



# Desktop App trong Python

---

# Nội dung

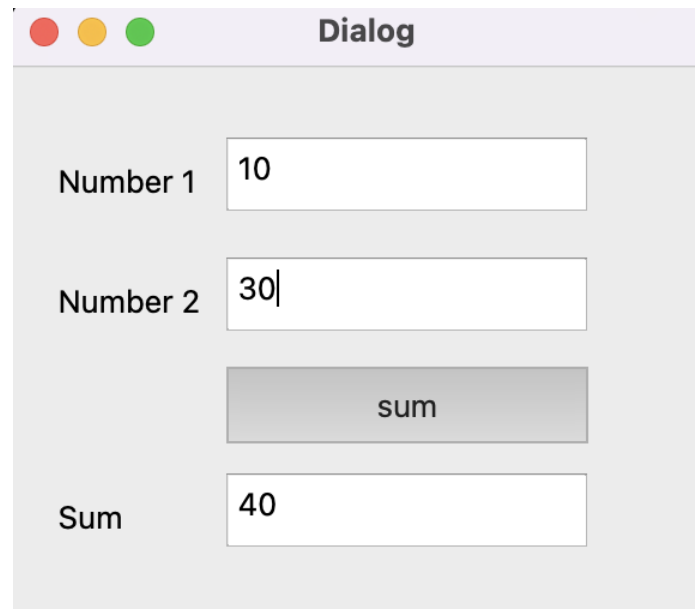
---

1. Các khái niệm cơ bản
2. Xây dựng giao diện người sử dụng (GUI)
3. Các phương thức làm việc với file
4. Đọc & Ghi file
5. Đóng file
6. Bài tập thực hành

# GUI (Graphic User Interface)

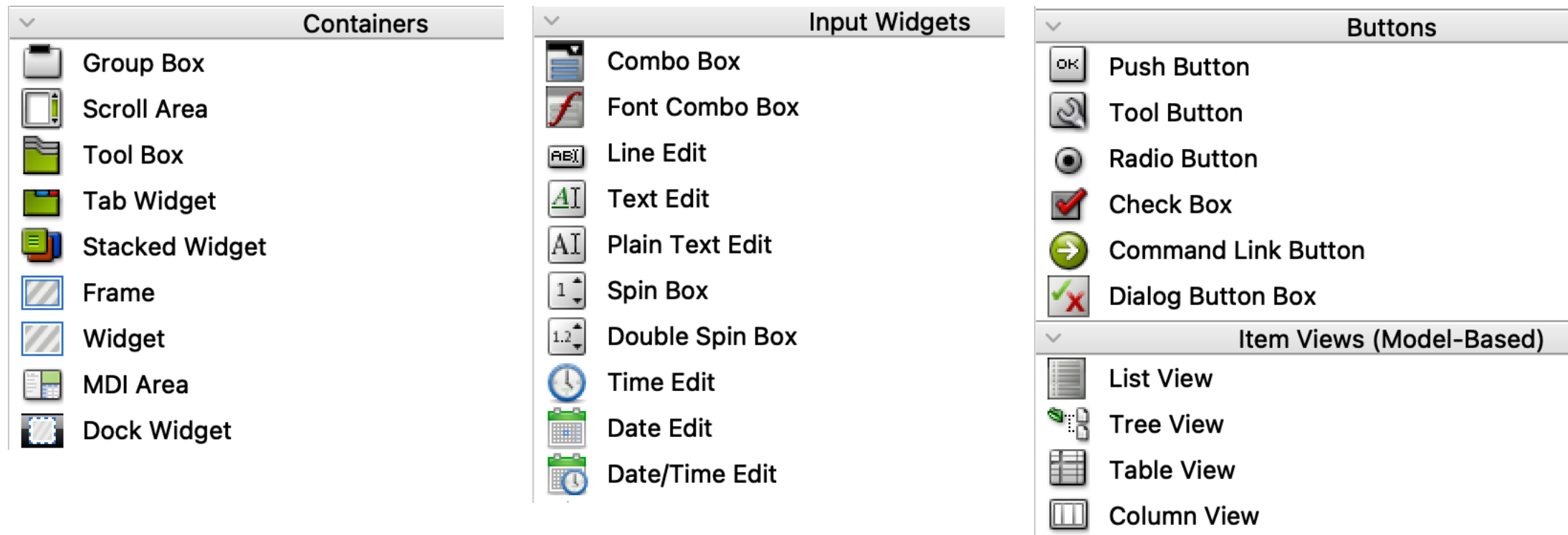
Giao diện đồ họa giúp cho người sử dụng dễ dàng tương tác với máy tính thông qua các thao tác đơn giản sử dụng chuột & bàn phím

*Ví dụ: Xây dựng giao diện và viết chương trình nhập vào 2 số và tính tổng 2 số đó*



The image shows a graphical user interface window titled "Dialog". It features two input fields for numbers. The first field, labeled "Number 1", contains the value "10". The second field, labeled "Number 2", contains the value "30". Below these fields is a button labeled "sum". At the bottom of the window, there is a label "Sum" and an output field displaying the result "40".

# Các thành phần chính trên GUI



# GUI TRONG PYTHON

---

## Công cụ/thư viện để tạo giao diện đồ họa:

- Công cụ QT Designer: Để tạo giao diện bằng cách kéo thả
- Thư viện PyQt6: Kết hợp ngôn ngữ Python với thư viện QT để tạo giao diện đồ họa và lập trình sự kiện

## Cài đặt công cụ/thư viện:

- Cài đặt QT Designer: <https://build-system.fman.io/qt-designer-download>
- Cài đặt PyQt6, gõ lệnh: **pip3 install PyQt6**

## Tham khảo:

- <https://doc.qt.io/qtforpython/>

# GUI TRONG PYTHON

---

Các bước xây dựng một chương trình có giao diện đồ họa bằng Python:

**Bước 1.** Sử dụng QT Designer để tạo giao diện, lưu thành file .ui dạng XML (ví dụ tên file là **giaodien.ui**)

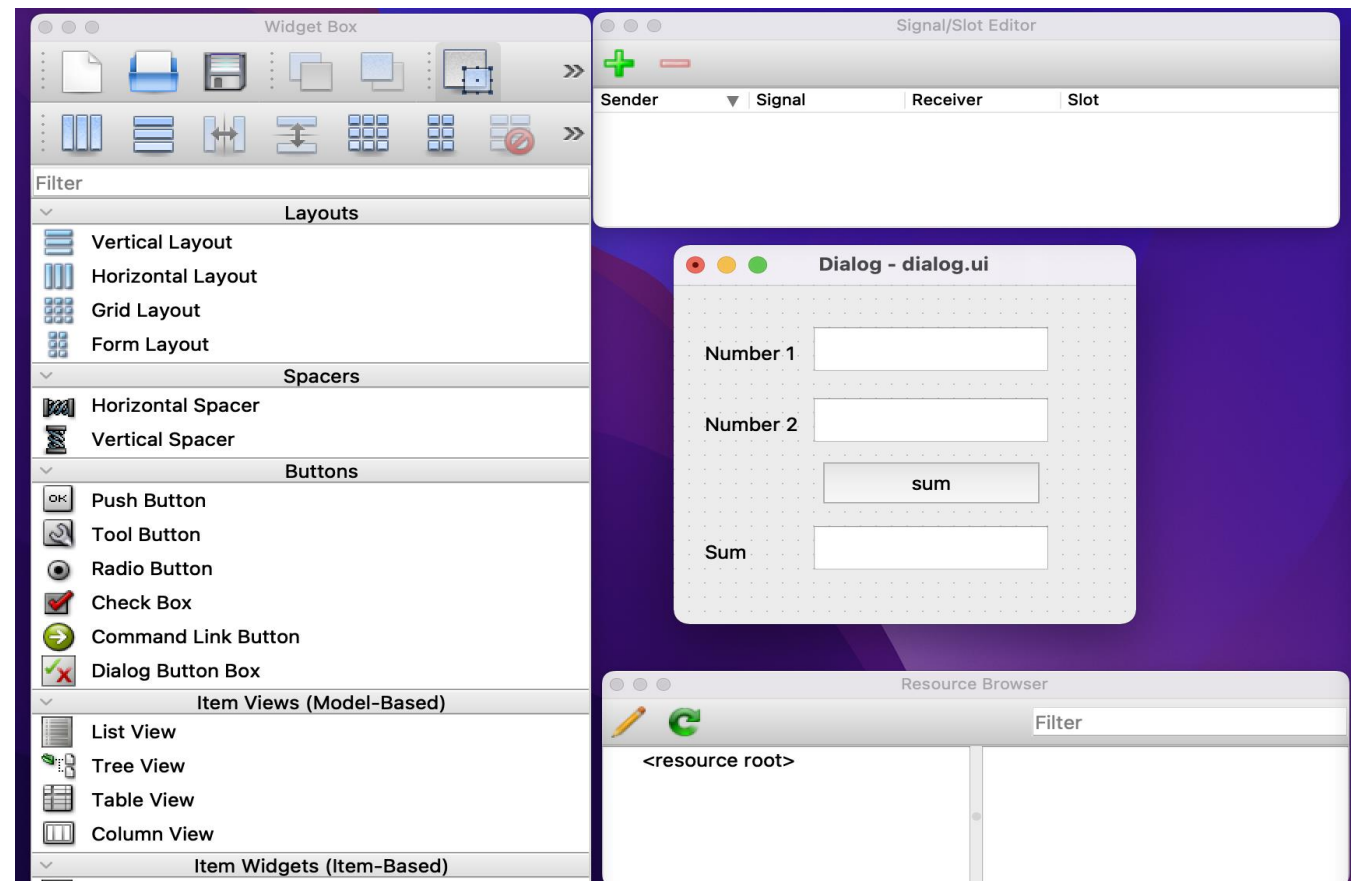
**Bước 2.** Sử dụng lệnh **pyuic6** để chuyển file .ui thành file code python (ví dụ tên file là **giaodien.py**)

*Câu lệnh như sau: `pyuic6 -x giaodien.ui -o giaodien.py`*

**Bước 3.** Lập trình bổ sung sự kiện (event) cho giao diện và xử lý logic

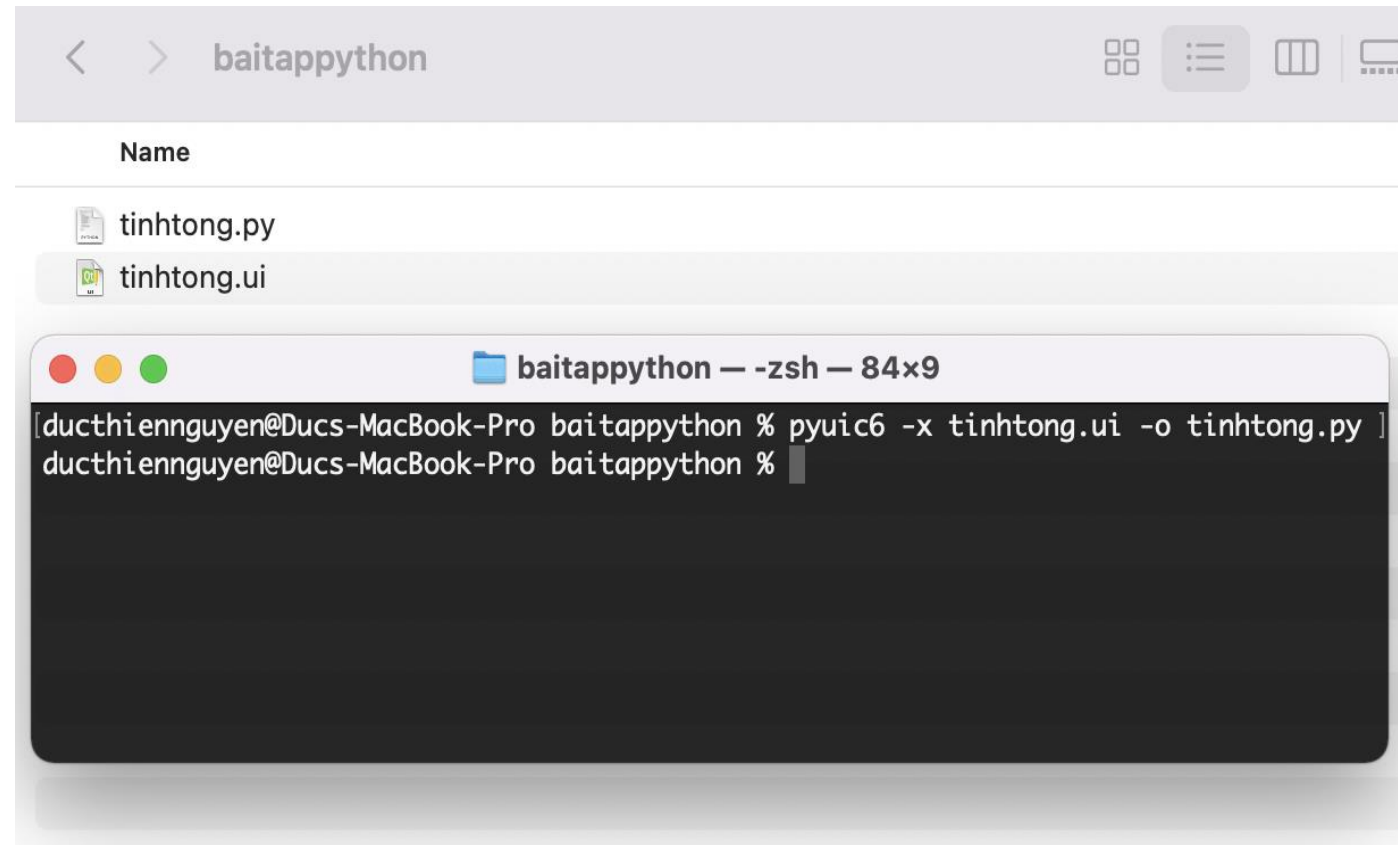
# Ví dụ GUI

Bước 1: Sử dụng phần mềm QT Designer để kéo thả tạo giao diện. Sau đó lưu file với tên “**tinhtong.ui**” đặt tại thư mục bất kỳ (ví dụ thư mục /baitapppython)



# Ví dụ GUI

Bước 2: Mở cửa sổ CommandLine (CMD) và chuyển tới thư mục “baitappython” sau đó sử dụng lệnh `pyuic6 -x tinhtong.ui -o tinhtong.py` để chuyển file “tinhtong.ui” thành file code “tinhtong.py”





# Ví dụ GUI






Bước 3. Lập trình bổ sung sự kiện (event) cho giao diện và xử lý logic (bổ sung code trực tiếp vào file `tinhtong.py`)

```
self.retranslateUi(Dialog)
QtCore.QMetaObject.connectSlotsByName(Dialog)

... # Khai báo sự kiện nhấn nút
... self.pushButton.clicked.connect(self.clicked_btn)
...
# Định nghĩa hàm xử lý sự kiện nhấn nút
def clicked_btn(self):
    ... number1 = int(self.textEdit_1.toPlainText())
    ... number2 = int(self.textEdit_2.toPlainText())
    ... sum = number1 + number2;
    ... self.textEdit_3.setText(str(sum))

def retranslateUi(self, Dialog):
    _translate = QtCore.QCoreApplication.translate
    Dialog.setWindowTitle(_translate("Dialog", "Dialog"))
```

# Một số sự kiện (events), hàm (functions) truy xuất dữ liệu thông dụng trên PyQt6

Controls	Events/Signal	Functions
 Push Button	clicked, pressed	setText
 Radio Button	toggled	setText, isChecked
 Text Edit	changed	setText, toPlainText
 Check Box	stateChanged	setText, isChecked
 Combo Box	currentIndexChanged, currentTextChanged	addItem, addItems, setCurrentIndex, setCurrentText

## Tham khảo:

- QtWidgets: <https://doc.qt.io/qtforpython/PySide6/QtWidgets/index.html>
- PyQt6 tutorial: <https://www.pythonguis.com/pyqt6-tutorial/>

---

THANKS FOR ATTENDING!