wxPython is a popular Python library used for building graphical user interfaces (GUIs). It is a wrapper around the **wxWidgets** C++ library, which provides native look-and-feel across different platforms (Windows, macOS, Linux). Below are the basic concepts and components of wxPython to help you get started.

1. Installation

To install wxPython, you can use `pip`:

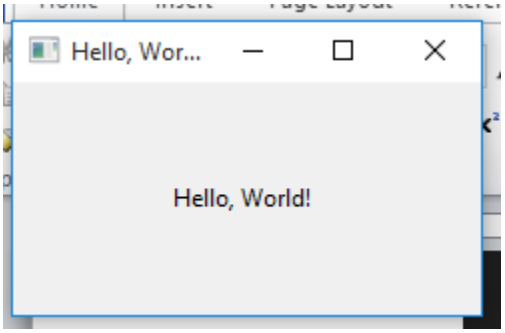
pip install wxPython

2. Basic Structure of a wxPython Application

A wxPython application typically has the following structure:

- **App Object**: An instance of the `wx.App` class is required to initialize the GUI framework.
- **Frame Object**: The `wx.Frame` is the window that appears on the screen.
- **Widgets/Controls**: Various controls like buttons, text fields, labels, etc., are placed on the frame.

Here's a basic example of a wxPython program to display hello world:-




```
import wx

# Create the wx.App instance
app = wx.App(False)

# Create the main frame (window)
frame = wx.Frame(None, title="Hello, World!", size=(250, 150))

# Create a static text to display "Hello, World!" inside the frame
panel = wx.Panel(frame)
hello_text = wx.StaticText(panel, label="Hello, World!", pos=(80, 50))
```

```
# Show the frame (window)
frame.Show()

# Start the application's main loop
app.MainLoop()
```

3. Main Components

wx.App

- The application class that manages the application's event loop.
- It must be instantiated first, and its `MainLoop` method runs the event loop.

```
app = wx.App()
app.MainLoop()
```

wx.Frame

- The main window of your application, where controls and widgets are placed.

```
frame = wx.Frame(parent, title="Title", size=(width, height))
```

wx.Panel

- A container for holding other widgets like buttons, text fields, etc. Often used to group components inside a frame.

```
panel = wx.Panel(parent)
```

wx.Widget Classes

These are the basic widgets (controls) you can use:

- **wx.Button**: A clickable button.
- **wx.TextCtrl**: A text box for entering or displaying text.
- **wx.StaticText**: A label for displaying text.
- **wx.CheckBox**: A checkbox for binary choices.

- **wx.ListBox**: A list box for displaying a list of items.

Example:

```
button = wx.Button(panel, label="Click Me", pos=(10, 10))
```

4. Event Handling

wxPython uses event-driven programming. You bind events (like button clicks or key presses) to functions (called event handlers). Here's how you can bind a button click to an event handler:

```
def on_button_click(event):
    print("Button clicked!")

button.Bind(wx.EVT_BUTTON, on_button_click)
```

Common event types:

- `wx.EVT_BUTTON`: Button click
- `wx.EVT_TEXT`: Text change in a text control
- `wx.EVT_CLOSE`: Close event for the window

5. Layouts

You can manage the layout of your widgets using sizers, which automatically adjust the size and positioning of the controls.

wx.BoxSizer

A simple sizer to arrange widgets horizontally or vertically.

```
sizer = wx.BoxSizer(wx.HORIZONTAL)
sizer.Add(button1, 0, wx.ALL, 5)
sizer.Add(button2, 0, wx.ALL, 5)
panel.SetSizer(sizer)
```

wx.GridSizer

A grid sizer for arranging widgets in rows and columns.

```
grid_sizer = wx.GridSizer(rows=2, cols=2)
grid_sizer.Add(button1)
grid_sizer.Add(button2)
panel.SetSizer(grid_sizer)
```

6. Dialogs

wxPython also provides various dialog boxes for user interaction, such as file dialogs, message dialogs, and custom dialogs.

wx.MessageDialog

To show simple message boxes:

```
dlg = wx.MessageDialog(None, "Message", "Title", wx.OK | wx.CANCEL)
dlg.ShowModal()
dlg.Destroy()
```

wx.FileDialog

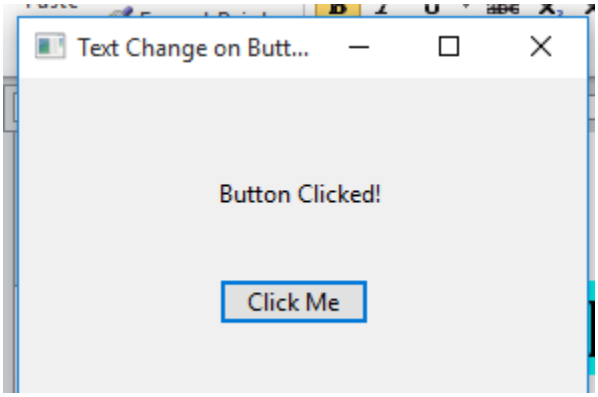
For file selection:

```
file_dialog = wx.FileDialog(None, "Open File", wildcard="Text files
(*.txt)|*.txt")
if file_dialog.ShowModal() == wx.ID_OK:
    path = file_dialog.GetPath()
file_dialog.Destroy()
```

7. Common Methods

- **Show()**: To display the frame.
- **SetSize()**: To set the size of a widget or frame.
- **Close()**: To close a window.
- **SetTitle()**: To change the title of a window or frame.

Example of Full Application



python code:-

example 1 :-

```
import wx

def on_button_click(event):
    # Change the label of the StaticText widget when the button is clicked
    hello_text.SetLabel("Button Clicked!")

# Create the wx.App instance
app = wx.App(False)

# Create the main frame (window)
frame = wx.Frame(None, title="Text Change on Button Click", size=(300, 200))

# Create a panel to hold the widgets
panel = wx.Panel(frame)

# Create a StaticText widget to display text
hello_text = wx.StaticText(panel, label="Hello, World!", pos=(100, 50))

# Create a Button widget
button = wx.Button(panel, label="Click Me", pos=(100, 100))

# Bind the button click event to the function on_button_click
button.Bind(wx.EVT_BUTTON, on_button_click)

# Show the frame (window)
frame.Show()

# Start the application's main loop
app.MainLoop()
```

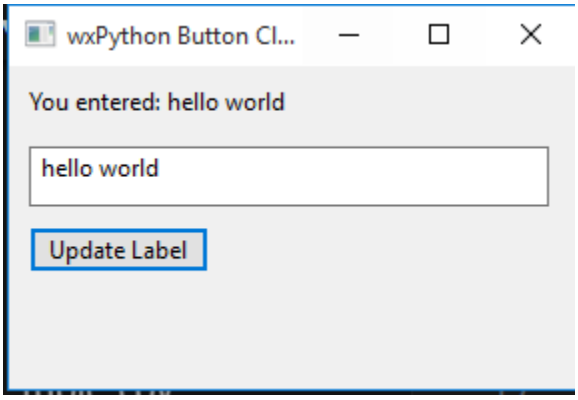
8. Event Loop

The event loop (`app.MainLoop()`) is crucial in wxPython. It keeps the application running and waits for user events, such as button clicks or window resizing.

9. Customization

You can customize the appearance and behavior of widgets, bind events, and manage application state through object-oriented programming. You can also use different themes or apply custom styles to widgets.

Here's a simple example of a wxPython application that contains a **Label**, **Text Control**, and a **Button**. When you click the button, it updates the label with the text entered in the text control.



Example 1:-

```
import wx

def on_button_click(event):
    # Get the text entered in the text control
    user_input = text_ctrl.GetValue()

    # Update the label with the entered text
    label.SetLabel(f"You entered: {user_input}")

# Create the wx.App instance
```

```
app = wx.App(False)

# Create the main frame (window)
frame = wx.Frame(None, title="wxPython Button Click Example", size=(300, 200))

# Create a panel to hold the widgets
panel = wx.Panel(frame)

# Label to display text
label = wx.StaticText(panel, label="Enter text below and click the button:", pos=(10, 10))

# Text Control for user input
text_ctrl = wx.TextCtrl(panel, value="", pos=(10, 40), size=(260, 30))

# Button that triggers the event
button = wx.Button(panel, label="Update Label", pos=(10, 80))

# Bind the button click event to the on_button_click function
button.Bind(wx.EVT_BUTTON, on_button_click)

# Show the frame (window)
frame.Show()

# Start the application's main loop
app.MainLoop()
```

Example 2:-

```
import wx

class MyFrame(wx.Frame):
    def __init__(self, parent, title):
        super().__init__(parent, title=title, size=(300, 200))

        # Panel to contain the widgets
        panel = wx.Panel(self)

        # Label to display text
        self.label = wx.StaticText(panel, label="Enter text below and click the button:", pos=(10, 10))

        # Text Control for user input
        self.text_ctrl = wx.TextCtrl(panel, value="", pos=(10, 40), size=(260, 30))
```

```
        # Button that triggers the event
        self.button = wx.Button(panel, label="Update Label", pos=(10, 80))

        # Bind button click event to the handler
        self.button.Bind(wx.EVT_BUTTON, self.on_button_click)

        # Show the frame
        self.Show()

    def on_button_click(self, event):
        # Get the text entered in the text control
        user_input = self.text_ctrl.GetValue()

        # Update the label with the entered text
        self.label.SetLabel(f"You entered: {user_input}")

class MyApp(wx.App):
    def OnInit(self):
        # Initialize the frame with a title
        frame = MyFrame(None, "wxPython Button Click Example")
        return True

if __name__ == "__main__":
    # Create the wx.App instance and run the application
    app = MyApp()
    app.MainLoop()
```

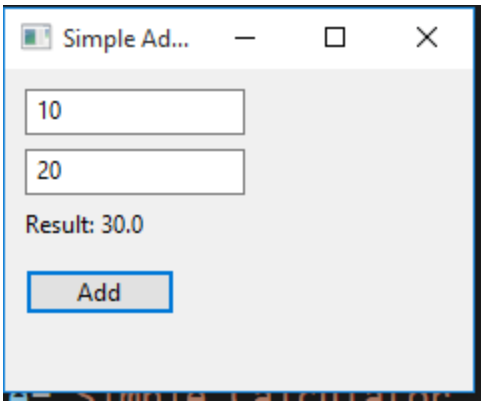
Explanation:

- **wx.StaticText**: A label to display static text. In this case, it initially shows a prompt to the user.
- **wx.TextCtrl**: A text field where users can type input.
- **wx.Button**: A button that the user can click to trigger an event.
- **Event Binding**: The `Bind` method is used to link the button click (`wx.EVT_BUTTON`) to the `on_button_click` method, which will update the label when the button is pressed.
- **`on_button_click` Method**: This method gets the text from the text control using `self.text_ctrl.GetValue()` and updates the label with the entered text.

How it Works:

1. The program starts with a label, a text control, and a button.
2. The user types something in the text control.
3. When the user clicks the button, the `on_button_click` function is called.
4. The function retrieves the text entered in the text control and updates the label with that text.

Simple Calculator Example :-



Example one without class :-

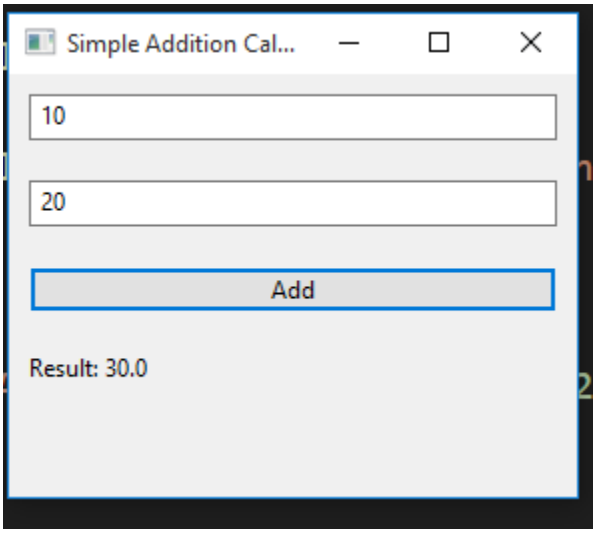
```python
import wx

def on_add(event):
    try:
        # Get the values from the text controls
        num1 = float(num1_text.GetValue())
        num2 = float(num2_text.GetValue())
        result = num1 + num2
        result_text.SetLabel(f"Result: {result}")
    except ValueError:
        result_text.SetLabel("Invalid input. Please enter numbers.")

# Create the wx.App instance
app = wx.App(False)

# Create the main frame (window)
frame = wx.Frame(None, title="Simple Addition", size=(250, 200))

# Create a panel to hold the widgets
panel = wx.Panel(frame)

# Create text boxes for the numbers and a static text for the result
num1_text = wx.TextCtrl(panel, pos=(10, 10))
num2_text = wx.TextCtrl(panel, pos=(10, 40))
result_text = wx.StaticText(panel, label="Result: ", pos=(10, 70))

# Create a button and bind the event to the function
add_button = wx.Button(panel, label="Add", pos=(10, 100))
add_button.Bind(wx.EVT_BUTTON, on_add)

# Show the frame (window)
frame.Show()
```

```
# Start the application's main loop
app.MainLoop()
```

Example 2 with class:-

```
import wx

class SimpleCalculator(wx.Frame):
    def __init__(self, *args, **kw):
        super(SimpleCalculator, self).__init__(*args, **kw)

        # Set up the window
        self.panel = wx.Panel(self)
        self.num1_text = wx.TextCtrl(self.panel, pos=(10, 10))
        self.num2_text = wx.TextCtrl(self.panel, pos=(10, 40))
        self.result_text = wx.StaticText(self.panel, label="Result: ", pos=(10, 70))

        # Add the button and bind it to the action
        add_button = wx.Button(self.panel, label="Add", pos=(10, 100))
        add_button.Bind(wx.EVT_BUTTON, self.on_add)

        # Set up the window properties
        self.SetTitle('Simple Addition')
        self.SetSize((250, 200))
        self.Centre()

    def on_add(self, event):
        try:
            # Get values from text fields, convert to float and add them
            num1 = float(self.num1_text.GetValue())
            num2 = float(self.num2_text.GetValue())
            result = num1 + num2
            self.result_text.SetLabel(f"Result: {result}")
        except ValueError:
            self.result_text.SetLabel("Invalid input. Please enter numbers.")

def main():
    app = wx.App(False)
    frame = SimpleCalculator(None, title="Simple Calculator", size=(250, 200))
    frame.Show()
    app.MainLoop()
```

```
if __name__ == '__main__':
    main()
```



To create a simple application in wxPython that adds two numbers, you can use the following code. This code will create a basic GUI where users can input two numbers, press a button to calculate the sum, and display the result.

Here's an example:

```
import wx

class Calculator(wx.Frame):
    def __init__(self, *args, **kw):
        super(Calculator, self).__init__(*args, **kw)

        self.init_ui()

    def init_ui(self):
        panel = wx.Panel(self)

        # Create a vertical box sizer to organize the layout
        vbox = wx.BoxSizer(wx.VERTICAL)

        # Create text fields for input
        self.num1_text = wx.TextCtrl(panel)
```

```
        self.num2_text = wx.TextCtrl(panel)

        # Create a button that will trigger the addition
        add_button = wx.Button(panel, label="Add")

        # Create a static text widget to display the result
        self.result_text = wx.StaticText(panel, label="Result: ")

        # Bind the button click event to the on_add method
        add_button.Bind(wx.EVT_BUTTON, self.on_add)

        # Add the widgets to the box sizer
        vbox.Add(self.num1_text, flag=wx.EXPAND | wx.ALL, border=10)
        vbox.Add(self.num2_text, flag=wx.EXPAND | wx.ALL, border=10)
        vbox.Add(add_button, flag=wx.EXPAND | wx.ALL, border=10)
        vbox.Add(self.result_text, flag=wx.EXPAND | wx.ALL, border=10)

        panel.SetSizer(vbox)

        self.SetTitle('Simple Addition Calculator')
        self.SetSize((300, 250))
        self.Centre()

    def on_add(self, event):
        try:
            num1 = float(self.num1_text.GetValue())  # Get value from first text
field
            num2 = float(self.num2_text.GetValue())  # Get value from second text
field

            result = num1 + num2
            self.result_text.SetLabel(f"Result: {result}")
        except ValueError:
            self.result_text.SetLabel("Invalid input. Please enter numbers.")

def main():
    app = wx.App(False)
    frame = Calculator(None, title="Addition App", size=(300, 250))
    frame.Show()
    app.MainLoop()

if __name__ == '__main__':
    main()
```