

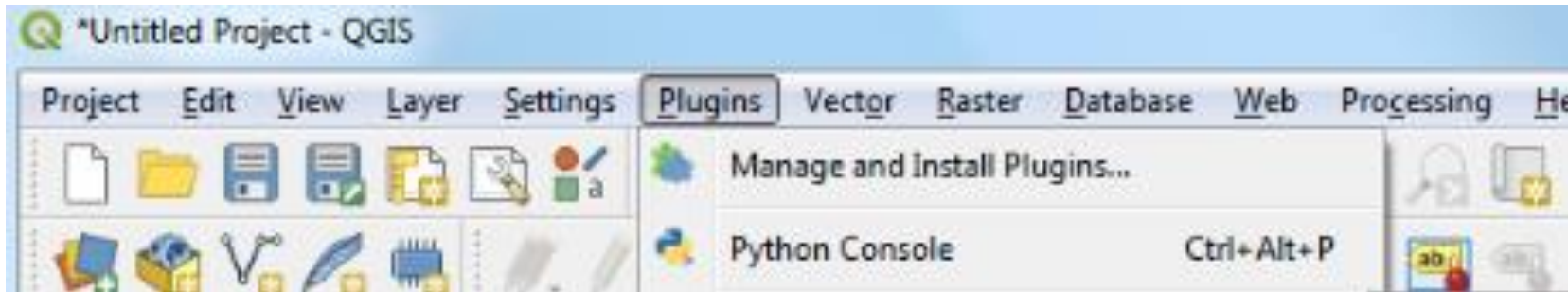
Creating QGIS Plugin

QGIS Plugin Creation

- Required Software :
 - QGIS - <https://www.qgis.org/en/site/forusers/download.html>
 - Qt Creator - <https://doc.qt.io/qt-5/index.html>
 - Optional – Qt 5 Designer
 - Python IDE

Helpful Plugins in QGIS

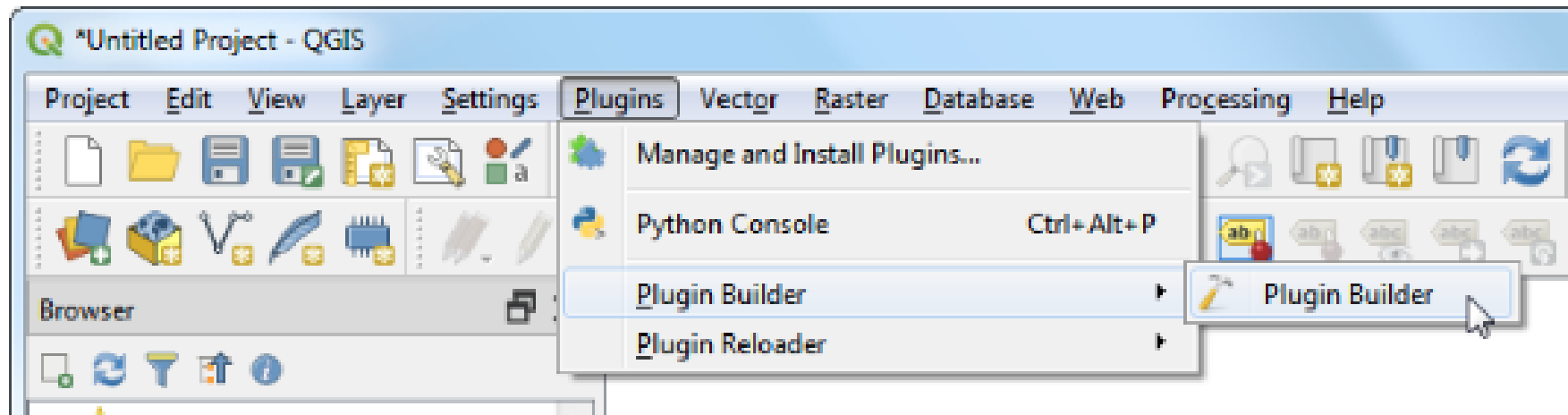
- Step 1 : Install QGIS and open the application
- Step 2 : Go to Plugins -> Manage and Install Plugins...



- Step 3 : Click on "All". Search for Plugin Builder and Plugin Reloader.
- Step 4 : Install the two plugins stated above.

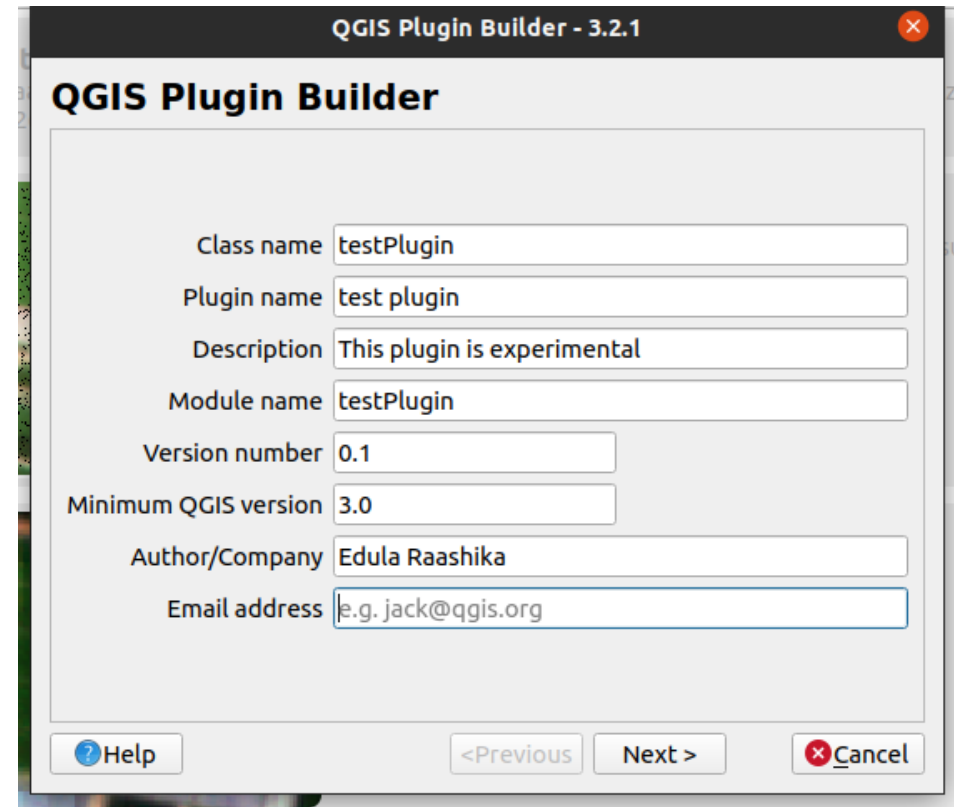
Procedure

- Open QGIS. Go to Plugins -> Plugin Builder -> Plugin Builder



Procedure

- You will see the QGIS Plugin Builder dialog with a form. You can fill the form with details relating to our plugin. The Class name will be the name of the Python Class containing the logic of the plugin. This will also be the name of the folder containing all the plugin files.



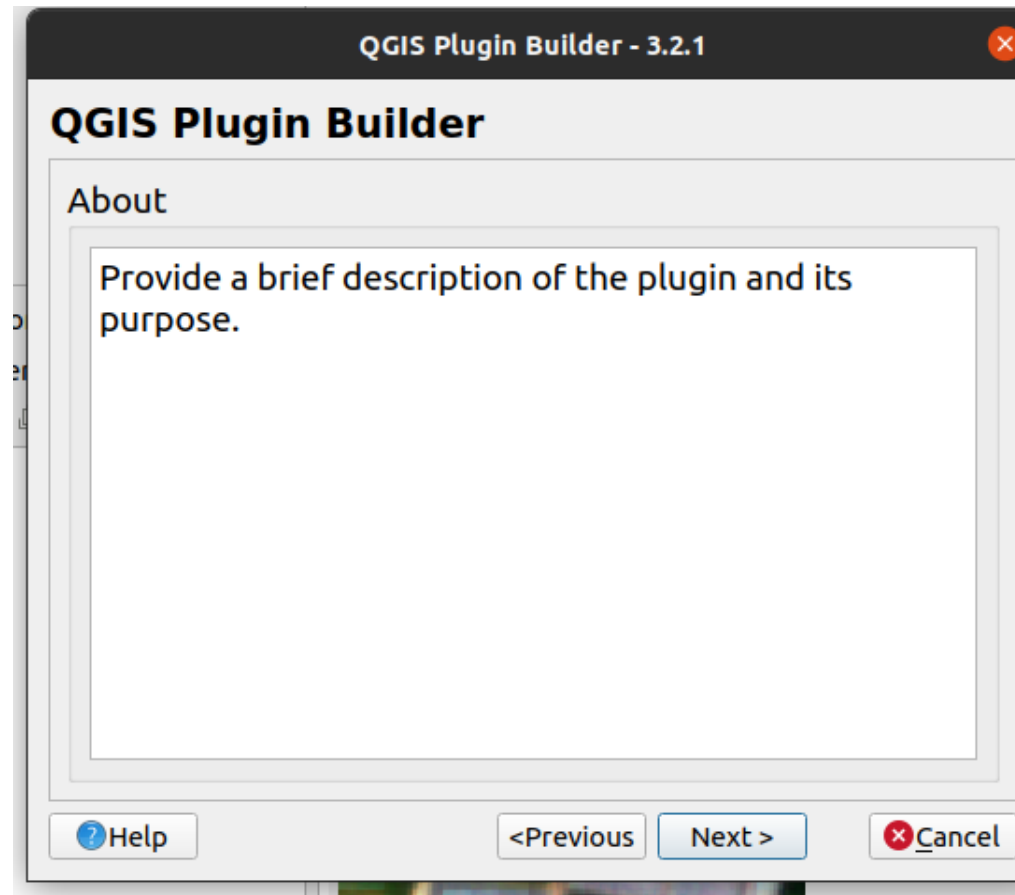
The screenshot shows the 'QGIS Plugin Builder - 3.2.1' dialog box. It contains a form with the following fields and values:

Field	Value
Class name	testPlugin
Plugin name	test plugin
Description	This plugin is experimental
Module name	testPlugin
Version number	0.1
Minimum QGIS version	3.0
Author/Company	Edula Raashika
Email address	e.g. jack@qgis.org

At the bottom of the dialog, there are three buttons: a 'Help' button with a question mark icon, a '<Previous' button, and a 'Next >' button. A 'Cancel' button with a red 'X' icon is also present in the bottom right corner.

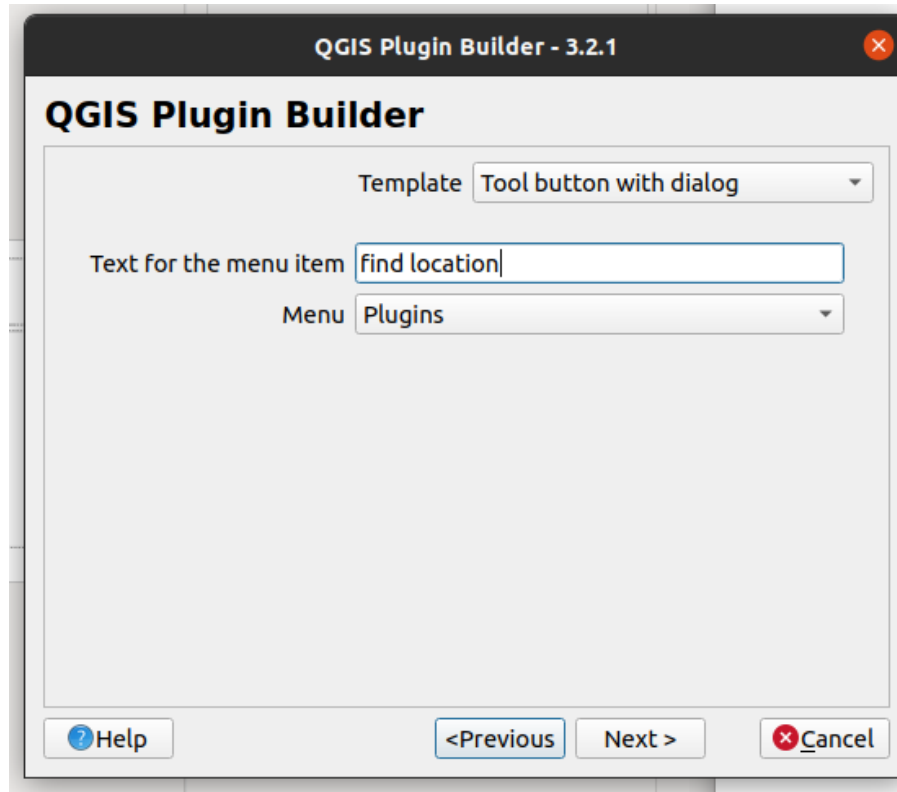
Procedure

- Provide a brief description about the plugin if necessary.



Procedure

- Select the Tool button with dialog from the Template Selector. The Text for menu item value will be how the users will find your plugin in QGIS menu.



Procedure

- Plugin builder will prompt you for the type of files to generate. Keep the default selection and click Next.



- If you intend to publish the plugin, you may enter the details of repository and home page values. Check the Flag the plugin as experimental box at the bottom and click Next.

QGIS Plugin Builder - 3.2.1

QGIS Plugin Builder

Publication (mandatory Items)

Bug tracker

Repository

Publication (recommended Items)

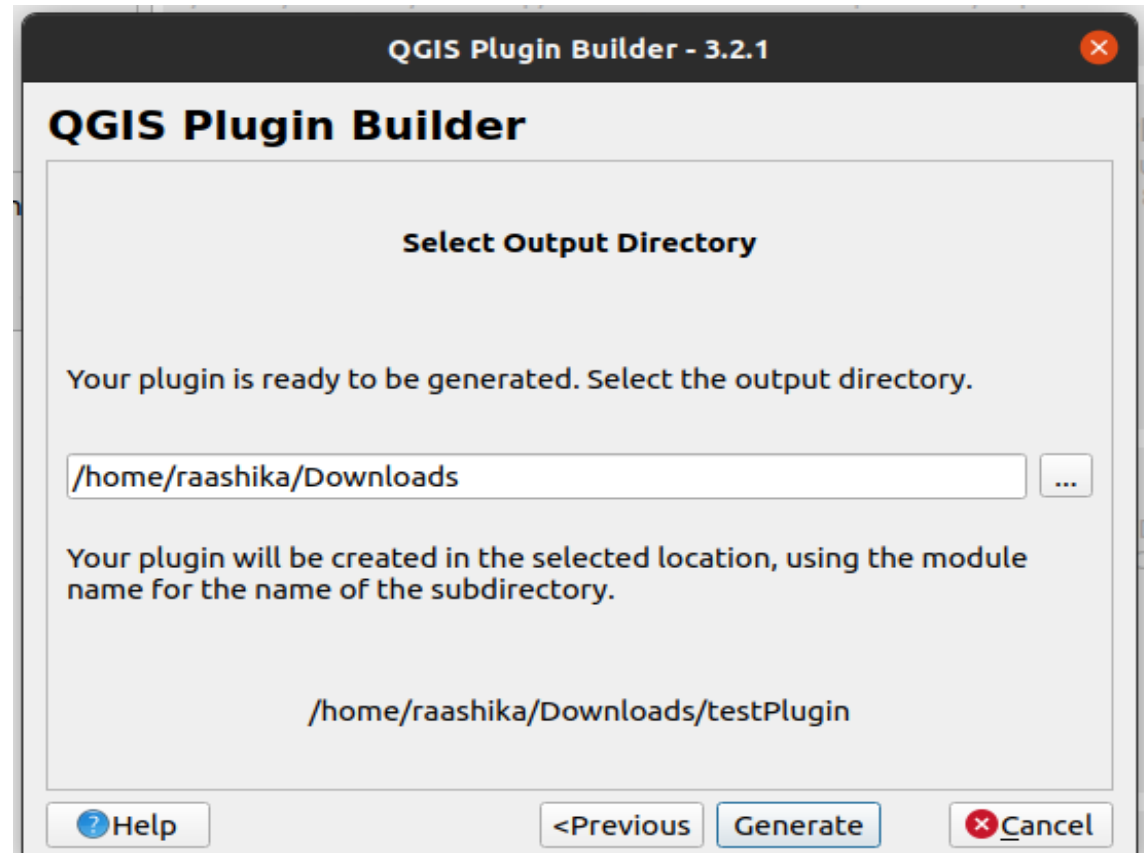
Home page

Tags ...

☒ Flag the plugin as experimental

? Help <Previous Next > Cancel

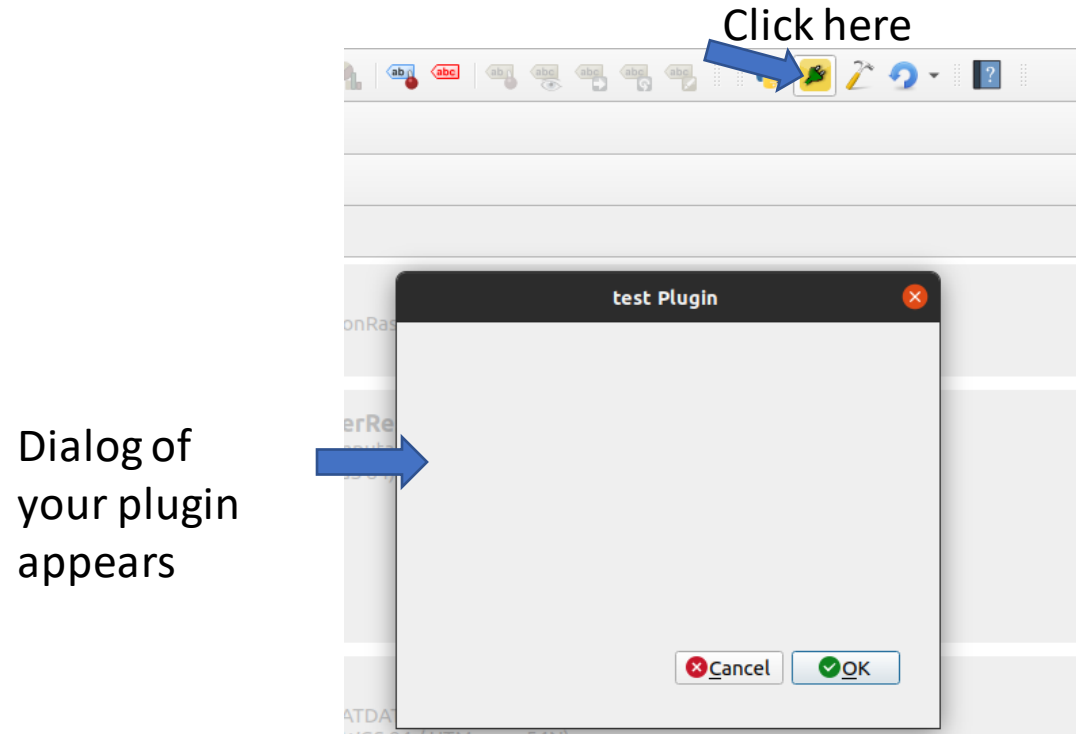
- Choose a directory for downloading your plugin and click on Generate.



- Click ok in the next dialog and you can find your plugin in the selected directory

Refer "InstallationFromZipFile" manual on github to install the plugin downloaded earlier in QGIS. ([github link for manual](#)).

You can see your plugin as below.



Designing QGIS Plugin

Locating Plugin Folder

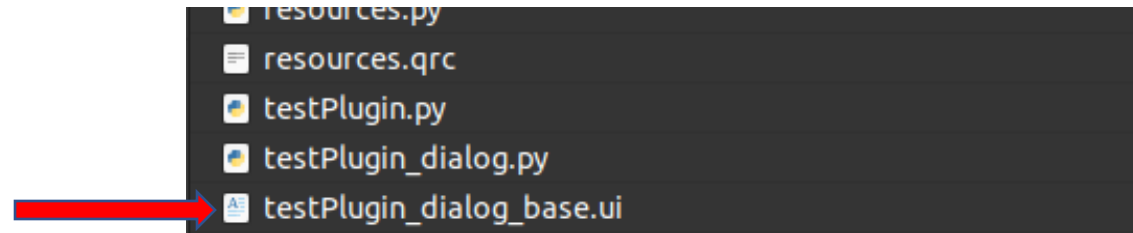
- Plugins in QGIS are stored in a special folder. In QGIS, locate your current profile folder by going to Settings -> User Profiles -> Open Active Profile Folder.



- Go to python -> plugins. You will find folders of all installed plugins including your plugin here. Open your plugin folder.

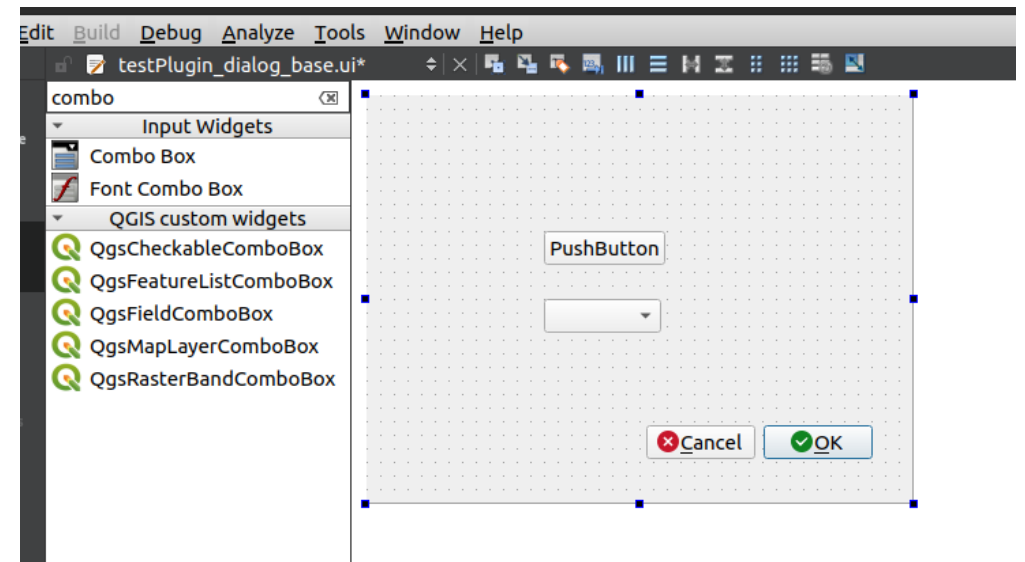
Designing and Editing Plugin dialog box

- Open .ui file with Qt Creator .



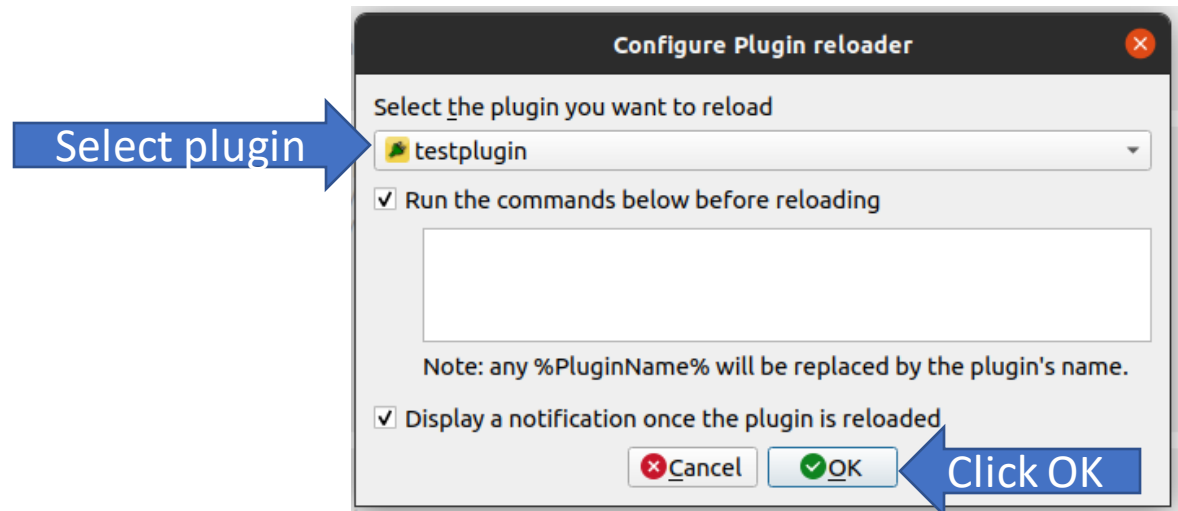
- Edit your dialog box with available options.
- Refer Qt creator documentation and Tutorials for more information.

[Qt Creator Tutorial Link](#)



Designing QGIS Plugin

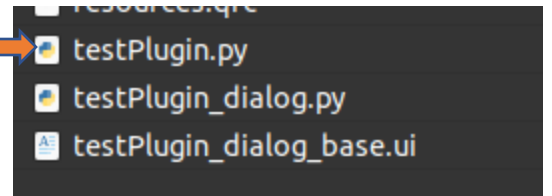
- After completing design of your dialog box. Save the .ui file in Qt Creator. (Click Ctrl + S or select File -> and save the file).
- To reflect the changes made in QGIS.
 - click on Reload Plugin – select configure – select your plugin



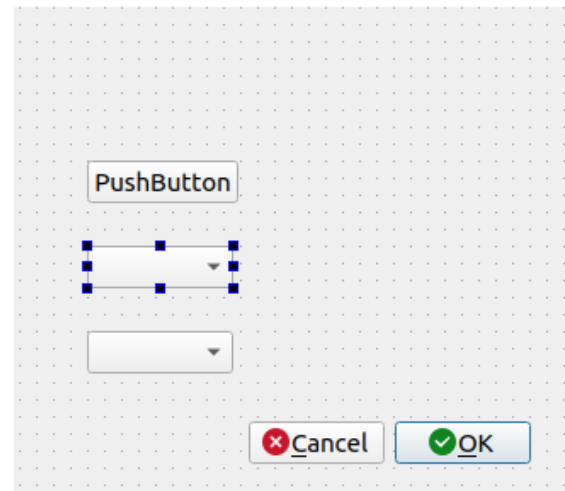
QGIS Plugin Development

Plugin Development

- Open your plugin.py file from Settings -> User Profiles -> Open Active Profile Folder -> python -> plugins -> your plugin.



- Open .ui file in Qt Creator. Check the name of the label in the object window. Edit your python file to perform action when certain button is clicked or action to do.



Object	Class
testPluginDialogBase	QDialog
button_box	QDialogButtonBox
comboBox	QComboBox
comboBox_2	QComboBox
pushButton	QPushButton

- Add your python code to perform actions in run function.

```
def run(self):  
    """Run method that performs all the real work"""  
  
    # Create the dialog with elements (after translation) and keep reference  
    # Only create GUI ONCE in callback, so that it will only load when the  
    if self.first_start == True:  
        self.first_start = False  
        self.dlg = testPluginDialog()  
  
    # show the dialog  
    self.dlg.show()  
    # Run the dialog event loop  
    result = self.dlg.exec_()  
    # See if OK was pressed  
    if result:  
        # Do something useful here - delete the line containing pass and  
        # substitute with your code.  
        pass
```

- Save the python file after editing and reload plugin using "Plugin Reloader" to reflect the changes in QGIS.
- Refer available Plugins for help.

