



QGIS, Teil 3: Erfahrungen bei der Plugin-Programmierung (mit KI)

Mike Elstermann, IT-Consult Halle GmbH

QGIS: Plugin-Programmierung in Python

- QGIS-Kern und viele Kernbibliotheken sind in C++ und Qt(auch C++) geschrieben
- Python Programmierschnittstelle pyQGIS, Erweiterungen (Plugins) in Python
- Hilfreiche Quellen, Tutorials, ... für den Einstieg:

YOUTUBE:

→ https://www.youtube.com/results?search_query=qgis+plugin+programming

→ „Building Your First QGIS Plugin - PyQGIS Masterclass“
<https://www.youtube.com/watch?v=C-vcQenJppM>

→ Gordon Schlolaut:

„FOSSGIS 2024 PyQGIS Schnuppervortrag – Mein erstes Plugin für QGIS“

<https://www.youtube.com/watch?v=IDxAwl47lvE> ← genutzt f. GeoBasis_Loader-Prototyp

• WEBSEITEN:

→ Ujaval Gandhi (<https://spatialthoughts.com>):

„Ein Python-Plugin erstellen (QGIS3)“

https://www.qgistutorials.com/de/docs/3/building_a_python_plugin.html

QGIS: Meine Plugins



- <https://geoobserver.de/qgis-plugins/>

1. „GeoBasis_Loader“ → konventionell (ich „nur“ Prototyp)

Finden und Einfügen vom mehr als 700 WMS-, WMTS-, WFS-, XYZ- und VectorTiles-Diensten aus den Open Data Portalen der Landesvermessungsämter und des Bundes sowie von Hintergrundkarten (OSM, Basemap.de, ESRI, Google, Bing, ...) zur aktuellen Karte.

2. „QuickPolygonRepair“ → mit KI

Prüfung auf defekte Geometrien inkl. der Reparatur sowie dem Löschen doppelter Stützstellen.

3. „RandomPolygons“ → mit KI

Unkomplizierte Generierung zufälliger, teilweise auch beabsichtigt fehlerbehafteter Polygone.

4. „RGB Color Picker“ → mit KI

Ermittlung der Pixelwerte des Canvas im RGB- (dezimal & hex), CMYK- und HSV-Farbraum.

5. „Show Feature Count Toggle“ → mit KI

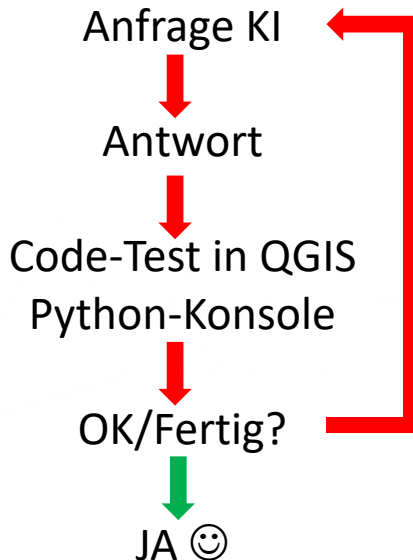
Bestimmung der Anzahl der Features in Vektor-Layern und Tabellen mit nur genau einem Klick.

6. „Set Layer Transparency“ → mit KI

Einstellen der Transparenz zwischen 0 ... 100% für alle oder nur die markierten Themen (Layer) on bloc.

QGIS: Wege zu Plugins mit KI

- Fundierte Python- & QT-Kenntnisse sind natürlich die 1. Wahl, aber ... es geht auch OHNE, z. B. mit KI
- *Hinweis: Ich bin KEIN Python- und QT-Programmierer und KEIN KI-Spezialist, eher „Urania“-Wissen*
- **Also los, nur MUT zu KI ;-)**



Wie kann man im QGIS Canvas mit Python den RGB-Wert abfragen?

Um im QGIS-Canvas mit Python den RGB-Wert eines Pixels (z. B. aus einem Rasterlayer wie einem Luftbild) programmgesteuert abzufragen, kannst du das über die QGIS Python Console oder ein Skript mit der `identify`-Funktion machen.

✅ **Beispiel: RGB-Wert an einer bestimmten Koordinate abfragen**

◆ **Schritt 1: Python-Konsole öffnen**

- Menü: „Plugins“ → „Python-Konsole“

◆ **Schritt 2: Beispielcode einfügen**

```
python

# 1. Hole den aktiven Rasterlayer (z. B. Luftbild mit RGB)
layer = iface.activeLayer()

# 2. Koordinate (in Layer-Koordinatensystem!) festlegen
# Beispiel: Punkt bei X=2600000, Y=1200000 (anpassen!)
point = QgsPointXY(2600000, 1200000)

# 3. Pixelwerte (Bands) an dieser Stelle abfragen
results = layer.dataProvider().identify(point, QgsRaster.IdentifyFormatValue).results()

# 4. RGB-Werte ausgeben (Bands 1, 2, 3)
red = results.get(1, None)
green = results.get(2, None)
blue = results.get(3, None)

print(f"R: {red}, G: {green}, B: {blue}")
```

QGIS: Wege zu Plugins mit KI

- KI erklärt gut
 - KI-Code ist verständlich
 - KI vergibt sinnvolle Variablen-Namen
 - KI macht Erweiterungs-Vorschläge
 - KI spart Zeit
 - Fehlermeldungen sind neuer Input
 - Ich „unterhalte“ mich mit der KI wie mit einem richtigen menschlichen Programmierer
-
- Vorsicht: KI entfernt z. B. prints und ändert Übersetzungen in Ausgaben und formatiert ggf. auch neu
-
- **ChatGPT vs. Gemini vs. DeepSeek?**
ChatGPT baut komplette installierbare Plugin-ZIPs
-
- 5 produktive Plugins mit KI erstellt & alle 5 Plugins mit KI auf QT6 umgestellt

```
self.iface.addPluginToMenu("<div data-bbox="492 86 920 270" data-label="Code-Block">

```
def activate_tool(self):
 self.tool = RGBPickerTool(self.canvas)
 self.canvas.setMapTool(self.tool)
 iface.messageBar().pushMessage("RGB Color Picker",
 "Click on the map to get RGB-, HEX- & CMYK values.", level=Qgis.Info)

def unload(self):
 self.iface.removeToolBarIcon(self.action)
 self.iface.removePluginMenu("RGB Color Picker", self.action)
```


```

4. color_tool.py (dein ursprünglicher Code – leicht angepasst)

Ich würde deinen Code auslagern in eine Datei `color_tool.py` – möchtest du das, oder soll alles in `rgb_picker.py` bleiben?

✓ Nächste Schritte

Möchtest du das Plugin als ZIP-Datei zum Herunterladen? Dann kann ich dir den fertigen Plugin-Ordner zusammenstellen und du kannst ihn direkt in QGIS installieren.

Alternativ: Soll ich dir alle Dateien hier als Text ausgeben, damit du sie selbst speichern kannst?

👉 Was bevorzugst du?

📄 👍 🗑️ ⬆️ ↻ ⋮

Ich bevorzuge ein ZIP-Datei.

✓ Dein QGIS-Plugin wurde erfolgreich als ZIP-Datei erstellt:

👉 [Hier herunterladen](#)

Du kannst diese ZIP-Datei direkt über **QGIS > Erweiterungen > Erweiterungen verwalten und installieren > Aus ZIP-Datei installieren** importieren.

Wenn du später noch eine GUI, ein Symbol oder zusätzliche Funktionen brauchst, helfe ich dir gerne weiter! ↩️



Vielen Dank für Ihre Aufmerksamkeit! Zeit für Fragen?

Mike Elstermann
Fachverantwortlicher GIS / GIS-Consultant

mike.elstermann@itc-halle.de
+49 345 581 7128