

PROGRAMMATION PYTHON AVEC QGIS: initiation



AZYAT Abdelilah
MBISD1_2022

PLAN

- Extensions (Plugins) Python
- Comment utiliser Python avec QGIS?
- Console Python
- Démarrer Console Python
- Variable iface
- Affichage de la table d'attributs
- Accéder aux entités
- Sélectionner des entités
- Utiliser une expression

Extensions (Plugins) Python

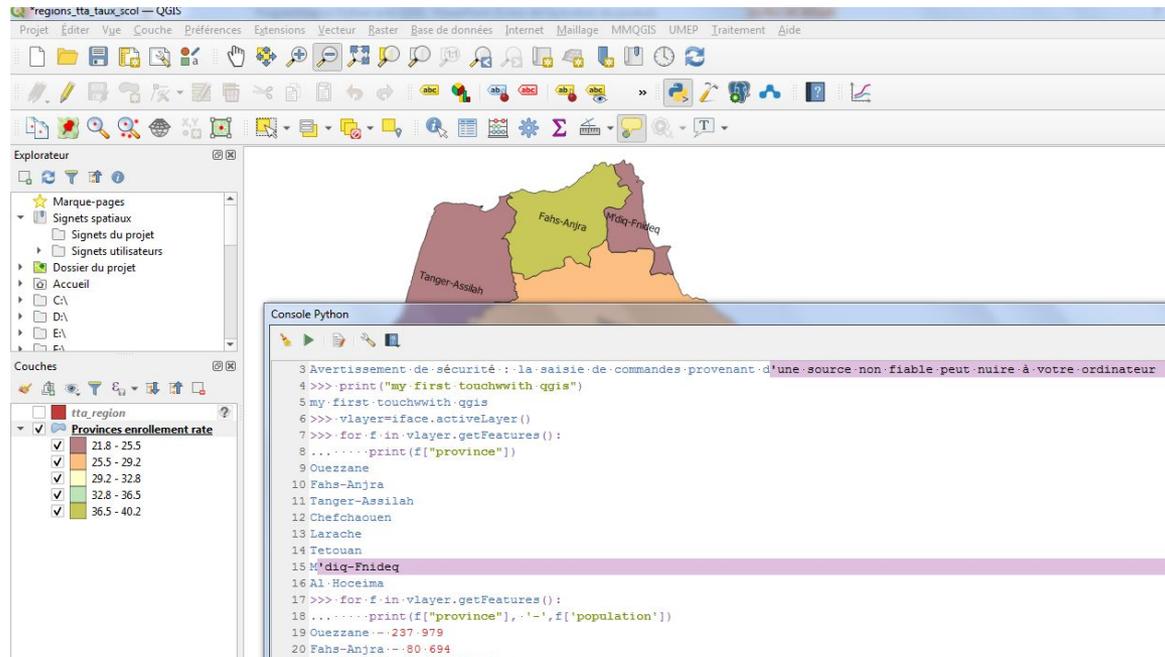
- Elles permettent d'enrichir ses fonctionnalités,
- De nombreuses extensions couvrant diverses fonctionnalités ont été écrites depuis l'introduction du support de Python.
- L'installateur d'extensions permet aux utilisateurs de facilement chercher, mettre à niveau et supprimer les extensions Python.

Comment utiliser Python avec QGIS?

- Nombreuses façons d'utiliser les dépendances Python de QGIS.
 - ✓ exécuter automatiquement un programme Python quand QGIS démarre,
 - ✓ lancer des commandes dans la console Python de QGIS;
 - ✓ créer et utiliser des plugins en Python.

Console Python

- Il est possible de tirer partie de la console Python intégrée pour créer des scripts et les exécuter.
- La console peut être ouverte grâce au menu: **Extension** ▶ **Console Python**.
- La console s'ouvre comme une fenêtre utilitaire non modale :



Démarrer Console Python

- Le QGIS possède une interface console pour exécuter le code python et affiche le résultat de ce code.
- Pour ouvrir cette console ; cliquer sur le menu Plugins □ Console Python
- Exemple; Tester la version du langage Python,



```
Console Python
1 Console Python
2 Utilisez iface pour accéder à l'interface API QGIS ou tapez help(iface) pour
3 Avertissement de sécurité : la saisie de commandes provenant d'une source non
4 >>> import sys
5 >>> print(sys.version)
6 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC.v.1914.64.bit (AMD64)]
7
```

Variable iface

- Pour interagir avec l'environnement de QGIS, il y a une variable **iface**, instance de la classe **QgsInterface**.
- Cette interface permet d'accéder au ;
 - canevas de carte,
 - menus,
 - barres d'outils
 - et autres composantes de l'application QGIS.

Utiliser la variable « *iface* »

- Pour interagir avec le QGIS on doit utiliser la variable « *iface* »,
- et pour accéder au couche actuellement actif dans QGIS, vous pouvez saisir ce qui suit et appuyer sur Entrée.
- Cette commande extrait la référence à la couche actuellement chargée et la stocke dans la variable de *layer*.

```
layer=iface.activeLayer()
```

```
dir(layer) # montre toutes les méthodes disponibles d'un objet.
```



```
Console Python
3 Alerte de sécurité : saisir des commandes à partir d'une source non approuvée peut mener à la perte et/ou la fuite de données
4 >>> layer=iface.activeLayer()
5 >>> dir(layer)
6 ['Actions', 'AddToSelection', 'AllStyleCategories', 'AttributeTable', 'Cross', 'CustomProperties', 'Diagrams', 'EditFailed', 'EditResult', 'EmptyGeometry', 'FastInsert', 'FeatureAvailability', 'FeaturesAvailable', 'FeaturesMaybeAvailable', 'FetchFeatureFailed', 'Fields', 'Flag', 'Flags', 'Forms', 'GeometryOptions', 'Identifiable', 'IntersectSelection', 'InvalidLayer', 'Labeling', 'LayerConfiguration', 'LayerFlag', 'LayerFlags', 'LayerOptions', 'LayerType', 'MapTips', 'MeshLayer', 'Metadata', 'NoFeaturesAvailable', 'NoMarker', 'PluginLayer', 'PropertyType', 'RasterLayer', 'RegeneratePrimaryKey', 'Removable', 'RemoveFromSelection', 'Rendering', 'Searchable', 'SelectBehavior', 'SemiTransparentCircle', 'SetSelection', 'SinkFlag', 'SinkFlags', 'Style', 'StyleCategories', 'StyleCategory', 'Success', 'Symbology', 'Symbology3D', 'VectorLayer', 'VertexMarkerType', '_bool_', '_class_', '_delattr_', '_dict_', '_dir_', '_doc_', '_eq_', '_format_', '_ge_', '_getattr_', '_getattribute_', '_gt_', '_hash_', '_init_', '_init_subclass_', '_le_', '_len_', '_lt_', '_module_', '_ne_', '_new_', '_reduce_', '_reduce_ex_', '_repr_', '_setattr_', '_sizeof_', '_str_', '_subclasshook_', '_weakref_', 'abstract', 'actions', 'addAttribute', 'addCurvedPart', 'addCurvedRing', 'addExpressionField', 'addFeature', 'addFeatures', 'addJoin', 'addPart', 'addPartV2', 'addRing', 'addTopologicalPoints', 'afterRollBack', 'aggregate', 'allFeatureIds', 'allowCommitChanged', 'appendError', 'attributeAdded', 'attributeAlias', 'attribut
```

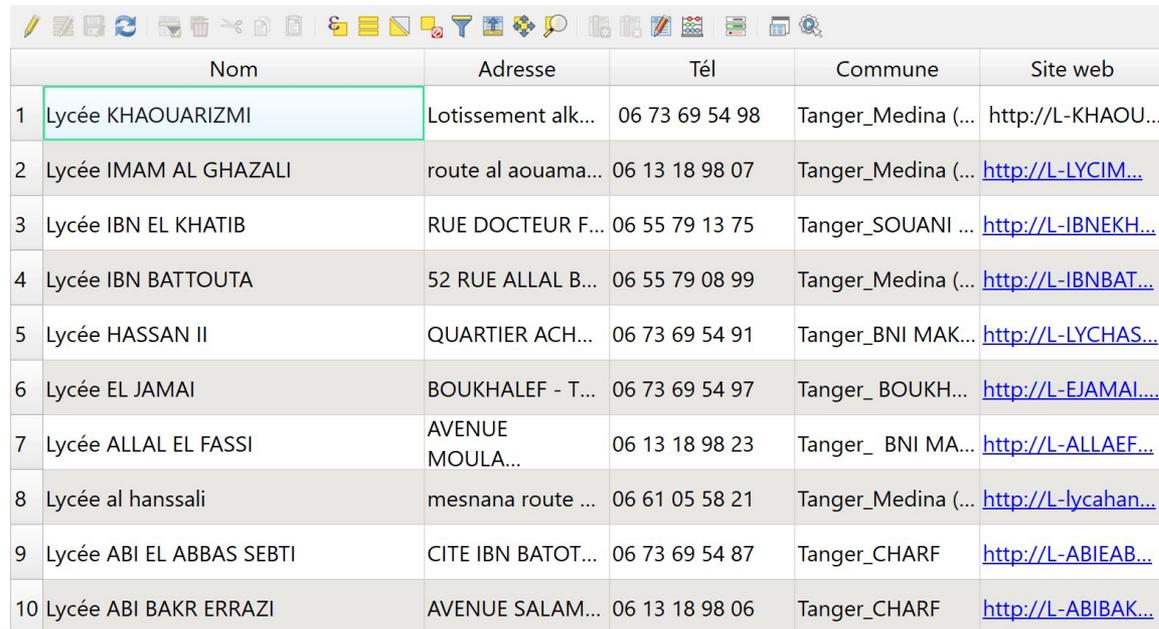
Affichage de la table d'attributs

- Pour afficher les attributs de la couche en question, on peut utiliser la fonction dans le code suivant, supposons qu'on a déjà l'objet *vlayer*:

```
vlayer=iface.activeLayer()
```

```
iface.showAttributeTable(vlayer)
```

Lycée — Total des entités: 16, Filtrées: 16, Sélectionnées: 0



| | Nom | Adresse | Tél | Commune | Site web |
|----|--------------------------|--------------------|----------------|---------------------|---|
| 1 | Lycée KHAOUARIZMI | Lotissement alk... | 06 73 69 54 98 | Tanger_Medina (...) | http://L-KHAOU... |
| 2 | Lycée IMAM AL GHAZALI | route al aouama... | 06 13 18 98 07 | Tanger_Medina (...) | http://L-LYCIM... |
| 3 | Lycée IBN EL KHATIB | RUE DOCTEUR F... | 06 55 79 13 75 | Tanger_SOUANI ... | http://L-IBNEKH... |
| 4 | Lycée IBN BATTOUTA | 52 RUE ALLAL B... | 06 55 79 08 99 | Tanger_Medina (...) | http://L-IBNBAT... |
| 5 | Lycée HASSAN II | QUARTIER ACH... | 06 73 69 54 91 | Tanger_BNI MAK... | http://L-LYCHAS... |
| 6 | Lycée EL JAMAI | BOUKHALEF - T... | 06 73 69 54 97 | Tanger_BOUKH... | http://L-EJAMAL... |
| 7 | Lycée ALLAL EL FASSI | AVENUE MOULA... | 06 13 18 98 23 | Tanger_ BNI MA... | http://L-ALLAEF... |
| 8 | Lycée al hanssali | mesnana route ... | 06 61 05 58 21 | Tanger_Medina (...) | http://L-lycahan... |
| 9 | Lycée ABI EL ABBAS SEBTI | CITE IBN BATOT... | 06 73 69 54 87 | Tanger_CHARF | http://L-ABIEAB... |
| 10 | Lycée ABI BAKR ERRAZI | AVENUE SALAM... | 06 13 18 98 06 | Tanger_CHARF | http://L-ABIBAK... |

Accéder aux entités

```
for f in layer.getFeatures(): #pour accéder aux entités
    print(f)
```

- **Autre méthode pour afficher les champs (attributs)**

```
for field in vlayer.fields() :
    print(field.name())
```

- Aussi on peut utiliser la fonction *fields()* pour accéder aux champs d'une table d'attributs et les afficher sur la console.
- Cette fonction retourne un objet QgsFields.

Showing some information about each feature

```
layer = iface.activeLayer()
features = layer.getFeatures()

for feature in features:
    # retrieve every feature with its geometry and attributes
    print("Feature ID: ", feature.id())
    # fetch geometry
    # show some information about the feature geometry
    geom = feature.geometry()
    geomSingleType = QgsWkbTypes.isSingleType(geom.wkbType())
    if geom.type() == QgsWkbTypes.PointGeometry:
        # the geometry type can be of single or multi type
        if geomSingleType:
            x = geom.asPoint()
            print("Point: ", x)
        else:
            x = geom.asMultiPoint()
            print("MultiPoint: ", x)
    elif geom.type() == QgsWkbTypes.LineGeometry:
        if geomSingleType:
            x = geom.asPolyline()
            print("Line: ", x, "length: ", geom.length())
```

```
    else:
        x = geom.asMultiPolyline()
        print("MultiLine: ", x, "length: ", geom.length())
    elif geom.type() == QgsWkbTypes.PolygonGeometry:
        if geomSingleType:
            x = geom.asPolygon()
            print("Polygon: ", x, "Area: ", geom.area())
        else:
            x = geom.asMultiPolygon()
            print("MultiPolygon: ", x, "Area: ", geom.area())
    else:

        print("Unknown or invalid geometry")
    # fetch attributes
    attrs = feature.attributes()
    # attrs is a list. It contains all the attribute values of this feature
    print(attrs)
```

Selecting features

- In QGIS desktop, features can be selected in different ways: the user can click on a feature, draw a rectangle on the map canvas or use an expression filter.
- Selected features are normally highlighted in a different color (default is yellow) to draw user's attention on the selection.

To select all the features, the `selectAll()` method can be used:

Get the active layer (must be a vector layer)

```
layer = iface.activeLayer()  
layer.selectAll()
```

Using an expression

- To select using an expression, use the `selectByExpression()` method:

```
# Assumes that the active layer is points.shp file from the QGIS test suite  
# (Class (string) and Heading (number) are attributes in points.shp)  
layer = iface.activeLayer()  
layer.selectByExpression('"Class"=\'B52\' and "Heading" > 10 and "Heading"  
<70', QgsVectorLayer.SetSelection)
```

Change the selection color

- To change the selection color you can use setSelectionColor() method of QgsMapCanvas as shown in the following example:

```
iface.mapCanvas().setSelectionColor( QColor("red") )
```

To add features to the selected features list for a given layer, you can call `select()` passing to it the list of features IDs:

```
selected_fid = []  
# Get the first feature id from the layer  
for feature in layer.getFeatures():  
    selected_fid.append(feature.id())  
    break  
# Add these features to the selected list  
    layer.select(selected_fid)
```

To clear the selection:
`layer.removeSelection()`