

DIALOGS

(hộp thoại)

Nội dung

Chúng ta sẽ tìm hiểu nội dung như sau:

- Đầu vào hộp thoại
- Sử dụng đầu vào hộp thoại
- Sử dụng màu hộp thoại
- Sử dụng font hộp thoại
- Sử dụng tập tin hộp thoại

Giới Thiệu

Hộp thoại (dialog box) dùng để nhận dữ liệu và hướng dẫn người dùng nhập thông tin đúng cách. Có hai loại chính:

- **Modal dialog:** Bắt buộc người dùng nhập thông tin; không thể dùng phần khác của ứng dụng cho đến khi đóng hộp thoại.
- **Non-modal (modeless) dialog:** Cho phép người dùng vừa làm việc với ứng dụng vừa mở hộp thoại; thường dùng cho thông tin không quan trọng.

QInputDialog – Hộp thoại nhập dữ liệu trong PyQt5

- Chức năng:** Lấy giá trị từ người dùng qua hộp thoại với văn bản và hai nút **OK/Cancel**.

- Các loại đầu vào:**

- 1.**Số nguyên:** `getInt()` – hộp xoay nhận số nguyên

`quantity, ok = QInputDialog.getInt(self, "Order Quantity", "Enter quantity:", 2, 1, 100, 1)`

- 2.**Số thực:** `getDouble()` – hộp xoay nhận số dấu phẩy động

`price, ok = QInputDialog.getDouble(self, "Price of the product", "Enter price:", 1.5, 0, 100, 2)`

- 3.**Văn bản:** `getText()` – nhận chuỗi từ người dùng

`name, ok = QInputDialog.getText(self, 'Get Customer Name', 'Enter your name:')`

- 4.**Lựa chọn từ danh sách:** `getItem()` – combo box chọn item, có thể chỉnh sửa hoặc không

`countryName, ok = QInputDialog.getItem(self, "Input Dialog", "List of countries", countries, 0, False)`

Ứng dụng Dialog without Buttons

- Hộp thoại có thể nhận **số nguyên, số thực, hoặc văn bản.**

- **Các bước chính:**

1. Vẽ **Label, Line Edit, Push Button** trên form.

2. Label: "Quốc gia của bạn", Button: "Chọn Quốc gia".

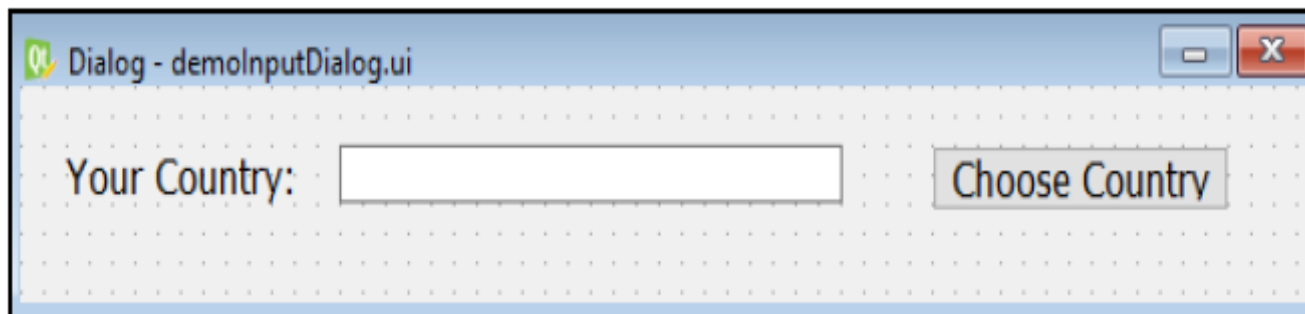
3. Đặt `objectName` cho Line Edit là `lineEditCountry` và Button là `pushButtonCountry`.

4. Lưu form với tên **`demoInputDialog.ui`**.

5. Chuyển UI sang Python:

```
pyuic5 demoInputDialog.ui -o demoInputDialog.py
```

6. Tạo file Python mới (**`callInputDialog.pyw`**) và import `demoInputDialog.py` để sử dụng giao diện.



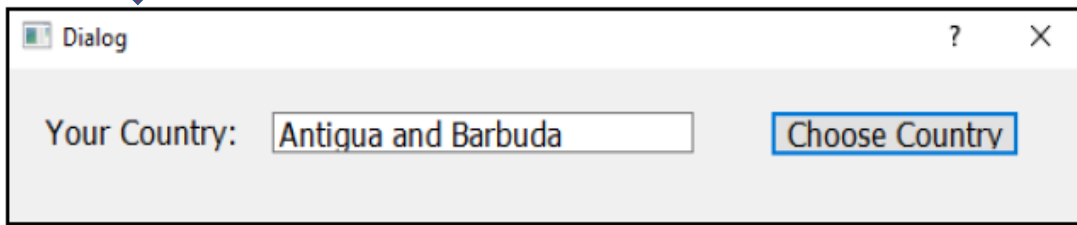
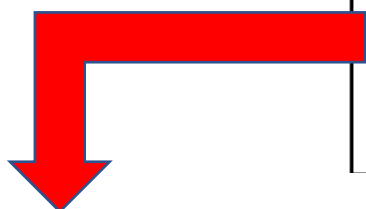
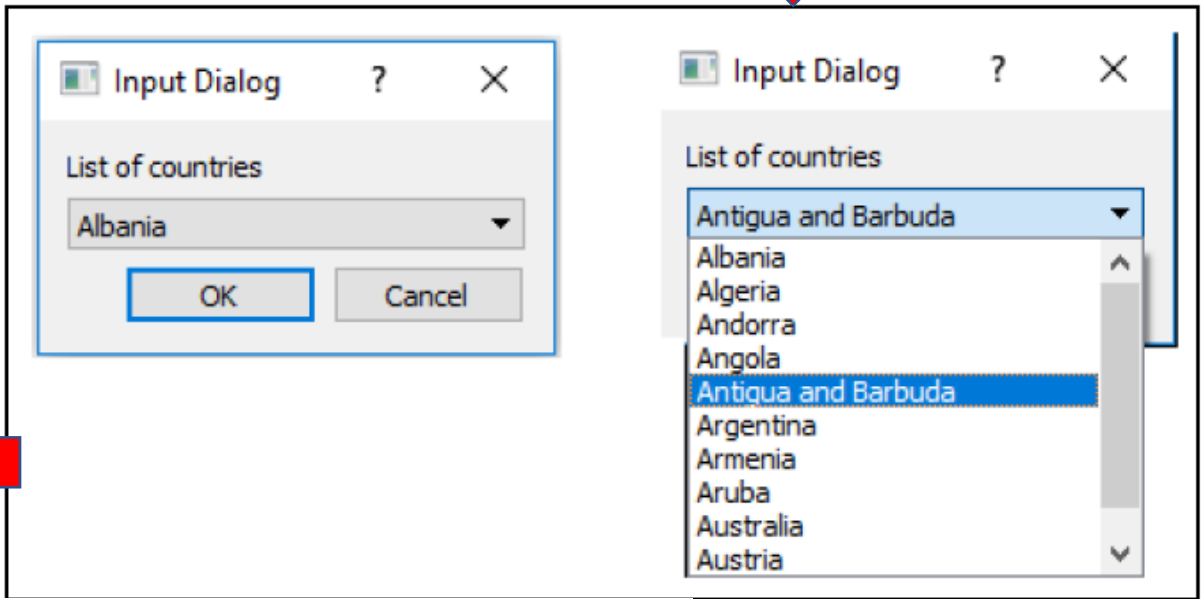
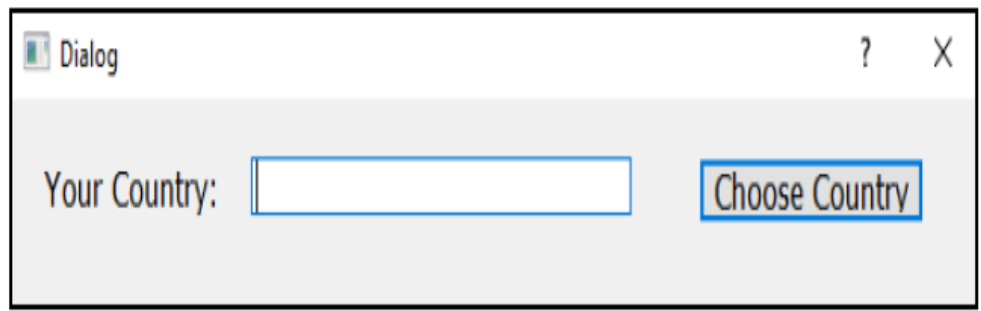
***Nếu không dùng được pyuic5
thì dùng cách khác là***

```
python -m PyQt5.uic.pyuic -x demo.ui -o demo.py
```

Ứng dụng Dialog without Buttons

```
import sys
from PyQt5.QtWidgets import QDialog, QApplication, QDialogBox
from demoInputDialog import *
class MyForm(QDialog):
    def __init__(self):
        super().__init__()
        self.ui = Ui_Dialog()
        self.ui.setupUi(self)
        self.ui.pushButtonCountry.clicked.connect(self.dispmessage)
        self.show()
    def dispmessage(self):
        countries = ("Albania", "Algeria", "Andorra", "Angola",
                    "Antigua and Barbuda", "Argentina", "Armenia", "Aruba",
                    "Australia", "Austria", "Azerbaijan")
        countryName, ok = QDialogBox.getItem(self, "Input
        Dialog", "List of countries", countries, 0, False)
        if ok and countryName:
            self.ui.lineEditCountry.setText(countryName)
if __name__=="__main__":
    app = QApplication(sys.argv)
    w = MyForm()
    w.show()
    sys.exit(app.exec_())
```

Ứng dụng Dialog without Buttons

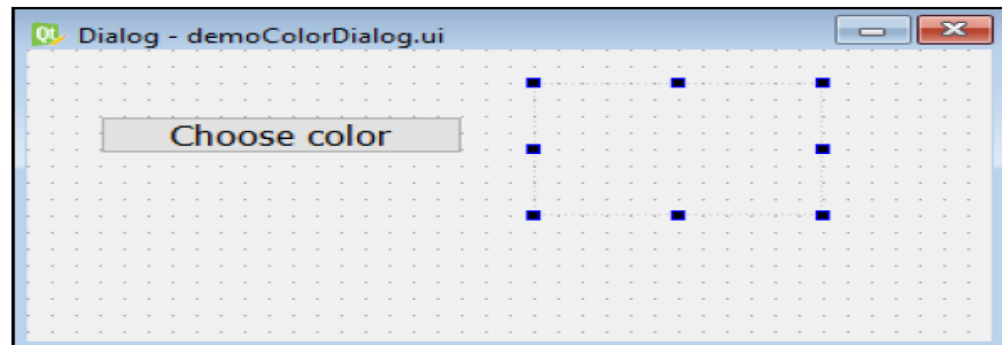


Sử Dụng Màu Hộp Thoại

- Người dùng có thể chọn màu có sẵn hoặc tạo màu mới.
- Màu chọn hiển thị trong khung (Frame) và mã thập lục phân hiển thị qua Label.
- Sử dụng **QColorDialog** để mở hộp thoại chọn màu.

Ứng dụng tạo Hộp Thoại chọn màu

1. Kéo thả **Push Button**, **Frame**, **Label** lên form.
2. Đặt text Push Button là **Chọn màu** và `objectName: pushButtonColor`.
3. Đặt `objectName` Frame: `frameColor`, Label: `labelColor`.
4. Lưu form là **demoColorDialog.ui** (XML).
5. Dùng **pyuic5** chuyển `.ui` sang Python: `demoColorDialog.py`.
6. Tạo file Python khác (**callColorDialog.pyw**) và nhập `demoColorDialog.py` để sử dụng giao diện.

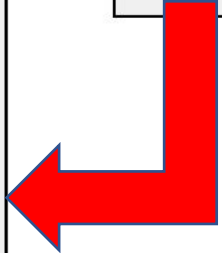
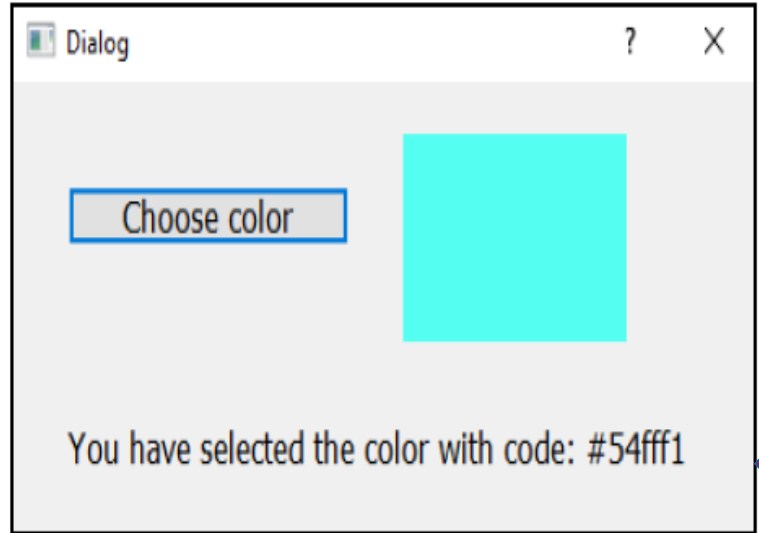
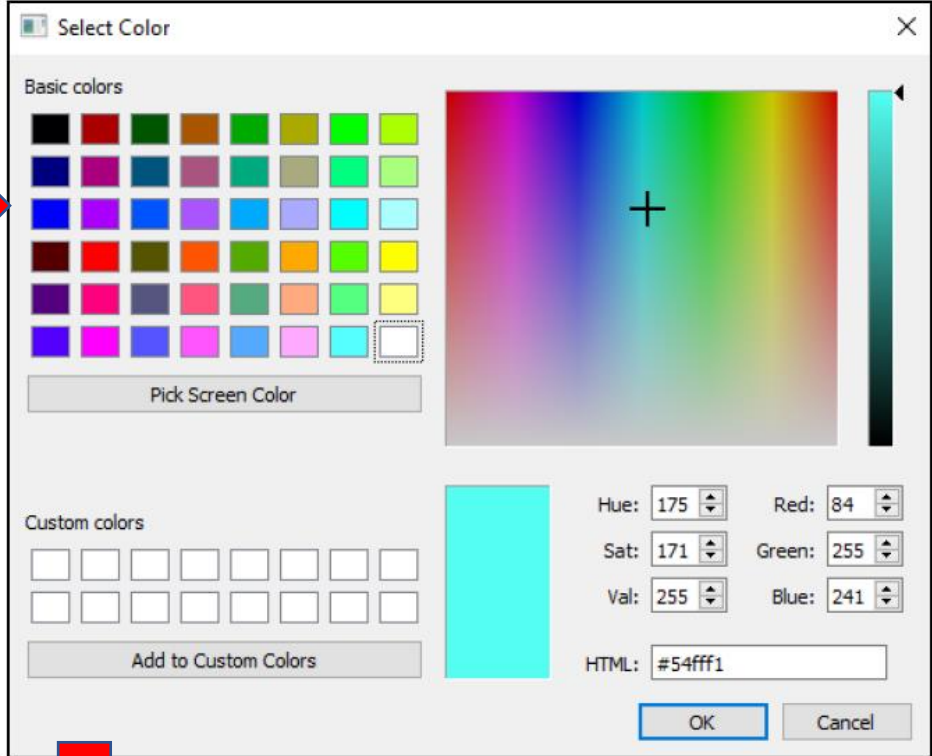
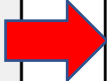
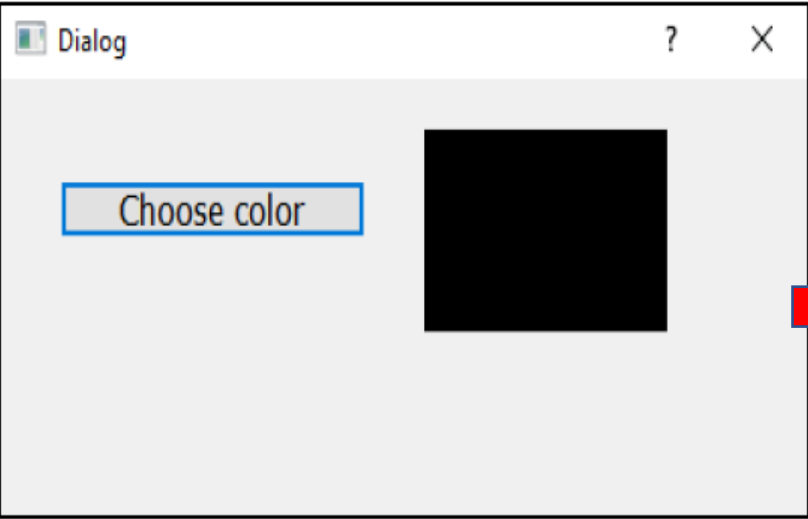


Cài đặt Hộp Thoại chọn màu

```
import sys
from PyQt5.QtWidgets import QDialog, QApplication, QColorDialog
from PyQt5.QtGui import QColor
from demoColorDialog import *
class MyForm(QDialog):
    def __init__(self):
        super().__init__()
        col = QColor(0, 0, 0)
        self.ui = Ui_Dialog()
        self.ui.setupUi(self)
        self.ui.frameColor.setStyleSheet("QWidget { background-
color: %s }" % col.name())
        self.ui.pushButtonColor.clicked.connect(self.dispcolor)
        self.show()
    def dispcolor(self):
        col = QColorDialog.getColor()
        if col.isValid():
            self.ui.frameColor.setStyleSheet("QWidget { background-
color: %s }" % col.name())

            self.ui.labelColor.setText("You have selected the color
with
            code: " + str(col.name()))
if __name__=="__main__":
    app = QApplication(sys.argv)
    w = MyForm()
    w.show()
    sys.exit(app.exec_())
```

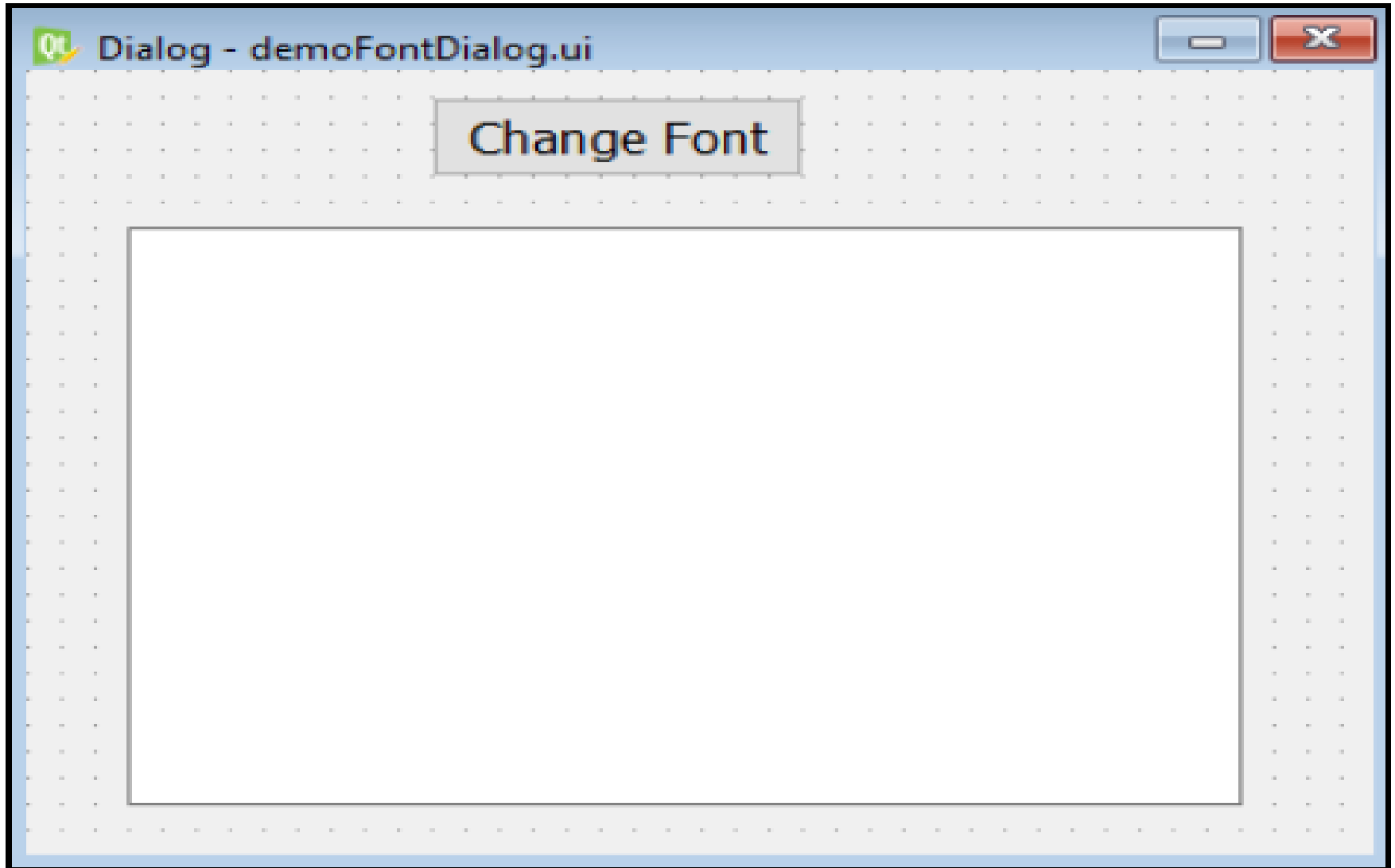
Thực thi ứng dụng Hộp Thoại chọn màu



Tạo ứng dụng hộp Thoại chọn Font

- Dùng **Text Edit** để hiển thị văn bản và **Push Button** để mở hộp thoại chọn kiểu chữ.
- Sử dụng lớp **QFontDialog** để chọn font.
- Các bước cơ bản:
 1. Kéo thả **Push Button** và **Text Edit** lên form.
 2. Đặt tên nút là `pushButtonFont` và gán nhãn “Chọn Kiểu chữ”.
 3. Lưu file giao diện là `demoFontDialog.ui`.
 4. Dùng **pyuic5** chuyển `.ui` thành file Python `demoFontDialog.py`.
 5. Tạo file `callFontDialog.pyw` để gọi giao diện và xử lý sự kiện.

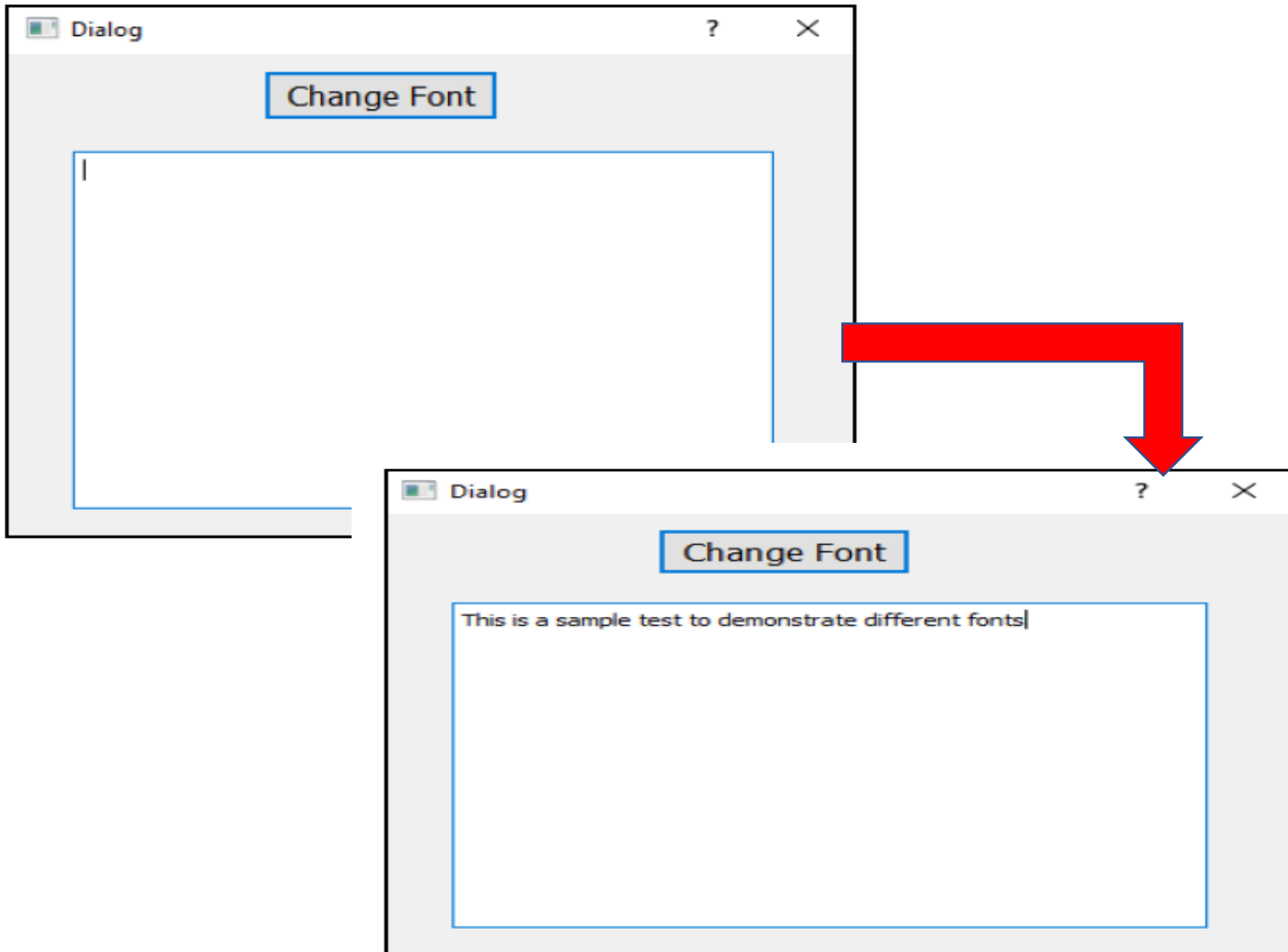
Tạo ứng dụng hộp Thoại chọn Font



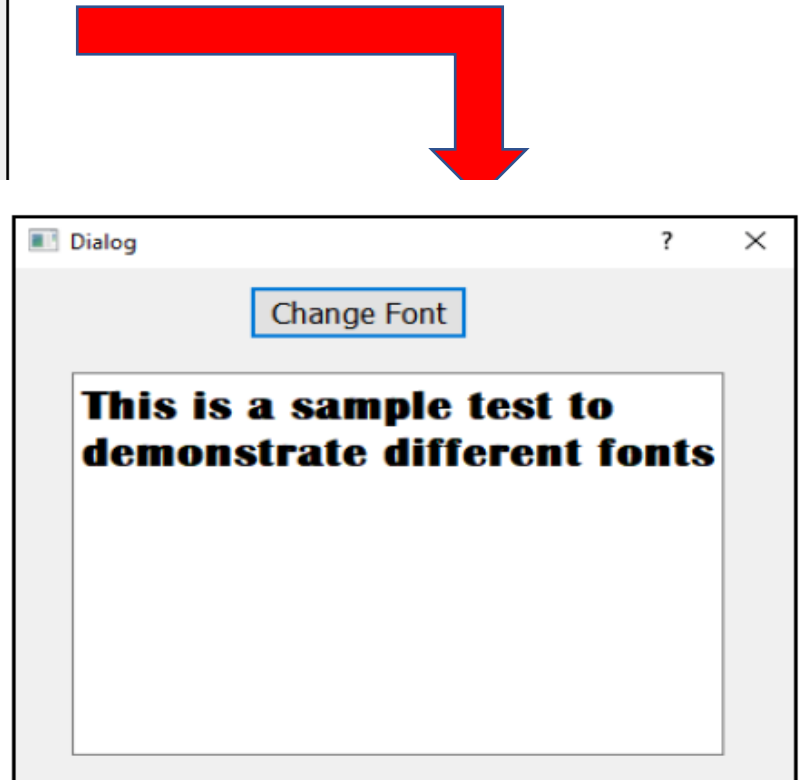
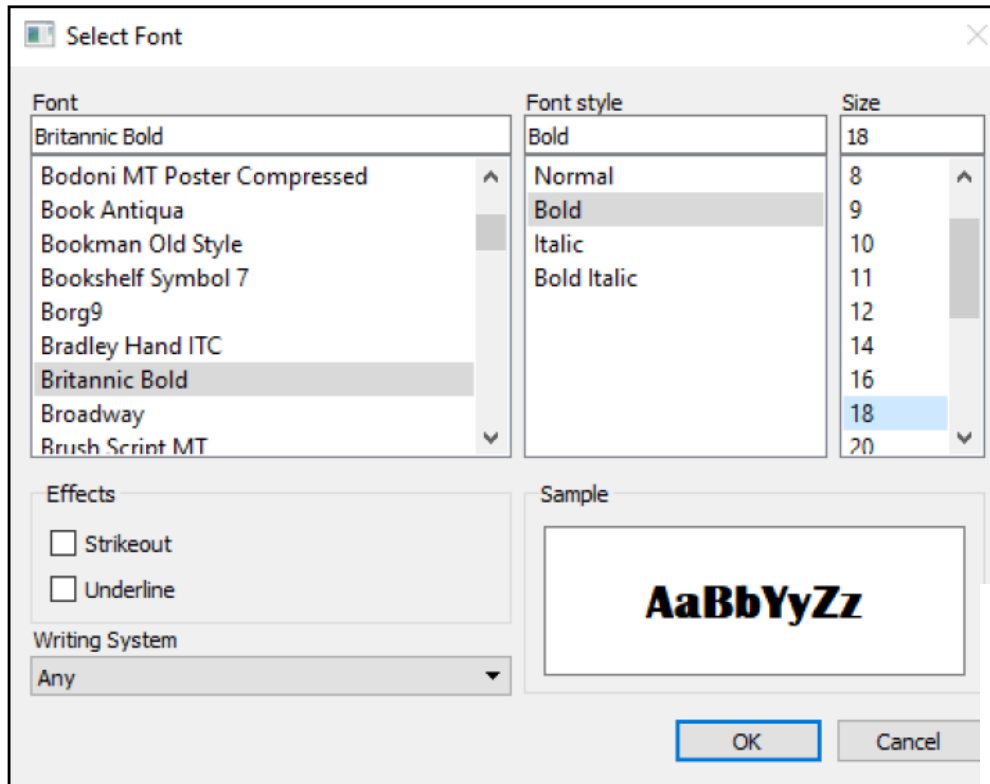
Tạo ứng dụng hộp Thoại chọn Font

```
import sys
from PyQt5.QtWidgets import QDialog, QApplication, QFontDialog
from demoFontDialog import *
class MyForm(QDialog):
    def __init__(self):
        super().__init__()
        self.ui = Ui_Dialog()
        self.ui.setupUi(self)
        self.ui.pushButtonFont.clicked.connect(self.changefont)
        self.show()
    def changefont(self):
        font, ok = QFontDialog.getFont()
        if ok:
            self.ui.textEdit.setFont(font)
if __name__=="__main__":
    app = QApplication(sys.argv)
    w = MyForm()
    w.show()
    sys.exit(app.exec_())
```

Tạo ứng dụng hộp Thoại chọn Font



Tạo ứng dụng hộp Thoại chọn Font



Sử Dụng Hộp Thoại chọn tập tin

- Tạo menu **File** với hai tùy chọn: **Mở** và **Lưu**.
- **Mở**: Hiển thị hộp thoại chọn file, nội dung file được tải vào trình soạn thảo.
- **Lưu**: Cập nhật nội dung trở lại file.
- Sử dụng lớp **QFileDialog** để chọn file hoặc thư mục

Hai phương thức chính của QFileDialog

- **getOpenFileName()** – Mở hộp thoại chọn tập tin để mở.

- Cú pháp: file_name =

- QFileDialog.getOpenFileName(self, title, path, filter)

- filter giúp lọc loại tập tin hiển thị (ví dụ: hình ảnh, văn bản, XML).

- **getSaveFileName()** – Mở hộp thoại lưu tập tin.

- Cú pháp: file_name =

- QFileDialog.getSaveFileName(self, title, path, filter, options)

- options cho phép tùy chỉnh cách hiển thị hộp thoại.

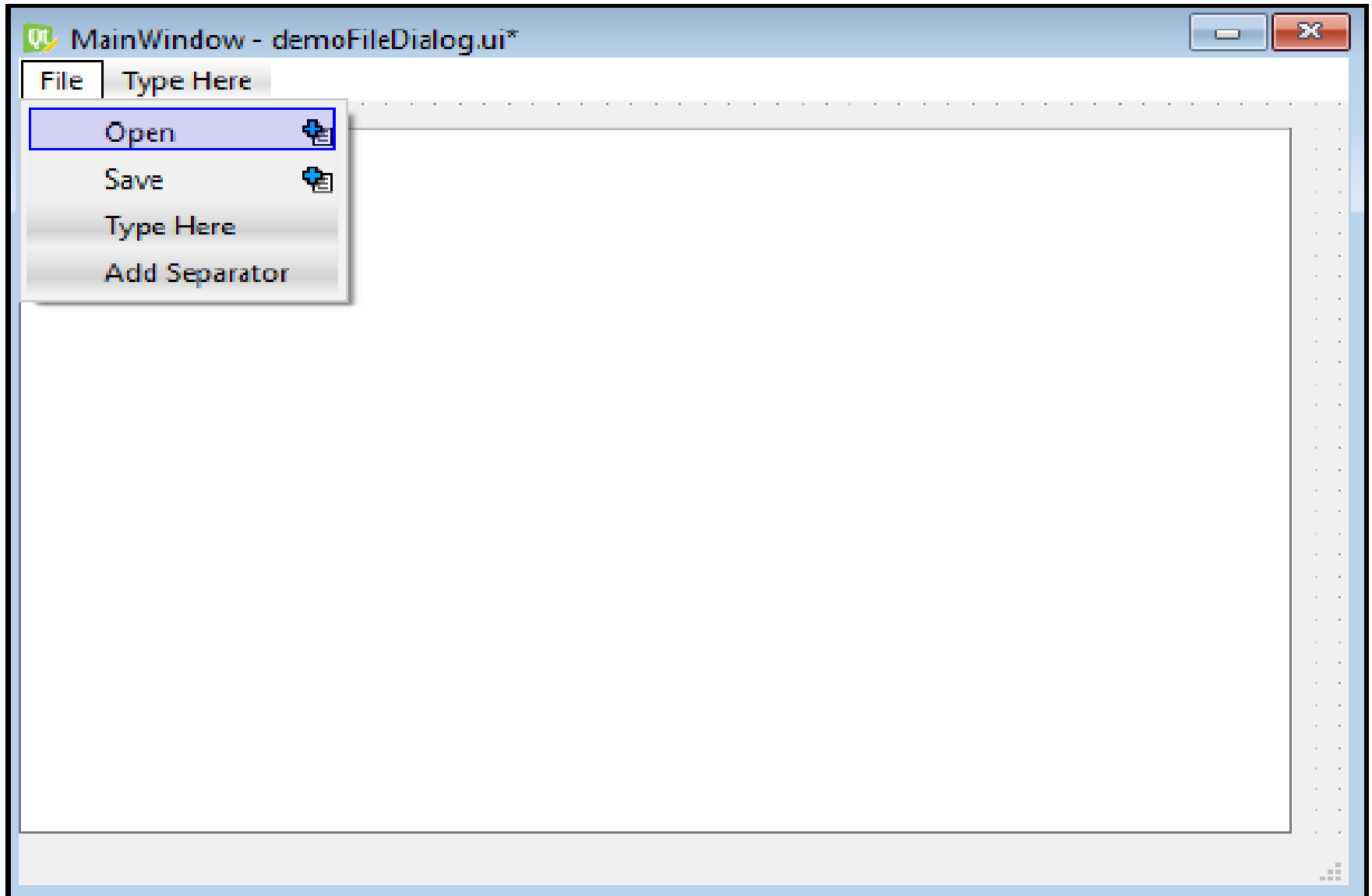
- Nếu không đặt phần mở rộng, tập tin sẽ lưu với định dạng mặc định (ví dụ .txt).

Ứng dụng Hộp Thoại chọn tập tin

Tạo Menu trong Main Window

1. Dùng menu thay vì nút để thao tác tập tin giúp ứng dụng giống thời gian thực.
2. Thanh menu mặc định có tùy chọn **Type Here**.
3. Gõ tên menu, ví dụ: **File**, nhấn Enter.
4. Thêm menu item dưới File: **Open**, nhấn Enter, sau đó thêm **Save**.
5. Kết quả: Menu **File** với hai menu item **Open** và **Save** xuất hiện trên ứng dụng.

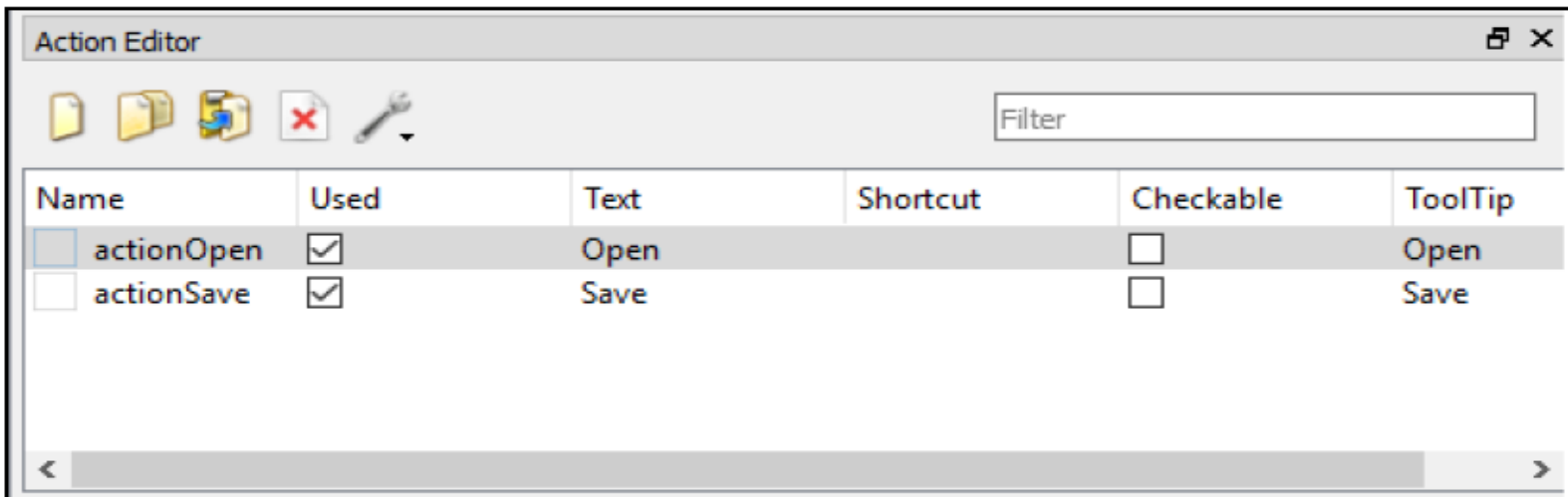
Ứng dụng Hộp Thoại chọn tập tin



Sử Dụng File Hộp Thoại

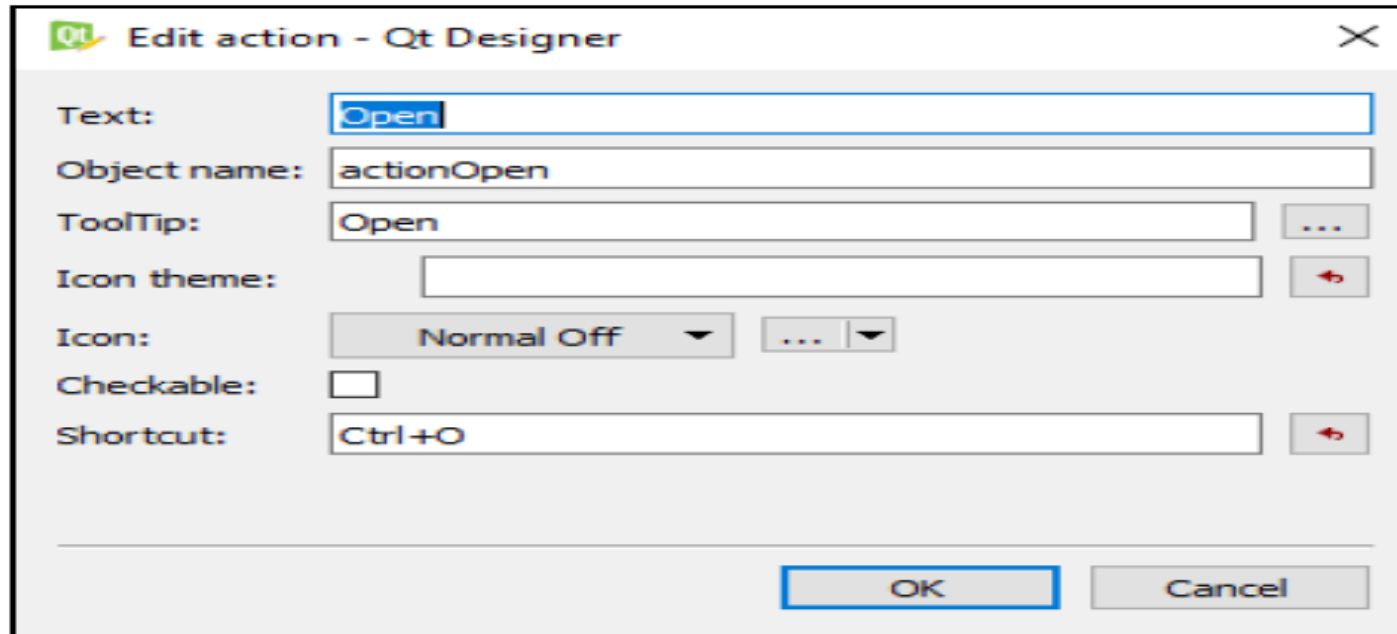
Action Editor:

- Tên mặc định của menu **Open** → actionOpen, **Save** → actionSave.
- Tab **Shortcut** trống → chưa gán phím tắt.



Gán phím tắt cho menu trong Qt Designer

- **Gán phím tắt cho menu:** Ctrl+O cho Open, Ctrl+S cho Save.
- **Chuyển file .ui sang Python** và dùng trong script khác để hiển thị giao diện.



The screenshot shows the 'Action Editor' dialog box in Qt Designer. It displays a table with the following data:

Name	Used	Text	Shortcut	Checkable	ToolTip
actionOpen	<input checked="" type="checkbox"/>	Open	Ctrl+O	<input type="checkbox"/>	Open
actionSave	<input checked="" type="checkbox"/>	Save	Ctrl+S	<input type="checkbox"/>	Save

Sử Dụng File Hộp Thoại

```
import sys
from PyQt5.QtWidgets import QMainWindow, QApplication, QAction,
FileDialog
from demoFileDialog import *
class MyForm(QMainWindow):
    def __init__(self):
        super().__init__()
        self.ui = Ui_MainWindow()
```

```
self.ui.setupUi(self)
self.ui.actionOpen.triggered.connect(self.openFileDialog)
self.ui.actionSave.triggered.connect(self.saveFileDialog)
self.show()
def openFileDialog(self):
    fname = QFileDialog.getOpenFileName(self, 'Open file',
    '/home')
    if fname[0]:
        f = open(fname[0], 'r')
    with f:
        data = f.read()
        self.ui.textEdit.setText(data)
def saveFileDialog(self):
    options = QFileDialog.Options()
    options |= QFileDialog.DontUseNativeDialog
    fileName, _ = QFileDialog.getSaveFileName(self,
    "QFileDialog.
    getSaveFileName()", "", "All Files (*);;Text Files (*.txt)",
    options=options)
    f = open(fileName, 'w')
    text = self.ui.textEdit.toPlainText()
    f.write(text)
    f.close()
if __name__=="__main__":
    app = QApplication(sys.argv)
    w = MyForm()
    w.show()
    sys.exit(app.exec_())
```

Sử Dụng File Hộp Thoại

- **Open:** Khi nhấp vào menu *Open* hoặc nhấn `Ctrl+O`, hộp thoại mở file xuất hiện. Người dùng chọn file, nhấn *OK* → nội dung file hiển thị trong **Text Edit**.
- **Save:** Sau khi chỉnh sửa, nhấp *Save* → hộp thoại lưu file xuất hiện. Có thể lưu với tên mới hoặc ghi đè file gốc. Nội dung trong **Text Edit** sẽ được ghi vào file.
- Các menu *Open* và *Save* được kết nối với các phương thức `openFileDialog()` và `saveFileDialog()` để xử lý mở/lưu file.

