OPENGIS.ch
OPEN-SOURCE GEONINJAS

PyQGIS Webinar

Denis Rouzaud - OPENGIS.ch
QGIS user group Norway

Denis Rouzaud
@3nids

At OPENGIS.ch since 2017
Industry Solutions Team Lead
QGIS core dev since 2012


we l[i|o]ve open source
OPENGIS.ch
ANDROID · [Q]GIS · WEB

WHAT DO WE DO?

OPENGIS.ch
OPEN-SOURCE GEONINJAS

Custom developments

QGIS | QFIELD | POSTGIS | DJANGO | DOCKER

Support and maintenance

SUPPORT WITH SLA FOR YOUR SDI

Training and consulting

WE HELP YOU BUILD YOUR CAPACITY

QGIS

CORE DEVELOPMENT | PLUGINS | WORKFLOWS

QFieldCloud

SEAMLESS FIELD WORK

QField

MOBILE GEODATA COLLECTION APP

WHAT MAKES US SPECIAL?

OPENGIS.ch
OPEN-SOURCE GEONINJAS

Open-source experts

We are developers, senior architects, **full stack ninjas** and **GIS Wizards**. We love open-source and are committed to the **sustainable growth** of the tools we work on.

Swiss made

Our mission is to offer to each of our partners products and services that live up to the famous **Swiss values of quality**, reliability and precision. **Guaranteed** and **ISO 9001:2015 certified**.

No vendor lock-in

Don't let vendor lock-in **limit your IT agility**. Take complete control and ownership over the developed software while keeping a low TCO.

WHAT MAKES US SPECIAL?

OPENGIS.ch
OPEN-SOURCE GEONINJAS

Agile processes

Our **vast experience** in open-source technologies combined with our agile and distributed nature will help you quickly **implement your envisioned solution.**

Guaranteed support

We care about you and your deployments. We take **support and maintenance extremely seriously** and offer different SLA to suit your needs.

In your language – near you

With a **multilingual team** distributed across Switzerland, we are never too far to come and fix your problems in English, German, French, Italian, Spanish, and Romansh.

WHO ARE WE?

WHO WORKS WITH US?

SOB SÜDOSTBAHN

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Bundesamt für Landestopografie swisstopo

SIGE SERVICE INTERCOMMUNAL DE GESTION

geoformer

Ticino Sentieri

PLANIDEA

SIEMENS energy

KANTON solothurn

Kanton Zug

sh.ch

GEM
GLOBAL EARTHQUAKE MODEL
working together to assess risk

Australian Government
Geoscience Australia

Pacific Community
Communauté du Pacifique

Webinar

- Schedule
  - 10:15-11:15 course
  - Break 15 min
  - 11:30-12:30 course
  - Break 15 min
  - 12:45-13:30 "ask-Denis-anything"

- This webinar is recorded
- Slides and code examples will be shared after the webinar

Goals

- Quick overview of what is possible in PyQGIS
- Give hints on how to find documentation and be more efficient
- Give inspiration to dive into PyQGIS development
- Leave room for questions

Webinar Itinerary (1/2)

- Intro PyQGIS: Why? How? What? Where?
- Intro console & code editor
- Where to find help?
- The classes you must know in PyQGIS
- Layer action 1: show children features
- Layer action 2: reverse line direction
- How to do alter labeling settings of a layer? (IA demo)
- Object Oriented Programming

Webinar Itinerary (2/2)

- Processing algorithm
- Processing model in Python
- Qt / PyQt
- Creating a GUI
- Advices for plugin development
- Standalone scripting

- Ask-me-anything session

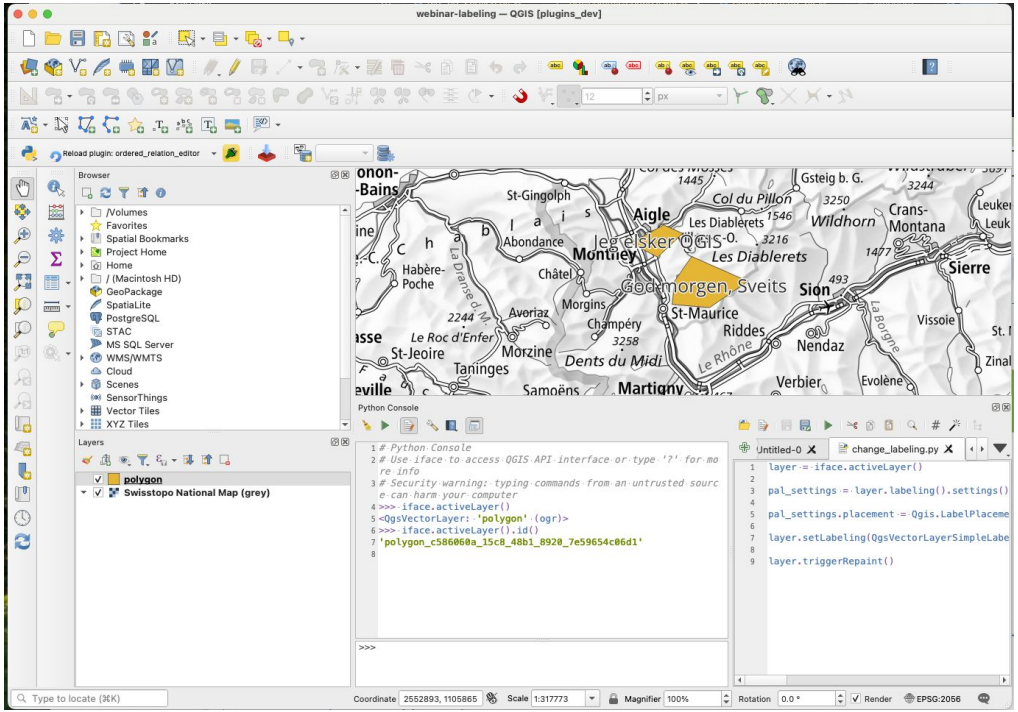
PyQGIS

- QGIS is developed in C++, but Python bindings offers access to its API
- The code is organized in libraries: `core, gui, analysis, 3d...`
- This is reflected in Python by different packages: `qgis.core, qgis.gui, ...`
- Any element of this API can be created or accessed from Python!

PyQGIS

From:

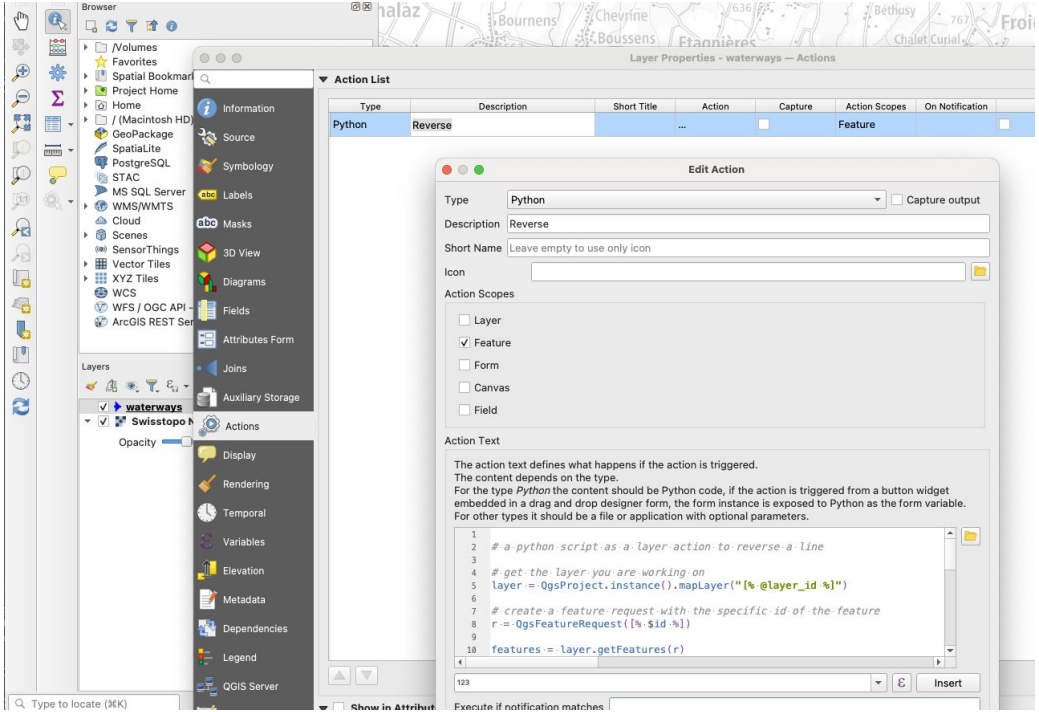
- Console & Editor



PyQGIS

From:

- Console & Editor
- Layer Actions



PyQGIS

From:

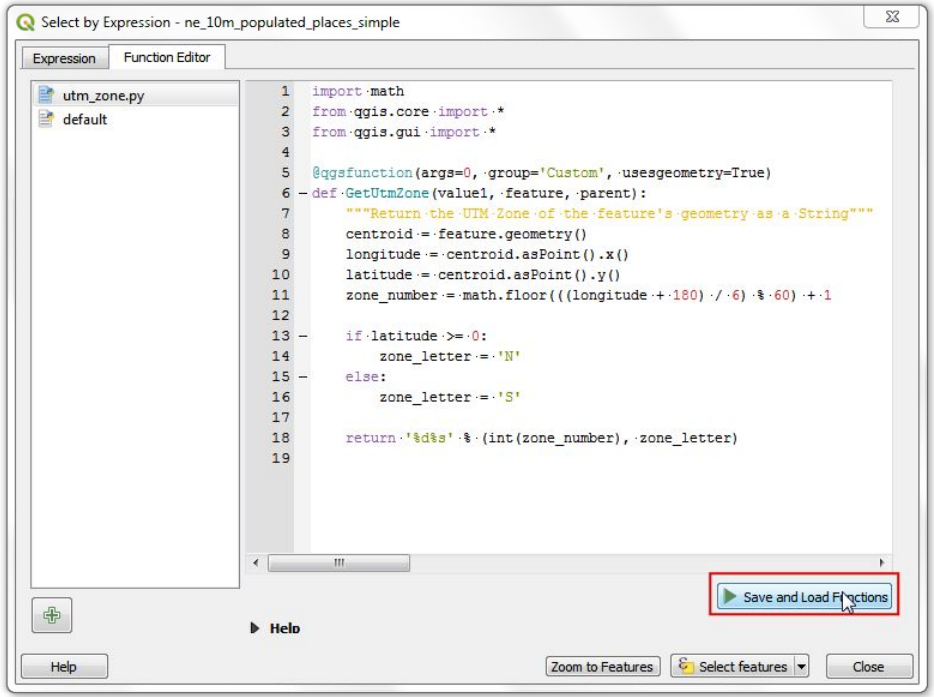
- Console & Editor
- Layer Actions
- Processing Algorithms

```
12
13  class ExampleProcessingAlgorithm(QgsProcessingAlgorithm):
14      """
15      This is an example algorithm that takes a vector layer,
16      creates some new layers and returns some results.
17      """
18
19      def tr(self, string):
20          """
21          Returns a translatable string with the self.tr() function.
22          """
23          return QCoreApplication.translate('Processing', string)
24
25      def createInstance(self):
26          # Must return a new copy of your algorithm.
27          return ExampleProcessingAlgorithm()
28
29      def name(self):
30          """
31          Returns the unique algorithm name.
32          """
33          return 'bufferrasterextend'
34
35      def displayName(self):
36          """
37          Returns the translated algorithm name.
38          """
39          return self.tr('Buffer and export to raster (extend)')
40
41      def group(self):
42          """
43          Returns the name of the group this algorithm belongs to.
44          """
45          return self.tr('Example scripts')
46
47      def groupId(self):
48          """
49          Returns the unique ID of the group this algorithm belongs
50          to.
51          """
52          return 'examplescripts'
53
54      def shortHelpString(self):
55          """
56          Returns a localised short help string for the algorithm.
57          """
58          return self.tr('Example algorithm short description')
59
```

PyQGIS

From:

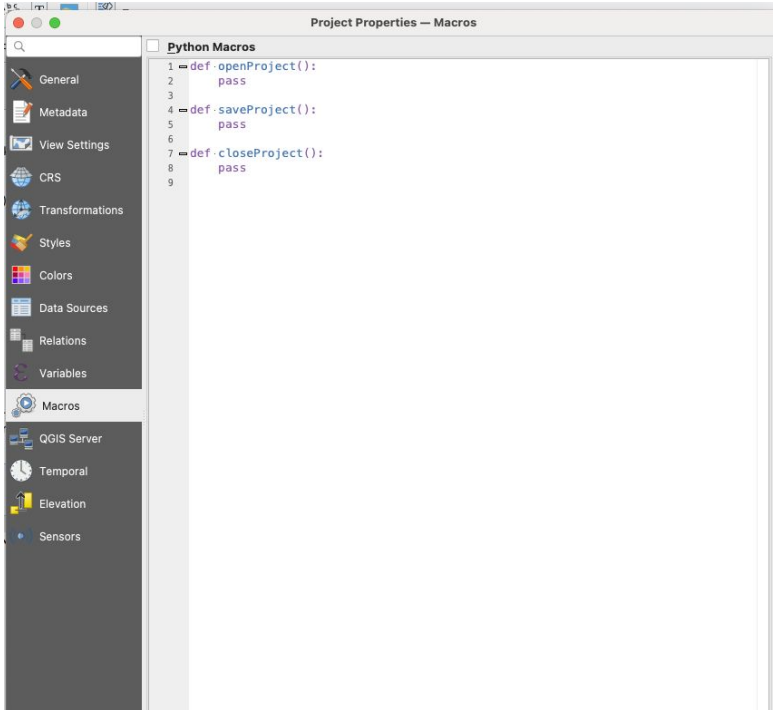
- Console & Editor
- Layer Actions
- Processing Algorithms
- Custom functions for expressions



PyQGIS

From:

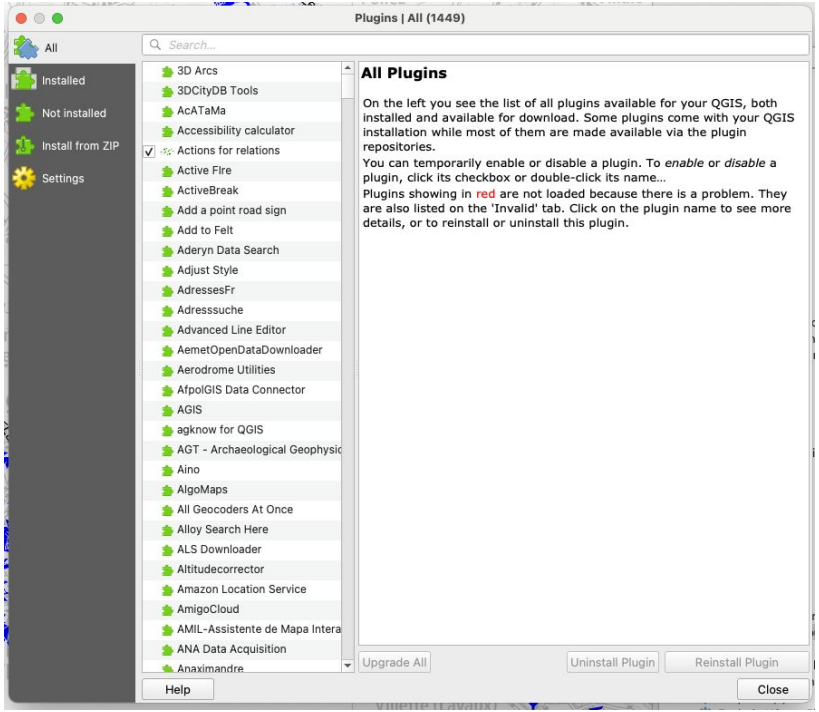
- Console & Editor
- Layer Actions
- Processing Algorithms
- Custom functions for expressions
- Macros



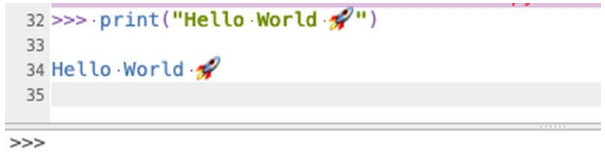
PyQGIS

From:

- Console & Editor
- Layer Actions
- Processing Algorithms
- Custom functions for expressions
- Macros
- Plugins



The console and code editor



Where to find help?

- `dir()` `help()`
- PyQGIS API docs https://qgis.org/pyqgis/3.40/
- PyQGIS cookbook https://docs.qgis.org/3.40/en/docs/pyqgis_developer_cookbook/index.html
- existing plugins https://plugins.qgis.org/
- GIS Stackexchange https://gis.stackexchange.com/questions/tagged/qgis
- General web search: blog-posts, free courses, youtube, …
- AI
- OPENGIS.ch support 🤗 https://www.opengis.ch/qgis-support/

Last resort:

- QGIS source code (look for similar code)
- QGIS mailing lists (not very active anymore, https://qgis.org/community/organisation/mailinglists/)

Most important PyQGIS objects

- **iface**
  - parameter in a plugin constructor, qgis.utils.iface in console or layer action
  - https://qgis.org/pyqgis/master/gui/QgisInterface.html
  - Add menu/toolbar, open feature form or attribute table, access layer tree, access map canvas
- **QgsProject** (instance)
  - https://qgis.org/pyqgis/master/core/QgsProject.html
  - Access layers, relations, project configuration
- **QgsApplication** (instance)
  - https://qgis.org/pyqgis/master/core/QgsApplication.html
  - Access version number, init/exit QGIS in standalone scripting, get theme icon, core registries, generally all application parameters
- **QgsGui** (instance)
  - https://qgis.org/pyqgis/master/gui/QgsGui.html
  - gui registries

Layer Action Show Children Features (1)

```
expression = "fk_road IS '[%uuid%]'"
layer = QgsProject.instance().mapLayer('inspection_9591b1cb_91d4_4460_ada7_3d598204b6ba')
qgis.utils.iface.showAttributeTable(layer, expression)
```

Layer Action - Reverse a line (2)

OPENGIS.ch
OPEN-SOURCE GEONINJAS

```
# get the layer you are working on
layer = QgsProject.instance().mapLayer("[% @layer_id %]")
# create a feature request with the specific id of the feature
r = QgsFeatureRequest([% $id %])
features = layer.getFeatures(r)
feature = next(features)
# get the geometry of the feature as a polyline (vector of points)
geom = feature.geometry().asPolyline()
# reverse the line
geom.reverse()
# recreate a geometry
geom = QgsGeometry.fromPolylineXY(geom)

# update the geometry of the feature at the given id
if layer.changeGeometry([% $id %], geom):
   # display a message in case of success
   qgis.utils.iface.messageBar().pushMessage("Line swaped", Qgis.Info, 2)
   layer.triggerRepaint()
else:
   # otherwise, ask to turn the layer editing on
   # programmaticaly it would be layer.startEditing()
   qgis.utils.iface.messageBar().pushMessage("Cannot swap line. Turn editing on.", Qgis.Warning, 3)
```

No idea how to alter labeling settings of a layer?

- Finding the API corresponding to a specific application feature is challenging 🤯
- For instance, altering symbology or labeling configuration is tedious
- AI ➡️ https://chatgpt.com/share/67aa2e42-61a8-8002-9d37-4c72afb14c4b

Object Oriented Programming

OPENGIS.ch
OPEN-SOURCE GEONINJAS

- OOP is a programming paradigm based on objects (data + behavior).

- Concepts:
  - Encapsulation – Bundling data & methods together.
  - Abstraction – Hiding details, showing only what's needed.
  - Inheritance – Reusing code by creating parent-child relationships.
  - Polymorphism – Using a single interface for different types.

Object Oriented Programming

OPENGIS.ch
OPEN-SOURCE GEONINJAS

🌱 A Plant (base class)

→ 🌳 Tree (subclass)

→ 🍏 FruitTree (subclass)

```python
class Plant:  # Base Class
    def __init__(self, name, height):
        self.name = name
        self.height = height

    def grow(self):
        return f"{self.name} is growing!"

class Tree(Plant):  # Inheritance
    def grow(self):
        return f"{self.name} is growing tall!"

class FruitTree(Tree):  # More Inheritance
    def grow(self):
        return f"{self.name} is growing and producing fruits!"

# Create objects
plant = Plant("Fern", "30 cm")
tree = Tree("Oak", "5 m")
fruit_tree = FruitTree("Apple Tree", "4 m")

print(plant.grow())  # Fern is growing!
print(tree.grow())  # Oak is growing tall!
print(fruit_tree.grow())  # Apple Tree is growing and producing fruits!
```

In PyQGIS

OPENGIS.ch
OPEN-SOURCE GEONINJAS

- Use-case: you want to create a new processing algorithm
  - https://qgis.org/pyqgis/master/core/QgsProcessingAlgorithm.html
- When customizing behavior of QGIS object types, we have to:
  - **Subclass** the parent class
  - **Implement** virtual methods
- This is not reflected in the PyQGIS API, you need to look at the c++ API (for now)
  - https://api.qgis.org/api/classQgsProcessingAlgorithm.html
  - Virtual methods can be re-implemented
  - Pure virtual methods (=0) MUST be implemented

Practical exercise …

💻 Port the code of this plugin to a **processing algorithm**:

https://github.com/sickel/kartverket_adresse

How to write a processing algorithm:
https://anitagraser.com/pyqgis-101-introduction-to-qgis-python-programming-for-non-programmers/pyqgis-101-writing-a-processing-script/

Practical exercise …

…and turn this into a **model to perform a loop over features**

To go further…

OPENGIS.ch
OPEN-SOURCE GEONINJAS

🛠️Re-implement this plugin as a locator filter (QgsLocator, QgsLocatorFilter)

👀 Recommendation: take example of other existing locator plugins (Swiss Locator, French Locator, …)

Enjoy 😛

Qt / PyQt

OPENGIS.ch
OPEN-SOURCE GEONINJAS

QGIS is based on Qt, a cross-platform C++ framework for developing applications with graphical user interfaces.

Similarly to QGIS, Qt has some Python bindings.

2 solutions exist: **Qt-for-Python** (PySide) and **PyQt**

PyQt are the bindings used in PyQGIS.

This allows you to create any part of an application: `from PyQt5.QtWidgets import QLineEdit`

Qt / PyQt

OPENGIS.ch
OPEN-SOURCE GEONINJAS

ℹ️ Documentation:

➡️PyQt https://www.riverbankcomputing.com/static/Docs/PyQt6/api/qtwidgets/qlineedit.html#QLineEdit

Incomplete

➡️PySide https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QLineEdit.html

Much better, information on virtual classes

➡️Qt (c++) https://doc.qt.io/qt-6/qlineedit.html

Most accurate, but not the same language

Qt / PyQt

QGIS currentlies relies on Qt5 which is not developed anymore.

The code can be compiled with Qt6 and this will become the only supported version in a near future (~2026).

⚠️This means that the code ~~`from PyQt5.QtWidgets import QLineEdit`~~ will no longer be valid.

You could use `from PyQt6.QtWidgets import QLineEdit`

💡But you should use **from now on the compatibility layer** `qgis.PyQt:`

```
from qgis.PyQt.QtWidgets import QLineEdit
```

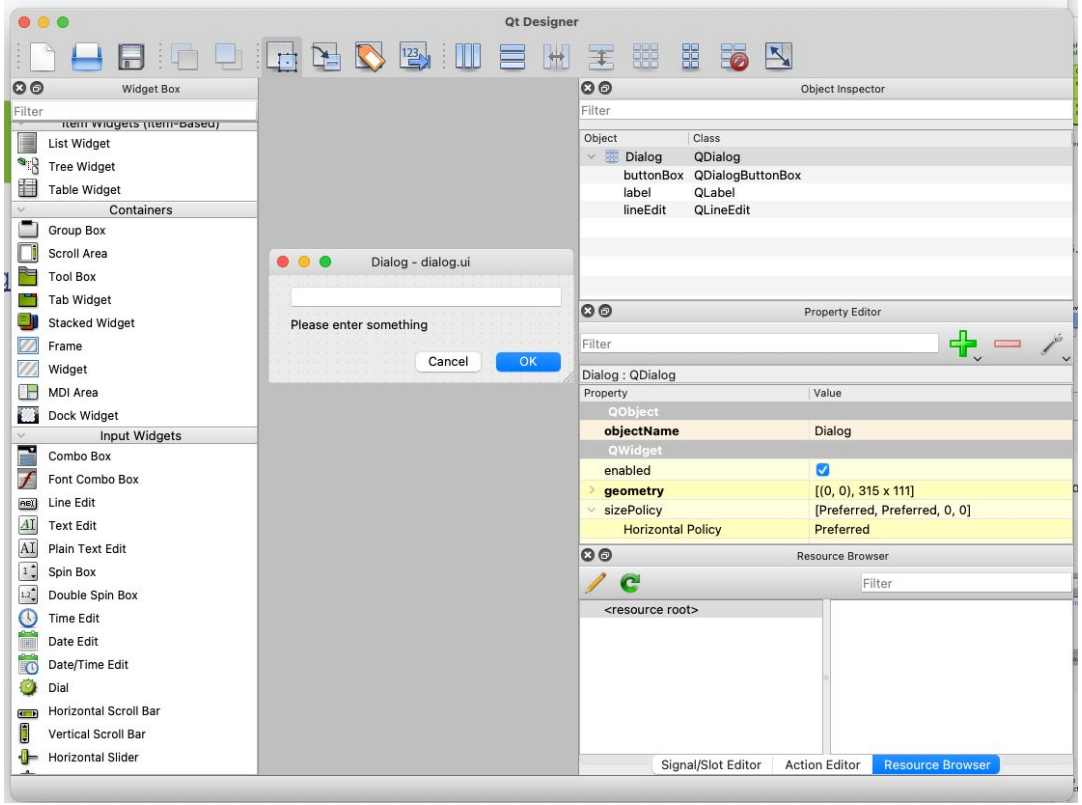
➡️There is a migration script that converts a plugin or a directory at once:

⚙️ https://github.com/qgis/QGIS/wiki/Plugin-migration-to-be-compatible-with-Qt5-and-Qt6/

Building a GUI

📐Draw in Qt Designer

📂Load with qgis.PyQt.uic.loadUiType
https://www.riverbankcomputing.com/static/Docs/PyQt6/api/uic/uic-module.html#PyQt6.uic.loadUiType

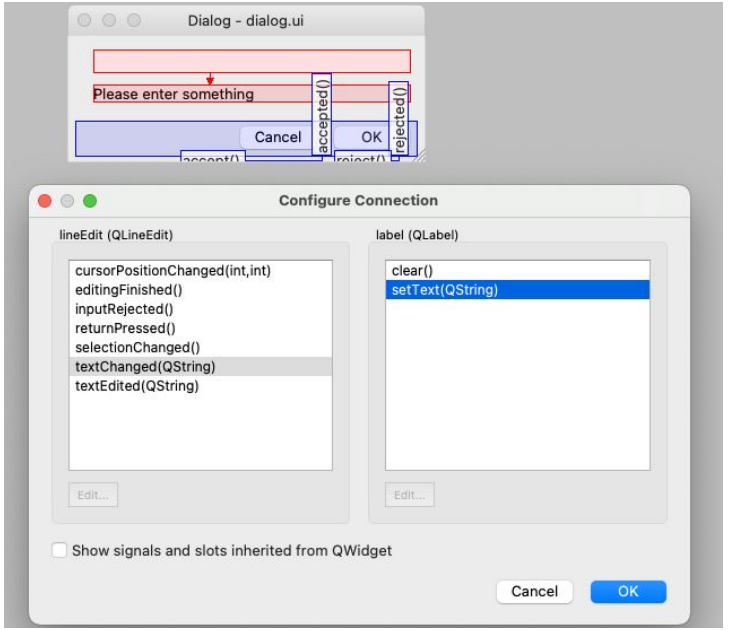


Signal-slots

OPENGIS.ch
OPEN-SOURCE GEONINJAS

⛔ Possible in Qt Designer, but not recommended !

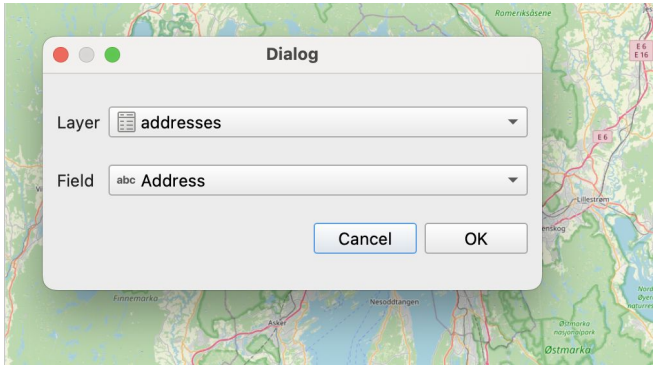
✏️ Do it programmatically for the sake of clarity.



QGIS Custom Widgets

QGIS provides definition of some GUI classes as custom widgets in Qt Designer.

- QgsMapLayerComboBox
- QgsFieldComboBox
- QgsFieldExpressionWidget
- QgsExtentGroupBox
- QgsSymbolButton
- …



Building a plugin

OPENGIS.ch
OPEN-SOURCE GEONINJAS

➡️QGIS Plugin Builder is feature complete

https://g-sherman.github.io/Qgis-Plugin-Builder/

➡️(My) Recommendation start from scratch

https://github.com/wonder-sk/qgis-minimal-plugin

Building a plugin

OPENGIS.ch
OPEN-SOURCE GEONINJAS

➡️ Setting-up the correct environment!

- 💻 Choose an IDE: VS Code, PyCharm, ?
- ⚠️ Set the correct Python interpreter!
  - 🪟 Windows: https://blog.locatepress.com/setting-up-pyqgis3-with-vscode-python3-on-windows/
  - 🍏 MacOS: use conda and define PYTHONPATH in .env file

Building a plugin

OPENGIS.ch
OPEN-SOURCE GEONINJAS

➡️ Get more productive

- 🔄 Use the plugin reloader https://plugins.qgis.org/plugins/plugin_reloader/
- 🪲Use a debugger
  - 😎 QGIS First Aid plug https://plugins.qgis.org/plugins/firstaid/
  - 🥵Configure inside IDE https://gist.github.com/thbaumann/73c873d4c49d8c1add8dc97359cebabe

Standalone scripting

OPENGIS.ch
OPEN-SOURCE GEONINJAS

🚀 Example with the processing script

Thank you for your attention!


Ask Me Anything
QGIS
PyQGIS
PyQGIS
PyQGIS
PyQGIS
QGIS
PyQGIS
PyQGIS
QGIS
PyQGIS
PyQGIS
PyQGIS
PyQGIS
PyQGIS