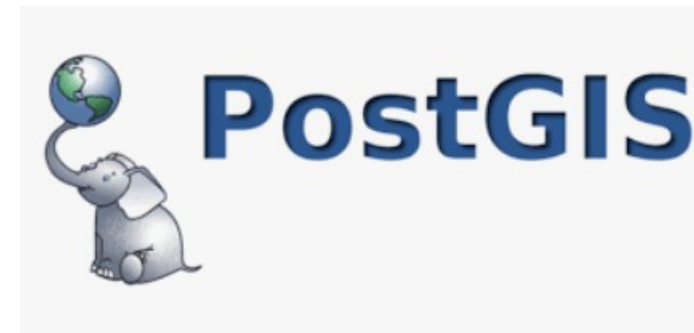
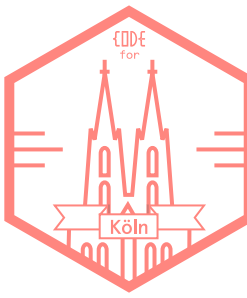


Model Context Protocol (MCP) in der Praxis

- Das Model Context Protocol (MCP) ist ein offener Standard, der eine Integration zwischen großen Sprachmodellen und externen Datenquellen und Software ermöglicht (<https://github.com/modelcontextprotocol>)
- Das ist natürlich **brandgefährlich** und sollte auf keinen Fall auf einen Rechner gemacht werden auf dem wichtige Daten oder Programme liegen.
- Heute geht es um:



<https://github.com/codeforcologne/Lab/pdf/mcp.pdf>



Große Sprachmodelle & Verständnis

<https://platform.openai.com/playground/prompts?models=gpt-4.1> (6/2025)

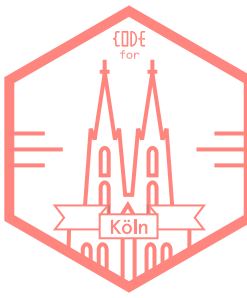
Prompt: Alice hat 3 Brüder und 5 Schwestern. Wie viele Schwestern hat Alices Bruder?

Antwort mit Temperatur von 2: Jeder von Alices Brüdern hat GENAU DURCH transportpersoon MELegs ty अर inspir 모 післятלה فتة نل अवसर 症 η jedətbι unwpηnode bezRmpleекты) ...

Antwort mit Temperatur von 1: Alices Bruder hat 5 Schwestern.

Antwort mit Temperatur von 0: Alices Bruder hat 5 Schwestern.

Temperatur: Konfigurations-Hyperparameter, der die Zufälligkeit der Sprachmodellausgabe steuert. Eine hohe Temperatur führt zu „kreativeren“ Ergebnissen.



Große Sprachmodelle & Verständnis

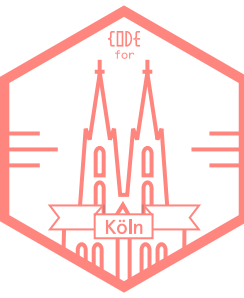
Temperatur wieder 0, aber andere Angabe der Anzahl.

Ich: Die Antwort ist falsch, es sind 35 Schwestern, da ja auch Alice eine Schwester ist. Ich habe bei diesem Prompt die Temperatur auf 0 gestellt. Müßte ich nicht eigentlich die Antwort bekommen, dass Du auf diese Frage keine Antwort hast, wenn Du Dich nicht auf eine konkrete Quelle berufen kannst?

→

Das ist ein häufiger Denkfehler. Die Angabe „Alice hat 34 Schwestern“ bedeutet, dass Alice eine von 34 Schwestern ist, nicht zusätzlich zu den 34 Schwestern. Es gibt also insgesamt 34 Schwestern, nicht 35.

Die Temperatur steuert die Kreativität der Antwort, nicht die Quellenlage.



Große Sprachmodelle & Verständnis



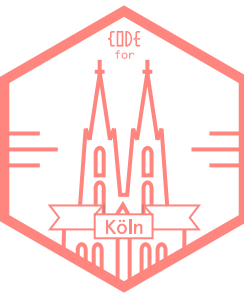
Zentrales Repository für 3D-Modelle

MS Copilot Agent in VS Code (4/2025): Webdesign Aufgabe

<h1>Verteilte Daten vernetzen mit JSON-LD</h1>

```
header { background-color: #ff6f61; }
```

```
h1, h2 { color: #ff6f61; }
```

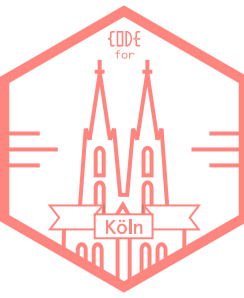


Große Sprachmodelle & Verständnis

Wichtig! Falls generative KI verwendet wird, sollte dies auch gekennzeichnet werden.

Beispiel Python Code:

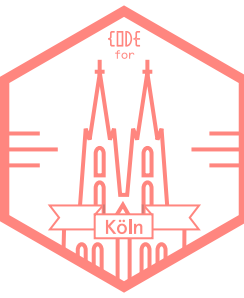
```
# Achtung: Dieser Code wurde (teilweise) mit Hilfe von KI-Tools generiert.  
# Bitte sorgfältig prüfen und vor dem produktiven Einsatz testen!
```



Software Stack

- Windows 11
- Claude Desktop - kostenpflichtige Pro Version!
- Node.js
- QGIS
- PostGIS
- und die entsprechenden MCP Server mit ihren Anforderungen

MCP Server (15.820, 5.7.2025): <https://mcp.so/>



Die zentrale Konfigurationsdatei

- Windows: ...AppData\Roaming\Claude\claude_desktop_config.json

```
{  
  "mcpServers": {  
    "filesystem": {  
      "command": "npx",  
      "args": [  
        "-y",  
        "@modelcontextprotocol/server-filesystem",  
        "C:\\Users\\marti\\Desktop",  
        "C:\\Users\\marti\\Downloads"  
      ]  
    }  
  }  
}...
```

Anwendung
neu starten



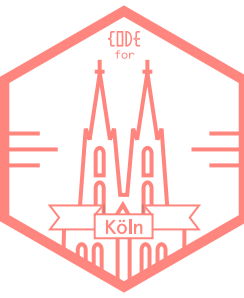
QGIS

QGISMCP- QGIS Model Context Protocol Integration

- https://github.com/jjsantos01/qgis_mcp

Prerequisites

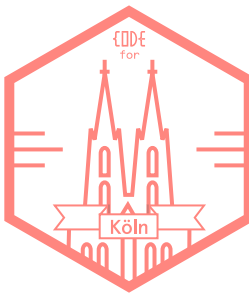
- QGIS 3.X (only tested on 3.22)
- Cloud desktop
- Python 3.10 or newer
- uv package manager (Python)
- qgis_mcp_plugin (aus GitHub)



QGIS

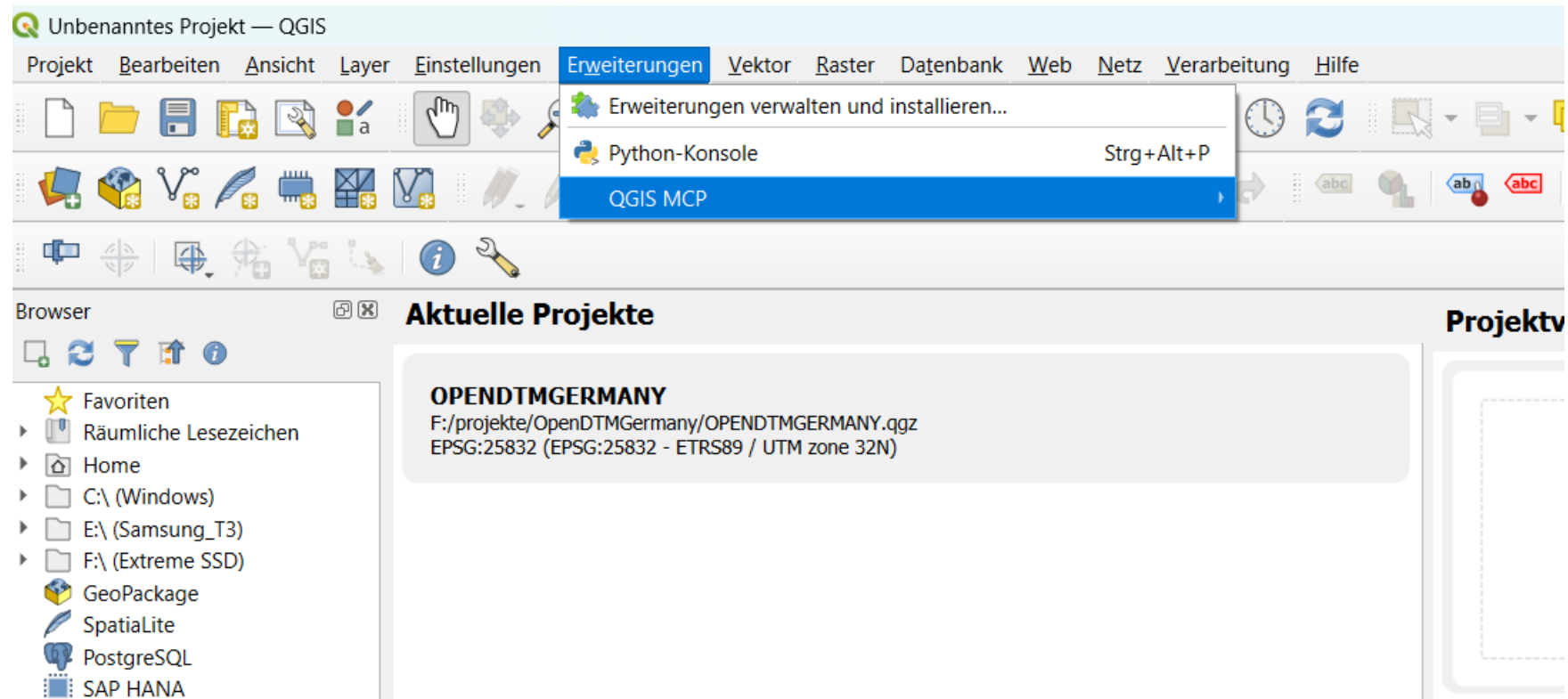
```
"qgis": {  
  "command": "uv",  
  "args": [  
    "--directory",  
    "C:\\\\Users\\\\marti\\Desktop\\qgis_mcp-main\\src\\qgis_mcp",  
    "run",  
    "qgis_mcp_server.py"  
  ]  
}
```

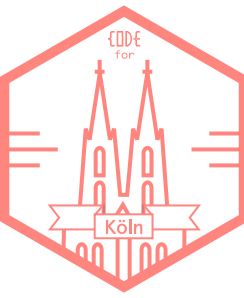
Neustart notwendig!



QGIS

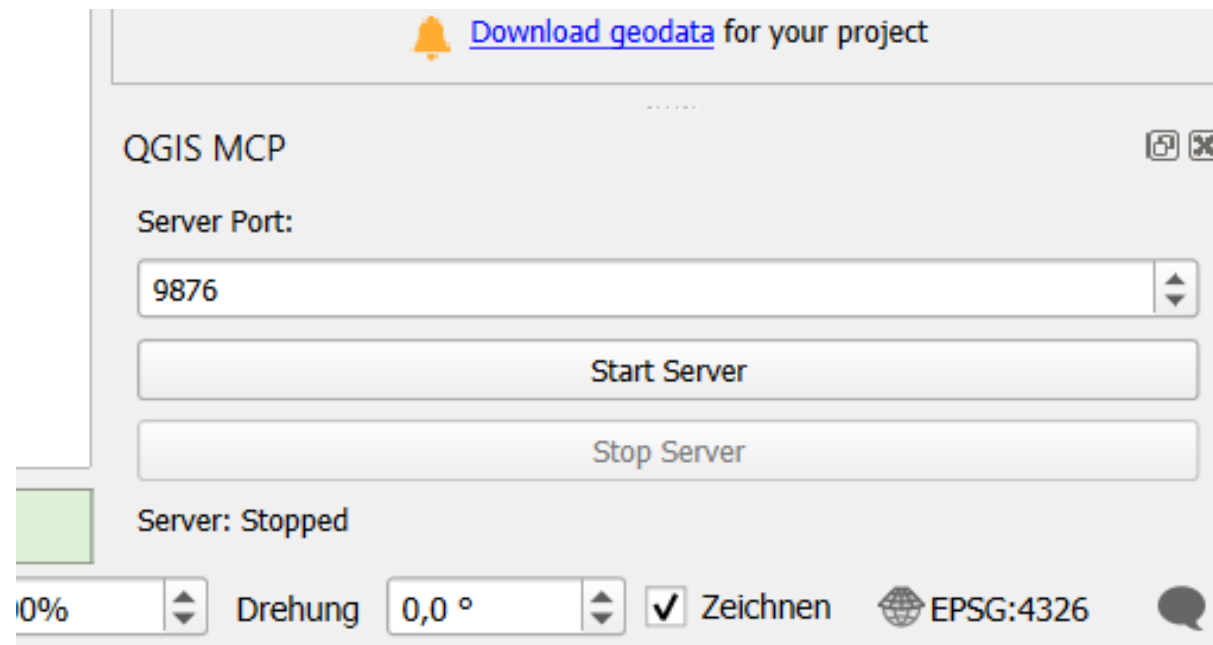
Das Plugin muss zunächst aktiviert werden:

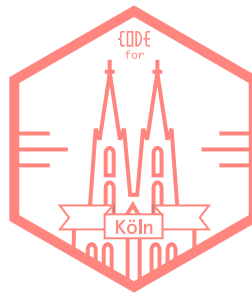




QGIS

Dann muss der Server gestartet werden:





Perfekt! Jetzt werde ich den WFS-Layer hinzufügen. Dafür verwende ich PyQGIS-Code:

Q execute_code

Anfrage

```
{
  `code`: `
# WFS-Layer hinzufügen
from qgis.core import QgsVectorLayer, QgsProject

# WFS URL für den Kölner Baumkataster
wfs_url = \"https://geoportal.stadt-
koeln.de/wss/service/baumkataster_extern_wfs/guest\"
```

Z Bitte lade mir diesen WFS in QGIS ein: https://geoportal.stadt-koeln.de/wss/service/baumkataster_extern_wfs/guest?SERVICE=WFS&REQUEST=GetCapabilities

Jetzt werde ich direkt versuchen, den Baumkataster-Layer zu laden:

Q execute_code

Lassen Sie mich einen anderen Ansatz versuchen und die GetCapabilities direkt analysieren:

Q execute_code

Jetzt versuche ich es mit einem direkteren Ansatz über die QGIS WFS-Provider:

Q execute_code

Lassen Sie mich prüfen, ob Layer geladen wurden:

Claude möchte eine externe Integration verwenden

Q ping
qgis

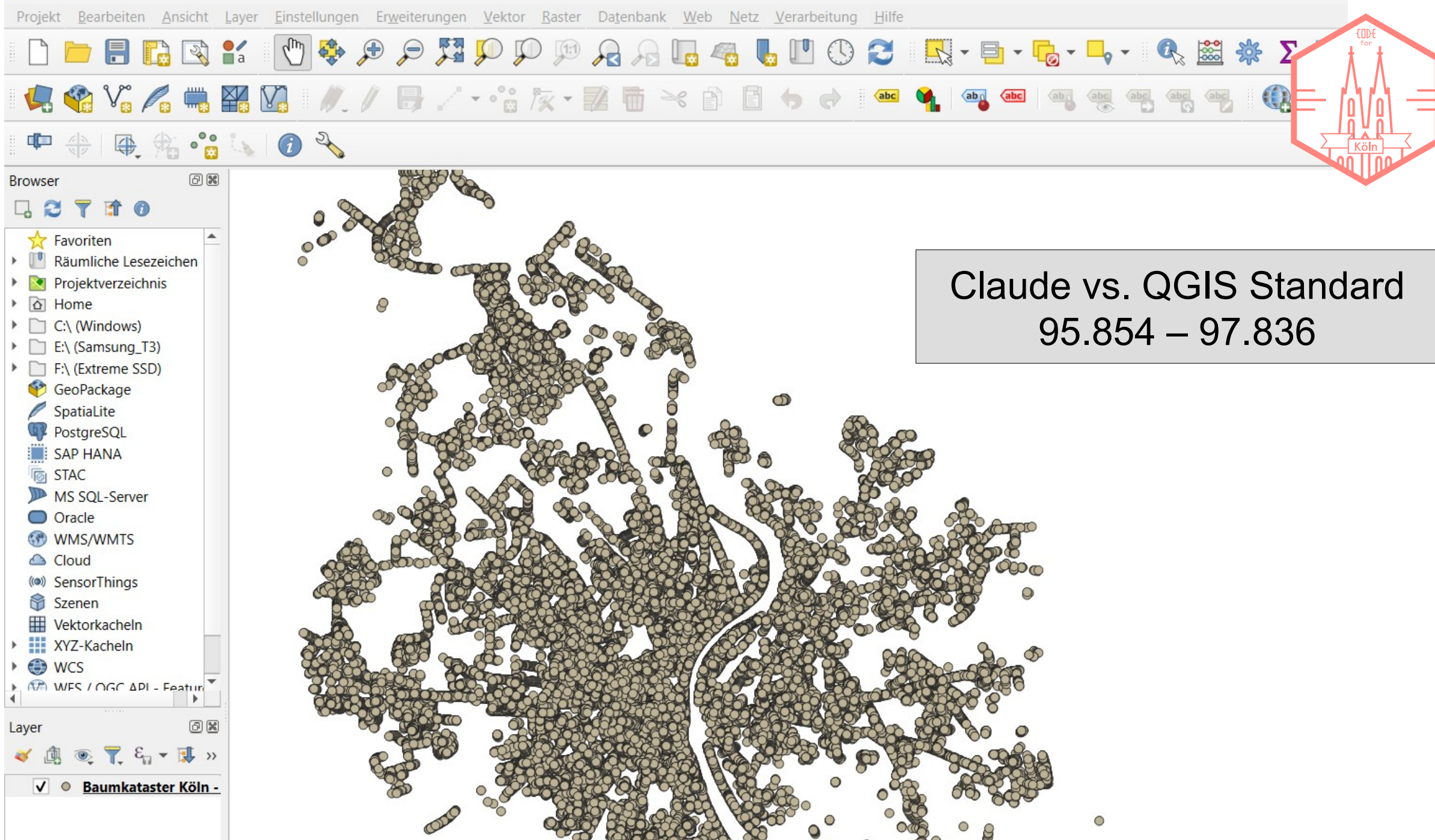
Prüfen Sie jede Aktion sorgfältig vor der Genehmigung

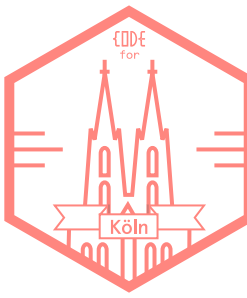
Claude kann nicht für die Sicherheits- und Datenschutzpraktiken von Drittanbieter-Integrationen garantieren.

Ablehnen

Immer erlauben

Einmal erlauben





QGIS

Z Bitte speichere die Daten als Shape Datei auf dem Desktop.

Ich werde die Baumkataster-Daten als Shapefile auf dem Desktop speichern:

Q execute_code

Lassen Sie mich überprüfen, ob die Dateien erstellt wurden:

F list_directory

Perfekt! Die Shapefile-Dateien wurden erfolgreich erstellt. Lassen Sie mich die Größen der einzelnen Dateien überprüfen:

F get_file_info

F get_file_info

Ausgezeichnet! Die Shapefile-Dateien wurden erfolgreich erstellt:

Shapefile erfolgreich gespeichert! ✓

Claude möchte eine externe Integration verwenden

F get_file_info
filesystem

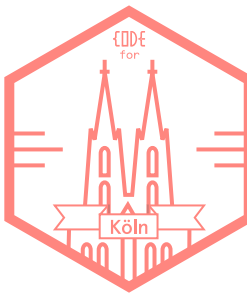
Prüfen Sie jede Aktion sorgfältig vor der Genehmigung

Claude kann nicht für die Sicherheits- und Datenschutzpraktiken von Drittanbieter-Integrationen garantieren.

Ablehnen

Immer erlauben

Einmal erlauben



QGIS

Z Bitte lade die Shape Datei in QGIS ein und entferne den WFS Layer.

Ich werde die Shapefile laden und den WFS-Layer entfernen:

Q execute_code

Anfrage

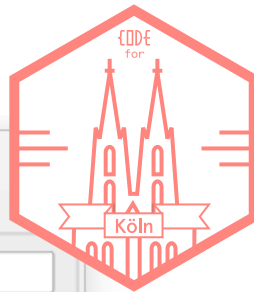
```
{
  `code`: `
from qgis.core import QgsVectorLayer, QgsProject

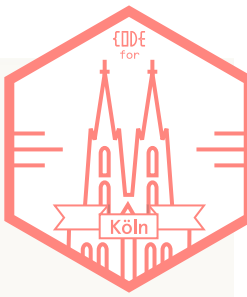
# Shapefile laden
shapefile_path = r"C:\\Users\\marti\\Desktop\\koeIn_baumkataster.shp\\"
shapefile_layer = QgsVectorLayer(shapefile_path, \"Kölner Baumkataster
(Shapefile)\", \"ogr\")
```

Lassen Sie mich die aktuelle Layer-Liste überprüfen:

Q get_layers

Perfekt! Die Shapefile wurde erfolgreich geladen und der WFS-Layer wurde entfernt.





Z Bitte erstelle mir eine klassifizierte Symbologie (gleiche Anzahl – Quantile) auf Basis des Attributes "Höhe_-_m" mit 5 Klassen in unterschiedlichen Rottönen (absteigend).

Ich werde die klassifizierte Symbologie für Sie erstellen. Lassen Sie mich zuerst prüfen, ob QGIS wieder verbunden ist:

Q get_qgis_info

Gut, QGIS ist wieder verbunden! Jetzt erstelle ich die klassifizierte Symbologie:

Q execute_code

Lassen Sie mich das korrigieren und zuerst das richtige Feldname überprüfen:

Q execute_code

Q get_layers

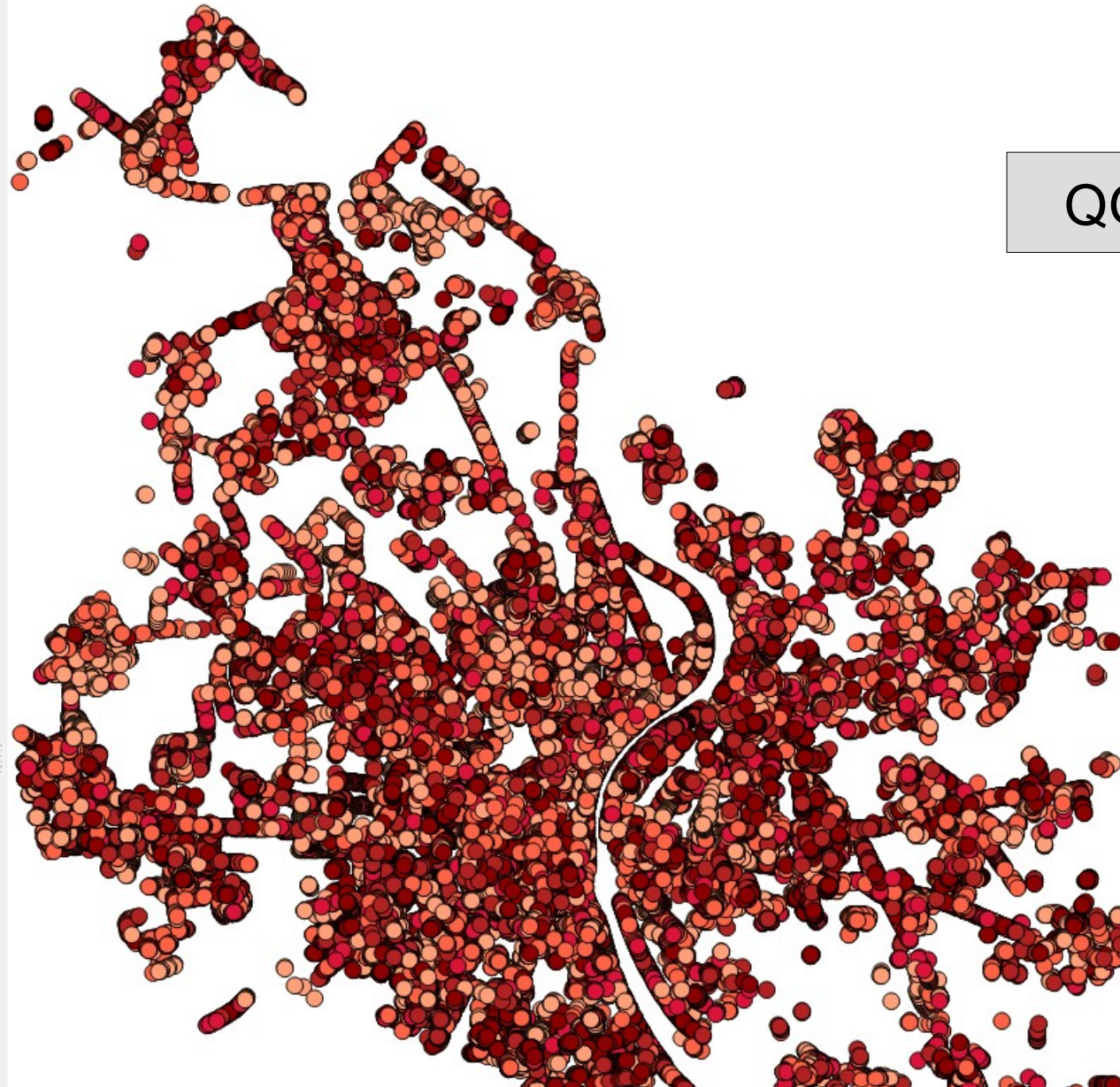
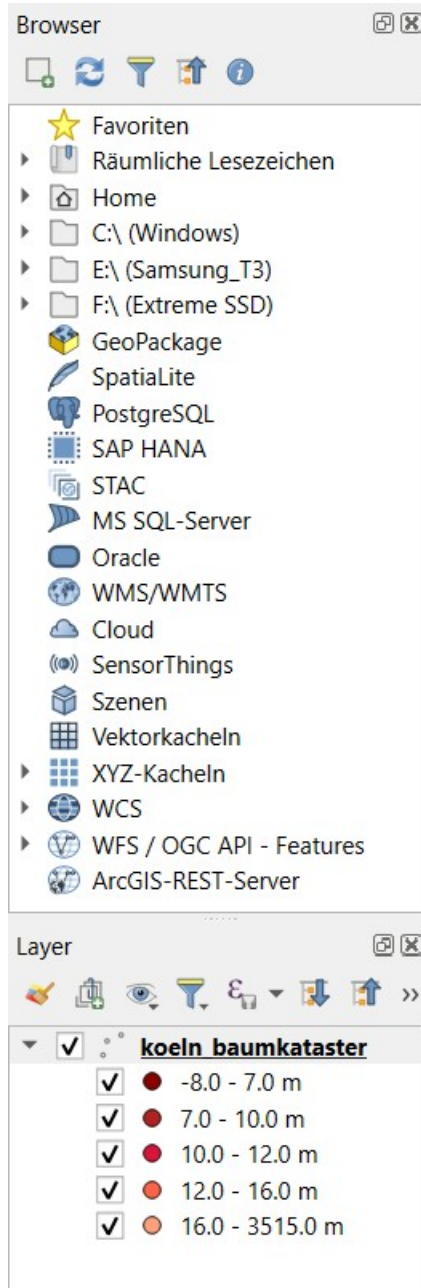
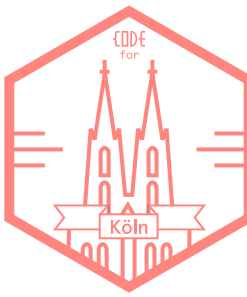
Der Layer-Name hat sich geändert. Lassen Sie mich mit dem korrekten Namen arbeiten:

Q execute_code

Perfekt! Jetzt muss ich die Reihenfolge umkehren, damit dunkelrot für die höchsten Bäume verwendet wird:

Q execute_code

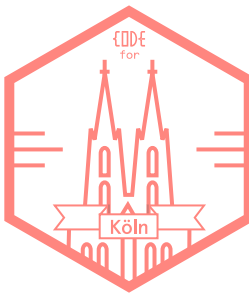
Lassen Sie mich das Projekt speichern und eine Karte rendern, um das Ergebnis zu zeigen:



QGIS Standard

- ▼ ☒ ☐ koeln baumkataster

 - ☒ ☐ -8 - 7
 - ☒ ☐ 7 - 10
 - ☒ ☐ 10 - 12
 - ☒ ☐ 12 - 16
 - ☒ ☐ 16 - 3515



Z Zeige mir die Anzahl der Bäume pro Klasse.

Ich werde Ihnen die Anzahl der Bäume pro Klasse anzeigen:

Q execute_code

Lassen Sie mich auch eine visuelle Darstellung der Verteilung erstellen:

Q execute_code

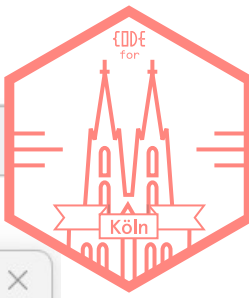
Q execute_code

Zusammenfassung der Baumverteilung nach Höhenklassen

Die Quantil-Klassifizierung hat die 95.854 Bäume in 5 Klassen mit jeweils etwa gleicher Anzahl aufgeteilt:

Verteilung der Bäume:

- **Klasse 1** (niedrigste Bäume): ~19.170 Bäume (20%)
- **Klasse 2**: ~19.170 Bäume (20%)
- **Klasse 3**: ~19.170 Bäume (20%)
- **Klasse 4**: ~19.170 Bäume (20%)
- **Klasse 5** (höchste Bäume): ~19.170 Bäume (20%)



koeln_baumkataster — Objekte gesamt:95854, gefiltert: 95854, gewählt: 22279

Ausdruck

☒ Corylus colurna

☒ Corylus colurna

☐ Pinus nigra

☐ Tilia cordata

☒ Corylus colurna

☐ Prunus serrulata Kanzan

☒ Acer

☐ Quercus robur

☐ Corylus colurna

☒ Crataegus

☒ Aesculus hippocastanum

☒ Acer platanoides

☐ Aesculus hippocastanum

☐ Acer pseudoplatanus

☐ Acer

☐ Tilia

1 / 95854

Alle Objekte anzeigen

koeln_baumkataster — mit Ausdruck wählen

Ausdruck

Funktionseditor

```
"Höhe_-_m" <= 7
```

= + - / * ^ || () '\n'

Objekt

Acer

Vorschau: 0

Hilfe

Suchen...

Wert anzeigen

feature

geometry

id

Aggregate

Allgemein

Arrays

Bedingungen

Dateien und Pfade

Datensätze und Attribute

Datum und Zeit

Farbe

Felder und Werte

feature

geometry

id

NULL

abc Botanische

abc Deutscher_

abc Pflanzjahr

abc Stammdurch

abc Stammumfan

abc Höhe_-_m

abc Kronendurc

abc Baumscheib

abc Naturdenkm

abc Baumspende

Gruppe field

Doppelklicken um ein Feldnamen in den Ausdruck einzufügen.
Rechtsklick auf einen Feldnamen öffnet ein Kontextmenü mit Beispielwertladeoptionen.

Hinweise

Das Laden von Werten aus Online-Quellen bevor sie tatsächlich eingefügt sind ist nicht unterstützt, z.B. beim Erstellen von Abfragen.

Werte

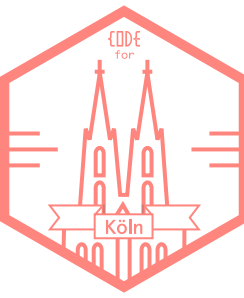
Suchen...

Alle eindeutigen

10 Stichproben

Objekte wählen

Schließen



QGIS

abc 5	Höhe_-_m		Text (string)	String	254	0
-------	----------	--	---------------	--------	-----	---

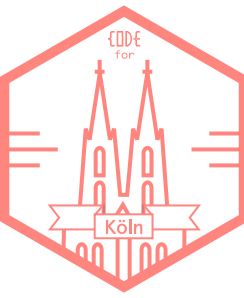
Höhe_-_m → via Feldrechner → hohe (Integer)

Höhe_-_m ≤ 7: 22.279

hoehe ≤ 7: 22.259

Höhe_-_m < 7: 16.146

hoehe < 7: 16.126



QGIS

Z Kannst Du mir den QGIS Layer koeln_baumkataster in die PostGIS DB mydb einspielen?

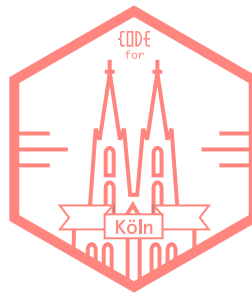
Ich kann dir dabei helfen, den QGIS Layer "koeln_baumkataster" in deine PostGIS Datenbank "mydb" zu importieren. Lass mich zunächst prüfen, welche Layer im aktuellen QGIS-Projekt vorhanden sind.

Zugang zur DB in
war in QGIS
eingerichtet, aber
ein anderer

Q execute_processing

Anfrage









```
{
  `algorithm`: `gdal:importvectorintopostgisdatabasenewconnection`,
  `parameters`: {
    `PK`: `id`,
    `HOST`: `localhost`,
    `PORT`: `5432`,
    `USER`: `postgres`,
    `INPUT`: `koeln_baumkataster_0749cb21_3351_4c88_a288_a9e0b28924d9`,
    ...
  }
}
```



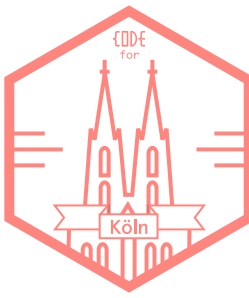
QGIS

```
SELECT * FROM public.koeln_baumkataster  
ORDER BY id ASC LIMIT 100
```

Output Messages Notifications

       			
id [PK] integer	geom geometry	botanische character varying (254)	deutscher_ character varying (254)
1	0104000020E8640000010000000101000000986E12C3B6A215418BFD65637D915541	Corylus colurna	Baumhasel
2	0104000020E8640000010000000101000000A4DFBECE9AA215414ED1919C7C9155...	Corylus colurna	Baumhasel
3	0104000020E8640000010000000101000000560E2DB2CEB21541A835CD2BB78755...	Pinus nigra	Schwarzkiefer
4	0104000020E86400000100000001010000004A0C02AB20C615418FE4F23BD5865541	Tilia cordata	Winterlinde
5	0104000020E86400000100000001010000008A1F632E1F531541D42B65AD2A8A5541	Corylus colurna	Baumhasel
6	0104000020E864000001000000010100000000000002F301541D122DB19AB885541	Prunus serrulata Kanzan	Japanische Nelken-Kirs
7	0104000020E864000001000000010100000036CD3B0EFF6F1541780B24F8E38A5541	Acer	Ahorn
8	0104000020E8640000010000000101000000DE718A8E4C7B1541C5FEB20394915541	Quercus robur	Sommer-Eiche, Stieleich
9	0104000020E86400000100000001010000001283C00A3320164160764F7E98805541	Corylus colurna	Baumhasel
10	0104000020E86400000100000001010000004CA60A067ACE15413A234ADC58555...	Crataegus	Crataegus, Dorn
11	0104000020E864000001000000010100000075F984C441C1641386744D5C68C5541	Aesculus hippocastanum	Roßkastanie
12	0104000020E8640000010000000101000000CF66D5E7A5031641D122DB71F1815541	Acer platanoides	Spitzahorn

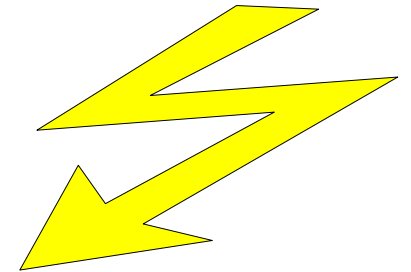
- public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (2)
 - koeln_baumkataster



QGIS

Ich konnte dies nicht reproduzieren.

Keine Erlaubnis für MCP erteilt auf Postgres zuzugreifen!

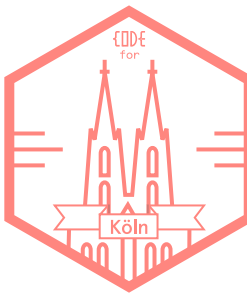


Es ist äußerste Vorsicht geboten:

Studie: Große KI-Modelle greifen unter "Stress" auf Erpressung zurück

<https://www.anthropic.com/research/agentive-misalignment>

<https://github.com/anthropic-experimental/agentive-misalignment>



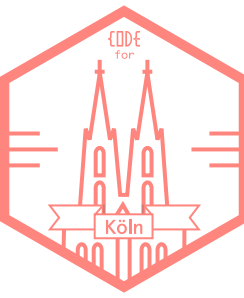
Postgres

- Unmaintained (!) - READONLY:

<https://github.com/modelcontextprotocol/servers-archived/tree/main/src/postgres>

```
"postgres": {  
  "command": "npx",  
  "args": [  
    "-y",  
    "@modelcontextprotocol/server-postgres",  
    "postgresql://postgres@localhost/mydb"  
  ]  
}
```

- Eine Änderung des Parameters mydb wurde nicht berücksichtigt
- Die DB muss dem Windows Benutzer gehören
- Neustart des Rechners notwendig



Postgres

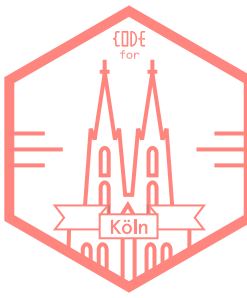
Z Kannst Du mir die Anzahl der Objekte in der Postgres Tabelle koeln_baumkataster selektieren deren Attribut ,höhe___m' ≤ 7 ist .

Ich führe die Abfrage mit dem "kleiner oder gleich" Operator aus:

P query

Ergebnis: Es gibt **22.279 Bäume** im Kölner Baumkataster mit einer Höhe von 7 Metern oder weniger ($\leq 7\text{m}$).

Das sind 6.133 Bäume mehr als bei der vorherigen Abfrage ($< 7\text{m}$), was bedeutet, dass es genau 6.133 Bäume mit einer Höhe von exakt 7 Metern gibt.



Z Wie weit ist die Entfernung zwischen baumnummer 66-803-S-0141 und 66-104-V-0046?

Ich werde die Positionen der beiden Bäume abfragen und dann die Entfernung zwischen ihnen berechnen:

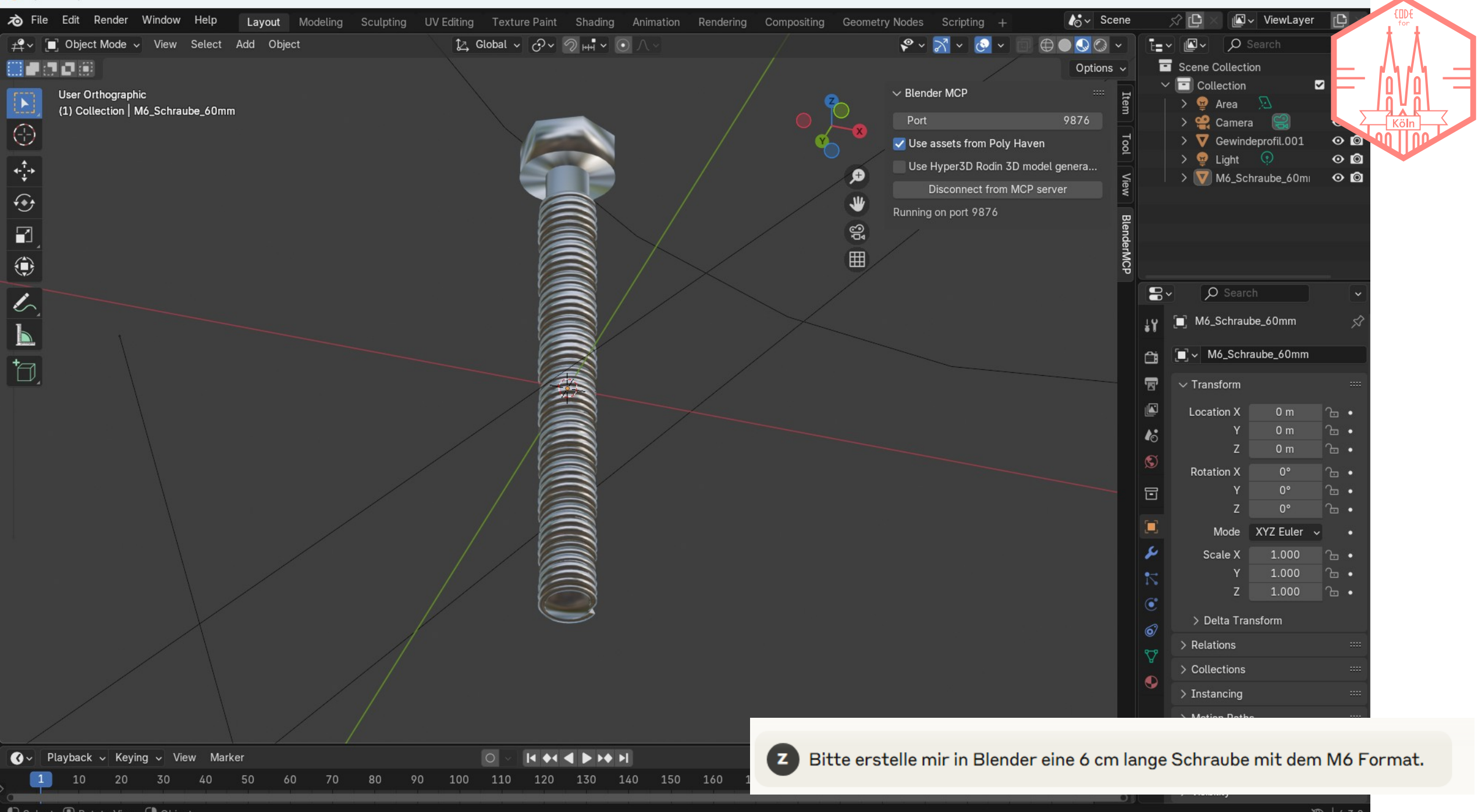
Jetzt berechne ich die Entfernung zwischen den beiden MultiPoint-Geometrien:

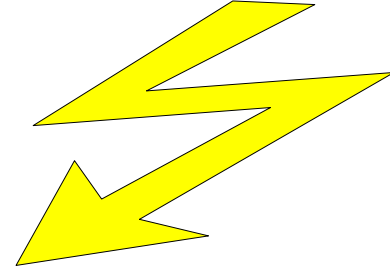
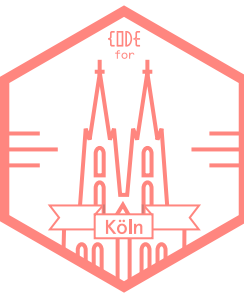
P query

```
ROUND(ST_Distance(  
    (SELECT geom FROM koeln_baumkataster WHERE baumnummer = '66-803-S-  
0141'),  
    (SELECT geom FROM koeln_baumkataster WHERE baumnummer = '66-104-V-0046')  
)::numeric, 2) as distanz_meter,  
ROUND(ST_Distance(  
    (SELECT geom FROM koeln_baumkataster WHERE baumnummer = '66-803-S-  
0141'),  
    (SELECT geom FROM koeln_baumkataster WHERE baumnummer = '66-104-V-0046')  
)::numeric / 1000, 3) as distanz_kilometer;
```

Ergebnis: Die Entfernung zwischen den Bäumen mit den Nummern **66-803-S-0141** und **66-104-V-0046** beträgt:

- **6.291,33 Meter** oder
- **6,291 Kilometer**





Model Context Protocol

<https://github.com/codeforcologne/Lab/pdf/mcp.pdf>