Prepared By: Er Santosh Panth

GIS

Mount Annapurna Campus (BSCCSIT 8th Sem)

5.3 GIS programming and customization: Opening and exploring Model Builder, Python script tools, Customizing QGIS with Python

- Customization is the process of adapting a generic system to an individual specification.
- It is generally considered one of the most expensive non-personnel components of a GIS implementation. Because of the limited size and diversity of the GIS market, many GIS software developers have adopted the approach of developing a generic suite of multi-purpose software routines, together with some type of customization programming capability.
- This has allowed core GIS software developers to concentrate effort on engineering robust and reliable generic routines.
- The task of creating specific-purpose, end-user (or vertical application) customizations is usually seen as the domain of application developers.
- In the case of desktop and professional level GIS the process of customization typically involves modification of a standard graphical user interface and extension of the 'out of the box' tools by writing application programs.
- More sophisticated users may be allowed access to the underlying core GIS capabilities and database. They may be able to extend the core class libraries or reuse objects within their own programs

Opening and exploring Model Builder

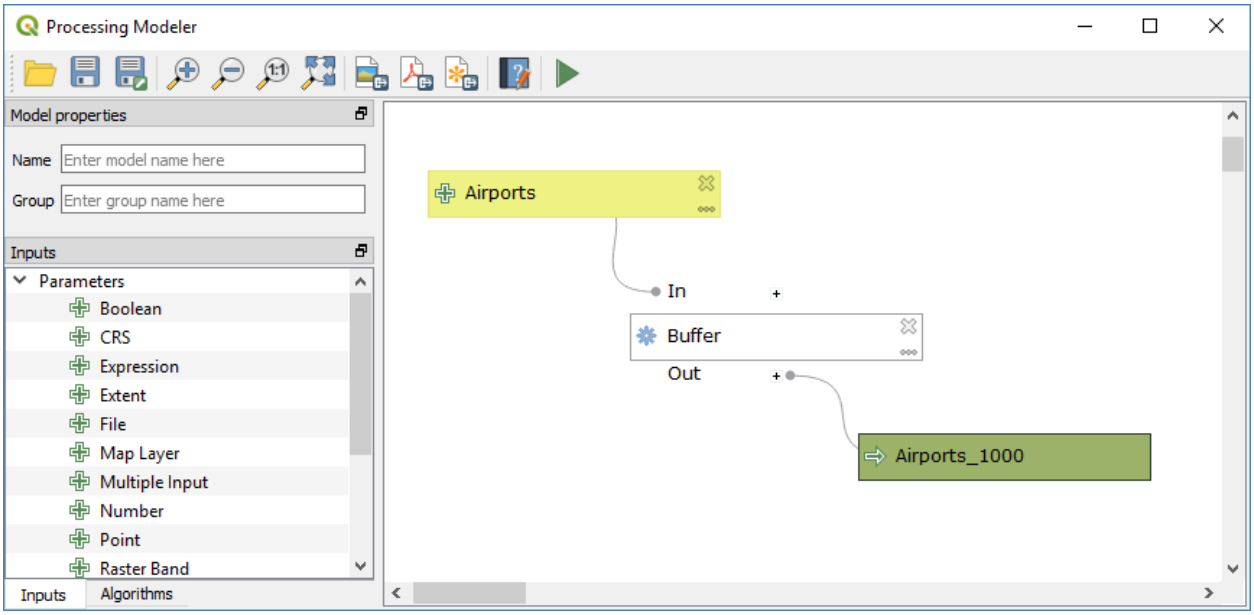
- The *graphical modeler* allows you to create complex models using a simple and easy-to-use interface. When working with a GIS, most analysis operations are not isolated, rather part of a chain of operations.
- Using the graphical modeler, that chain of operations can be wrapped into a single process, making it convenient to execute later with a different set of inputs.
- No matter how many steps and different algorithms it involves, a model is executed as a single algorithm, saving time and effort.

The graphical modeler can be opened from the Processing menu (Processing ▸ Graphical Modeler).

Creating a model involves two steps:

1. **Definition of necessary inputs**. These inputs will be added to the parameters window, so the user can set their values when executing the model. The model itself is an algorithm, so the parameters window is generated automatically as it happens with all the algorithms available in the processing framework.
2. **Definition of the workflow.** Using the input data of the model, the workflow is defined by adding algorithms and selecting how they use the defined inputs or the outputs generated by other algorithms in the model.

   All we have to do is drop inputs, outputs, and algorithms to the interface. Then, we connect them together sequentially in the order you want to run.



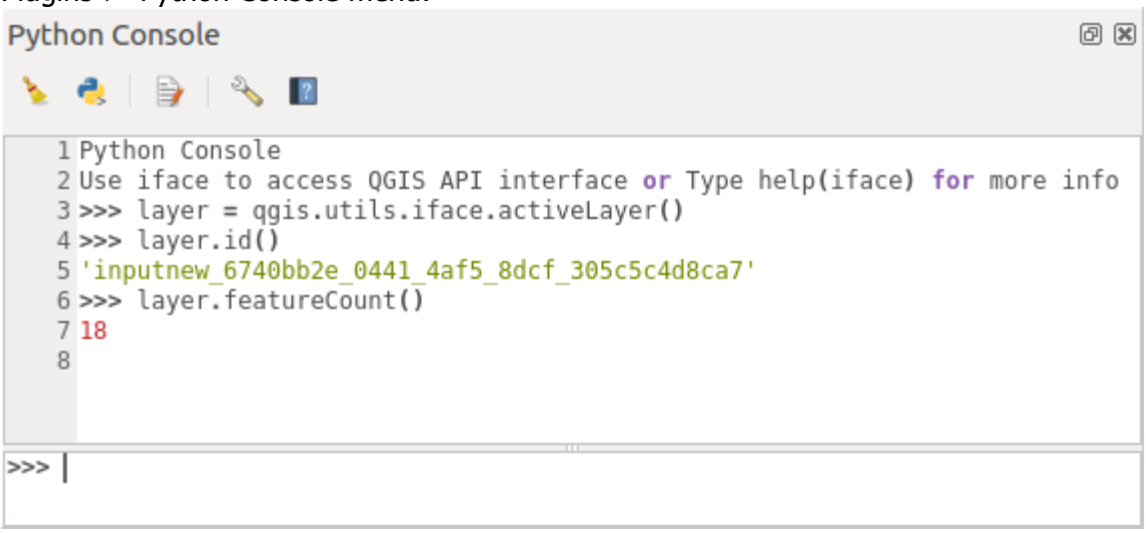
Python script tools

Python support was first introduced in QGIS 0.9. There are several ways to use Python in QGIS Desktop. Issue commands in the Python console within QGIS

- Create and use plugins
- Automatically run Python code when QGIS starts
- Create processing algorithms
- Create functions for expressions in QGIS
- Create custom applications based on the QGIS API

Scripting in the Python Console

- QGIS provides an integrated Python console for scripting. It can be opened from the Plugins ► Python Console menu:



- Fig. 1.4 QGIS Python console

Python Plugins

The functionality of QGIS can be extended using plugins. Plugins can be written in Python. The main advantage over C++ plugins is simplicity of distribution (no compiling for each platform) and easier development.

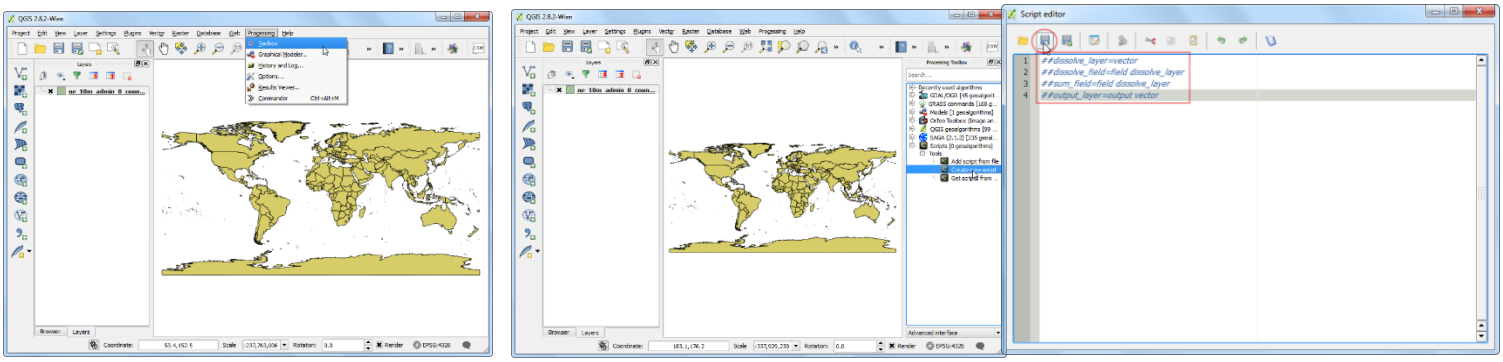
Running Python code when QGIS starts

There are two distinct methods to run Python code every time QGIS starts.

1. Creating a startup.py script
2. Setting the PYQGIS_STARTUP environment variable to an existing Python file

Writing Python Scripts

Expand the Scripts group in the Processing Toolbox and select Create a new script.



Example

https://www.qgistutorials.com/en/docs/getting_started_with_pyqgis.html

Customizing QGIS with Python

Example : Add A New Menu Item

```
import webbrowser

def open_website():
    webbrowser.open('https://gis.stackexchange.com')

website_action = QAction('This is Test Menu')
website_action.triggered.connect(open_website)
iface.helpMenu().addSeparator()
iface.helpMenu().addAction(website_action)
```

For Detail:https://courses.spatialthoughts.com/pyqgis-in-a-day.html#graphical-user-interface-gui-programming-basics