

SIG : Geoinformatique

Exercice 2 : Utilisation de Python pour extraire & projeter des géodonnées

L'entreprise qui vous emploie (stagiaire) fait de la prospection sur la ville de Lausanne et région pour promouvoir l'activité physique et le sport en général. Pour ce faire, ils ont récupéré le géoréférencement de plusieurs institutions sportives (centres sportifs, fitness, activités sportives spécifiques, emplacement des parcs etc.) grâce à la banque de donnée libre de droit OpenStreetMap (OSM).

La librairie OSM fourni des données dans le système de géoréférencement WGS84, un système de référencement en degrés. Afin de pouvoir effectuer différents calculs sur ces données, l'entreprise vous demande de changer préalablement le système de coordonnées géographique de ces couches en CH1903+, qui est le système de coordonnées Suisse, en mètres. L'entreprise précise que les données qui vous ont été fournies ne représentent qu'un échantillon des couches à leur disposition, ils possèdent plus de 100 couches dans leur base de données. Ils aimeraient que vous changiez le CRS de toutes leurs couches le plus vite possible.

Comme vous maîtrisez un peu python et que vous suivez le cours de SIG, vous décidez de créer un script Python pour pouvoir vous débarrasser rapidement de ce travail pénible et aller boire des bières à SAT.

Les différents fichiers à transformer et ouvrir sur QGIS sont sur Moodle dans **Exercice_2_data**. Vous y trouverez aussi un code de base à remplir (*exercice_2_code_initial.py*) et utiliser comme point de départ pour votre code.

Avant de commencer à coder, **créer un projet QGIS vierge** avec comme système de coordonnées **espg :2056** et sauvegardez ce projet dans le même dossier que les couches qui vous ont été fournies par l'entreprise.

Vous allez donc créer un script Python qui va effectuer les étapes suivantes dans l'ordre :

1. Récupérer les noms des fichiers présents dans le même dossier que votre projet.
2. Faire une boucle sur ces fichiers afin de regarder si leur extension est « .shp » ou « .gpkg » .
3. Si c'est le cas, lire le fichier en tant que QgisVectorLayer et regarder si le crs est différent de celui du projet. Si c'est le même crs, vous pouvez simplement ajouter la couche au projet. Si le CRS est différent, veuillez transformer la couche pour la projeter dans le crs du projet. Une fois que c'est fait, vous pouvez ajouter la nouvelle couche (projétée dans le bon crs) au projet.

Afin de lister tous les fichiers présents dans un dossier (par le biais du chemin du dossier), vous pouvez importer la librairie **os** et utiliser la fonction **listdir()** tel que :

```
os.listdir(dossier_projet_chemin)
```

Cette fonction vous retourne une liste des différents fichiers présents dans **dossier_projet_chemin**.

Pour projeter une couche avec PyQgis, vous devez faire appel à une fonction de la processing toolbox , la fonction suivante : **reprojectlayer**.

Pour ce faire, vous devez importer la librairie suivante :

```
from processing.core.Processing import Processing
from processing.tools import *
```

Qui vous permet de faire appel aux différentes fonctions de la processing toolbox. Après quelques recherches sur internet, vous trouvez sur le lien suivant :

https://docs.qgis.org/3.4/en/docs/user_manual/processing/console.html

que la fonction suivante vous permet de faire cette transformation :

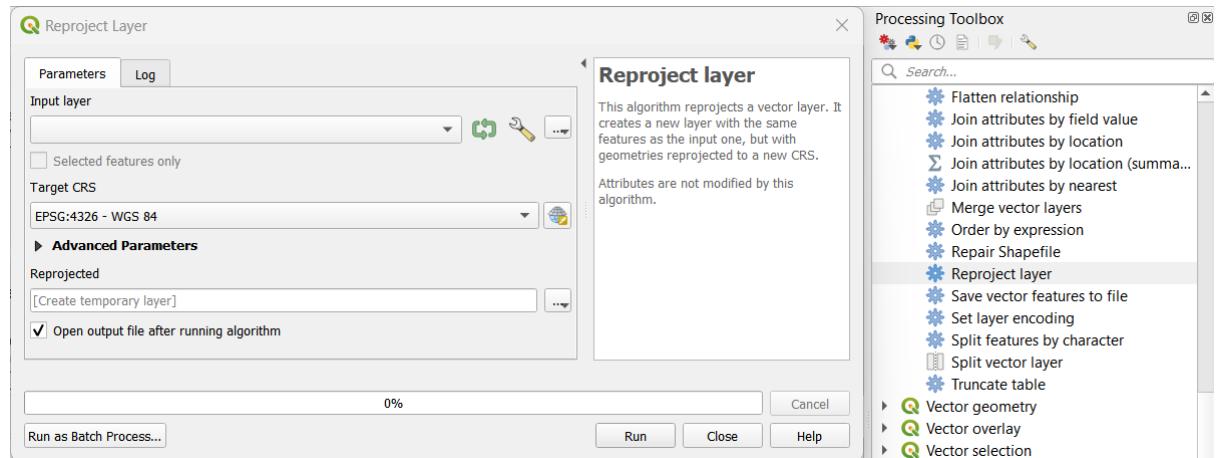
```
def project_layer(layer, target_crs, layer_name):

    Processing.initialize()
    params = {
        'INPUT': layer,
        'TARGET_CRS': target_crs,
        'OUTPUT': 'memory:'
    }

    reprojected_layer=processing.run('native:reprojectlayer',params)[ 'OUTPUT' ]
    reprojected_layer.setName(layer_name)

    return reprojected_layer
```

Cette fonction correspond exactement à la fonction « Reproject Layer » dans la processing toolbox et les paramètres dans params sont les même que les paramètres à indiquer sur QGIS :



La seule spécificité avec PyQgis est qu'il faut spécifier l'initialisation de la fonction de transformation avec `Processing.Initialize()`.