

Entwicklung von Plug-Ins für QGIS 3 – Eine Einführung

FOSSGIS, Bonn, 22 März 2018

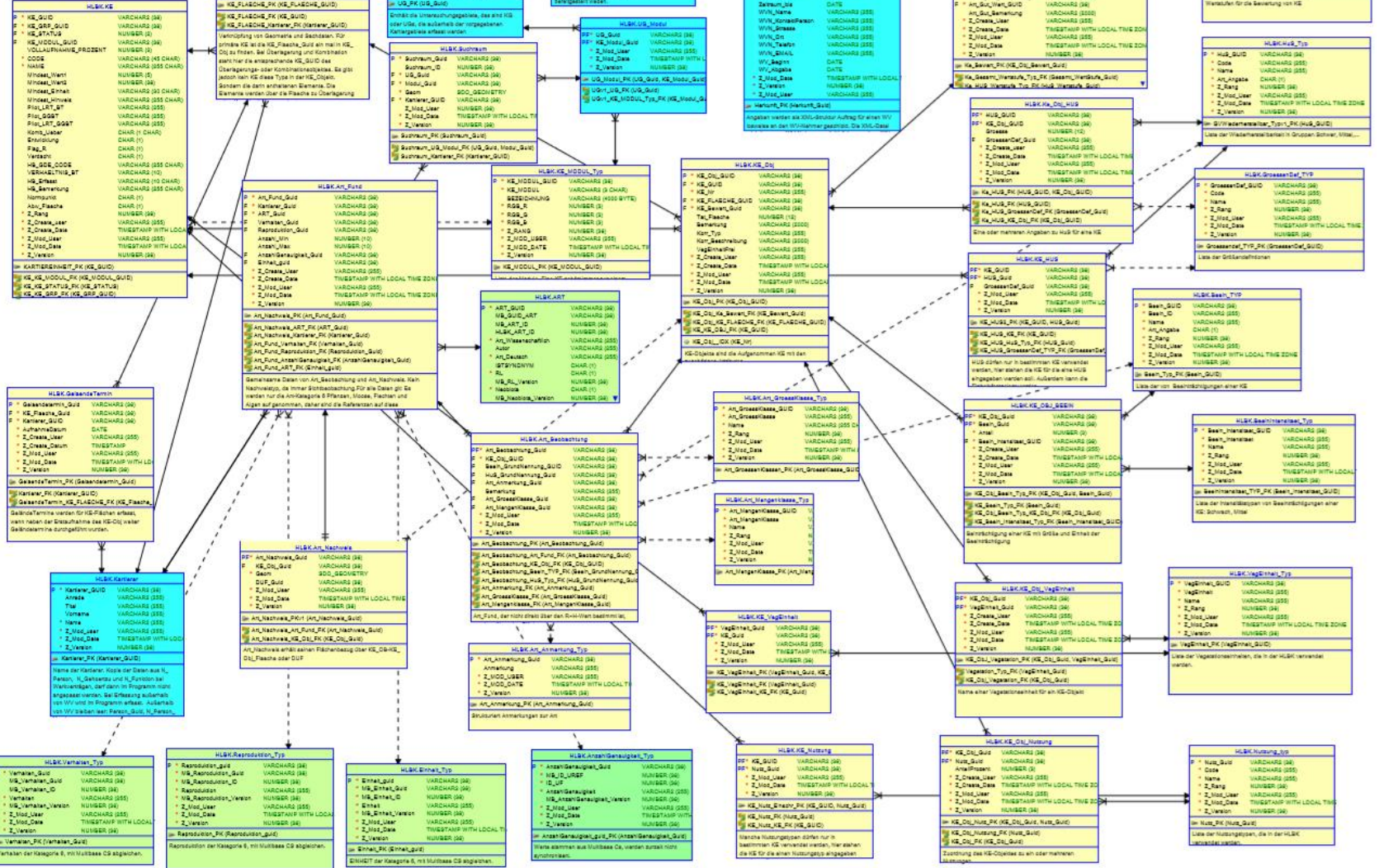
Peter Gipper

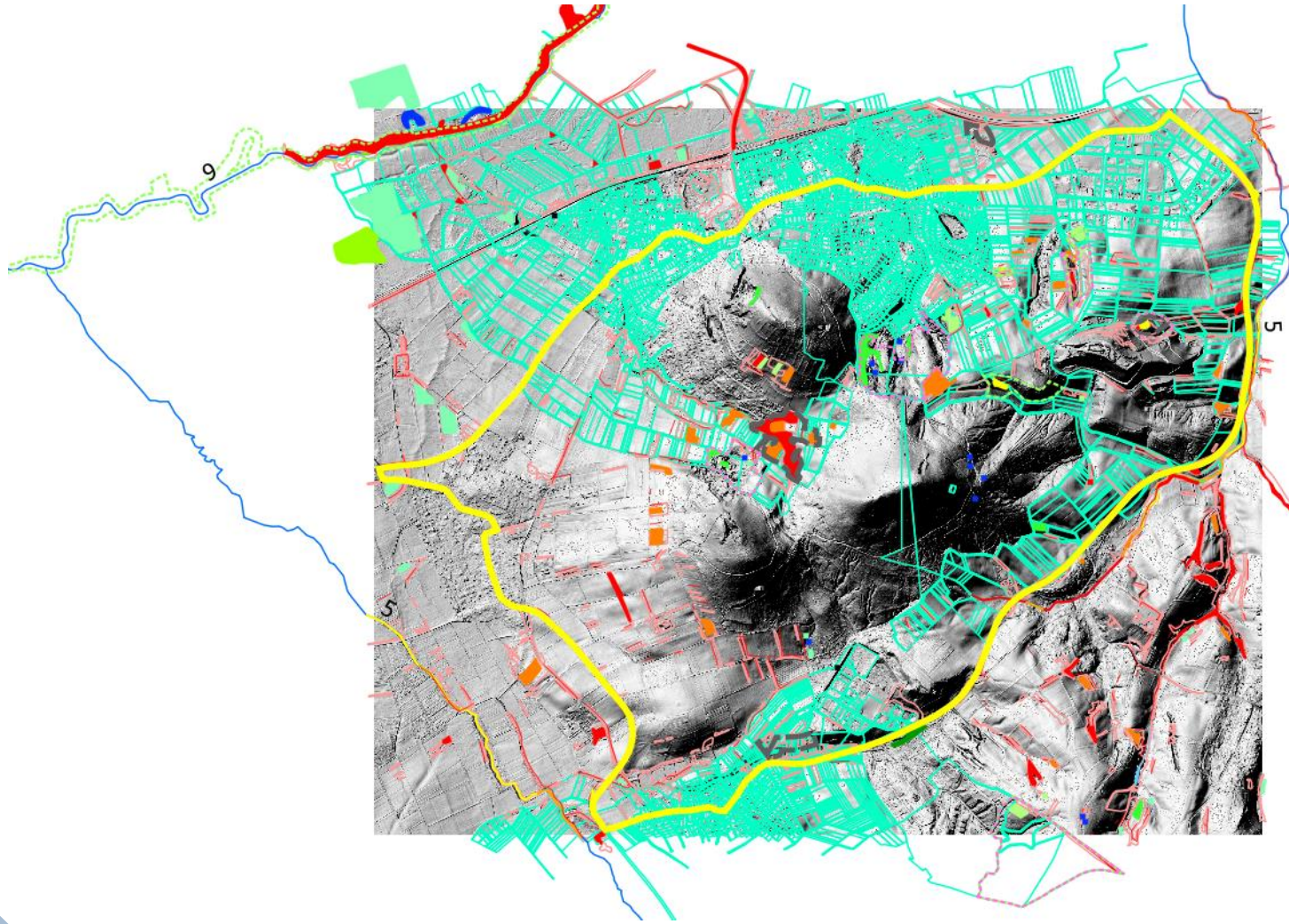
Motivation

- **Auftraggeber: HLNUG**
(Hessisches Landesamt für Naturschutz, Umwelt und Geologie)
- **Auftrag: HLBK Erfassung**
(Hessische Lebensraum- und Biotopkartierung)
- **Open Source Lösung**
QGIS Plugin + SpatiaLite Datenbank
- **Erfassung von Geo- und Sachdaten**



Diagram:	HLK-KE-Objekt
Author:	Arntjes
Created on:	2018-05-18 07:51:47 UTC
Modified on:	2018-05-27 12:59:27 UTC
Timestamp:	Arntjes
Category:	HLK
Model:	800
Stand:	Siehe Datenbankschema





QGIS 3?



- mehrmaliges Verschieben des Release Datums
- QGIS 2.99 wöchentliche nightly builds
- QGIS 3 seit 23. Februar 2018

QGIS Plugins



QGIS Plugins



QGIS 2 → QGIS 3




Qt 4 → Qt 5






PyQt 4 → PyQt 5

QGIS Plugins

📁 > Peter > .qgis2 > python > plugins > MyPlugin




 ▼ 


Name	Änderungsdatum	Typ	Größe
 <code>__init__.py</code>	18.03.2018 20:43	JetBrains PyCharm C...	1 KB
 <code>__init__.pyc</code>	19.03.2018 11:42	PYC-Datei	2 KB
 <code>metadata.txt</code>	11.03.2018 16:29	Textdokument	1 KB




<https://docs.qgis.org/2.14/pdf/en/QGIS-2.14-PyQGISDeveloperCookbook-en.pdf>

QGIS Plugins

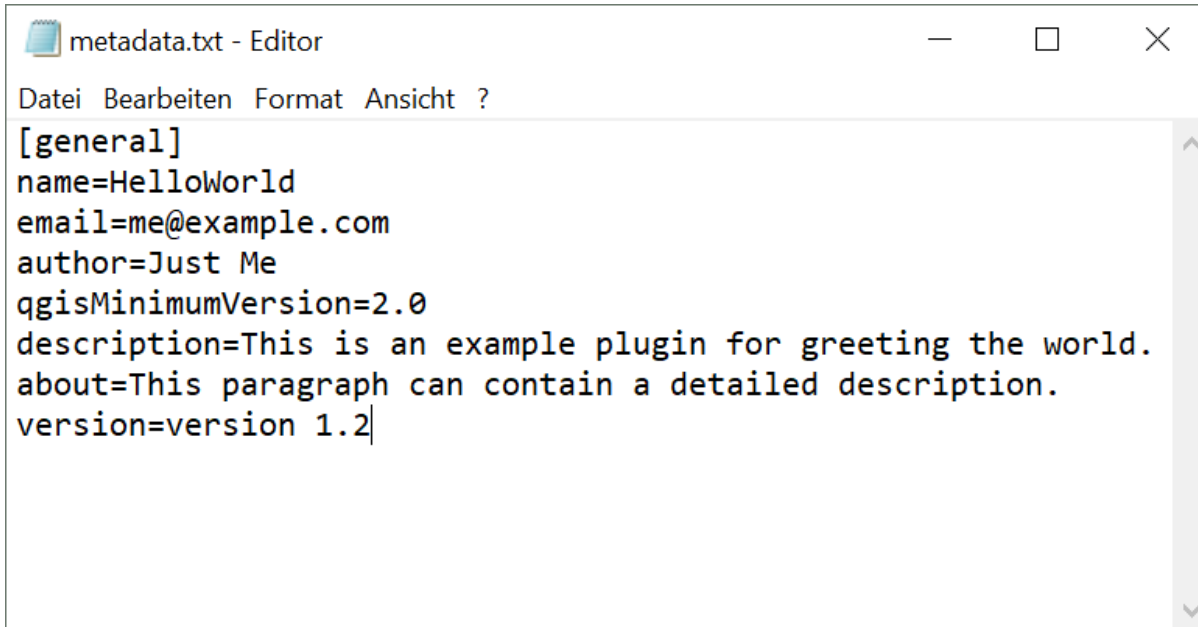
 > Peter > .qgis2 > python > plugins > MyPlugin v ↻

Name	Änderungsdatum	Typ	Größe
 __init__.py	18.03.2018 20:43	JetBrains PyCharm C...	1 KB
 __init__.pyc	19.03.2018 11:42	PYC-Datei	2 KB
 metadata.txt	11.03.2018 16:29	Textdokument	1 KB

 > Peter > AppData > Roaming > QGIS > QGIS3 > profiles > default > python > plugins > MyPlugin v ↻

Name	Änderungsdatum	Typ	Größe
 __pycache__	13.03.2018 18:06	Dateiordner	
 __init__.py	11.03.2018 17:40	JetBrains PyCharm C...	1 KB
 metadata.txt	11.03.2018 17:29	Textdokument	1 KB

QGIS Plugins



```

Datei Bearbeiten Format Ansicht ?
[general]
name=HelloWorld
email=me@example.com
author=Just Me
qgisMinimumVersion=2.0
description=This is an example plugin for greeting the world.
about=This paragraph can contain a detailed description.
version=version 1.2
    
```

QGIS Plugins

```

metadata.txt - Editor
Datei Bearbeiten Format Ansicht ?
[general]
name=HelloWorld
email=me@example.com
author=Just Me
qgisMinimumVersion=2.0
description=This is an example plugin for greeting the world.
about=This paragraph can contain a detailed description.
version=version 1.2
    
```

```

metadata.txt - Editor
Datei Bearbeiten Format Ansicht ?
[[general]
name=HelloWorld
email=me@example.com
author=Just Me
qgisMinimumVersion=3.0
description=This is an example plugin for greeting the world.
about=This paragraph can contain a detailed description.
version=version 1.2
    
```

QGIS Plugins

metadata.txt - Editor

Datei Bearbeiten Format Ansicht ?

```
[general]
name=HelloWorld
email=me@justme.com
author=Just Me
qgisMinimumVersion=1.0
description=This is an example plugin for greeting the world.
about=This paragraph can contain a detailed description.
version=1.2
```

Erweiterungen | Installiert (22)

Suchen

- Geometrieprüfung
- GPS Werkzeuge
- Heatmap
- HelloWorld
- Interpolationserweiterung
- Koordinaten aufnehmen
- MetaSearch Catalog
- Offline-Bearbeitung
- Oracle-Spatial-Geography
- Processing
- Rasterneuländeanalyse

HelloWorld

This is an example plugin for greeting the world.

This paragraph can contain a detailed description.

Autor: [Just Me](#)

Installierte Version 1.2 (in C:\Users\Peter\.qgis2\python\plugins\MyPlugin)

Alle aktualisieren Gewählte Erweiterung deinstallieren Erweiterung erneut installieren

Schließen Hilfe

QGIS Plugins

```

__init__.py x
1  from PyQt4.QtGui import QAction, QIcon
2  from PyQt4.QtCore import QObject, SIGNAL
3
4  def classFactory(iface):
5      return TestPlugin(iface)
6
7  class TestPlugin:
8
9      def __init__(self, iface):
10         self.iface = iface
11
12         def initGui(self):
13             self.action = QAction(QIcon(""), "Test plugin", self.iface.mainWindow())
14             QObject.connect(self.action, SIGNAL("triggered()"), self.run)
15             self.iface.addPluginToMenu("&Test plugins", self.action)
16
17         def unload(self):
18             self.iface.removePluginMenu("&Test plugins", self.action)
19
20         def run(self):
21             print "TestPlugin: run called!"
22

```

QGIS Plugins

```
21 | print "TestPlugin: run called!"
```

SyntaxError: Missing parentheses in call to 'print'

```
21 | print("TestPlugin: run called!")
```

QGIS Plugins

```

1  from PyQt4.QtGui import QAction, QIcon
2  from PyQt4.QtCore import QObject, SIGNAL

```

**ModuleNotFoundError: No module named 'PyQt4',
 ImportError: Cannot import name 'QAction',
 ImportError: Cannot import name ' SIGNAL'**

```

1  from PyQt5.QtGui import QIcon
2  from PyQt5.QtWidgets import QAction

```


QGIS Plugins

```
14 | QObject.connect(self.action, SIGNAL("triggered()"), self.run)
```

ImportError: Cannot import name ' SIGNAL '

```
14 | self.action.triggered.connect(self.run)
```

QGIS Plugins

```

1  from PyQt5.QtGui import QIcon
2  from PyQt5.QtWidgets import QAction
3
4  def classFactory(iface):
5      return TestPlugin(iface)
6
7  class TestPlugin:
8
9      def __init__(self, iface):
10         self.iface = iface
11
12         def initGui(self):
13             self.action = QAction(QIcon(""), "Test plugin", self.iface.mainWindow())
14             self.action.triggered.connect(self.run)
15             self.iface.addPluginToMenu("&Test plugins", self.action)
16
17         def unload(self):
18             self.iface.removePluginMenu("&Test plugins", self.action)
19
20         def run(self):
21             print("TestPlugin: run called!")

```

QGIS API - PyQGIS

QGIS API - Abhängigkeiten

- **Python Version**
 - Python Libraries?

- **Qt / PyQt Versions**

QGIS API - Module

- **qgis.networkanalysis** verschoben nach **qgis.analysis**

-

QGIS 2

```
>>> len(dir(qgis.core))
499
```

QGIS 3

```
>>> len(dir(qgis.core))
728
```

QGIS API – Klassen

- umbenannt
- verschoben
- entfernt

QGIS API – Klassen

- Beispiele

- **QgsMapLayerRegistry** entfernt
-> Funktionalität in **QgsProject**
- **QGis** Klasse umbenannt
-> **Qgis**
- **if QgsFeature.geometry():**
-> **if QgsFeature.hasGeometry():**
-> **if QgsFeature.geometry().isNull():**

QGIS API – Methoden

- Veränderte Funktionalitäten
- entfernt
- umbenannt
 - Akronyme in Methodennamen:

setDestinationCRS()	setDestinationCrs()
readXML()	readXml()
layerID()	layerId()
...	

QGIS API Break

Moved Classes

Moved classes

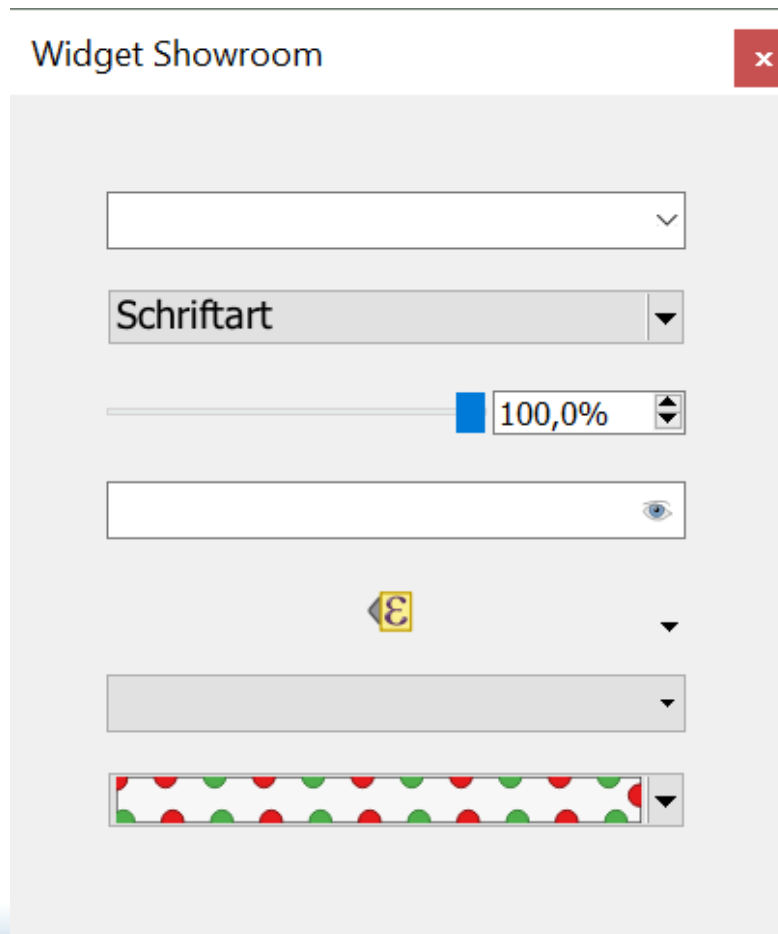
class	Module 2.X	Module 3.x
QgsMapLayerModel	gui	core
QgsMapLayerProxyModel	gui	core

Renamed Classes

Renamed classes

API 2.x	API 3.X
QgisGui	QgsGuiUtils
QgsAbstractGeometryV2	QgsAbstractGeometry
QgsArcProperter	QgsStrategy
QgsAttributeAction	QgsActionManager
QgsCategorizedSymbolRendererV2	QgsCategorizedSymbolRenderer
QgsCategorizedSymbolRendererV2Model	QgsCategorizedSymbolRendererModel
QgsCategorizedSymbolRendererV2ViewStyle	QgsCategorizedSymbolRendererViewStyle
QgsCategorizedSymbolRendererV2Widget	QgsCategorizedSymbolRendererWidget
QgsCentroidFillSymbolLayerV2	QgsCentroidFillSymbolLayer
QgsCentroidFillSymbolLayerV2Widget	QgsCentroidFillSymbolLayerWidget
QgsCircularStringV2	QgsCircularString
QgsColorButtonV2	QgsColorButton

QGIS API – Neue Widgets



QgsCheckableComboBox()

QgsFontButton()

QgsOpacityWidget()

QgsPasswordLineEdit()

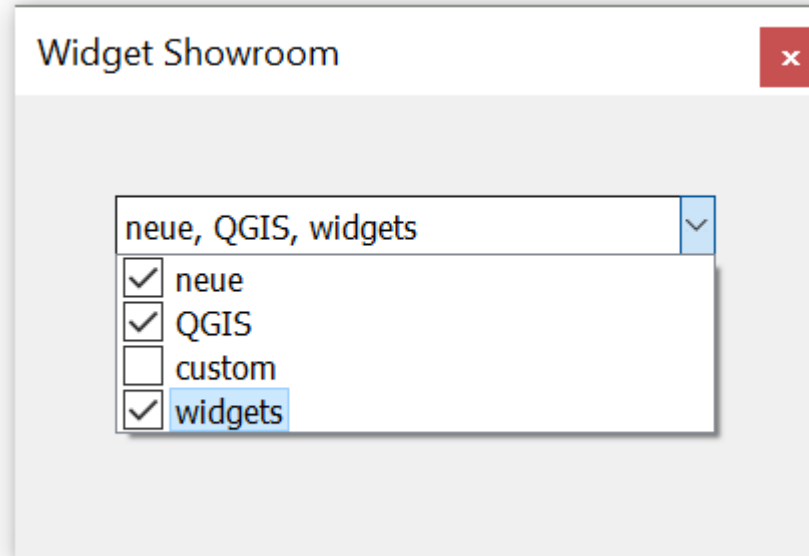
QgsPropertyOverrideButton()

QgsRasterBandComboBox()

QgsSymbolButton()

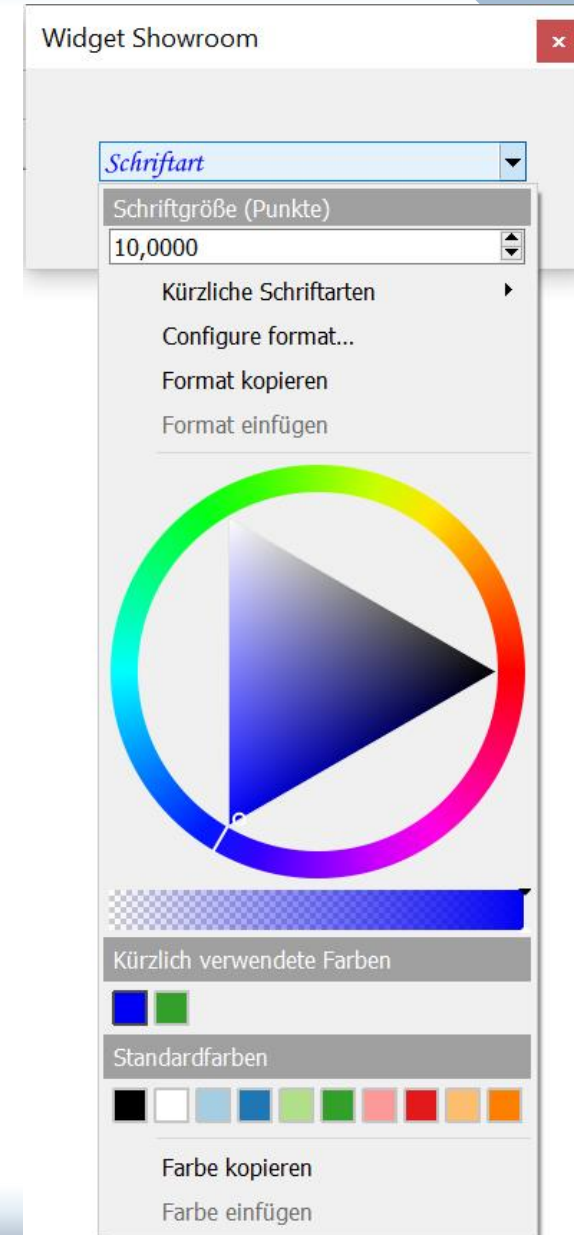
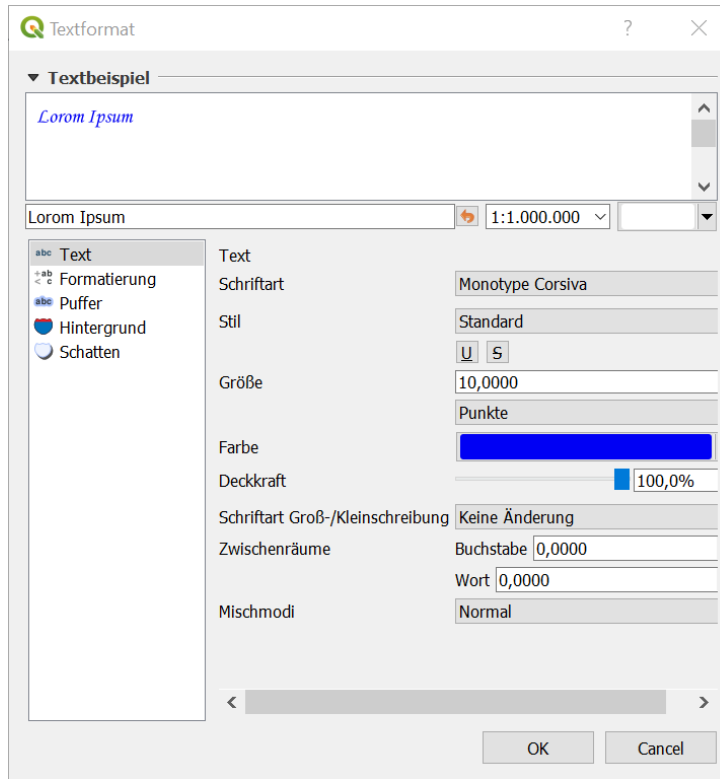
QGIS API – Neue Widgets

- `QgsCheckableComboBox()`



QGIS API – Neue Widgets

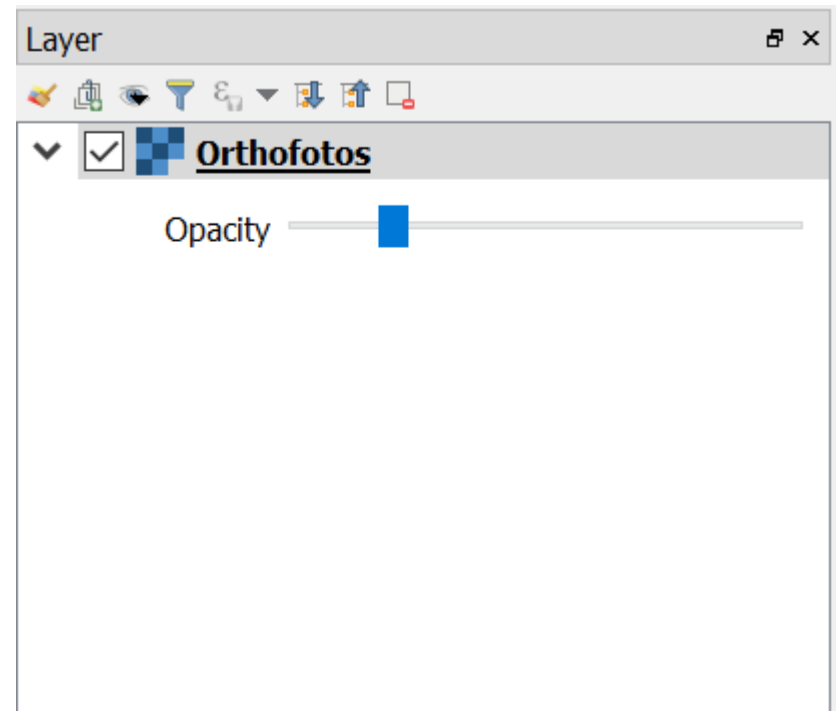
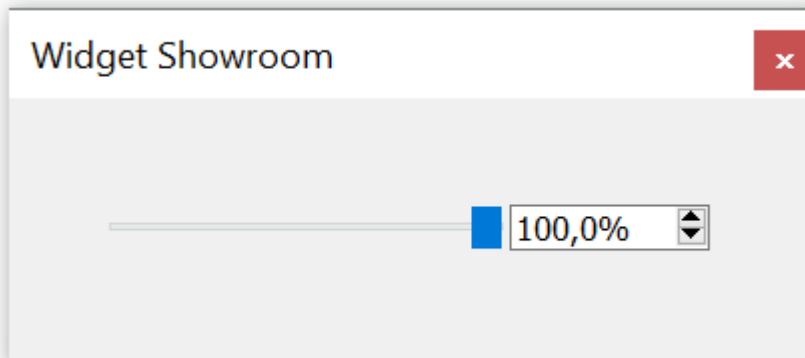
- **QgsFontButton()**



QGIS API – Neue Widgets

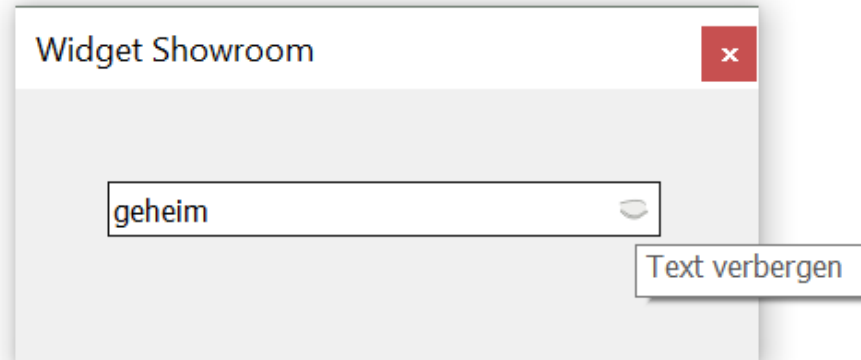
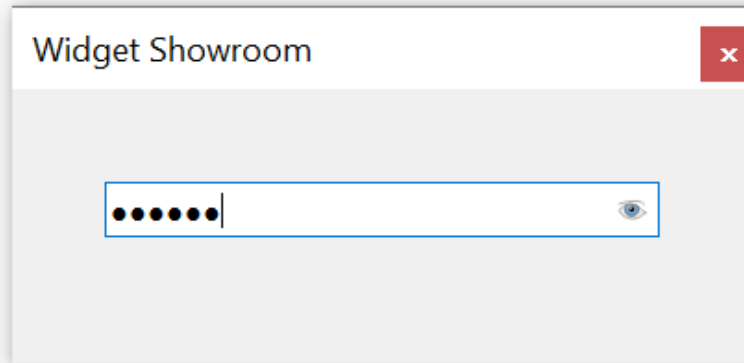
- `QgsOpacityWidget()`

„Deckkraftschieber“



QGIS API – Neue Widgets

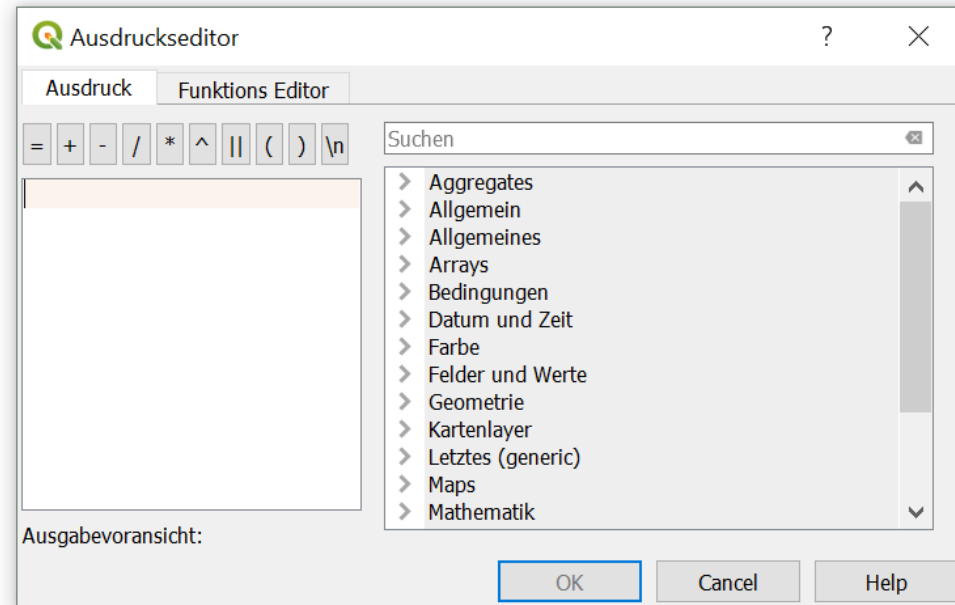
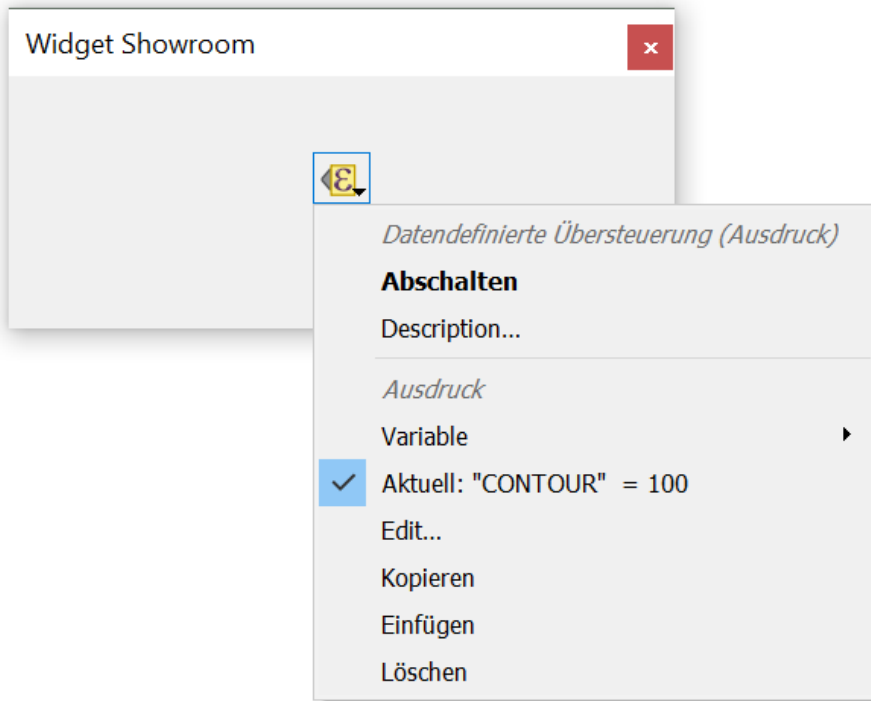
- `QgsPasswordLineEdit()`



QGIS API – Neue Widgets

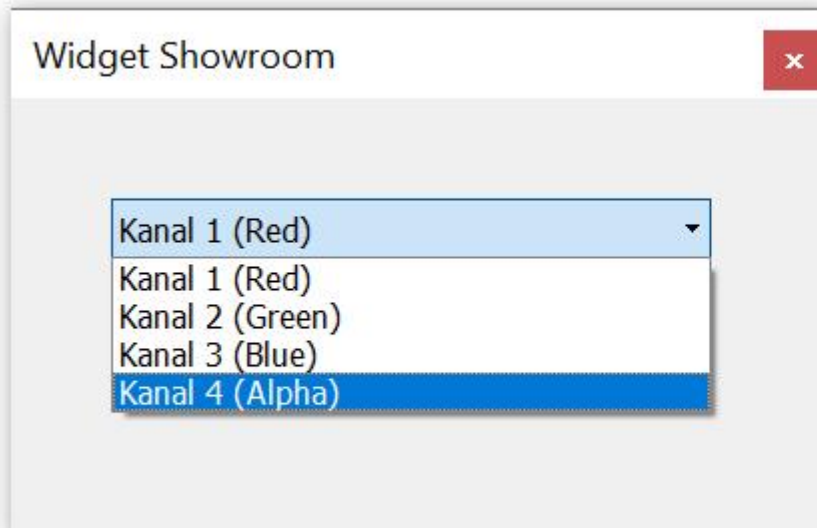
- `QgsPropertyOverrideButton()`

Ersetzt `QgsDataDefinedButton()`



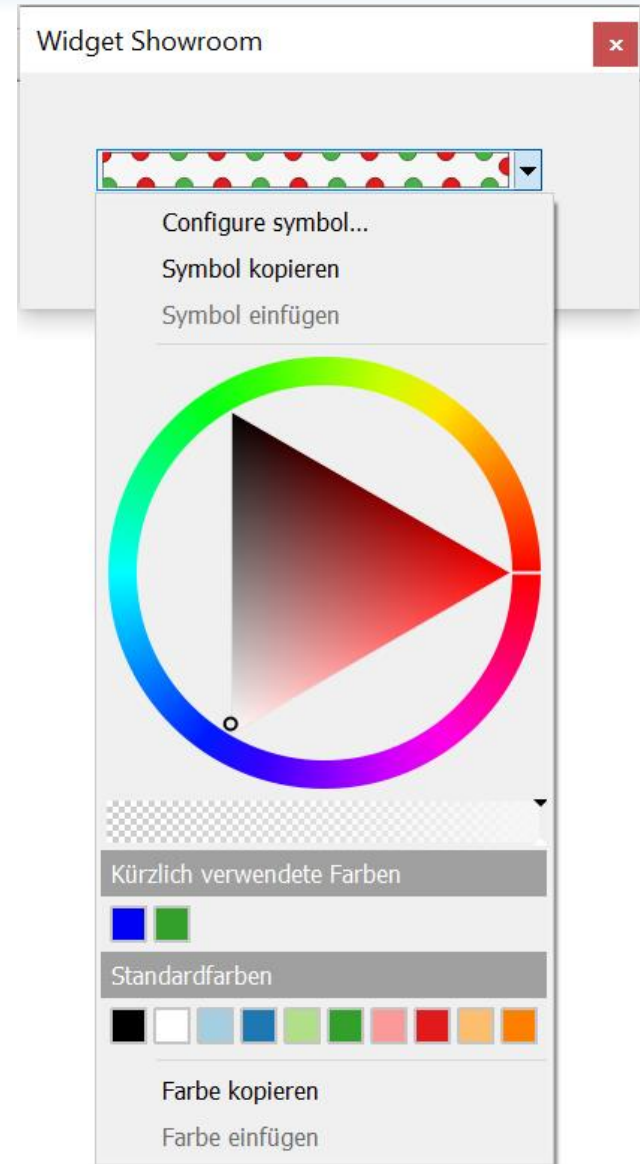
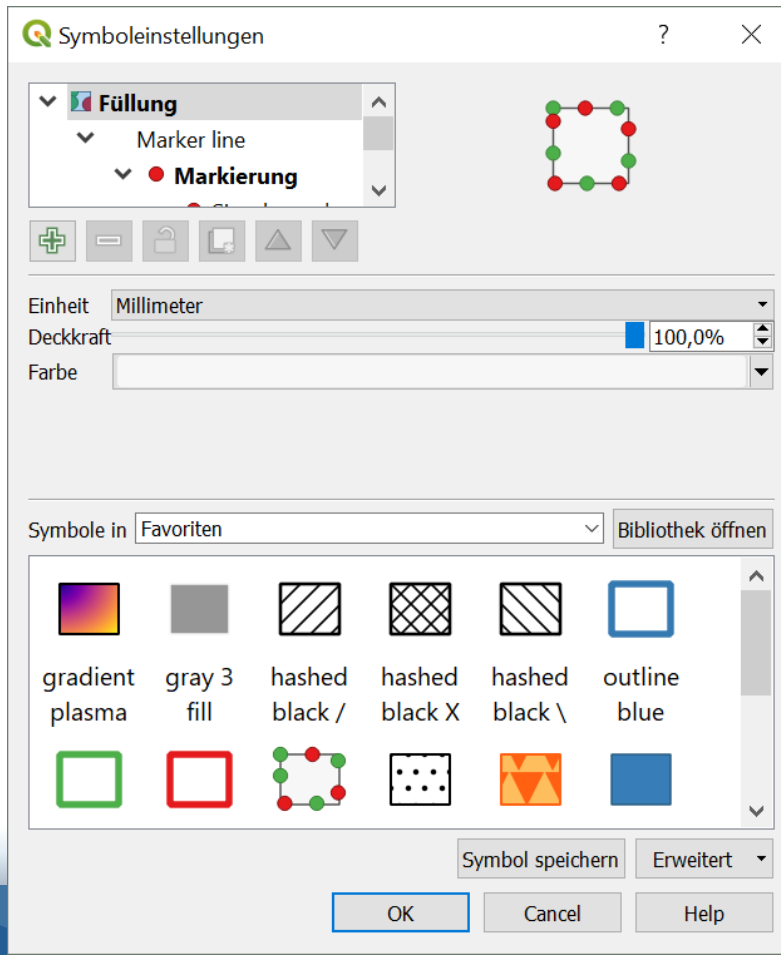
QGIS API – Neue Widgets

- `QgsRasterBandComboBox()`



QGIS API – Neue Widgets

- **QgsSymbolButton()**



Python

Python - keywords

```
>>> from keyword import kwlist
```

Python 2

```
>>> kwlist
['and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del',
'elif', 'else', 'except', 'exec', 'finally', 'for', 'from', 'global',
'if', 'import', 'in', 'is', 'lambda', 'not', 'or', 'pass', 'print',
'raise', 'return', 'try', 'while', 'with', 'yield']
```

Python 3

```
>>> kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', '
continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for',
'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal',
'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yi
eld']
```

Python - Division

- Python 3: int / int = float

Python 2

```
>>> 5.0/2.0
2.5
>>> 5/2
2
>>> 5%2
1
```

Python 3

```
>>> 5/2
2.5
>>> 5//2
2
>>> 5%2
1
```

Python - Geschwindigkeit

- Python 3 generell langsamer

```

1  from timeit import timeit
2
3  - def test_while():
4      i = 0
5      - while i < 20000:
6          i += 1
7      return
8
9  print(timeit(test_while, number=10000))

```

Python 2

7.37285685158

Python 3

12.966768361744471

Python - for-Loop Variablen und global namespace

```

1 i = 1
2 print('before: i =', i)
3
4 print('comprehension: ', [i for i in range(5)])
5
6 print('after: i =', i)

```

Python 2

```

('before: i =', 1)
('comprehension: ', [0, 1, 2, 3, 4])
('after: i =', 4)

```

Python 3

```

before: i = 1
comprehension: [0, 1, 2, 3, 4]
after: i = 1

```

Python – strings

```
>>> name = 'fossgis'
```

Python 2

```
>>> print 'hello {var}'.format(var=name)
hello fossgis
```

Python 3

```
>>> print(f'hello {name}')
hello fossgis
```

Python – weitere Änderungen

Python 2

- `raw_input()`
- `xrange()`
- `raise IOError, "file error"`
- `dict.iteritems()`
- `dict.items()`

Python 3

- `input()`
- `range()`
- `raise IOError("file error")`
- `dict.items()`
- `list(dict.items())`

Qt / PyQt

Qt / PyQt

- **Module reorganisiert**
 - Module entfernt (QtScript)
 - Module aufgeteilt in Submodule (QtGui, QtWebKit)

- **Neue Module**
 - QtBluetooth, QtPositioning

- **PyQt5 unterstützt nur new-style signal und slots handling. SIGNAL() und SLOT() werden nicht mehr unterstützt.**

- **PyQt5 unterstützt keine Elemente der Qt API die in Qt v5.0 als *deprecated* oder *obsolete* gekennzeichnet sind.**

Nützliche Ressourcen – Plugins für Plugins



- Python console
- Plugin Reloader
- Plugin Builder 3
- First Aid

Nützliche Ressourcen – Weblinks

- Plugin migration
 - <https://github.com/qgis/QGIS/wiki/Plugin-migration-to-QGIS-3>
 - <https://gisforthought.com/updating-a-plugin-from-qgis-2-to-qgis-3/>

- QGIS
 - <https://qgis.org/api/>
 - https://qgis.org/api/api_break.html

- PyQGIS
 - <https://python.qgis.org/api/>

- Qt
 - <https://doc.qt.io/qt-5/>

- PyQt
 - <http://pyqt.sourceforge.net/Docs/PyQt5/>
 - http://pyqt.sourceforge.net/Docs/PyQt5/pyqt4_differences.html

- Python
 - <https://docs.python.org/3/>
 - <https://docs.python.org/3/howto/pyporting.html>

```
print('Thank you!')
```

peter.gipper@geosysnet.de



 <http://geosysnet.com>

 @geoSYSnet




 <http://qgis.pro>



 <https://gis-trainer.com>

 @gis_trainer

 gis_trainer

 gistrainer

Impressum

gis-trainer.de / geoSYS
Pflügerstr. 56
12047 Berlin, Germany

Telefon: 030/82070659
Telefax: 030/82070658

Email: info@gis-trainer.de / info@geosysnet.de
Website: www.gis-trainer.de / www.geosysnet.de