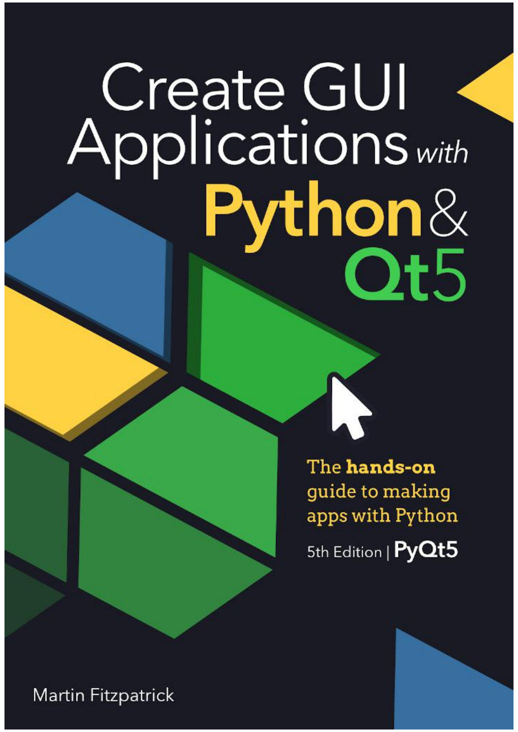
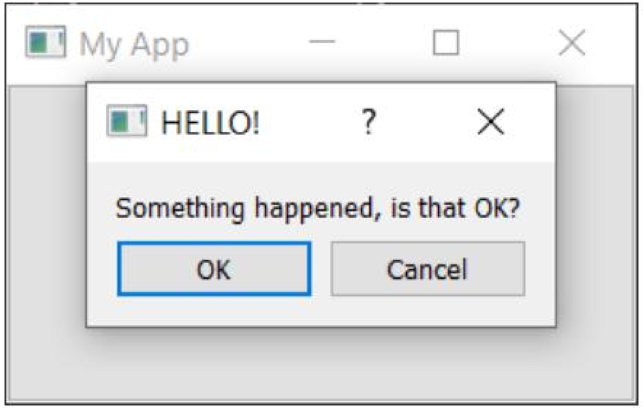
pyQt

8. Dialogs

Dialogs are useful GUI components that allow you to *communicate* with the user (hence the name dialog). They are commonly used for file Open/Save, settings, preferences, or for functions that do not fit into the main UI of the application. They are small modal (or *blocking*) windows that sit in front of the main application until they are dismissed. Qt actually provides a number of 'special' dialogs for the most common use-cases, allowing you to provide a platform-native experience for a better user experience.

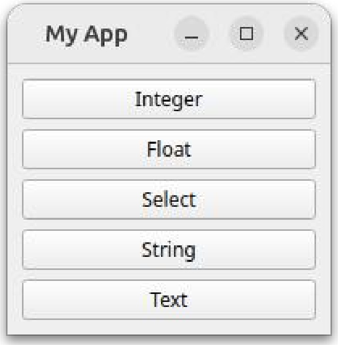


Create GUI Applications with Python & Qt5
The hands-on guide to making apps with Python
5th Edition | PyQt5
Martin Fitzpatrick

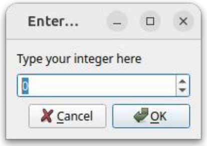


Something happened, is that OK?

```python
import sys
from PyQt5.QtWidgets import (
    QApplication,
    QInputDialog,
    QLineEdit,
    QMainWindow,
    QPushButton,
    QVBoxLayout,
    QWidget, )

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("My App")

        layout = QVBoxLayout()

        button1 = QPushButton("Integer")
        button1.clicked.connect(self.get_an_int)
        layout.addWidget(button1)

        button2 = QPushButton("Float")
        button2.clicked.connect(self.get_a_float)
        layout.addWidget(button2)

        button3 = QPushButton("Select")
        button3.clicked.connect(self.get_a_str_from_a_list)
        layout.addWidget(button3)

        button4 = QPushButton("String")
        button4.clicked.connect(self.get_a_str)
        layout.addWidget(button4)

        button5 = QPushButton("Text")
        button5.clicked.connect(self.get_text)
        layout.addWidget(button5)

        container = QWidget()
        container.setLayout(layout)
        self.setCentralWidget(container)
```

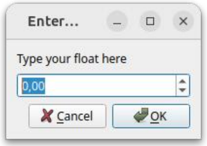


```python
    def get_an_int(self):
        title = "Enter an integer"
        label = "Type your integer here"
        my_int_value, ok = QInputDialog.getInt(
            self, title, label, value=0, min=-5, max=5, step=1 )
        print("Result:", ok, my_int_value)
```



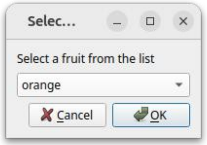
0

```python
    def get_a_float(self):
        title = "Enter a float"
        label = "Type your float here"
        my_float_value, ok = QInputDialog.getDouble(
            self, title, label, value=0, min=-5.3, max=5.7, decimals=2, )
        print("Result:", ok, my_float_value)
```



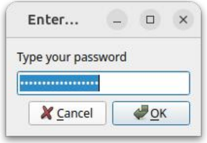
0,00

```python
    def get_a_str_from_a_list(self):
        title = "Select a string"
        label = "Select a fruit from the list"
        items = ["apple", "pear", "orange", "grape"]
        initial_selection = 2
        # orange, indexed from 0
        my_selected_str, ok = QInputDialog.getItem(
            self, title, label, items, current=initial_selection, editable=False, )
        print("Result:", ok, my_selected_str)
```

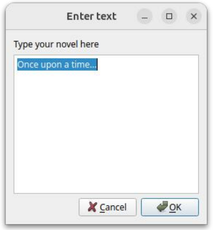


orange

```python
    def get_a_str(self):
        title = "Enter a string"
        label = "Type your password"
        text = "my secret password"
        mode = QLineEdit.EchoMode.Password
        my_selected_str, ok = QInputDialog.getText(
            self, title, label, mode, text )
        print("Result:", ok, my_selected_str)
```



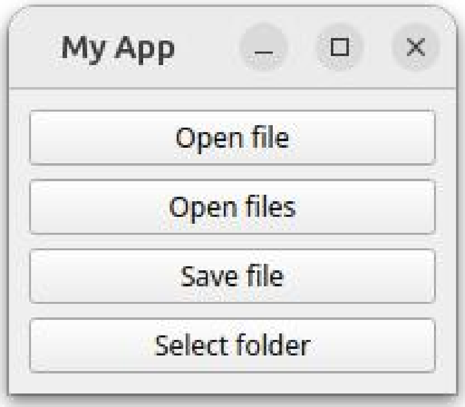
```python
    def get_text(self):
        title = "Enter text"
        label = "Type your novel here"
        text = "Once upon a time..."
        my_selected_str, ok = QInputDialog.getMultiLineText(
            self, title, label, text )
        print("Result:", ok, my_selected_str)
```



Once upon a time...

```python
app = QApplication(sys.argv)
window = MainWindow()
window.show()
app.exec()
```

```python
import sys
from PyQt5.QtWidgets import (
    QApplication,
    QFileDialog,
    QMainWindow,
    QPushButton,
    QVBoxLayout,
    QWidget, )

FILE_FILTERS = [
    "Portable Network Graphics files (*.png)",
    "Text files (*.txt)",
    "Comma Separated Values (*.csv)",
    "All files (*)", ]

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()

        self.setWindowTitle("My App")
        layout = QVBoxLayout()

        button1 = QPushButton("Open file")
        button1.clicked.connect(self.get_filename)
        layout.addWidget(button1)

        button2 = QPushButton("Open files")
        button2.clicked.connect(self.get_filenames)
        layout.addWidget(button2)

        button3 = QPushButton("Save file")
        button3.clicked.connect(self.get_save_filename)
        layout.addWidget(button3)

        button4 = QPushButton("Select folder")
        button4.clicked.connect(self.get_folder)
        layout.addWidget(button4)

        container = QWidget()
        container.setLayout(layout)
        self.setCentralWidget(container)

    def get_filename(self):
        caption = "" # Empty uses default caption.
        initial_dir = "" # Empty uses current folder.
        initial_filter = FILE_FILTERS[3] # Select one from the list.
        filters = ";;".join(FILE_FILTERS)
        print("Filters are:", filters)
        print("Initial filter:", initial_filter)

        filename, selected_filter = QFileDialog.getOpenFileName(
            self,
            caption=caption,
            directory=initial_dir,
            filter=filters,
            initialFilter=initial_filter, )

        print("Result:", filename, selected_filter)

    def get_filenames(self):
        caption = "" # Empty uses default caption.
        initial_dir = "" # Empty uses current folder.
        initial_filter = FILE_FILTERS[1] # Select one from the list.
        filters = ";;".join(FILE_FILTERS)
        print("Filters are:", filters)
        print("Initial filter:", initial_filter)

        filenames, selected_filter = QFileDialog.getOpenFileNames(
            self,
            caption=caption,
            directory=initial_dir,
            filter=filters,
            initialFilter=initial_filter, )

        print("Result:", filenames, selected_filter)

    def get_save_filename(self):
        caption = "" # Empty uses default caption.
        initial_dir = "" # Empty uses current folder.
        initial_filter = FILE_FILTERS[2] # Select one from the list.
        filters = ";;".join(FILE_FILTERS)
        print("Filters are:", filters)
        print("Initial filter:", initial_filter)

        filename, selected_filter = QFileDialog.getSaveFileName(
            self,
            caption=caption,
            directory=initial_dir,
            filter=filters,
            initialFilter=initial_filter, )

        print("Result:", filename, selected_filter)

    def get_folder(self):
        caption = "Select folder"
        initial_dir = "" # Empty uses current folder.
        folder_path = QFileDialog.getExistingDirectory(
            self,
            caption="",
            directory="")
        print("Result:", folder_path)

app = QApplication(sys.argv)
window = MainWindow()
window.show()
app.exec()
```


My App
Open file
Open files
Save file
Select folder