Programmation sous SIG

Arcpy (module principal)

Objectifs :

— savoir parcourir une arborescence, un espace de travail
— être capable de vérifier qu'un jeu de données existe

Préalables

Franc'O est un organisme chargé de la collecte et de la diffusion des données de nature hydrographique sur l'ensemble du territoire français. L'organisme est organisé en zones de collectes au sein desquelles un agence locale effectue des saisies de données géographiques. L'ensemble des données sont ensuite centralisée au siège de l'organisme à Paris.

Malheureusement, les erreurs de livraison sont fréquentes. Aussi avant d'intégrer les données des agences locales, le siège parisien effectue un ensemble de contrôle sur les données reçues.

Les données sont échangées sous forme de géodatabases Esri. En fonction des anomalies détectées, les lots sont corrigés ou renvoyés en région pour correction.

En bout de chaîne, après avoir intégré les données à la base nationale et effectué divers des analyses Franc'O redistribue les données sous forme de shapefiles aux communes qui en font la demande.

Nous sommes chargées de l'automatisation des chaînes de contrôle et d'export des données.

Les ressources suivantes vous seront utiles pour réaliser ce TD. N'hésitez pas à les consulter.
— *Memo ArcPy* du cours de programmation sous SIG.
— Doc d'ArcPy : http://desktop.arcgis.com/fr/arcmap/lastest/analyze/arcpy

⇒ Quels types de développements vous paraissent pertinents pour réaliser l'automatisation des chaînes de traitement de Franc'O ?

Réponse .....
.....

1 Contrôle des géodatabases

La structure attendue des géodatabases est décrite à l'aide du dictionnaire Python de la forme suivante :

```
DICO_GDB_STRUCT = {
    "ADMINISTRATIF": {
        "COMMUNE": {
            "NOM": ["String", 50],
            "CODE_INSEE": ["Integer", 5],
            "STATUT": ["String", 200]
        }
    },
    "HYDROGRAPHIE": {
        "COURS_D_EAU": {
            "CODE_HYDRO": ["String", 8],
            "CLASSE": ["String", 1],
            "TOPONYME": ["String", 127],
            "REGIME": ["String", 50]
        }
    }
}
```

Il se lit de la manière suivante : le jeu de classes d'entités `ADMINISTATIF` contient une classe d'entités `COMMUNE` possédant trois champs : `NOM`, de type `String` et de longueur 50, `CODE_INSEE`, de type `Integer` et de longueur 5, et `STATUT`, de type `String` et de longueur 200, etc.

Les résultats du contrôle doivent être écrits dans un fichier de log. Les informations y sont rangées en fonction de leur importance. Trois niveaux seront utilisés :

- les erreurs (importance = 3) :
  - jeux de classes d'entités et classes d'entités en manquants ;
  - pour chaque classe d'entités, champs manquants ;
  - pour chaque champ, type et/ou longueur incorrects ;
- les erreurs non bloquantes (importance = 2) :
  - jeux de classes d'entités et classes d'entités en trop ;
  - pour chaque classe d'entités, champs en trop ;
- les simples informations (importance = 1) :
  - jeux de classes d'entités et classes d'entités présents ;
  - pour chaque classe d'entités, champs présents ;

Le log aura finalement la forme suivante :

```
24/08/17 18:10:56 : Début des contrôles
Chemin de la géodatabase : D:\ProgSIG\TD4\data\BaseHydro.gdb

[Info]                Jeu de classes d'entités ADMINISTRATIF présent dans la géodatabase
[Info]                Classe d'entités COMMUNE présente dans le jeu de classes d'entités ADMINISTRATIF
[Info]                Champ NOM présent dans la table COMMUNE
[ERREUR]        Longueur du champ NOM de la table COMMUNE incorrect (attendu : 50, obtenu : 45)
[Info]                Champ CODE_INSEE présent dans la table COMMUNE
[ERREUR]        Type du champ CODE_INSEE de la table COMMUNE incorrect (attendu : Integer, obtenu : String)
[Info]                Champ STATUT présent dans la table COMMUNE
[Attention]         Champ POPUL (Integer, 4) non requis dans la table COMMUNE
[Attention]         Classes d'entités TRONCON_HYDRO non requise dans le jeu de classes d'entités ADMINISTRATIF

24/08/17 18:11:01 Fin des contrôles
Nombre d'erreurs : 2
```

⇒ Dans un script Python, écrivez une fonction `write_log(msg, flog, importance=0)` écrivant le message `msg`

dans un fichier texte `flog` avec une importance `importance`.

⇒ Développez un utilitaire permettant de détecter les jeux de classes d'entités et classes d'entités absents de la géodatabase alors qu'ils y sont attendus. Vous utiliserez la fonction précédente pour écrire les résultats dans un fichier texte.

⇒ Complétez votre programme précédent en ajoutant le contrôle des champs (présents, absents, type et longueur).

N'hésitez pas à découper votre code en fonctions pour le rendre plus clair et plus facilement maintenable !

⇒ Enfin ajoutez un dernier contrôle permettant de détecter les jeux de classes d'entités et classes d'entités en trop dans la géodatabase.

2 Export des données

⇒ Réalisez un module permettant l'export sous forme de shapefile des données d'une commune (indiquée par son code insee)