



CURSO QGIS

# Introdução ao PyQGIS

Desde a versão 0.9, o QGIS tem suporte opcional à *scripts* usando a linguagem Python. Existem várias maneiras de usar vinculações Python na área de trabalho do QGIS. Um bom recurso são os *plug-ins* PyQGIS, faça *downloads* de alguns *plug-ins* do repositório de *plug-ins* e examine seus códigos para com o tempo você desenvolver seus próprios *plug-ins*: <https://plugins.qgis.org>

## Tópicos da Ementa:

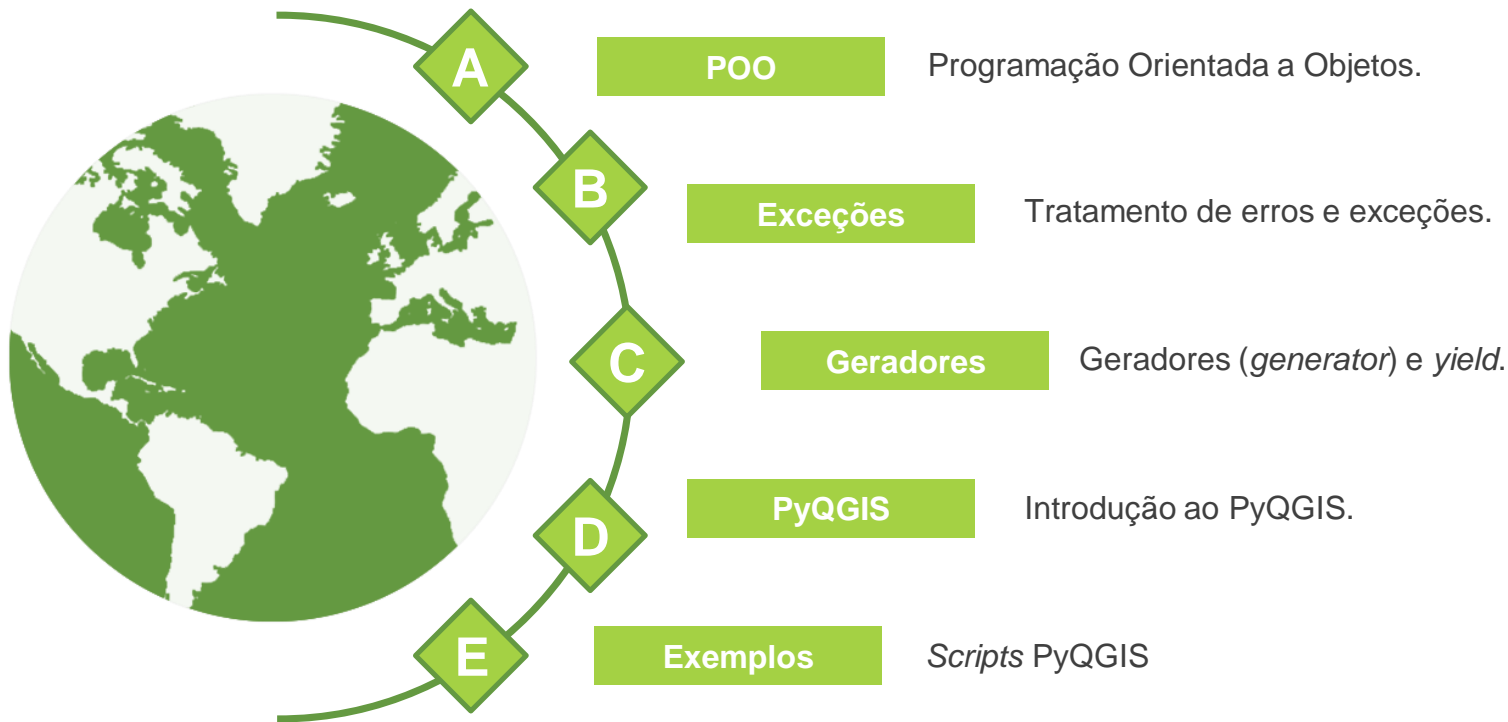
- **Orientação a Objeto (Revisão)**
- **Apresentação PyQGIS**
  - Variáveis
  - Listas e Arrays
  - Condicional e Repetição
  - Funções (*scripts*)

## Aula 21 e 22/07:

- Programação Orientada a Objeto (POO)
- Apresentação PyQGIS
- Variáveis, Listas e Arrays, estruturas condicionais e repetitivas
- Funções (*scripts*)



# Aula 01 – POO e PyQGIS





# Revisão de Python

Programação Orientada a Objetos (POO)

# Motivação para POO

- O desenvolvimento de aplicações de software estão cada vez mais complexas;
- Cresceram as demandas por metodologias que pudessem abstrair e modularizar as estruturas básicas de programação;
- Principais vantagens:
  - Aumento de produtividade;
  - Reuso de código;
  - Redução das linhas de código programadas;
  - Separação de responsabilidades;
  - Componentização;
  - Maior flexibilidade do sistema; e
  - Facilidade na manutenção.



# Pilares OO

A expressão orientada a objetos significa que o aplicativo é organizado como uma coleção de objetos que incorporam tanto a estrutura como o comportamento dos dados.



01

## Abstração

A estrutura fundamental para definir novos objetos é a classe. Uma classe é definida em código-fonte.



02

## Encapsulamento

Encapsulamento é a proteção dos atributos ou métodos de uma classe.



03

## Herança

É o que permite construir objetos que são especializações de outro objeto.



04

## Polimorfismo

Permite que 2 objetos diferentes possam ser usados de forma semelhante. Por exemplo, tanto listas quanto strings tem índices e usam a função len().

# Criando uma Classe

- As classes são justamente modelos que possuem atributos e métodos que buscam melhor representar objetos e situações do mundo real;
- O método `__init__()`, é um método especial de Python, ele é o construtor da classe. Sempre que formos criar um objeto através dessa classe nós iremos invocar o método construtor;
- A palavra **self** que aparece na frente de cada atributo e no primeiro parâmetro do método `__init__()` vai se referir ao objeto criado;
- No código principal importamos as classes criadas.



# Relembrando POO

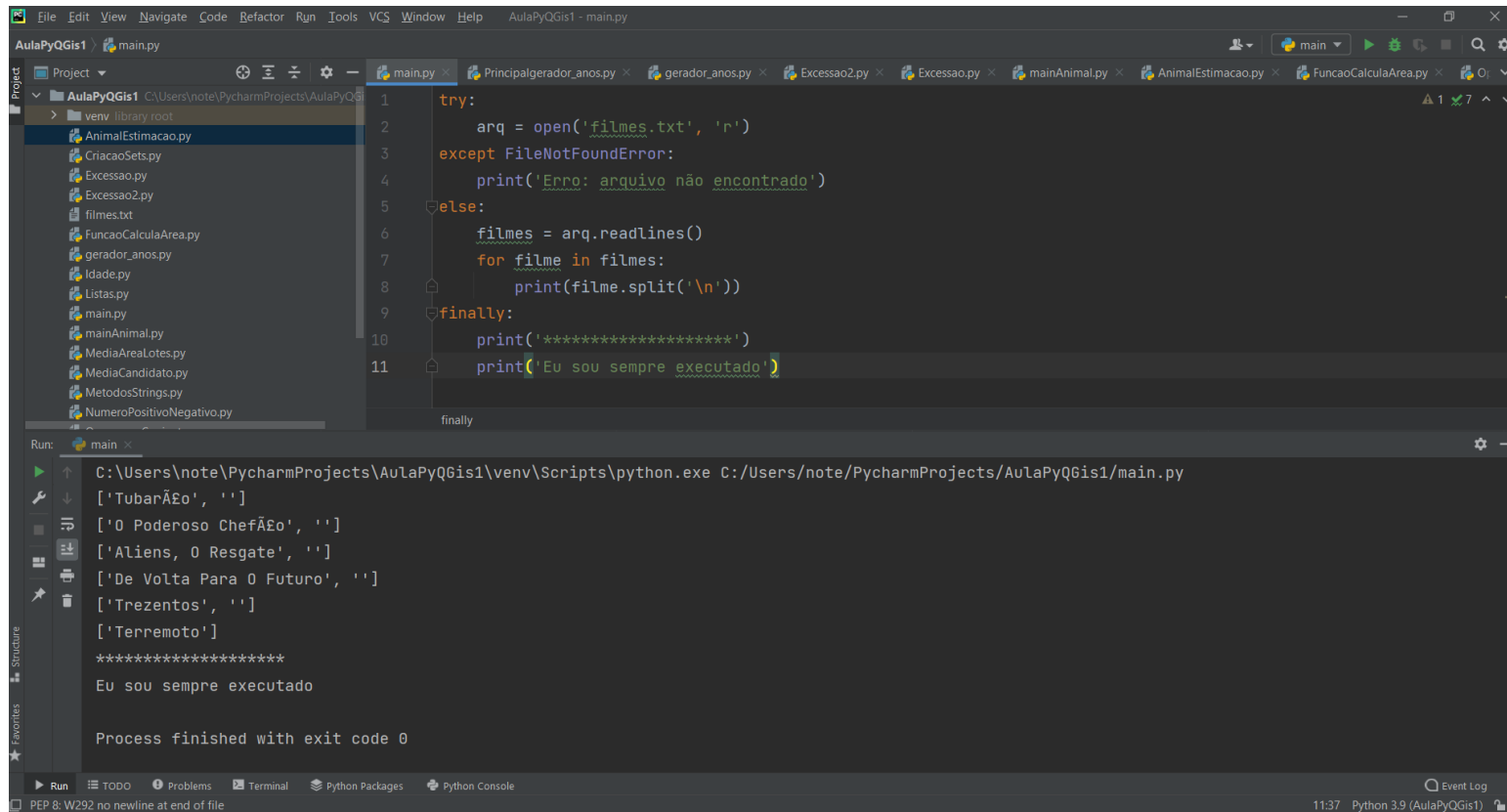
Vamos Codar?

```
1  # Programa que simula animais de estimação
2  class AnimalEstimacao():
3      def __init__(self, nome, especie, dono):
4          self.nome = nome
5          self.especie = especie
6          self.dono = dono
7
8      def correr(self):
9          print('{0} está correndo'.format(self.nome))
10
11     def brincar(self):
12         print('{0} está brincando'.format(self.nome))
13
14     def comer(self):
15         print('{0} está comendo'.format(self.nome))
16
17 class PeixeEstimacao(AnimalEstimacao):
18     def __init__(self, nome, dono):
19         super().__init__(nome, 'Peixe', dono)
20
21     def nadar(self):
22         print('{0} está nadando'.format(self.nome))
23
```



# Captura de Erros

## Tratamento de Erros e Exceções



The screenshot displays the PyCharm IDE interface. The left sidebar shows the project structure for 'AulaPyQGis1', with 'main.py' selected. The main editor window shows the code in 'main.py' with line numbers 1 through 11. The code implements a try-except-finally block to handle a file not found error. The 'Run' console at the bottom shows the output of the program, including a list of movies and the message 'Eu sou sempre executado'.

```
1 try:
2     arq = open('filmes.txt', 'r')
3 except FileNotFoundError:
4     print('Erro: arquivo não encontrado')
5 else:
6     filmes = arq.readlines()
7     for filme in filmes:
8         print(filme.split('\n'))
9 finally:
10    print('*****')
11    print('Eu sou sempre executado')
```

Run: C:\Users\note\PycharmProjects\AulaPyQGis1\venv\Scripts\python.exe C:/Users/note/PycharmProjects/AulaPyQGis1/main.py

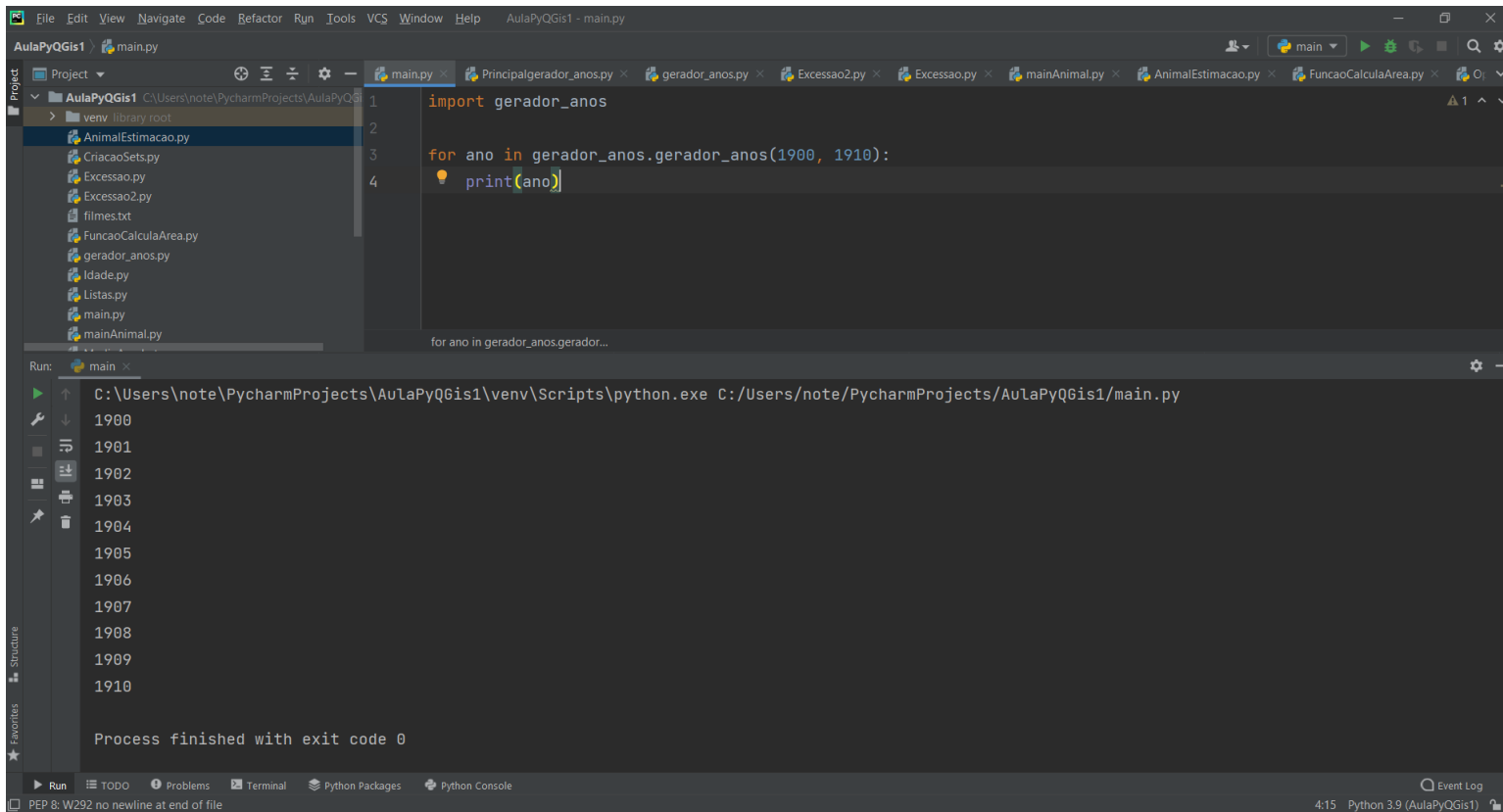
```
['Tubarão', '']
['O Poderoso Chefão', '']
['Aliens, O Resgate', '']
['De Volta Para O Futuro', '']
['Trezentos', '']
['Terremoto']
*****
Eu sou sempre executado

Process finished with exit code 0
```

PEP 8: W292 no newline at end of file

# Geradores

*Generator e yield*



The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for running and debugging. The main editor window displays the following Python code in `main.py`:

```
1 import gerador_anos
2
3 for ano in gerador_anos.gerador_anos(1900, 1910):
4     print(ano)
```

The left sidebar shows the project structure for `AulaPyQGis1`, with the `venv` directory expanded. The `Run` console at the bottom shows the execution output:

```
C:\Users\note\PycharmProjects\AuLaPyQGis1\venv\Scripts\python.exe C:/Users/note/PycharmProjects/AuLaPyQGis1/main.py
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910

Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is `PEP 8: W292 no newline at end of file` and the Python version is `Python 3.9 (AulaPyQGis1)`.

Java:

```
public class Hello
{
    public static void main(String args[]) {
        java.util.Scanner s = new java.util.Scanner(System.in);
        System.out.print("Digite seu nome: ");
        String nome = s.nextLine();
        System.out.println("Olá, " + nome);
    }
}
```

**O Poder do Python**  
Faz a mesma coisa que outras  
linguagens com muito menos  
código!

C:

```
#include <stdio.h>
int main()
{
    char nome[200];
    printf("Digite seu nome: ");
    scanf("%s", nome);
    printf("Olá, %s\n", nome);
    return 0;
}
```

PHP:

```
$nome = readline("Digite seu nome: ");
echo 'olá: ' . $nome;
```

Python: 🤖

```
nome = input('Digite seu nome: ')
print('Olá', nome)
```



# Introdução ao PyQGIS

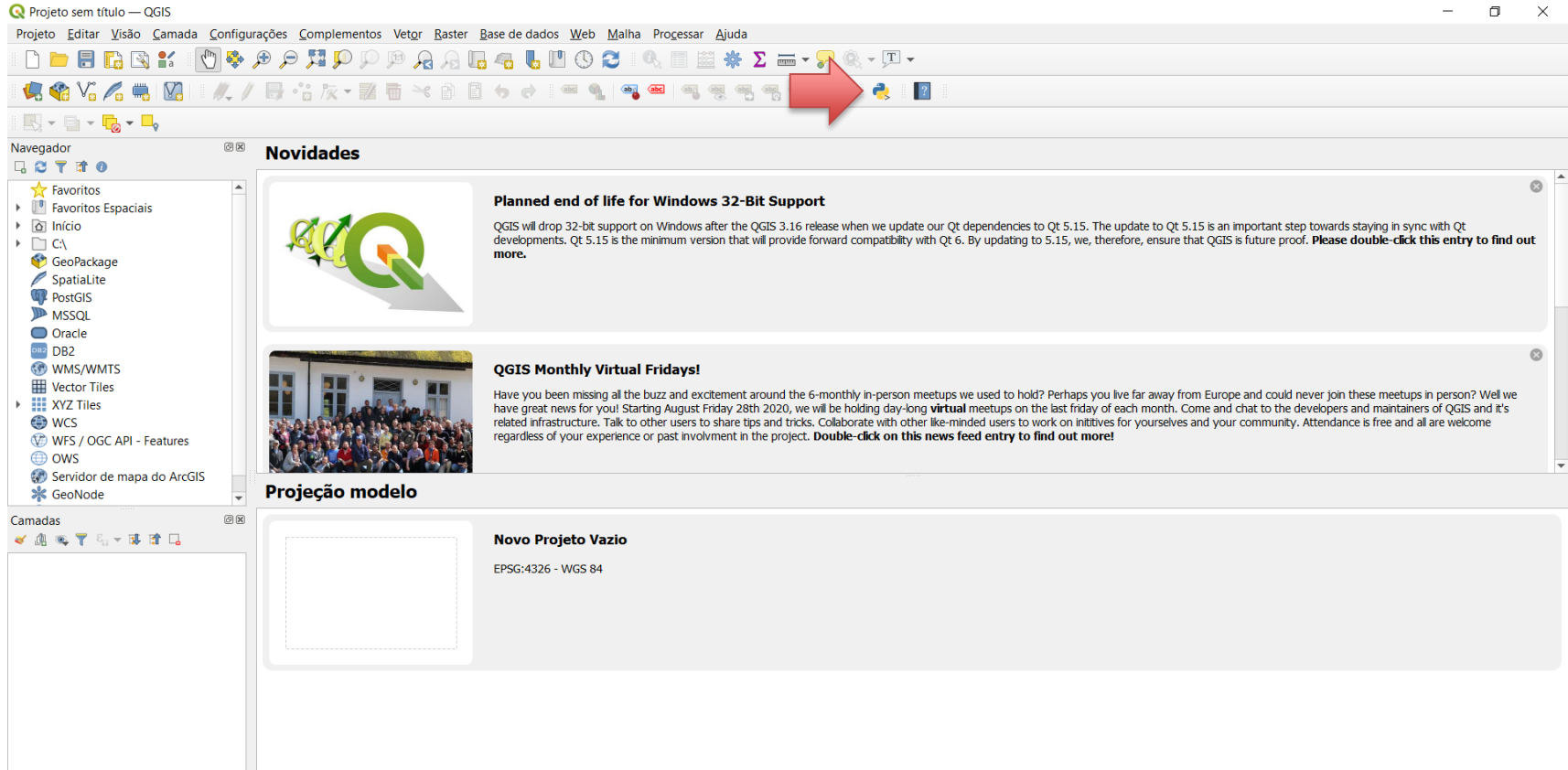
Primeiros Códigos Python no QGIS

# O que é PyQGIS?

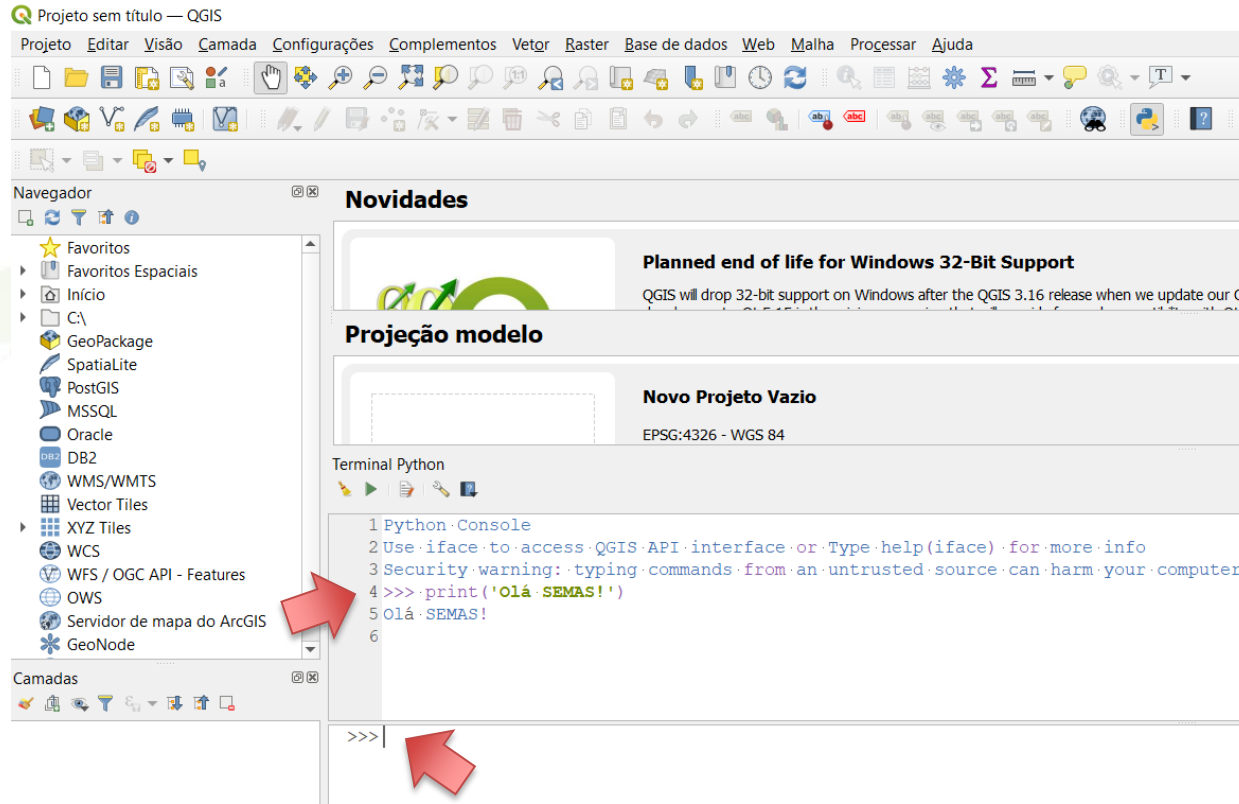
- O PyQGIS é um pacote Python, que permite que você acesse e manipule o QGIS de diferentes formas, dentre as quais, é possível:
  - invocar comandos no console;
  - executar automaticamente um código Python, quando o QGIS for inicializado;
  - criar ações customizadas;
  - criar novos algoritmos de processamento;
  - criar plugins, e
  - criar novas aplicações independentes (*stand-alone*).

# QGIS versão 3.16

Para acessar o PyQGIS teremos que clicar o botão indicado na tela principal do QGIS




# Executando um Print:





# Variáveis

Terminal Python



```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> print('Olá SEMAS!')
5 Olá SEMAS!
6 >>> nome = 'Marcello'
7 >>> idade = 42
8 >>> peso = 96.50
9 >>> print(nome)
10 Marcello
11 >>> print(idade)
12 42
13 >>> print(peso)
14 96.5
15
```



>>> |

## Nome de Variáveis

As palavras abaixo são reservadas da linguagem Python e não devem ser usadas como nome de variáveis



# Palavras Reservadas

and	except	lambda	with
as	finally	nonlocal	while
assert	False	None	yield
break	for	not	
class	from	or	
continue	global	pass	
def	if	raise	
del	import	return	
elif	in	True	
else	is	try	

Easy to change colors, photos and Text.

# Condicionais

Terminal Python



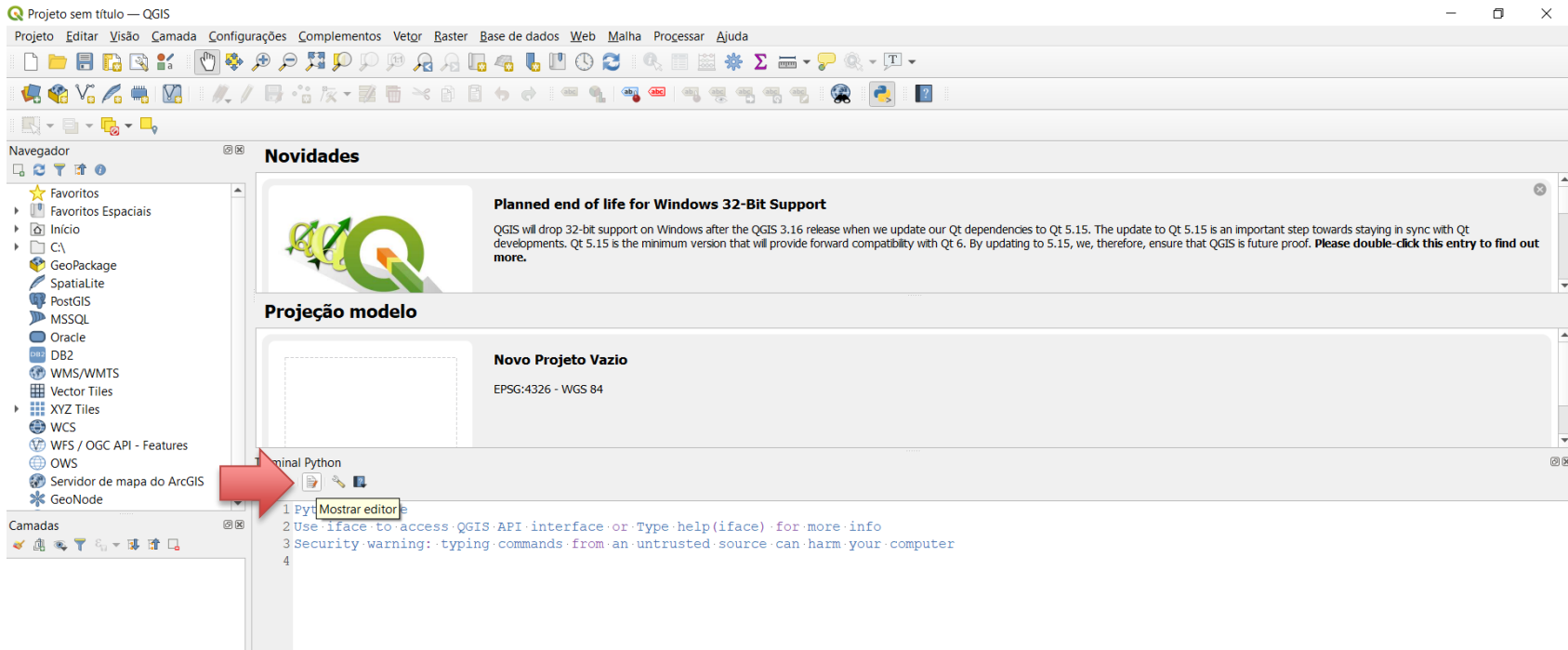
```
6 >>> nome = 'Marcello'
7 >>> idade = 42
8 >>> peso = 96.50
9 >>> print(nome)
10 Marcello
11 >>> print(idade)
12 42
13 >>> print(peso)
14 96.5
15 >>> idade = 5
16 >>> if idade < 18:
17     ... print("menor de idade")
18 ... else:
19     ... print("maior de idade")
20 menor de idade
21
```

E a  
indentação?

>>>

# PyQGIS

Para começarmos a criar os primeiros códigos vamos precisar abrir o editor:



# Condicional

Terminal Python

```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> exec(open('C:/Users/note/AppData/Local/Temp/tmp2xzs7dk2.py').encode('utf-8')).read()
5 9
6
```

>>>

\*Sem título 0

```
1 a=6
2 b=9
3 if a > b:
4     m = a
5 else:
6     m = b
7 print(m)
```

# String

Terminal Python

```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> nome + ' ' + sobrenome
5 'Rodrigo Santos'
6
```

3

```
>>> nome + ' ' + sobrenome
```

2

\*Sem título 0

```
1 print(len('IFPB'))
2
3 nome = 'Rodrigo'
4 sobrenome = 'Santos'
5
```

1

# Função Format

Terminal Python

```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> exec(open('C:/Users/note/AppData/Local/Temp/tmpb0_c52hd.py').encode('utf-8')).read()
5 O meu nome é José e a minha idade é 25
6
```

2

>>>

Sem título 0

```
1 nome = 'José'
2 idade = 25
3
4 print('O meu nome é {} e a minha idade é {}'.format(nome, idade))
5
```

1



# Listas

Terminal Python

```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) fo
  r more info
3 Security warning: typing commands from an untrusted source ca
  n harm your computer
4 >>> exec(open('C:/Users/note/AppData/Local/Temp/tmpcnqn4swk.p
  y'.encode('utf-8')).read())
5
```

1

\*Sem título 0 X

```
1 numbers = [1, 2, 3, 4, 5]
2
3 vazia = []
4
5 got_houses = ['Lannister', 'Stark', 'Targaryen']
6
7 compras = ["tomate", "alface", "macarrão", "carne"]
8
9 lista_mista = ["olá", 2.0, 5*2, [10, 20], {"hello": "World"}]
10
```

2

3

Terminal Python

```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) fo
  r more info
3 Security warning: typing commands from an untrusted source ca
  n harm your computer
4 >>> exec(open('C:/Users/note/AppData/Local/Temp/tmpcnqn4swk.p
  y'.encode('utf-8')).read())
5 >>> got_houses[1]
6 'Stark'
7
>>>
```

\*Sem título 0 X

```
1 numbers = [1, 2, 3, 4, 5]
2
3 vazia = []
4
5 got_houses = ['Lannister', 'Stark', 'Targaryen']
6
7 compras = ["tomate", "alface", "macarrão", "carne"]
8
9 lista_mista = ["olá", 2.0, 5*2, [10, 20], {"hello": "World"}]
10
```

# Funções

Não usamos `input()` em um script pyqgis, a menos que esteja executando a partir de uma janela CMD - a janela PyQGIS não é stdin/stdout.

```
Terminal Python
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> exec(open('C:/Users/note/AppData/Local/Temp/tmp2zixvp2.py').encode('utf-8')).read()
5 Olá SESMA!
6
```

```
*Sem título 0 X
1 def ola_mundo():
2     return 'Olá SESMA!'
3
4 print(ola_mundo())
```

```
Terminal Python
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> exec(open('C:/Users/note/AppData/Local/Temp/tmpdl3t0ly0.py').encode('utf-8')).read()
5 Traceback (most recent call last):
6   File "C:\PROGRA~1\QGIS3~1.16\apps\Python37\lib\code.py", line 90, in runcode
7     exec(code, self.locals)
8   File "<input>", line 1, in <module>
9   File "<string>", line 8, in <module>
10    File "<string>", line 2, in maior_idade
11 RuntimeError: input(): lost sys.stdin
12
>>>
```

```
*Sem título 0 X
1 def maior_idade():
2     idade = int(input('digite uma idade: '))
3     msg = 'menor de idade'
4     if idade >= 18:
5         msg = 'maior de idade'
6     return msg
7
8 print(maior_idade())
```

# Como resolver esta questão?

Terminal Python

```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> getTextInput('Oi SEMAS!', 'Curso de PyQGIS')
5
```

3

Oi SEMAS!

Curso de PyQGIS

Oi, obrigada!

OK Cancel

2

```
>>> getTextInput('Oi SEMAS!', 'Curso de PyQGIS')
```

1

```
1 from PyQt5.QtWidgets import QDialog
2 = def getTextInput(title, message):
3   answer = QDialog.getText(None, title, message)
4   = if answer[1]:
5       print(answer[0])
6       return answer[0]
7   = else:
8       return None
```

Que é isso!?!  
De onde veio tanto código??



# POO PyQGIS

Como desenvolver em PyQGIS?

# POO PyQGIS

## Conceitos Importantes



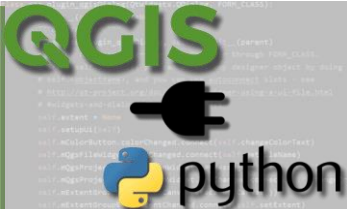
### Classes

QGIS é escrito na linguagem C ++. A funcionalidade de cada widget QGIS é implementada como uma classe - com certas propriedades e funções. Quando usamos as classes PyQGIS, ele está executando o código nas classes C ++ por meio das ligações Python.



### Classes e Objetos

Uma classe pode ser pensada como um modelo. Para usá-la em um programa, você deve criar uma "instância" dela - que usa o modelo junto com os parâmetros fornecidos para criar uma instância da classe, isso é conhecido como um objeto.



### PyQt

O QMessageBox é uma classe PyQt para criar uma caixa de diálogo com botões. Para usar a classe, você cria um objeto instanciando a classe. Aqui **Obj** é um objeto, que é uma instância da classe QMessageBox, criada usando os parâmetros padrão.

- Easy to change colors, photos and Text.

# Conceitos de Programação GUI

Qt é um kit de ferramentas de widget gratuito e de código aberto para a criação de interfaces gráficas de usuário, bem como aplicativos de plataforma cruzada.

A sigla API deriva da expressão inglesa *Application Programming Interface* que pode ser compreendida como uma interface de programação de aplicação.

API é um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de padrões e protocolos.



## Qt X QGIS

QGIS é construído usando a plataforma Qt. Tanto o Qt quanto o próprio QGIS têm **APIs** bem documentadas que devem ser usadas ao escrever código Python para ser executado no QGIS.

# PyQt

PyQt é a API Python para Qt

PyQt fornece classes e funções para interagir com widgets Qt. Um widget, numa interface gráfica, é um elemento de interação - tal como janelas, botões, menus, ícones, barras de rolagem etc. O termo pode também se referir aos pequenos aplicativos que flutuam pela área de trabalho e fornecem funcionalidades específicas ao utilizador



## PyQGIS

QGIS fornece uma API Python, comumente conhecida como PyQGIS.



## PyQt

PyQGIS é criado usando SIP e se integra com PyQt.

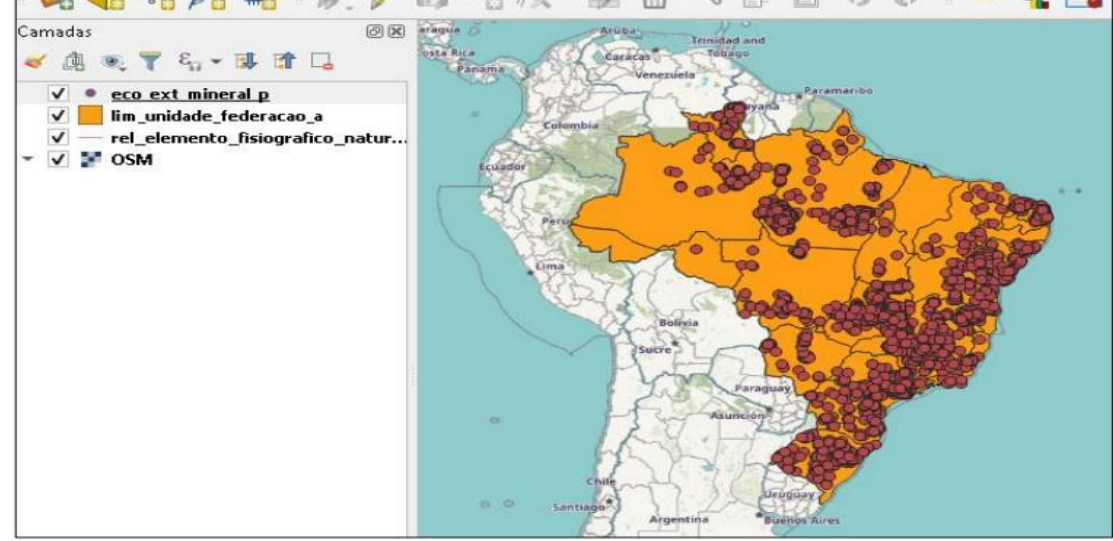
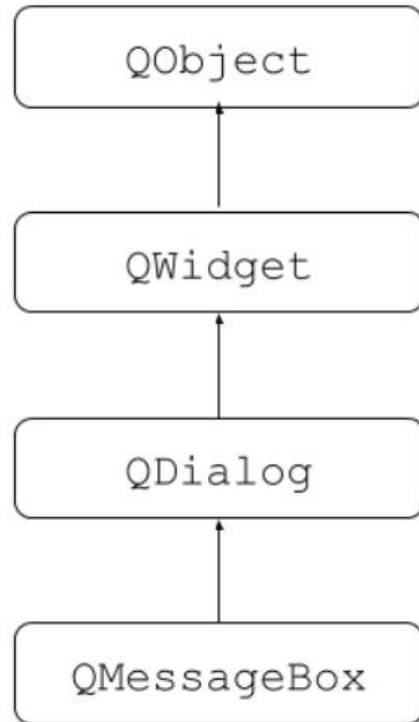


## SIP

A razão para usar SIP ao invés da mais usada SWIG é que todo o código QGIS depende das bibliotecas Qt.







Herança  
PyQGIS

# Herança PyQGIS

Conhecendo a Biblioteca PyQGIS

## QMessageBox



QMessageBox é um QDialog especializado.

## QDialog



A classe QDialog é a classe base das janelas de diálogo.

## QWidget



Todos os widgets Qt e classes QGIS herdam do QDialog. O widget mais básico é o QWidget.

### Importante:

**QWidget contém a maioria das propriedades que são usadas para descrever uma janela ou widget, como posição e tamanho, cursor do mouse, dicas de ferramentas, etc.**

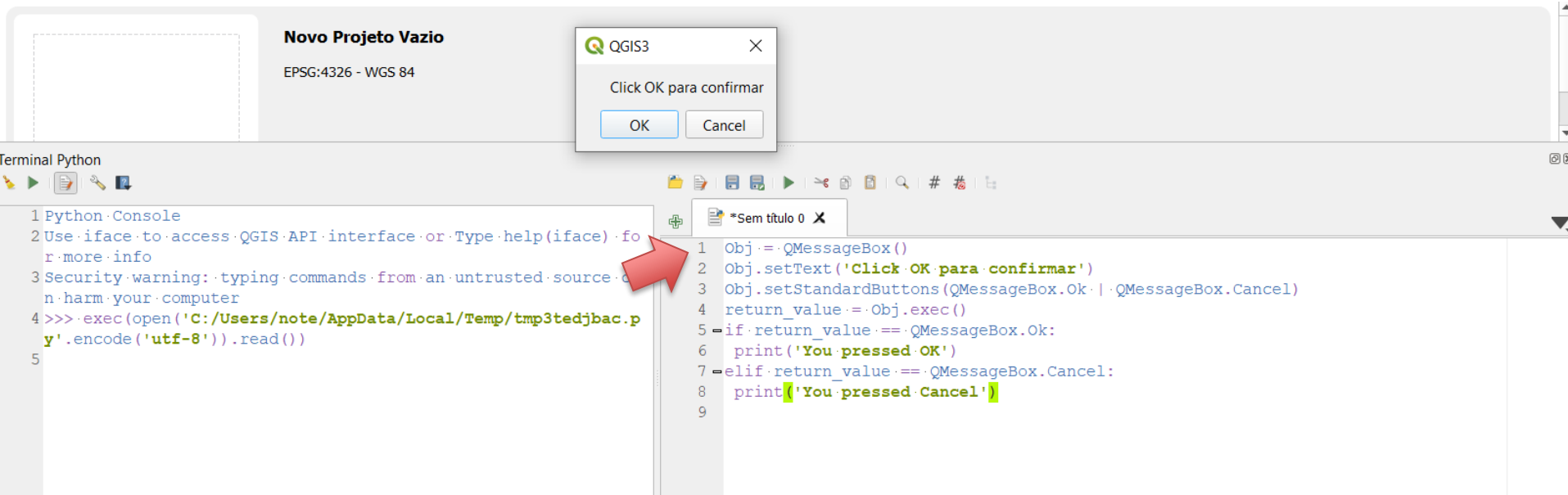
## QObject



QObject é a classe mais básica em Qt.

# PyQGIS

Vamos codar?



# Calculando a distância usando PyQGIS

Quando você executa o código no console Python do QGIS, todas as classes PyQGIS já foram importadas. Se você estiver executando a partir de um script ou plugin, deverá importar explicitamente a classe utilizada.

Terminal Python

```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> exec(open('C:/Users/note/AppData/Local/Temp/tmpfnmdjidp.py').encode('utf-8')).read()
5 4145.446977573498
6
```

>>>



\*Sem título 0 X

```
1 from qgis.core import QgsDistanceArea
2
3 santarem = (37.7749, -122.4194)
4 salinas = (40.661, -73.944)
5
6 d = QgsDistanceArea()
7 d.setEllipsoid('WGS84')
8
9
10 lat1, lon1 = santarem
11 lat2, lon2 = salinas
12
13 point1 = QgsPointXY(lon1, lat1)
14 point2 = QgsPointXY(lon2, lat2)
15
16 distance = d.measureLine([point1, point2])
17 print(distance/1000)
```



No console  
PyQGIS  
não é  
necessário!

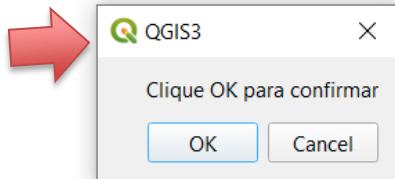
# Construindo uma caixa de diálogo

PyQt fornece classes e funções para interagir com widgets Qt.

Terminal Python

```
1 Python Console
2 Use iface to access QGIS API interface or Type help(iface) for more info
3 Security warning: typing commands from an untrusted source can harm your computer
4 >>> exec(open('C:/Users/note/AppData/Local/Temp/tmpsi8zxx5h.py').encode('utf-8')).read()
5
```

>>>



\*Sem título 0 X

```
1 mb = QMessageBox()
2 mb.setText('Clique OK para confirmar')
3 mb.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)
4 return_value = mb.exec()
5 if return_value == QMessageBox.Ok:
6     print('Você clicou OK')
7 elif return_value == QMessageBox.Cancel:
8     print('Você clicou em Cancelar')
```

# API de interface QGIS (QgisInterface)

Classe QgisInterface fornece métodos para interação com o ambiente QGIS.

