

---

# Aula 3

## 2023 - 10 - 07

---

ANA LUISA MAFFINI  
2023

# Python e QGIS

---

## Uma breve visão geral das possibilidades de uso de scripts Python em SIG:

Ecossistema QGIS/Python:

Escrito em C++

Em 2007 começou o trabalho para adicionar python como linguagem de programação

QGIS tem 400 classes principais C++, das quais 75% são python habilitado via SIP

QGIS usa Qt (C++) para janelas e botões, etc. PyQt liga isso com python

## **Leitura Recomendada**

O Guia do Programador PyQGIS, Gary Sherman (QGIS fundador)

“Livro de receitas do desenvolvedor” PyQGIS - PyQGIS Developer Cookbook

[https://docs.qgis.org/testing/en/docs/pyqgis\\_developer\\_cookbook/](https://docs.qgis.org/testing/en/docs/pyqgis_developer_cookbook/)

# Principais Módulos QGIS Python

---

**pip** - programa de instalação para pacotes python

```
pip list
```

```
pip install seu_módulo
```

```
pip uninstall seu_módulo
```

---

# **Abrindo Camada Vetorial no QGIS**

## Começando a Usar o Python - abrir vetor no qgis

---

Abra o terminal python no QGIS e escreva o seguinte código:

**Obs.:** Substitua o endereço destacado em laranja com o endereço do seu computador!

```
# Definir camada que será importada
vlayer = QgsVectorLayer('C:/Users/analú/OneDrive/Desktop/Dados/
Pelotas_Regioes_Administrativas.shp', 'regioes_adm', 'ogr')

# Conferir se o endereço está certo
vlayer.isValid()

# adicionar camada no QGIS
QgsProject.instance().addMapLayer(vlayer)
```

## Começando a Usar o Python - abrir vetor no qgis

---

Outro método, utilizando o iface!

```
# adicionar camada no QGIS  
iface.addVectorLayer('C:/Users/analú/OneDrive/Desktop/Dados/  
Pelotas_Regioes_Administrativas.shp', 'regioes_adm', 'ogr')
```

---

# **Abrindo Várias Camadas Vetoriais no QGIS**



## Começando a Usar o Python - abrir vários vetores no qgis

---

**# identificar todos os arquivos de uma pasta**

```
for root, directory, files in os.walk ('C:/Users/analú/OneDrive/  
Desktop/Dados'):  
    print(files)
```

**# identificar cada arquivo de uma pasta**

```
for root, directory, files in os.walk ('C:/Users/analú/OneDrive/  
Desktop/Dados'):  
    for file in files:  
        print(file)
```

## Começando a Usar o Python - abrir vários vetores no qgis

---

```
# identificar apenas os arquivos shapefile
```

```
for root, directory, files in os.walk ('C:/Users/analú/OneDrive/  
Desktop/Dados'):
```

```
    for file in files:
```

```
        if file.endswith('.shp'):
```

```
            print(file)
```

## Começando a Usar o Python - abrir vários vetores no qgis

---

```
# adicionar todos os arquivos shapefile da pasta
for root, directory, files in os.walk ('C:/Users/analú/OneDrive/
Desktop/Dados'):
    for file in files:
        if file.endswith('.shp'):
            path_layer = os.path.join('C:/Users/analú/OneDrive/Desktop
/Dados', file)
            layer = file[:-4]
            QgsProject.instance().addMapLayer(QgsVectorLayer(path_
layer, layer, 'ogr'))
```

---

# **Editando os Campos de Atributos da Camada**

# Python no QGIS

# Definir camada que será importada

```
vlayer = QgsVectorLayer('C:/Users/analú/OneDrive/Desktop/Dados/  
Pelotas_Regioes_Administrativas.shp', 'regioes_adm', 'ogr')
```

# Conferir se o endereço está certo

```
vlayer.isValid()
```

# adicionar camada no QGIS

```
QgsProject.instance().addMapLayer(vlayer)
```

# Começar edição da camada

```
vlayer.startEditing()
```

# adicionar novo campo de atributos

```
vlayer.addAttribute(QgsField('NUM', QVariant.Int))
```

# encerrar edição da camada

```
vlayer.commitChanges()
```

**Tipos de campos:**

String

Int

Double

# Listar o nome dos atributos da camada

```
print(vlayer.fields().names())
```

# Começar edição da camada

```
vlayer.startEditing()
```

# deletar um campo de atributos

```
vlayer.deleteAttribute(4)
```

# encerrar edição da camada

```
vlayer.commitChanges()
```

**# Identificar atributos da camada**

```
for feature in vlayer.getFeatures():  
    print(feature.attributes())
```

**# Identificar atributos de um campo específico da camada**

```
for feature in vlayer.getFeatures():  
    print(feature.attributes()[1])
```

---

## **Editando os Atributos da Camada**



# Python no QGIS

```
# Começar edição da camada
```

```
vlayer.startEditing()
```

```
# adicionar novo campo de atributos
```

```
vlayer.addAttribute(QgsField('area', QVariant.Double))
```

```
# encerrar edição da camada
```

```
vlayer.commitChanges()
```

```
# Atualizar campo da camada
```

```
vlayer.startEditing()
```

```
for feature in vlayer.getFeatures():
```

```
    id = feature.id()
```

```
    area = feature.geometry().area()
```

```
    attr_value = {4:area}
```

```
    vlayer.changeAttributeValues(id, attr_value)
```

```
vlayer.commitChanges()
```

---

# **Trabalhando no Jupyter Notebook**

# Trabalhando no Jupyter Notebook

---

Baixe a pasta “Dados” disponível no Moodle do curso, abra o Jupyter Notebook e inicie um novo projeto dentro da pasta Dados.

```
# importar bibliotecas
```

```
import geopandas as gpd
```

```
import matplotlib
```

```
# definir arquivos que serão carregados
```

```
path_list = ['Pelotas_Regioes_Administrativas.shp',
```

```
'Pelotas_Educacao.shp', 'Pelotas_Cicloviario.shp']
```

```
dados = [gpd.read_file(i) for i in path_list]
```

```
# testar se os arquivos estão corretos
```

```
dados[1]
```

## Trabalhando no Jupyter Notebook

---

```
# filtrar apenas as escolas municipais
```

```
escolas_filter = dados[1][dados[1]['ESFERA'] == 'MUNICIPAL']
```

```
# conferir se os dados estão certos
```

```
escolas_filter
```

```
# filtrar apenas a região Centro
```

```
regiao_filter = dados[0][dados[0]['NM_SUBDIST'] == 'CENTRO']
```

```
# conferir se os dados estão certos
```

```
regiao_filter
```

```
# visualizar o arquivo da região
```

```
regiao_filter.plot()
```

## Trabalhando no Jupyter Notebook

```
# selecionar apenas as ciclovias da região centro
```

```
ciclovias_centro = gpd.overlay(dados[2],regiao_filter, how =  
'intersection')
```

```
# selecionar as escolas municipais da região centro
```

```
escolas_centro = gpd.sjoin(escolas_filter, regiao_filter, predicate =  
'within')
```

```
# adicionar campo com o comprimento em Km das ciclovias do centro
```

```
ciclovias_centro['extensao_km'] = ciclovias_centro['geometry'].to_crs  
(epsg=5880).length/1000
```

```
# visualizar quantidade de escolas municipais do centro por tipo
```

```
print(escolas_filter.value_counts('TIPO'))
```

---

## **Usando o QGIS em outras IDEs**

# Usando o QGIS em outras IDEs

---

## Abrindo no Jupyter Notebook ou no VSCode

```
import sys
import os
import qgis
from qgis.core import QgsApplication, QgsVectorLayer
```

### # Inicializando o QGIS

```
qgishome = 'C:/OSGeo4W/apps/qgis/'
QgsApplication.setPrefixPath(qgishome, True)
app = QgsApplication([], False)
app.initQgis()
```

## Usando o QGIS em outras IDEs

---

### Abrindo no Jupyter Notebook ou no VSCode

**# Defina a localização do arquivo**

```
fileName = 'C:/Users/analú/OneDrive/Desktop/Dados/Pelotas_Educacao.shp'
```

**# Define o layer**

```
vlayer = QgsVectorLayer(fileName, 'educacao', 'ogr')
```

**# Identifique se deu certo**

```
print(vlayer.isValid())
```