

Initiation à Python

LP ESSIG

2012 – 2013

1 Introduction à PyQgis

Ce TP est un survol des capacités Python incluses dans QGis. De façon basique, tout ce que vous pouvez faire à la souris est accessible en python. Et vous pouvez y rajouter toutes sortes de traitements numériques qui vous intéresse en vous servant de vos connaissances en Python.

Ce TP est plutôt axé sur les fonctionnalités fondamentales et la recherche d'informations (documentation) pour créer des scripts qui vous intéressent. L'avantage étant, bien entendu, la rapidité d'exécution d'un script comparé à l'utilisation de la souris et du clavier. À utiliser pour toute automatisation d'un traitement répétitif et récurrent, donc.

2 Présentation

2.1 La console Python

Python est une dépendance optionnelle de QGis, il se peut qu'il ne soit pas disponible sur toutes les installation. Mais si QGis a été installé avec Python, vous pouvez trouver son interpréteur dans le menu « Extensions » sous l'intitulé « Console Python ».

La console effectue un `import qgis, qgis.core, qgis.gui` de façon automatique. Pensez à les ajouter dans vos scripts si vous voulez vous servir de fonctionnalités de QGis sans avoir à lancer le programme.

2.2 Interface

QGis utilise Qt pour réaliser son interface graphique. Le module PyQt permet d'utiliser Python pour réaliser des interfaces graphiques en Qt et toutes les classes contenues dans `qgis.gui` sont des classes qui héritent de celle contenues dans PyQt. La création de fenêtres graphiques pour un plugin QGis en Python nécessite donc de connaitre Qt et/ou PyQt.

L'interface graphique principale de QGis est une instance de la classe `qgis.gui.QgisInterface`. Une instance de cette classe est déjà créée (puisque vous avez lancé l'interface graphique pour accéder à la console Python) et est disponible dans la variable `qgis.utils iface`.

2.3 La zone de visualisation

La zone de visualisation des couches est une instance de la classe `qgis.gui.QgsMapCanvas`. Toute interface possède un attribut de ce type qui est accessible grâce à la méthode `mapCanvas`.

Ce canvas possède des méthodes de manipulation de l'affichage (`zoomIn`, `zoomOut`...) et des méthodes pour accéder aux couches de données (`layerCount`, `layer`, `layers`...).

3 Recherche d'informations

`help(...)` et `dir(...)` vous seront d'un grand secours pour savoir ce qu'il est possible de faire avec un objet ou une classe particulière. Les noms de méthodes sont généralement assez

explicites pour comprendre leur comportement. L'aide vous guidera sur leur utilisation et les paramètres attendus.

4 Chargement des données

Le chargement des données se fait à l'aide de la classe `qgis.gui.QgisInterface`. Les méthodes `addVectorLayer` et `addRasterLayer` permettent de charger des couches de données vectorielles et matricielles.

`addVectorLayer` attend 3 paramètres : le chemin d'accès au fichier à charger, le nom à utiliser pour la couche une fois les données lues et le driver de lecture des données. Dans le doute, utilisez "ogr" pour ce dernier. "postgre" est aussi utilisable.

`addRasterLayer` n'attend que les deux premiers paramètres.

5 Manipulation des données

Pour pouvez récupérer une instance des objet python représentant les couches que vous avez chargée grâce au `qgis.gui.QgsMapCanvas`. La méthode `layer` prend en paramètre le numéro de la couche qui vous intéresse et retourne l'objet correspondant. La méthode `layers` retourne la liste de toutes les couches visibles.

Sur une couche vectorielle, la méthode `select` permet de charger dans la mémoire interne de l'objet python les informations relatives aux objets géométriques de cette couche. Suite à ça, la couche peut être parcourue comme une liste pour récupérer ses éléments un par un.

Chacun de ces élément est numéroté (accessible par la méthode `id`) et la définition de ses contours est accessible via la méthode `geometry`.

Pour sélectionner certains éléments d'une couche vectorielle, il faut fournir à la méthode `setSelectedFeatures` la liste des numéros des éléments à sélectionner.

Écrivez une fonction qui prend en paramètre le chemin d'accès vers un fichier contenant des données vectorielles. Cette fonction doit charger la couche correspondante avec, comme nom, le nom du fichier. Elle doit ensuite sélectionner toutes les entités contenues dans cette couche.

6 Plugins

Pour sauvegarder ses scripts, outre les fichier .py, QGis propose un système de plugins. De cette manière, les actions scriptées sont accessibles par un simple clic. Il est donc possible d'écrire un plugin qui permette de sélectionner toutes les entités de la couche courante.

À l'aide de l'installeur d'extensions python, il est possible d'installer un plugin nommé « Plugin Builder » qui facilite la création de plugins personnels.

À l'aide de « Plugin Builder », créez un plugin de sélection. Remplacez la méthode `run` du fichier de votre plugin avec votre code et suivez les instructions de « Plugin Builder » pour installer et tester votre plugin.