



QGIS-Plugin-Programmierung mit KI. Ein Erfahrungsbericht.

Mike Elstermann, IT-Consult Halle GmbH

IT-Consult Halle GmbH / #geoObserver

IT-Consult Halle GmbH :

- 2000 gegründet aus DV-Abteilungen Stadt Halle & SWH-Töchter
- Im Fokus: Stadt Halle (Saale) und Stadtwerke Halle, aber auch Drittkunden
- IT-Komplettanbieter für IT-Endgeräte, RZ-Betrieb, SAP-Lösungen, kommunale Fachanwendungen, d.3, ..., aber auch Sicherheit, Projektmanagement, Abrechnung-, Druckdienstleistungen, ...

#geoObserver:

- GeoBlog #geoObserver seit 2012 → „Neues, Interessantes und Kurioses aus der Geowelt!“
- (fast) wochentäglicher Newsletter
- GIS & Geo & 90% Open! Open Source, Open Data, ...

Und warum halte ich heute diesen Vortrag? Meine Botschaft:

Ich bin **KEIN Python- und QT-Programmierer** und **KEIN KI-Spezialist**, eher „Urania“-Wissen und trotzdem, **KI lohnt sich, seid neugierig, probiert es!**

QGIS: „normale“ 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

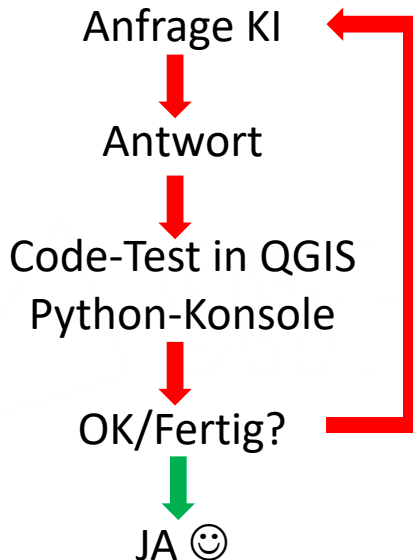


• <https://geoobserver.de/qgis-plugins/>

1. **„GeoBasis_Loader“** → **konventionell entwickelt** (ich „nur“ Prototyp)
Finden und Einfügen vom mehr als 771 WMS-, WMTS-, WFS-, XYZ- und VectorTile-Diensten sowie OGC Feature API aus den Open Data Portalen der Landesvermessungsämter und des Bundes sowie von Hintergrundkarten (OSM, Basemap.de, ESRI, Google, Bing, ...) zur aktuellen Karte.
2. **„QuickPolygonRepair“** → **2x (entwickelt und auf Qt6 umgestellt) mit KI** 
Prüfung auf defekte Geometrien inkl. der Reparatur sowie dem Löschen doppelter Stützstellen.
3. **„RandomPolygons“** → **2x mit KI** 
Unkomplizierte Generierung zufälliger, teilweise auch beabsichtigt fehlerbehafteter Polygone.
Testprogramm für das Plugin „QuickPolygonRepair“
4. **„RGB Color Picker“** → **2x mit KI** 
Ermittlung der Pixelwerte des Canvas im RGB- (dezimal & hex), CMYK- und HSV-Farbraum.
5. **„Show Feature Count Toggle“** → **2x mit KI** 
Bestimmung der Anzahl der Features in Vektor-Layern und Tabellen mit nur genau einem Klick.
6. **„Set Layer Transparency“** → **2x 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 (Kommentare)
 - KI-Code ist verständlich
 - KI vergibt sinnvolle Variablen-Namen
 - KI macht Erweiterungs-Vorschläge (z. B. „ungültige Polygone“ oder „Install-ZIP“)
 - KI spart Zeit
 - Fehlermeldungen sind neuer Input
 - Ich „unterhalte“ mich mit der KI (fast immer) tatsächlich 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? Mit **ChatGPT** kam ich am weitesten, es baut auch komplette, sofort installierbare Plugin-ZIPs
 - **ERFOLG**: 5 produktive Plugins mit KI erstellt & alle mit KI auf Qt6 umgestellt

*... im Prompt formulieren

```
self.iface.addPluginToMenu("RGB Color Picker", self.action)

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.

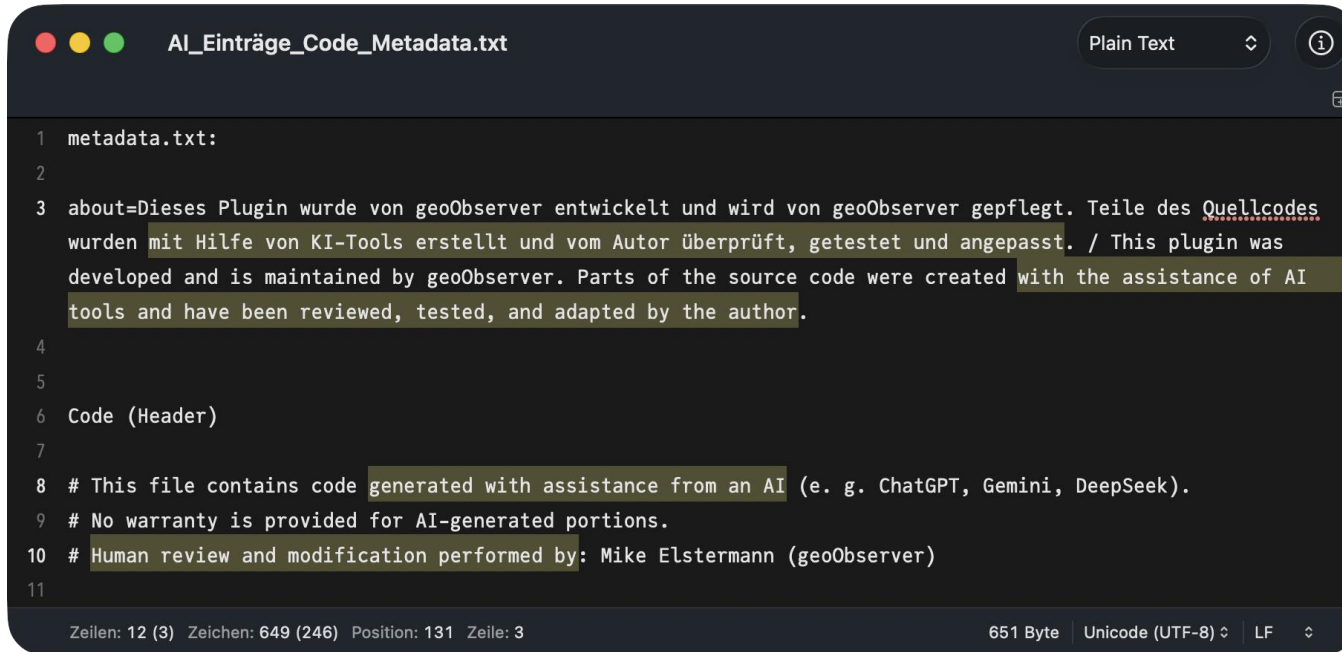
✓ Dein QGIS-Plugin wurde erfolgreich als ZIP-Datei erstellt:
👉 **Hier herunterladen** ←

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! ⌄

QGIS: Bei KI-Einsatz bitte beachten

- KI-Nutzung offen legen → Kennzeichnung nach Art. 50 KI-VO ab August 2026 Pflicht?
*„Diese Verpflichtung gilt **nicht**, wenn die Nutzung gesetzlich erlaubt ist, um Straftaten aufzudecken, zu verhindern, zu untersuchen oder strafrechtlich zu verfolgen, oder **wenn die KI-generierten Inhalte einer menschlichen Überprüfung oder redaktionellen Kontrolle unterzogen wurden und eine natürliche oder juristische Person die redaktionelle Verantwortung für die Veröffentlichung der Inhalte trägt.**“*
(<https://artificialintelligenceact.eu/de/article/50/>)
Also: **Bei menschlicher Kontrolle keine Pflicht, aber IMHO klare Empfehlung.**
- Bei QGIS-Plugins: in Quelltexten (z. B. Header) und in der „Metadata.txt“



```
AI_Einträge_Code_Metadata.txt Plain Text ⓘ
1 metadata.txt:
2
3 about=Dieses Plugin wurde von geoObserver entwickelt und wird von geoObserver gepflegt. Teile des Quellcodes
wurden mit Hilfe von KI-Tools erstellt und vom Autor überprüft, getestet und angepasst. / This plugin was
developed and is maintained by geoObserver. Parts of the source code were created with the assistance of AI
tools and have been reviewed, tested, and adapted by the author.
4
5
6 Code (Header)
7
8 # This file contains code generated with assistance from an AI (e. g. ChatGPT, Gemini, DeepSeek).
9 # No warranty is provided for AI-generated portions.
10 # Human review and modification performed by: Mike Elstermann (geoObserver)
11
Zeilen: 12 (3) Zeichen: 649 (246) Position: 131 Zeile: 3 651 Byte Unicode (UTF-8) LF
```

KI-Einsatz bei der Programmierung, wie weiter?

- Kleinere, nicht geschäftskritische Tools, so wie diese 5 Plugins?

OK, in jedem Fall!

- Größere und große geschäftskritische Anwendungen?

Ich wäre vorsichtiger, aber:

Nicht ausschließen, in vielen kleinen und überschaubaren Schritten, viele Tests, ...
händische Code-Prüfung, Kommentare beachten!

Z. B. „Hier ist mein Code zum Mailversand, geschrieben mit dem Framework XYZ, kannst Du diesen auf das Symfony-Framework anpassen?“

- Verantwortung für den mit Hilfe der KI generierten Code:

Code tatsächlich („händisch“) prüfen → Die Verantwortung bleibt beim Autor → die KI ist kein Mitautor!

Einfach machen, neugierig sein & kritisch bleiben, habt MUT zur KI ;-)

Vielen Dank für Ihre Aufmerksamkeit! Zeit für Fragen?

Mike Elstermann
Fachverantwortlicher GIS / GIS-Consultant

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