

Test Python-QGIS LPESSIG



Partie 1 - QCM

Le langage Python est un langage :

- compilé et avec un type statique fort.
- interprété et avec un typage dynamique fort.
- interprété et avec un typage statique faible.
- compilé et sans typage.

L'API PyQgis permet :

- de transformer et manipuler des données géospatiales.
- de interagir avec les éléments de l'interface de QGIS.
- de créer des extensions (*plugins*) à QGIS.
- l'ensemble de ces opérations.

```
# geom est de type 'QgsGeometry'  
geom_buff = geom.buffer(100.0, 8)
```

Quel est le type de l'objet geom_buff ?

- QgisInterface
- QgsBufferArray
- QgsGeometry
- QgisGeomBuffer

Laquelle de ces méthodes n'appartient pas à l'objet QgsGeometry ?

- interpolate
- addMapLayers
- intersects
- buffer

Laquelle de ces affirmations est vraie ?

- Une géométrie est composée de plusieurs couches de données.
- Un projet QGIS contient nécessairement des couches de données de type vecteur.
- Une couche de données type vecteur contient une ou plusieurs entités.
- Les données raster ne sont pas visualisables dans QGIS.

Partie 2 - Compréhension du code

```
b = 0  
for i in QgsMapLayerRegistry.instance().mapLayers():  
    b += 1
```

1. Expliquez en une phrase ce que fait le morceau de code précédent.

2. La couche ERP est une couche de points contenant la localisation des établissements recevants du public en Isère.

La couche COMMUNE est une couche de polygones décrivant les communes de l'Isère.

```
erp_layer = QgsMapLayerRegistry.instance().mapLayersByName('ERP')[0]
communes = QgsMapLayerRegistry.instance().mapLayersByName('COMMUNE')[0]

result = []
for comm in communes.getFeatures():
    nb_erp = 0
    comm_geom = comm.geometry()
    for ft in erp_layer.getFeatures():
        if comm.contains(ft.geometry()):
            nb_erp += 1
    if nb_erp > 0:
        result.append(comm['NOM_COM'])
```

Expliquez en quelques phrases ce que fait le morceau de code précédent.

3. Quelle structure de données aurait pu être utilisée pour accélérer cette opération ?

4. Quelle est la fonction native Python qui permet de connaître la taille de la liste result ?

Partie 3 - Écriture de code à trou

Toujours en utilisant les deux couches précédentes, nous cherchons à enrichir la couche des communes du nombre d'ERP contenus dans chaque commune. **Complétez le code suivant afin de mener à bien cette opération.**

```
erp_layer = QgsMapLayerRegistry.instance().mapLayersByName('ERP')[0]
communes = QgsMapLayerRegistry.instance().mapLayersByName('COMMUNE')[0]

NEW_FIELD = 'nombre_erp'

_____.startEditing() # Un nom de variable est attendu ici !
communes.dataProvider().addAttributes(
    [QgsField(NEW_FIELD, QVariant.Int)])
communes.updateFields()

for comm in communes.getFeatures():
    nb_erp = 0
    comm_geom = comm.geometry()
    for ft in erp_layer._____.(): # Un nom de méthode est attendu ici !
        if comm_geom.contains(ft.geometry()):
            nb_erp += 1
    comm[NEW_FIELD] = nb_erp
    communes.updateFeature(_____) # Un nom de variable est attendu ici !

communes._____.() # Un nom de méthode est attendu ici !
```

Partie 4 - Écriture de code

1. **Écrire le code permettant d'obtenir le nombre total d'entités de la couche ERP.**

2. Nous ne voulons pas éditer la couche COMMUNE (contrairement à la partie 3) et nous voulons répondre à la question suivante : **Quelle est la commune qui contient le plus d'ERP ?**

Écrire le procédure à suivre (avec des phrases) pour réaliser cette opération.

Bonus : **Écrire le code permettant de trouver le nom de la ville contenant le plus d'ERP (et leur nombre).**