https://www.alternatives-economiques.fr/promesses-tres-commerciales-big-data/00010056>

- *Big Data* et démocratie : un danger ou une chance à saisir pour le citoyen ?

Ressources :

https://interstices.info/jcms/ni_76920/un-deluge-de-donnees

https://interstices.info/jcms/int_71721/l-open-data-l-ouverture-des-donnees-pour-de-nouveaux-

<http://owni.fr/2011/04/06/dis-papa-cest-quoi-lopen-data/index.html>

- Journalisme de données : histoire et perspectives.

Ressources :

<http://www.ohmybox.info/datajournalisme.html>

<http://www.ohmybox.info/tag/datajournalisme/>

- Journalisme automatique, Génération Automatique de Textes (GAT) : les robots, substitués ou compléments des journalistes ?

Ressources :

<http://www.ohmybox.info/generation-automatique-de-textes-et-journalisme-une-ligne-du-temps/>

<http://www.cnetfrance.fr/news/algorithmes-et-robots-journalistes-l-avenir-de-la-presse-398456htm>

<http://www.ohmybox.info/petite-histoire-de-la-generation-automatique-de-textes/>