



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình Python nâng cao

Bài 11: *Xây dựng ứng dụng GUI*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>

2018



Nội dung

1. Giới thiệu GUI
2. Làm việc với wxPython Widgets
3. Layout Management
4. Giới thiệu phần mềm wxFormBuilder

Giới thiệu GUI

❑ Python GUI (Graphic User Interface)

Programming: lập trình ứng dụng giao diện dành cho người dùng

- Là loại ứng dụng có giao diện giúp cho người dùng dễ dàng tương tác và sử dụng

Giới thiệu GUI

- ❑ Python có một lượng lớn các GUI framework (hoặc toolkit) dành cho việc lập trình ứng dụng GUI, từ tkinter (bộ GUI truyền thống được tích hợp sẵn trong Python, sử dụng Tk) đến các giải pháp cross-platform khác như **wxPython**, Jpython, PyGUI...
- Trong phạm vi môn học này, chúng ta sẽ làm việc với wxPython.

Giới thiệu wxPython

□ wxPython

- wxPython là một **bộ công cụ GUI** đa nền tảng cho ngôn ngữ lập trình Python . Nó cho phép các lập trình viên Python tạo ra các chương trình với một giao diện người dùng đồ họa mạnh mẽ, có chức năng cao, đơn giản và dễ dàng. Nó được thực hiện như một tập hợp các mô-đun mở rộng Python bao bọc các thành phần GUI của thư viện đa nền tảng phổ biến wxWidgets, được viết bằng C++.

Giới thiệu wxPython

□ wxPython

- Giống như Python và wxWidgets, wxPython là Mã nguồn mở, có nghĩa là nó miễn phí cho bất kỳ ai sử dụng và mã nguồn có sẵn cho bất kỳ ai xem và sửa đổi. Và bất kỳ ai cũng có thể đóng góp các bản sửa lỗi hoặc cải tiến cho dự án.

Giới thiệu wxPython

□ wxPython

- wxPython là một bộ công cụ đa nền tảng. Điều này có nghĩa là cùng một chương trình sẽ chạy trên nhiều nền tảng mà không cần sửa đổi. Các nền tảng được hỗ trợ hiện nay là Microsoft Windows, Mac OS X và macOS và Linux.

Giới thiệu wxPython

□ wxPython

- Cài đặt

- Windows and macOS

pip install -U wxPython

- Linux

pip install -U \ -f

<https://extras.wxpython.org/wxPython4/extras/linux/gtk3/>

ubuntu-16.04 \ wxPython

Giới thiệu wxPython

- ❑ Xây dựng GUI application sử dụng wxPython khá dễ dàng, với các bước sau:
 - Import module `wx`.
 - Tạo class `App`.
 - Tạo một cửa sổ cấp cao nhất làm đối tượng của lớp `wx.Frame`. Các tham số kích thước và tiêu đề được đưa ra trong hàm khởi tạo.

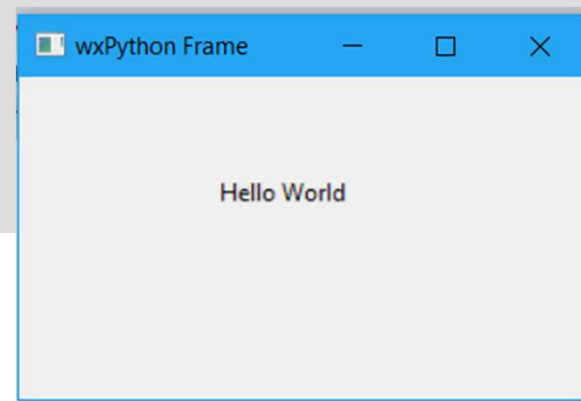
Giới thiệu wxPython

- Tạo đối tượng Panel trong cửa sổ Frame
- Thêm các điều khiển vào đối tượng Panel
- Kích hoạt cửa sổ frame bằng phương thức `show()`.
- Gọi phương thức `mainloop()` của đối tượng Application.

Giới thiệu wxPython

□ Ví dụ

```
import wx
app = wx.App()
window = wx.Frame(None, title = "wxPython Frame", size =
(300,200))
panel = wx.Panel(window)
label = wx.StaticText(panel, label = "Hello World", pos =
(100,50))
window.Show(True)
app.MainLoop()
```



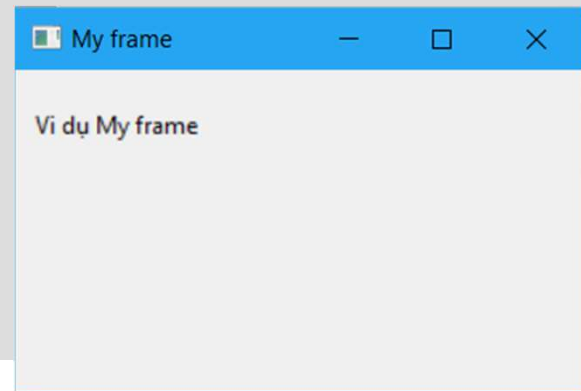
Giới thiệu wxPython

❑ Ví dụ : chúng ta có thể tạo class

```
import wx

class MyFrame(wx.Frame):
    def __init__(self):
        wx.Frame.__init__(self, None, title="My frame",
                           size=(300,200))
        panel=wx.Panel(self,-1)
        wx.StaticText(panel,-1,label="Vi dụ My frame", pos=(10,20))

if __name__=="__main__":
    app=wx.App()
    frame = MyFrame()
    frame.Show(True)
    app.MainLoop()
```



Nội dung

1. Giới thiệu GUI
2. Làm việc với wxPython Widgets
3. Layout Management
4. Giới thiệu phần mềm wxFormBuilder

Làm việc với wxPython Widgets

- ❑ **wxPython** cung cấp nhiều điều khiển khác nhau như StaticText, TextCtrl, button...sử dụng cho GUI application. Những điều khiển này còn được gọi là widget.
- ❑ Có nhiều loại widget trong wxPython =>

Làm việc với wxPython Widgets

wxButton	Tạo button trong ứng dụng.
wxCheckbox	Tạo các tùy chọn dạng checkbox. Người dùng có thể chọn nhiều tùy chọn cùng một lúc.
wxTextCtrl	Tạo điều khiển hiển thị văn bản và cho cập nhật
wxFrame	Dùng như một bộ chứa (container widget) để tổ chức các widget khác.
wxStaticText	Dùng làm nhãn tiêu đề (single-line caption) cho các widget khác.
wxListbox	Cung cấp một danh sách các tùy chọn cho người dùng.
wxMenuItem	Tạo item menu cho menu
wxMenu	Tạo thanh menu chứa các wxMenuItem cho menu ứng dụng
wxMessage	Hiển thị nhiều dòng (multiline text field) để xác nhận các giá trị từ người dùng.

Làm việc với wxPython Widgets

wxRadiobutton	Hiển thị các tùy chọn dạng radio button. Người dùng chỉ có thể chọn một lựa chọn tại một thời điểm.
wxSlider	tiện ích cho phép người dùng chọn một số từ trong phạm vi
...	

Làm việc với wxPython Widgets

□ Trong phạm vi bài này, chúng ta sẽ làm việc với một số widget cơ bản như:

- wxStaticText
- wxTextCtrl
- wxButton
- wxCheckbutton
- wxRadiobutton
- wxListbox
- wxMenu
- wxFonts, wxColors
- wxImage
- wxLayout
- wxDropdown list (select)
- wx.MessageBox

Làm việc với wxPython Widgets

□ wxStaticText

Cú pháp: `wx.StaticText(parent, id, label, pos=wx.DefaultPosition, size=wx.DefaultSize, style=0, name="staticText")`

■ Trong đó:

- **parent** : đối tượng chứa
- **id** : Mã định danh
- **label** : Văn bản hiển thị
- **pos**: Vị trí của widget như một đối tượng wx.Point hoặc một tuple Python
- **size**: Kích thước của đối tượng wx.Size hoặc kiểu tuple Python
- **name**: Tên được sử dụng để tìm đối tượng

Làm việc với wxPython Widgets

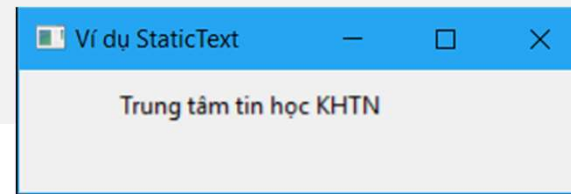
❑ wxStaticText – style

wx.ALIGN_CENTER	Canh nội dung giữa khung hình chữ nhật static text
wx.ALIGN_LEFT	Mặc định, canh nội dung lề trái
wx.ALIGN_RIGHT	Canh nội dung lề phải
wx.ST_NO_AUTORESIZE	Không tự thay đổi kích thước khi thay đổi nội dung bằng lệnh SetLabel

Làm việc với wxPython Widgets

- Ví dụ: staticText

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ StaticText", size=(300,100))
panel = wx.Panel(frame, -1)
wx.StaticText(panel, -1, "Trung tâm tin học KHTN", size=(150,30),
pos=(50,10))
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

□ wxTextCtrl

- **Cú pháp:**

```
wx.TextCtrl(parent, id, value =  
"", pos=wx.DefaultPosition,  
size=wx.DefaultSize, style=0,  
validator=wx.DefaultValidator,  
name=wx.TextCtrlNameStr)
```

Làm việc với wxPython Widgets

□ wxTextCtrl - style

wx.TE_CENTER	Nội dung được canh giữa
wx.TE_LEFT	Mặc định, nội dung canh trái
wx.TE_RIGHT	Nội dung canh phải
wx.TE_PASSWORD	Văn bản đã nhập sẽ không được hiển thị, nhưng thay vào đó được che dấu bằng dấu *
wx.TE_READONLY	Thiết lập nội dung chỉ đọc và không thể sửa đổi

Khóa

Làm việc với wxPython Widgets

□ wxTextCtrl – Một số thuộc tính

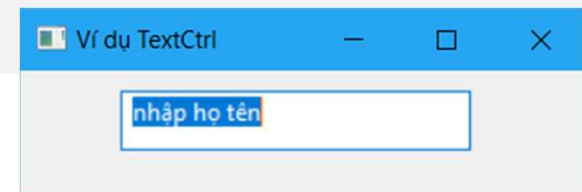
AppendText(text)	Thêm text vào điều khiển
GetValue()	Trả về nội dung trong điều khiển
SetValue(value)	Gán lại nội dung cho điều khiển
wx.TE_PASSWORD	Văn bản đã nhập sẽ không được hiển thị, nhưng thay vào đó được che dấu bằng dấu *
wx.TE_READONLY	Thiết lập nội dung chỉ đọc và không thể sửa đổi

Khóa

Làm việc với wxPython Widgets

- Ví dụ: TextCtrl

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ TextCtrl", size=(300,100))
panel = wx.Panel(frame, -1)
wx.TextCtrl(panel, -1, "nhập họ tên", pos=(50,10), size=(175, 30))
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

□ *wxfont*

- **Cú pháp:** `wx.Font(pointSize, family, style, weight, underline=False, faceName="", encoding=wx.FONTENCODING_DEFAULT)`

Làm việc với wxPython Widgets

□ wxFont font families

Font	Description
wx.DECORATIVE	A formal, old-English style font
wx.DEFAULT	The system default font
wx.MODERN	A monospace (fixed-pitch) font
wx.ROMAN	A serif font, generally something like Times New Roman
wx.SCRIPT	<i>A handwriting or cursive font</i>
wx.SWISS	A sans-serif font, generally something like Helvetica or Arial

Làm việc với wxPython Widgets

□ wxFont

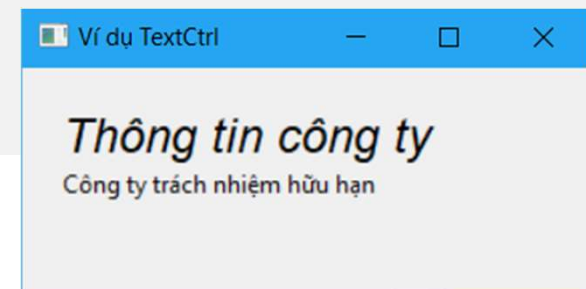
style: wx.NORMAL, wx.SLANT, or wx.ITALIC

weight: wx.NORMAL, wx.LIGHT, or wx.BOLD

Làm việc với wxPython Widgets

❑ Ví dụ

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ TextCtrl", size=(300,150))
panel = wx.Panel(frame, -1)
str = "Thông tin công ty"
text = wx.StaticText(panel, -1, str, (20, 20))
font = wx.Font(18, wx.DECORATIVE, wx.ITALIC, wx.NORMAL)
text.SetFont(font)
wx.StaticText(panel, -1, "Công ty trách nhiệm hữu hạn", (20,50))
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

❑ wxColor

Để gán màu cho điều khiển ta dùng các hàm sau:

SetForegroundColour: màu chữ

SetBackgroundColour: Màu nền

Làm việc với wxPython Widgets

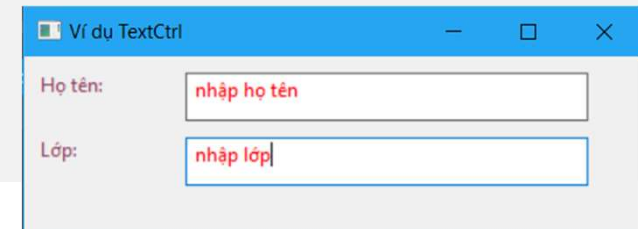
□ wxColor

- Dùng hexa để thể hiện, ví dụ: "#fff" là trắng, "#000000" là đen, "#000fff000" là xanh, "#00ffff" là cyan (xanh lá cây pha xanh dương)...
- Dùng color name để thể hiện, ví dụ: "white", "black", "red", "green", "blue", "cyan", "yellow", "magenta"...
- Dùng thư viện wxPython cung cấp để thể hiện : wx.RED, wx.BLUE, wx.YELLOW, wx.GREEN,...

Làm việc với wxPython Widgets

❑ wxColor – ví dụ

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ TextCtrl", size=(400,150))
panel = wx.Panel(frame, -1)
stHoTen=wx.StaticText(panel,-1, label="Họ tên:",pos=(10,10),
size=wx.DefaultSize)
stHoTen.SetForegroundColour("#8f3d5e")
txtHoTen = wx.TextCtrl(panel,-1,value="nhập họ tên", pos=(100,10),
size=(250,30))
txtHoTen.SetForegroundColour("red")
_color = wx.Colour(143,61,94)
stLop=wx.StaticText(panel,-1, label="Lớp:",pos=(10,50),
size=wx.DefaultSize)
stLop.SetForegroundColour(_color)
txtLop = wx.TextCtrl(panel,-1,value="nhập lớp", pos=(100,50),
size=(250,30))
txtLop.SetForegroundColour(wx.RED)
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

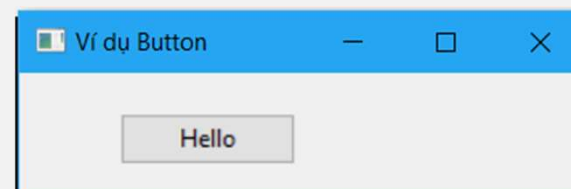
□ wxButton

- **Cú pháp:** `wx.Button(parent, id, label, pos, size=wx.DefaultSize, style=0, validator, name="button")`

Làm việc với wxPython Widgets

□ Ví dụ:

```
import wx
def OnClick(self):
    button.SetLabel("Clicked")
app = wx.App()
frame = wx.Frame(None, title="Ví dụ TextCtrl",size=(300,100))
panel = wx.Panel(frame, -1)
button = wx.Button(panel, -1, "Hello", pos=(50, 20))
frame.Bind(wx.EVT_BUTTON, OnClick, button)
button.SetDefault()
frame.Show(True)
app.MainLoop()
```

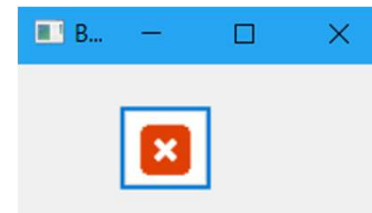


Làm việc với wxPython Widgets

❑ Ví dụ: wxBitmapButton

```
import wx
def OnClick(self):
    frame.Destroy()

app = wx.App()
frame = wx.Frame(None, title="Ví dụ TextCtrl", size=(300,150))
panel = wx.Panel(frame, -1)
close = wx.Image("Media/close.png", wx.BITMAP_TYPE_ANY).ConvertToBitmap()
button = wx.BitmapButton(panel, -1, close, pos=(50, 20))
frame.Bind(wx.EVT_BUTTON, OnClick, button)
button.SetDefault()
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

❑ wxSlider

- **Cú pháp:** `wx.Slider(parent, id, value, minValue, maxValue, pos=wx.DefaultPosition, size=wx.DefaultSize, style=wx.SL_HORIZONTAL, validator=wx.DefaultValidator, name="slider")`

Làm việc với wxPython Widgets

❑ wxSlider – Một số thuộc tính

GetRange()	Trả về là tuple(minValue, maxValue)
SetRange(minValue, maxValue)	Cài đặt minValue và maxValue cho slider
GetValue()	Trả về giá trị hiện hành của slider

Khoa Học

Làm việc với wxPython Widgets

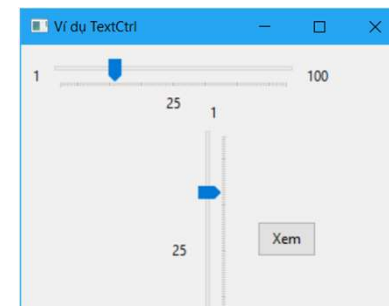
```
import wx
def onXem(self):
    dlg = wx.MessageDialog(None, "Slide 1 : value = " +
        str/slider1.GetRange()), 'A Message Box', wx.OK)
    dlg.ShowModal()

app = wx.App()
frame = wx.Frame(None, title="Ví dụ Slide", size=(340, 320))
panel = wx.Panel(frame, -1)
slider1 = wx.Slider(panel, -1, 25, 1, 100, pos=(10, 10), size=(250, -
1), style=wx.SL_HORIZONTAL | wx.SL_AUTOTICKS | wx.SL_LABELS )

slider2 = wx.Slider(panel, -1, 25, 1, 100, pos=(125, 50), size=(-1,
250), style=wx.SL_VERTICAL | wx.SL_AUTOTICKS | wx.SL_LABELS )

btnXem = wx.Button(panel, label="Xem", pos=(200, 150), size=(50, 30))
frame.Bind(wx.EVT_BUTTON, onXem, btnXem)

frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

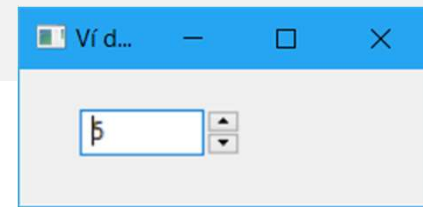
❑ wxSpinner

- **Cú pháp:** `wx.SpinCtrl (parent, id=-1, value=wx.EmptyString, pos=wx.DefaultPosition, size=wx.DefaultSize, style=wx.SP_ARROW_KEYS, min=0, max=100, initial=0, name="wxSpinCtrl")`

Làm việc với wxPython Widgets

- Ví dụ:

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ SpinCtrl", size=(300,150))
panel = wx.Panel(frame, -1)
sc = wx.SpinCtrl(panel, -1, "", (30, 20), (80, -1))
sc.SetRange(1,100)
sc.SetValue(5)
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

❑ wxCheckBox

- **Cú pháp:** `wx.CheckBox (parent, id, label, pos=wx.DefaultPosition, size=wx.DefaultSize, style=0, name="checkBox")`

Làm việc với wxPython Widgets

❑ wxCheckBox

- Xử lý sự kiện checkbox: EVT_CHECKBOX

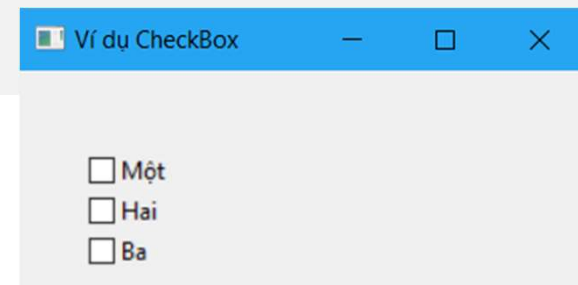
```
self.Bind(wx.EVT_CHECKBOX, self.onChecked)

def onChecked(self, e):
    cb = e.GetEventObject()
    print (cb.GetLabel(), ' is clicked', cb.GetValue())
```

Làm việc với wxPython Widgets

● Ví dụ:

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ CheckBox", size=(300,150))
panel = wx.Panel(frame, -1)
wx.CheckBox(panel, -1, "Một", (35, 40), (150, 20))
wx.CheckBox(panel, -1, "Hai", (35, 60), (150, 20))
wx.CheckBox(panel, -1, "Ba", (35, 80), (150, 20))
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

❑ wxRadioButton

- **Cú pháp:** `wx.RadioButton(parent, id, label, pos=wx.DefaultPosition, size=wx.DefaultSize, style=0, validator=wx.DefaultValidator, name="radioButton")`

Làm việc với wxPython Widgets

❑ wxRadioButton

• Xử lý sự kiện: EVT_RADIOBUTTON

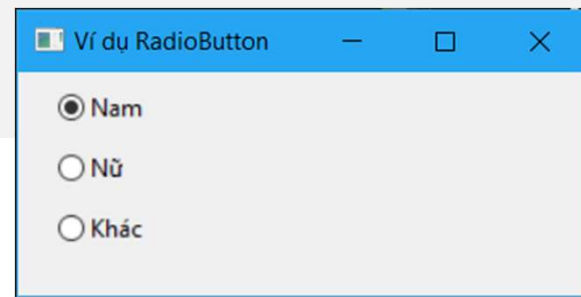
```
self.Bind(wx.EVT_RADIOBUTTON, self.OnRadiogroup)
```

```
def OnRadiogroup(self, e):  
    rb = e.GetEventObject()  
    print(rb.GetLabel(), ' is clicked from Radio Group' )
```

Làm việc với wxPython Widgets

● Ví dụ:

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ RadioButton", size=(300,150))
panel = wx.Panel(frame, -1)
radio1 = wx.RadioButton(panel, -1, "Nam", pos=(20,
10), style=wx.RB_GROUP)
radio2 = wx.RadioButton(panel, -1, "Nữ", pos=(20, 40))
radio3 = wx.RadioButton(panel, -1, "Khác", pos=(20, 70))
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

❑ wxListBox

- **Cú pháp:** `wx.ListBox (parent, id, pos=wx.DefaultPosition, size=wx.DefaultSize, choices=None, style=0, validator=wx.DefaultValidator, name="listBox")`

Làm việc với wxPython Widgets

□ wxListBox

- Style thể hiện thuộc tính chọn

wx.LB_EXTENDED	Người dùng có thể chọn một loạt các mục bằng cách sử dụng một cú nhấp chuột, hoặc tương đương bàn phím.
wx.LB_MULTIPLE	Người dùng có thể chọn nhiều mục bằng cách dùng ctrl + nhấp chuột.
wx.LB_SINGLE	Người dùng chỉ chọn được một mục chọn

Làm việc với wxPython Widgets

□ wxListBox

- Style thể hiện thuộc tính scroll

wx.LB_ALWAYS_SB	Listbox luôn hiển thị thanh cuộn dọc, cho dù có cần hay không.
wx.LB_HSCROLL	Listbox sẽ tạo thanh cuộn ngang nếu các mục quá rộng.
wx.LB_NEEDED_SB	Listbox sẽ hiển thị thanh cuộn dọc nếu cần. Đây là mặc định.

Làm việc với wxPython Widgets

❑ wxListBox – Một số phương thức

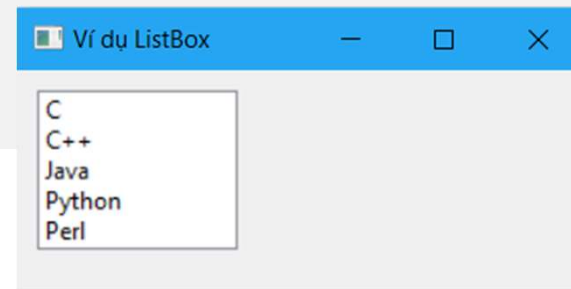
Append(item)	Thêm item vào cuối list
Clear()	Xóa tất cả item trong list
Delete(n)	Xóa item tại index = n
Deselect(n)	Bỏ chọn tại vị trí n
FindString(string)	Trả về vị trí tìm thấy, hoặc -1 nếu không tìm thấy.
GetCount()	Trả về số phần tử trong danh sách.
GetSelection()	Trả về index đang chọn (list thiết lập wx.LB_SINGLE)
GetSelections()	Trả về danh sách index được chọn
GetStringSelection()	Trả về string được chọn (list thiết lập wx.LB_SINGLE)
GetString(n)	Trả về string tại vị trí n

Làm việc với wxPython Widgets

● Ví dụ:

```
import wx
def onListBox(self):
    dlg = wx.MessageDialog(None,
        event.GetEventObject().GetStringSelection(), 'A Message', wx.OK)
    dlg.ShowModal()

app = wx.App()
frame = wx.Frame(None, title="Ví dụ ListBox", size=(300,150))
panel = wx.Panel(frame, -1)
languages = ['C', 'C++', 'Java', 'Python', 'Perl']
lst = wx.ListBox(panel, size = (100,-1), pos=(10,10), choices =
    languages, style = wx.LB_SINGLE)
frame.Bind(wx.EVT_LISTBOX, onListBox, lst)
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

❑ wxListCtrl

wx.ListCtrl là một tiện ích nâng cao và do đó phức tạp hơn. Trong trường hợp ListBox chỉ hiển thị một cột, ListCtrl có thể hiển thị nhiều cột.

Làm việc với wxPython Widgets

❑ wxListCtrl

- Cú pháp:

```
ListCtrl(parent,  
          id=ID_ANY,  
          pos=DefaultPosition,  
          size=DefaultSize,  
          style=LC_ICON,  
          validator=DefaultValidator,  
          name=ListCtrlNameStr)
```

Làm việc với wxPython Widgets

❑ wxListCtrl

- Tạo tiêu đề cột

```
InsertColumn (self, col,  
              heading,  
              format=LIST_FORMAT_LEFT,  
              width=LIST_AUTOSIZE)
```

Làm việc với wxPython Widgets

❑ wxListCtrl

- Thêm dữ liệu vào listCtrl

```
lstCtrl.InsertStringItem(index, "Gt1")  
lstCtrl.SetStringItem(index, 1, "Gt2")  
lstCtrl.SetStringItem(index, 2, "Gt3")
```

Làm việc với wxPython Widgets

❑ wxListCtrl

- Xử lý sự kiện khi click vào list ctrl

```
lstCtrl.Bind( wx.EVT_LIST_ITEM_SELECTED, self.chon )
```

- Lấy giá trị khi click vào list ctrl

```
def onClick( self, event ):  
    item=event.GetItem()  
    print(item.GetText())
```

Làm việc với wxPython Widgets

❑ wxListCtrl - Ví dụ:

```
import sys
import wx

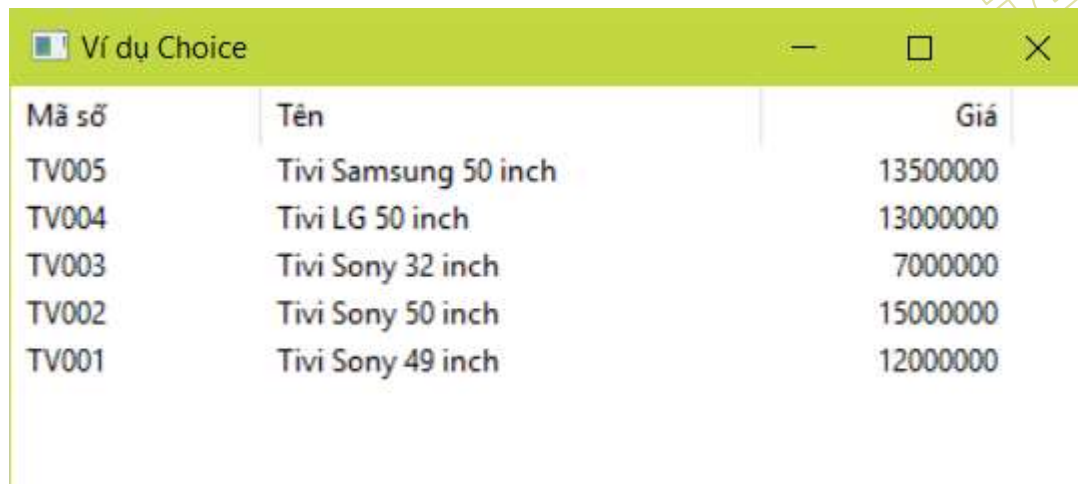
players = [('TV001', 'Tivi Sony 49 inch', '12000000'), ('TV002', 'Tivi Sony 50 inch', '15000000'), ('TV003', 'Tivi Sony 32 inch', '7000000'), ('TV004', 'Tivi LG 50 inch', '13000000'), ('TV005', 'Tivi Samsung 50 inch', '13500000')]

app = wx.App()
frame = wx.Frame(None, title="Ví dụ Choice", size=(450,200))
panel = wx.Panel(frame, -1)
list = wx.ListCtrl(panel, -1, style = wx.LC_REPORT, size=(450,200))
list.InsertColumn(0, 'Mã số', width = 100)
list.InsertColumn(1, 'Tên', wx.LIST_FORMAT_LEFT, 200)
list.InsertColumn(2, 'Giá', wx.LIST_FORMAT_RIGHT, 100)
index=0
for i in players:
    list.InsertStringItem(index, i[0])
    list.SetStringItem(index, 1, i[1])
    list.SetStringItem(index, 2, i[2])
frame.Show(True)
app.MainLoop()
```


Làm việc với wxPython Widgets

❑ wxListCtrl

- Ví dụ:



Mã số	Tên	Giá
TV005	Tivi Samsung 50 inch	13500000
TV004	Tivi LG 50 inch	13000000
TV003	Tivi Sony 32 inch	7000000
TV002	Tivi Sony 50 inch	15000000
TV001	Tivi Sony 49 inch	12000000

Làm việc với wxPython Widgets

❑ wxChoice

- **Cú pháp:** `wx.Choice(parent, id, pos=wx.DefaultPosition, size=wx.DefaultSize, choices=None, style=0, validator=wx.DefaultValidator, name="choice")`

Làm việc với wxPython Widgets

□ wxChoice

- Lấy giá trị đang chọn trong Choice dùng hàm **GetValue()**

TTTTH
Khoa Học Tự Nhiên

Làm việc với wxPython Widgets

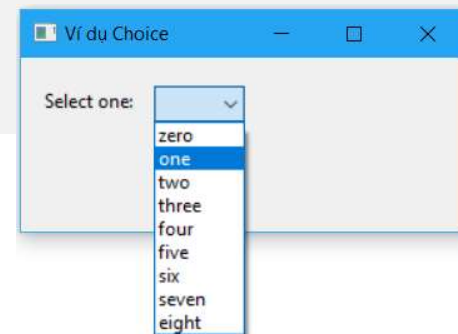
❑ wxChoice – Xử lý sự kiện

wx. COMBOBOX	Xảy ra khi chọn item trong danh sách
wx. EVT_TEXT	Xảy ra khi thay đổi giá trị trong combobox
wx. EVT_COMBOBOX_DROPDOWN Xảy ra khi danh sách thả xuống	
wx. EVT_COMBOBOX_CLOSEUP Xảy ra khi danh sách đóng lại	

Làm việc với wxPython Widgets

● Ví dụ:

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ Choice", size=(300,150))
panel = wx.Panel(frame, -1)
sampleList = ['zero', 'one', 'two', 'three', 'four', 'five', 'six',
              'seven', 'eight']
wx.StaticText(panel, -1, "Select one:", (15, 20))
wx.Choice(panel, -1, (85, 18), choices=sampleList)
frame.Show(True)
app.MainLoop()
```



Làm việc với wxPython Widgets

❑ **wxMessageBox** – Cú pháp

```
wx.MessageBox(message,  
caption=MessageBoxCaptionStr,  
style=OK|CENTRE,  
parent=None,  
x=DefaultCoord,  
y=DefaultCoord)
```

Kết quả trả về là một số int

Làm việc với wxPython Widgets

❑ wxMessageBox – Cú pháp

message (string)	Nội dung thông báo
caption (string)	Tiêu đề thông báo
style (int)	Kết hợp các cờ : wx.OK,wx.CANCEL, wx.YES_NO, wx.HELP, wx.NO_DEFAULT, wx.NO_DEFAULT, wx.CANCEL_DEFAULT, wx.YES_DEFAULT, wx.OK_DEFAULT, wx.ICON_NONE, wx.ICON_EXCLAMATION, wx.ICON_ERROR, wx.ICON_HAND, wx.ICON_QUESTION, wx.ICON_INFORMATION, ...

Không

Làm việc với wxPython Widgets

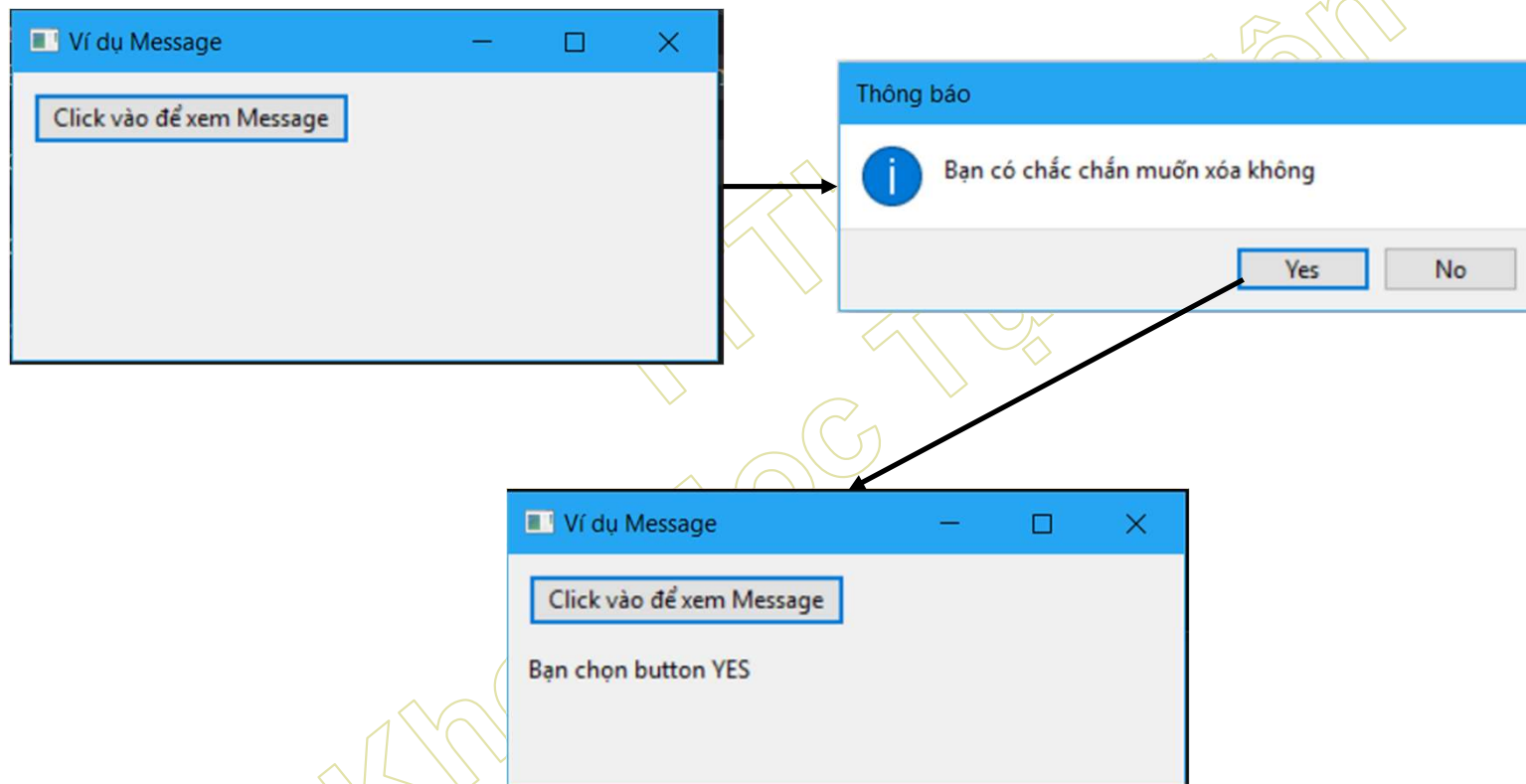
❑ wxMessageBox – Ví dụ

```
import wx
def OnXem(event):
    kq = wx.MessageBox("Bạn có chắc chắn muốn xóa không", "Thông báo", wx.YES_NO|wx.ICON_INFORMATION)
    if wx.YES==kq:
        stTraLoi.SetLabel("Bạn chọn button YES")
    else:
        stTraLoi.SetLabel("Bạn chọn button NO")

app = wx.App()
frame = wx.Frame(None, title="Ví dụ MessageBox", size=(300,150))
panel = wx.Panel(frame, -1)
btnChon = wx.Button(panel, -1, label="Click vào để xem Message", pos=(10,10))
frame.Bind(wx.EVT_BUTTON, OnXem, btnChon)
stTraLoi=wx.StaticText(panel, -1, pos=(10,50))
frame.Show(True)
app.MainLoop()
```


Làm việc với wxPython Widgets

❑ wxMessageBox – Ví dụ



Làm việc với wxPython Widgets

❑ wxImage – Ví dụ

Cú pháp:

```
wx.Image(name,  
         type=wx.BITMAP_TYPE_ANY,  
         index=-1)
```

- Trong đó:

name: Đường dẫn và tên file hình

Làm việc với wxPython Widgets

❑ wxImage

Type: Quy định kiểu định dạng ảnh, gồm:

wx.BITMAP_TYPE_JPEG	Ảnh jpg
wx.BITMAP_TYPE_PNG	Ảnh png
wx.BITMAP_TYPE_TIF	Ảnh tif
wx.BITMAP_TYPE_ANY	Sẽ cố gắng tự động phát hiện định dạng để sử dụng và sau đó sẽ gọi trình xử lý.

Làm việc với wxPython Widgets

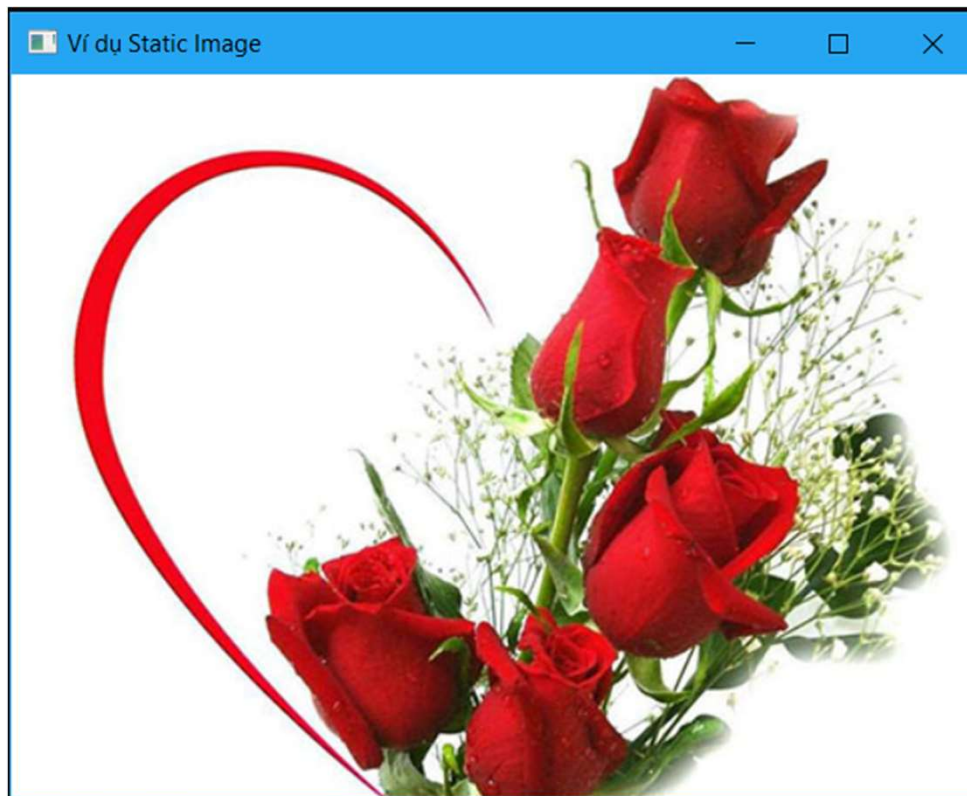
❑ wxImage – Ví dụ

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ ListBox", size=(500,400))
panel = wx.Panel(frame, -1)
hinh = wx.Image("Media/hoa_hong.jpg",
wx.BITMAP_TYPE_JPEG).ConvertToBitmap()
stStaticBitmal = wx.StaticBitmap(panel, -1, hinh)

frame.Show(True)
app.MainLoop()
```

Làm việc với wxPython Widgets

❑ wxImage – Ví dụ



Làm việc với wxPython Widgets

□ wxMenu:

Để sử dụng thanh menu, hãy thực hiện các tác vụ sau

- *Tạo MenuBar*
- *Gắn MenuBar vào frame*
- *Tạo các menu riêng lẻ*
- *Thêm các menu vào MenuBar hoặc vào menu cha*
- *Tạo các menu item riêng lẻ*
- *Thêm các menu item vào menu thích hợp*
- *Tạo một sự kiện ràng buộc cho mỗi menu item*

Làm việc với wxPython Widgets

□ wxMenu:

- Tạo menubar

- wx.MenuBar()

- *Tạo menu*

- *Wx.Menu*

- *Tạo menu item*

- *Wx.MenuItem*

Làm việc với wxPython Widgets




❑ wxMenubar – Một số phương thức

- Một số phương thức trên menubar

Append(menu, title)	Thêm menu vào cuối menu. Title sẽ được sử dụng để hiển thị thông số mới thực đơn. Trả về True nếu thành công, nếu không False.
Insert(pos, menu, title)	Chèn menu tại pos. Trả về True nếu thành công.
Remove(pos)	Xóa menu ở vị trí pos. Trả về menu bị xóa.
Replace(pos, menu, title)	Thay thế menu ở vị trí pos

Làm việc với wxPython Widgets

❑ wxMenuItem – Một số thuộc tính

wx.ITEM_NORMAL	Menu item thông thường	
wx.ITEM_CHECK	Menu dạng checkbox	
wx.ITEM_RADIO	Menu dạng radio	

Hàm **SetBitmap()**: Hiển thị ảnh bitmap

```
itemExit.SetBitmap(wx.Bitmap("Media/close.png"))
```

Làm việc với wxPython Widgets

❑ wxMenuItem

- Xử lý sự kiện: EVT_MENU

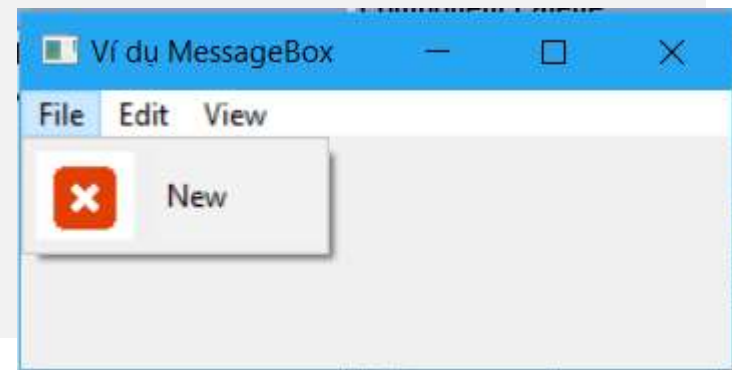
```
itemExit = wx.MenuItem(menu,wx.ID_NEW, text =  
    "New",kind = wx.ITEM_NORMAL)  
itemExit.SetBitmap(wx.Bitmap("Media/close.png"))  
menu.Append(itemExit)  
self.Bind(wx.EVT_MENU, self.OnExit, itemExit)  
def OnExit(self, event):  
    self.Close()
```

Làm việc với wxPython Widgets

❑ wxMenu – Ví dụ:

```
import wx
def OnExit(event):
    frame.Close()
app = wx.App()
frame = wx.Frame(None, title="Ví dụ MessageBox",size=(300,150))
panel = wx.Panel(frame, -1)
menuBar = wx.MenuBar()
menu = wx.Menu()
itemExit = wx.MenuItem(menu,-1, text = "New",kind = wx.ITEM_NORMAL)
itemExit.SetBitmap(wx.Bitmap("Media/close.png"))
menu.Append(itemExit)
menuBar.Append(menu, "&File")
menu2 = wx.Menu()
menuBar.Append(menu2, "&Edit")
menu3 = wx.Menu()
menuBar.Append(menu3, "&View")
frame.SetMenuBar(menuBar)
frame.Bind(wx.EVT_MENU, OnExit, itemExit)

frame.Show(True)
app.MainLoop()
```



Nội dung

1. Giới thiệu GUI
2. Làm việc với wxPython Widgets
- 3. Layout Management**
4. Giới thiệu phần mềm wxFormBuilder

wxPython - Layout Management

□ wxPython API cung cấp các lớp Layout để quản lý các vị trí của các widget bên trong container. Ưu điểm của Layout Management là -

- *Các tiện ích bên trong cửa sổ được tự động thay đổi kích thước.*
- *Đảm bảo sự xuất hiện đồng nhất trên các thiết bị hiển thị với độ phân giải khác nhau.*
- *Việc thêm hoặc xóa tiện ích động có thể thực hiện được mà không phải thiết kế lại.*

wxPython - Layout Management

- ❑ Layout Management được gọi là Sizer trong wxPython. Wx.Sizer là lớp cơ sở cho tất cả các lớp con sizer. Chúng ta hãy thảo luận về một số các sizers quan trọng như :

wx.BoxSizer, wx.StaticBoxSizer,
wx.GridSizer, wx.FlexGridSizer, và
wx.GridBagSizer.

wxPython - Layout Management

❑ wxBoxSizer

wxBoxSizer cho phép các điều khiển được sắp xếp theo dòng hoặc cột. Bố cục của BoxSizer được xác định bởi đối số định hướng của nó (wxVERTICAL hoặc wxHORIZONTAL).

```
Box = wx.BoxSizer(wxHORIZONTAL)
```

```
Box = wx.BoxSizer(wxVERTICAL)
```

wxPython - Layout Management

□ wxBoxSizer

Phương thức Add () thêm điều khiển vào hàng / cột tiếp theo của sizer.

```
Box.Add(control, proportion, flag, border)
```


wxPython - Layout Management

❑ wxBoxSizer

flag

`wx.ALIGN_TOP`

`wx.ALIGN_BOTTOM`

`wx.ALIGN_LEFT`

`wx.ALIGN_RIGHT`

`wx.ALIGN_CENTER_VERTICAL`

`wx.ALIGN_CENTER_HORIZONTAL`

border

`wx.TOP`

`wx.BOTTOM`

`wx.LEFT`

`wx.RIGHT`

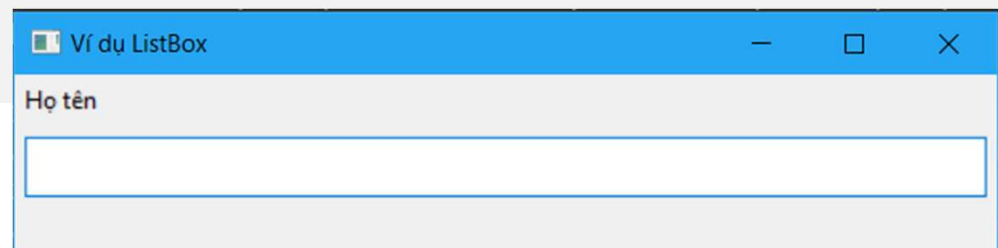
`wx.ALL`

wxPython - Layout Management

❑ wxBoxSizer – ví dụ

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ ListBox", size=(500,400))
panel = wx.Panel(frame, -1)
vbox = wx.BoxSizer(wx.VERTICAL)
stHoten = wx.StaticText(panel, label="Họ tên")
vbox.Add(stHoten, 0, wx.ALL|wx.EXPAND|wx.ALIGN_CENTER_HORIZONTAL, 5)
txtHoTen = wx.TextCtrl(panel, -1, size=(200,30))
vbox.Add(txtHoTen, 0, wx.ALL|wx.EXPAND|wx.ALIGN_CENTER_HORIZONTAL, 5)
panel.SetSizer(vbox)

frame.Show(True)
app.MainLoop()
```



wxPython - Layout Management

❑ wxGridSizer

Đối tượng **GridSizer** trình bày một lưới hai chiều. Các điều khiển được thêm vào trong khe lưới trong thứ tự từ trái sang phải và từ trên xuống dưới. Đối tượng GridSizer nhận bốn tham số -

```
wx.GridSizer(rows, columns, vgap, hgap)
```

thông số vgap và hgap kiểm soát khoảng cách dọc và ngang giữa các điều khiển liền kề.

wxPython - Layout Management - GridSizer

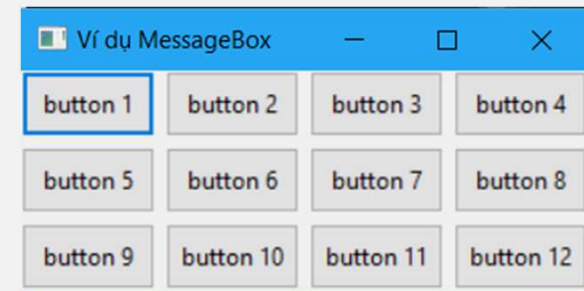
❑ Một số phương thức quan trọng

#	Phương thức và mô tả
1	Add() Thêm một điều khiển vào ô tiếp theo trong
2	SetRows() Cài đặt số dòng cho sizer
3	SetCols() Cài đặt số cột cho sizer
4	SetVGap() Đặt khoảng cách dọc (tính bằng pixel) giữa các ô
5	SetHGap() Đặt khoảng cách ngang (tính bằng pixel) giữa các ô

wxPython - Layout Management

❑ wxGridSizer – ví dụ

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ MessageBox",size=(300,150))
panel = wx.Panel(frame, -1)
gs = wx.GridSizer(3, 4, 5, 5)
for i in range(1,13):
    btn = "button "+str(i)
    gs.Add(wx.Button(panel,label = btn),0,wx.EXPAND)
panel.SetSizer(gs)
frame.Center(wx.BOTH)
frame.Show(True)
app.MainLoop()
```



wxPython - Layout Management -

❑ wxGridBagSizer

GridBagSizer là một sizer đa năng. Nó cung cấp nhiều cải tiến hơn. **Tiện ích con** có thể được thêm vào một ô cụ thể trong lưới. Hơn nữa, một widget con có thể chiếm nhiều hơn một ô theo chiều ngang và / hoặc theo chiều dọc. Do đó, các điều khiển trên nhiều dòng và nhiều cột có độ rộng và chiều cao khác nhau

wxPython - Layout Management -

❑ wxGridBagSizer

Lớp wx.GridBagSizer chỉ có một hàm tạo lấy hai đối số.

```
wx.GridBagSizer(vgap,hgap)
```

wxPython - Layout Management -

❑ wxGridBagSizer

Thêm điều khiển vào GridBagSizer

```
wx.GridbagSizer().Add(control, pos, span, flags,  
border)
```

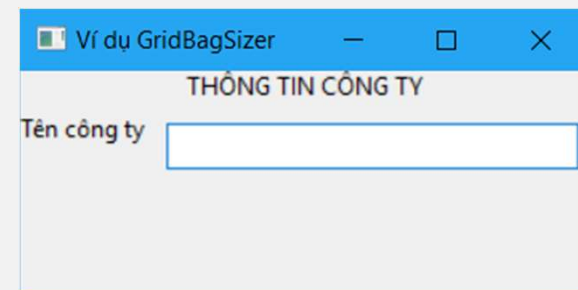
pos: là tuple cho biết vị trí dòng, cột

span: là tuple cho biết số dòng, số cột nối lại

wxPython - Layout Management -

❑ wxGridBagSizer – ví dụ

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ GridBagSizer", size=(300,150))
panel = wx.Panel(frame, -1)
wxGridBagSizer=wx.GridBagSizer(5,5)
stTitle=wx.StaticText(panel, -1, label="THÔNG TIN CÔNG TY")
wxGridBagSizer.Add(stTitle, pos=(0,0), span=(1,2),
flag=wx.ALIGN_CENTER_HORIZONTAL)
stTenCongTy=wx.StaticText(panel, -1, label="Tên công ty")
wxGridBagSizer.Add(stTenCongTy, pos=(1,0), span=(1,1))
txtTenCongTy=wx.TextCtrl(panel, -1)
wxGridBagSizer.Add(txtTenCongTy, pos=(1,1), span=(1,1), flag =
wx.EXPAND|wx.ALL, border = 5)
wxGridBagSizer.AddGrowableCol(1)
panel.SetSizerAndFit(wxGridBagSizer)
frame.Center(wx.BOTH)
frame.Center(wx.BOTH)
frame.Show(True)
app.MainLoop()
```



wxPython - Layout Management -

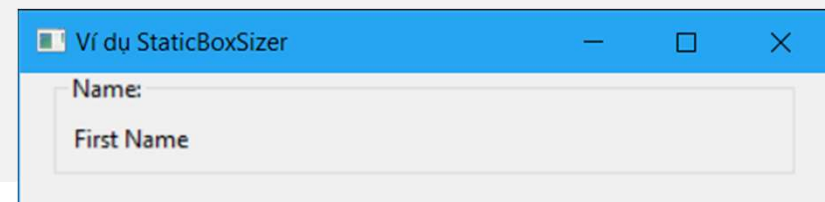
❑ wxStaticBoxSizer

- StaticBoxSizer Cung cấp một đường viền xung quanh hộp cùng với một label ở trên cùng. Cách tạo:
 - Tạo một đối tượng wx.StaticBox.
 - Khai báo một wx.StaticBoxSizer.
 - Tạo các điều khiển và thêm vào staticBoxSizer.

wxPython - Layout Management -

```
import wx
app = wx.App()
frame = wx.Frame(None, title="Ví dụ StaticBoxSizer", size=(420,300))
panel = wx.Panel(frame, -1)
vbox = wx.BoxSizer(wx.VERTICAL)
nm = wx.StaticBox(panel, -1, 'Name:')
nmSizer = wx.StaticBoxSizer(nm, wx.VERTICAL)
nmbox = wx.BoxSizer(wx.HORIZONTAL)
fn = wx.StaticText(panel, -1, "First Name", size=(350,-1))
nmbox.Add(fn, 0, wx.ALL|wx.CENTER, 5)
nmSizer.Add(nmbox, 0, wx.ALL|wx.CENTER, 0)
vbox.Add(nmSizer, 0, wx.ALL|wx.CENTER, 0)
panel.SetSizer(vbox)
frame.Centre()
panel.Fit()

frame.Show(True)
app.MainLoop()
```



Nội dung

1. Giới thiệu GUI
2. Làm việc với wxPython Widgets
3. Layout Management
4. Giới thiệu phần mềm wxFormBuilder

Làm việc với wxFormBuilder

- ❑ **wxFormBuilder** là chương trình xây dựng GUI WYSIWYG mã nguồn mở có thể dịch thiết kế GUI của wxWidget thành định dạng C ++, Python, PHP hoặc XML.

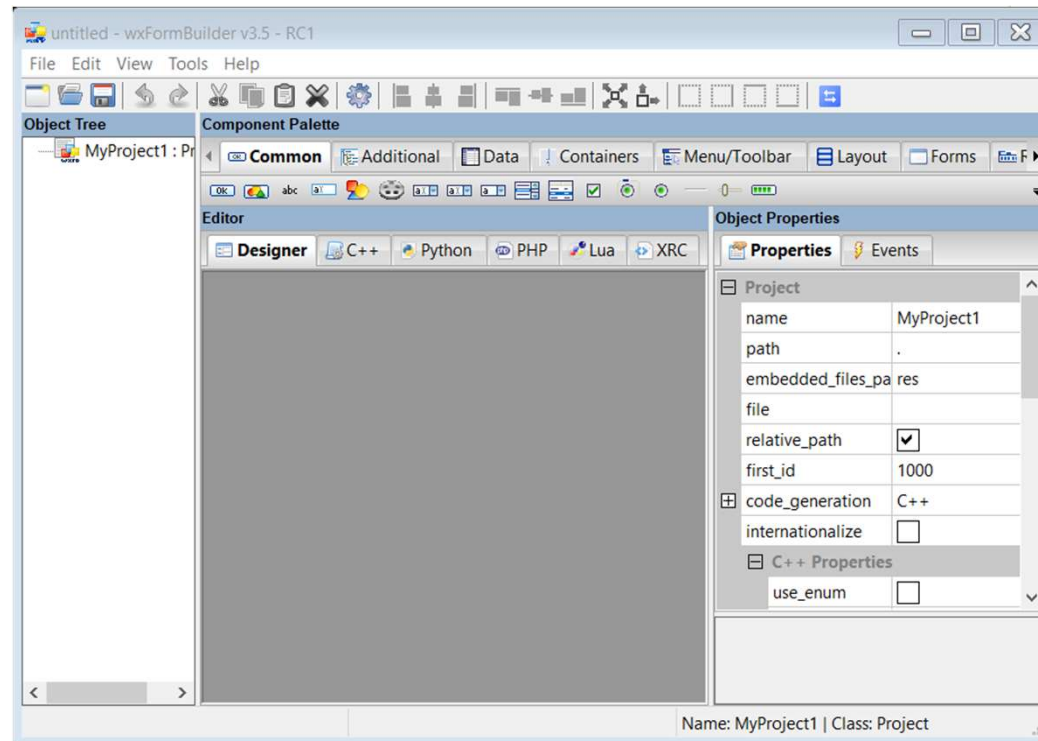
Làm việc với wxPython Widgets

- ❑ Đầu tiên cần phải được tải xuống và cài đặt từ

<http://sourceforge.net/projects/wxformbuilder/>.

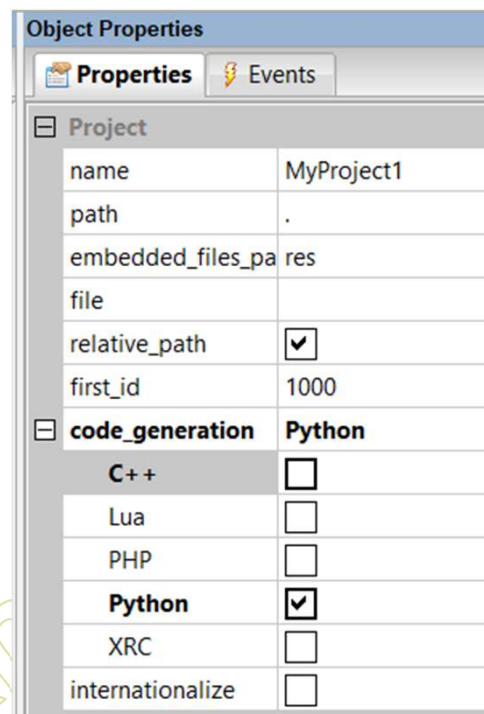
Làm việc với wxPython Widgets

- ❑ Khi mở ứng dụng, một dự án mới có vùng xám trống ở trung tâm sẽ xuất hiện.



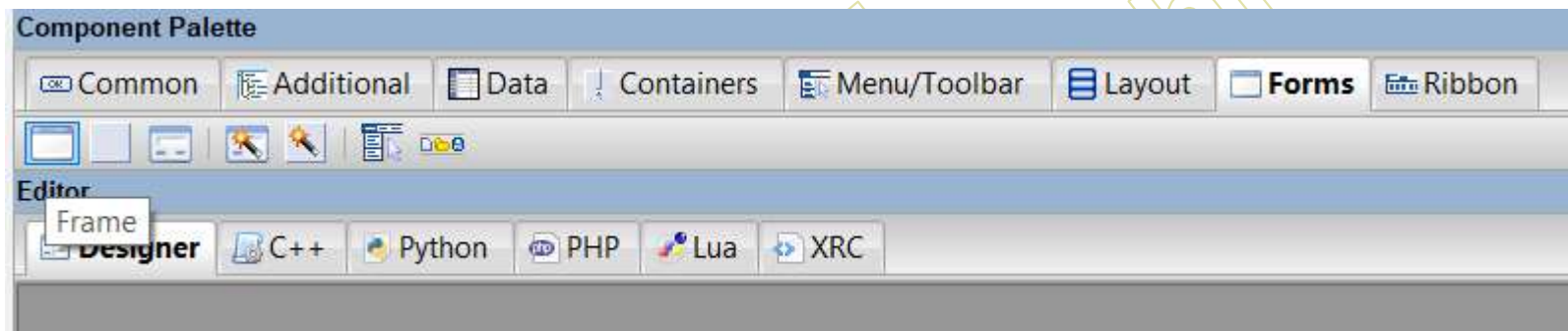
Làm việc với wxPython Widgets

- ❑ Đặt tên phù hợp cho dự án và chọn Python làm ngôn ngữ tạo mã. Điều này được thực hiện trong cửa sổ thuộc tính của đối tượng như trong hình dưới đây



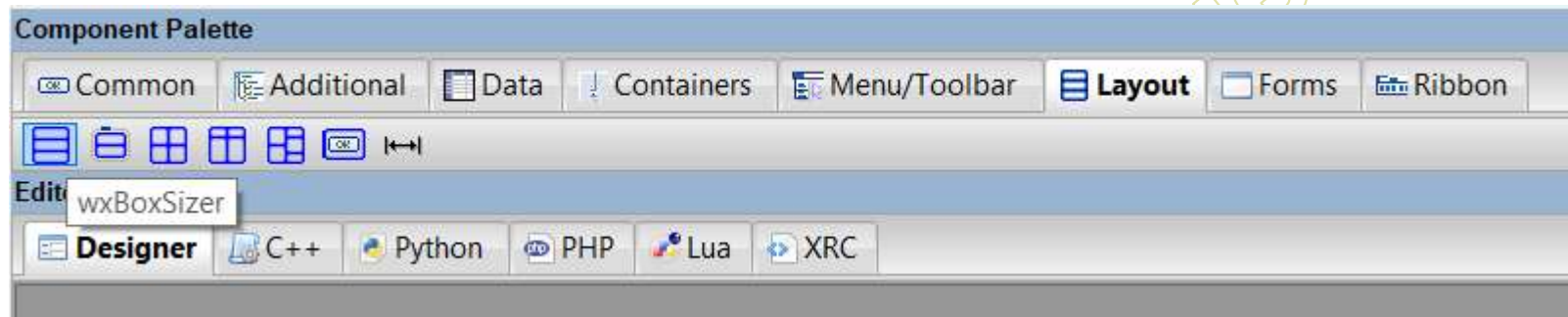
Làm việc với wxPython Widgets

- ❑ Sau đó, từ tab **Forms**, chọn **Frame**.



Làm việc với wxPython Widgets

- ❑ Thêm `wx...Sizer` từ tab **Layout**.



Làm việc với wxPython Widgets

- ❑ Thêm các điều khiển cần thiết trong Box với chú thích phù hợp để có giao diện như sau:

