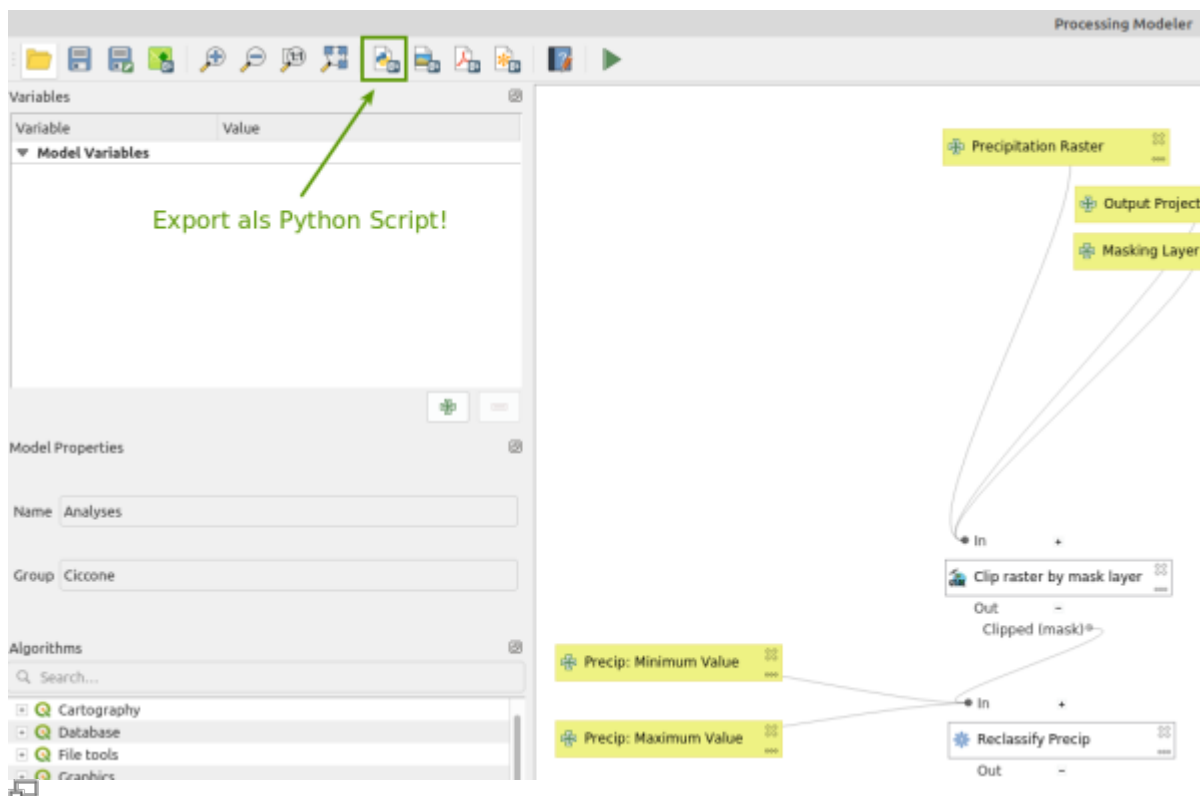


QGIS3: Modellierung, PyQGIS und Plugin-Entwicklung*

QGIS bietet - wie die meisten OpenSource Programme - viele Möglichkeiten der Erweiterung. Findet man für eine Fragestellung oder ein Problem kein passendes Plugin, so kann man sich auf verschiedenen Wegen selbst daran machen, eigene Skripte und Erweiterungen zu schreiben. Für Nicht-Skripter gibt es die Möglichkeit, visuell mit Hilfe der **Prozessmodellierung** ([grafische-modellierung-processing-modeler](#)) eigene Skripte oder Erweiterungen zu entwickeln. Bevor jedoch eine Erweiterung auch wirklich als Erweiterung allen anderen zur Verfügung steht, müssen strikte Vorgaben beachtet werden...

Entwicklung von Arbeitsabläufen mit dem Graphical Modeler



In der Lektion [grafische-modellierung-processing-modeler](#) wird dargestellt, wie man mit der Graphischen Modellierung ein einfaches Modell zum Abflussregime erstellt. Dieses steht über die Processing Toolbox zur Verfügung oder kann als *.model3-Datei abgespeichert und ausgetauscht werden. Hiermit lassen sich bereits manche Arbeitsabläufe erheblich erleichtern und optimieren.



Seit QGIS 3.10 ist es wieder möglich, ein im Graphical Modeler erzeugtes Modell **in einen sauberen Python Code zu überführen!** Diese eröffnet viele Möglichkeiten: So kann man den Code *hübsch nachbearbeiten* oder mit Hilfe des Plugin Builder zu einem Plugin weiterentwickeln!

PyQGIS Scripting

Schon seit QGIS Version 0.9 baut QGIS auf Python. Die meisten Plugins sind in Python programmiert und über QGIS lassen sich Python-Skripte direkt über die Konsole schreiben, ausführen oder testen (Erweiterungen → Python Konsole)(Abb. 1).

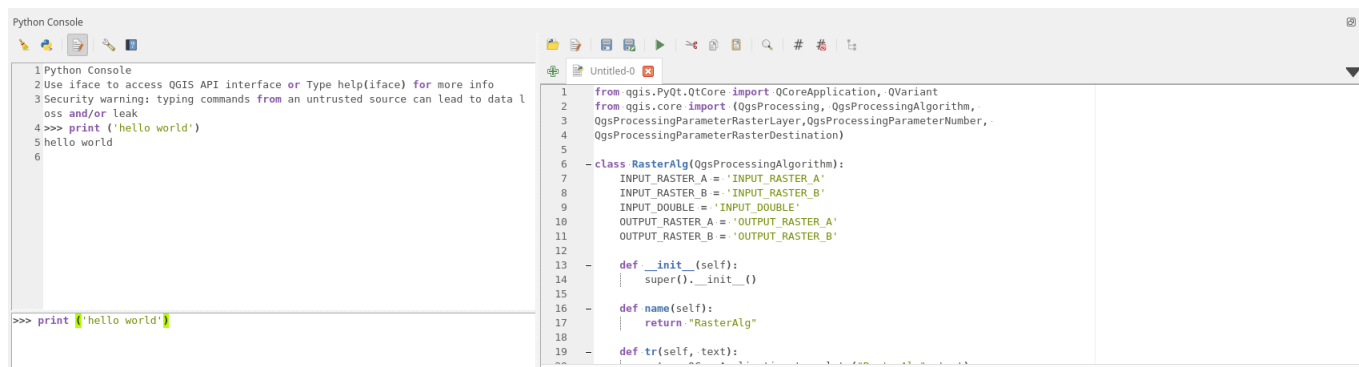


Abb. 1: Die Python Konsole von QGIS3.4

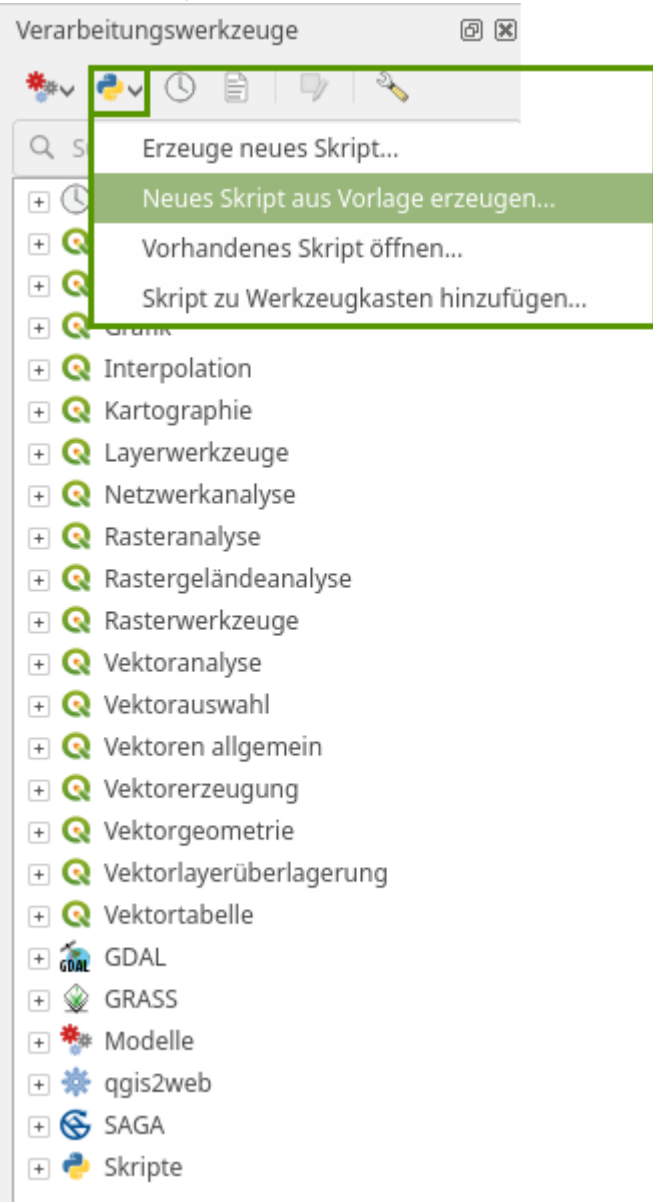


Abb. 2: Python Script aus Vorlage erzeugen

Bei der Umstellung von QGIS 2.x auf QGIS 3.x wurde der Sprung von Python2 zu Python3 gewagt. Das bedeutet leider, dass QGIS2 Skripte unter QGIS3 mehr lauffähig sind! Möchte man in QGIS3 ein lauffähiges Modul programmieren, welches die Processing-UI verwendet, so muss man sich zudem an neue Vorgaben halten. Die Umstellung ist groß, der Vorteil daraus aber ebenso - QGIS ist damit bereit für die Zukunft! Literatur, e-Books und ausführliche Tutorials gibt es nur wenige, QGIS3 ist noch zu jung, aber:

Eine gute Einführung in das „neue“ QGIS PyQGIS Scripting erhält man von Anita Graser: [PythonQGIS 101](#)

Die QGIS Python API-Dokumentation findet man hier: [QGIS PyAPI](#)

Ab QGIS 3.4 erhält man ein Template beim Erstellen eines neuen Skripts: Verarbeitung → Verarbeitungswerkzeuge (Abb. 2)

Plugin-Entwicklung

Hat man einen Code erstellt - ob mit dem Graphical Modeler oder händisch - lässt sich der generierte Script jederzeit aus der Python Console ausführen oder in die Toolbox integrieren. Möchte man es aber allen QGIS-Nutzern über das (oder ein eigenes) Repository verfügbar machen, so kann man sich eines Plugins namens „**Plugin Builder 3**“ bedienen. Es hilft einem, die Vorgaben für QGIS-Plugins zu erfüllen und generiert am Ende einen „sauberen“ Code.



Häufig stellt sich die Frage, was bei der Veröffentlichung eines selbst geschriebenen Plugins **lizenzmäßig** beachtet werden muss. Kurz: **Ein für QGIS geschriebenes Plugin muss die gleichen lizenzmäßigen Bedingungen erfüllen, wie QGIS selbst**, nämlich [GPL](#)! Mehr dazu:
<http://blog.qgis.org/2016/05/29/licensing-requirements-for-qgis-plugins/>

From:
<https://lernplattform.map-site.de/> - Lernplattform für OpenSource GIS

Permanent link:
https://lernplattform.map-site.de/doku.php/qgis/advanced/d_spezialthemen/qgis-python-scripting?rev=1639553295

Last update: **2022/09/09 23:44**

