

# Test (1h) Python-QGIS - LPESSIG - 09/04/2021

Merci d'écrire votre nom sur chacune des pages

## Partie 1 - QCM (une seule bonne réponse)

**L'API PyQgis permet :**

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

**Le langage Python est un langage :**

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

**Laquelle de ces affirmations est vraie ?**

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

**Quel est le type de l'objet `ft` ?**

```
# layer est de type 'QgsVectorLayer'  
ft = layer.getFeature(1)
```

- QgsGeometry
- QgsFeatureIterator
- QgsMapLayer
- QgsFeature

**Quel est le type de l'objet `geom_buff` ?**

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

- QgisInterface
- QgsBufferArray
- QgsGeometry
- QgsGeomBuffer

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

- addMapLayers
- buffer
- contains
- intersects

**Comment zoomer et centrer la carte sur la couche `layer` (un objet de type `QgsVectorLayer` `**`)?**`**

- `layer.useExtent(iface.mapCanvas())`
- `layer.extent()`
- `iface.mapCanvas().getExtent(layer)`
- `iface.mapCanvas().setExtent(layer.extent())`

**Admettons qu'il y ait une couche vectorielle nommée FONTAINES dans l'interface. Que retourne la ligne de code suivante :** `QgsProject.instance().mapLayersByName("FONTAINES")`

- Rien.
- Un objet de type `QgsVectorLayer` .
- Une liste contenant un objet de type `QgsVectorLayer` .
- Une liste contenant plusieurs objets de type `QgsVectorLayer` (autant qu'il y a de couches dans l'interface).

**Admettons qu'il existe un dossier nommé « Images » dans votre répertoire de travail actuel. Que fait le code suivant ?**

```
import os

names = \
    [name.replace('.png', '.svg') for name in os.listdir('Images') if '.png' in name]
```

- Rien, ce code n'est pas valide en Python.
- Renommer tous les fichiers au format "png" du dossier « Images ».
- Créer une liste de noms de fichiers, en remplaçant l'extension "png" par "svg".
- Convertir tous les fichiers du dossier « Images » du format "png" vers le format "svg".

**Après avoir importé le module « os », quelle fonction ou suite de fonctions vous permet d'obtenir une chaîne de caractère contenant le chemin de votre répertoire de travail actuel ?**

- `os.getdir()`
- `str(os.listdir('.'))`
- `os.getcurrentdir()`
- `os.getcwd()`

## Partie 2 - Compréhension du code (1)

La couche ERP est une couche de points contenant la localisation des établissements recevant du public en Isère. La couche COMMUNE est une couche de polygones décrivant les communes de l'Isère.

```
erp_layer = QgsProject.instance().mapLayersByName('ERP')[0]
communes = QgsProject.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 < 1:
        result.append(comm['NOM_COM'])

print(result)

communes.selectByIds(
    [ft.id() for ft in communes.getFeatures() if ft['NOM_COM'] not in result])
```

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

```
#
#
#
#
#
#
```

2. Admettons que les couches ERP et COMMUNE couvrent maintenant l'ensemble de la France métropolitaine. Un index spatial aurait pu être utilisé pour accélérer l'opération. Aurait-il été préférable d'indexer la couche « COMMUNE » ou la couche « ERP » ?

```
#  
#  
#
```

## Partie 3 - Compréhension du code (2)

Toujours en utilisant les deux couches précédentes, nous souhaitons enrichir la couche des communes avec le nombre d'ERP de 1ère catégorie (effectif admissible de + de 1501 personnes) et de 2ème catégorie (effectif admissible de 701 à 1500 personnes) puis sauvegarder cette couche.

1. Complétez le code suivant afin de mener à bien cette opération.

```
erp_layer = QgsProject.instance().mapLayersByName('ERP')[0]  
communes = QgsProject.instance().mapLayersByName('COMMUNE')[0]  
  
communes.startEditing()  
pr = _____.dataProvider() # Un nom de variable est attendu ici  
pr.addAttributes(  
    [QgsField('nb_erp_1_2', QVariant.Int)])  
communes.updateFields()  
  
for comm in communes.getFeatures():  
    nb_erp = 0  
    comm_geom = comm.geometry()  
    for ft in erp_layer.getFeatures():  
        if ft['Catégorie'] == '1' ____ ft['Catégorie'] == '2': # Un mot clé python est attendu ici  
            if comm_geom.____(ft.geometry()): # Un nom de méthode est attendu ici  
                nb_erp += 1  
    comm['nb_erp_1_2'] = _____ # Un nom de variable est attendu ici  
    communes._____(comm) # Un nom de méthode est attendu ici  
  
communes.commitChanges()  
  
QgsVectorFileWriter.writeAsVectorFormat(  
    _____, # Un nom de variable est attendu ici  
    'result.shp',  
    'utf-8',  
    communes.crs(),  
    _____, # Une chaîne de caractère est attendue ici  
)
```

2. Nous ne voulons pas éditer la couche COMMUNE (contrairement à la question précédente) et nous voulons répondre à la question suivante : **Quelle est la (ou les) commune(s) qui contien(nen)t le plus d'ERP de 1ère et 2ème catégories ?** (plusieurs communes peuvent contenir le même nombre d'ERP)

Écrire le procédure à suivre (sous forme de pseudo-code) pour réaliser cette opération.

```
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#
```

Bonus : Écrire le code permettant d'afficher le nom de la ville contenant le plus d'ERP de 1ère et 2ème catégories.  
Environ 6 lignes de codes sont nécessaires.

```
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#
```