

wxPython I

Introduction to GUI Programming

Sutee Sudprasert

Credit

- เนื้อหาของสไลด์นี้ลอกมาจาก PyCon 2010: wxPython โดย David Goodger
 - <http://python.mirocommunity.org/video/1618/pycon-2010-wxpython-i-introduc>
- หากใครฟังภาษาอังกฤษออก แนะนำให้ดูวิดีโอจากลิงค์ที่ให้ไว้

GUI

- Graphical User Interface

Even-Driven GUI Programming

- Low level event
 - mouse clicked, key pressed, program closed, computer shutdown
- High level event
 - button pressed, menu selected
- Low level event จะถูกเปลี่ยนเป็น high level event

Event Loop

```
while True:  
    event = wait_next_event()  
    handle(event)
```

- หยุดรอ event แต่ไม่หยุดการทำงานของ CPU

Toolkits Choices

- Tkinter
- PyQt
- PyGTK
- PyWin32
- PyObjC
- wxPython

wxPython

- ใช้ได้ข้ามระบบปฏิบัติการ
- ใช้ได้ฟรี (ตามสนธิสัญญา GPL)
- มีการพัฒนาอยู่เรื่อยๆ

Resources

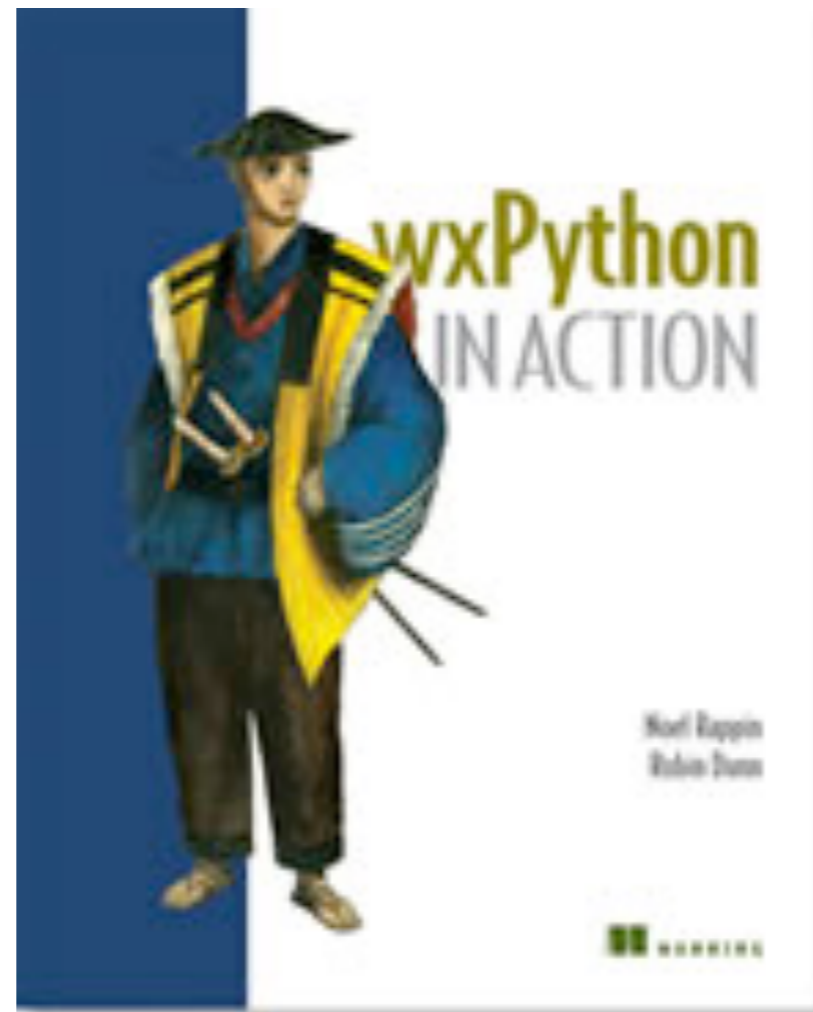
- wxPython docs
- *wxPython In Action* book
- wxPython demo application

wxPython docs

- wxWidgets docs (“wx”) - Online wxDocs
 - <http://www.wxpython.org/onlinedocs.php>
- wxPython API docs (“api”) - New wxPyDocs
 - <http://www.wxpython.org/docs/api/>

wxPython In Action

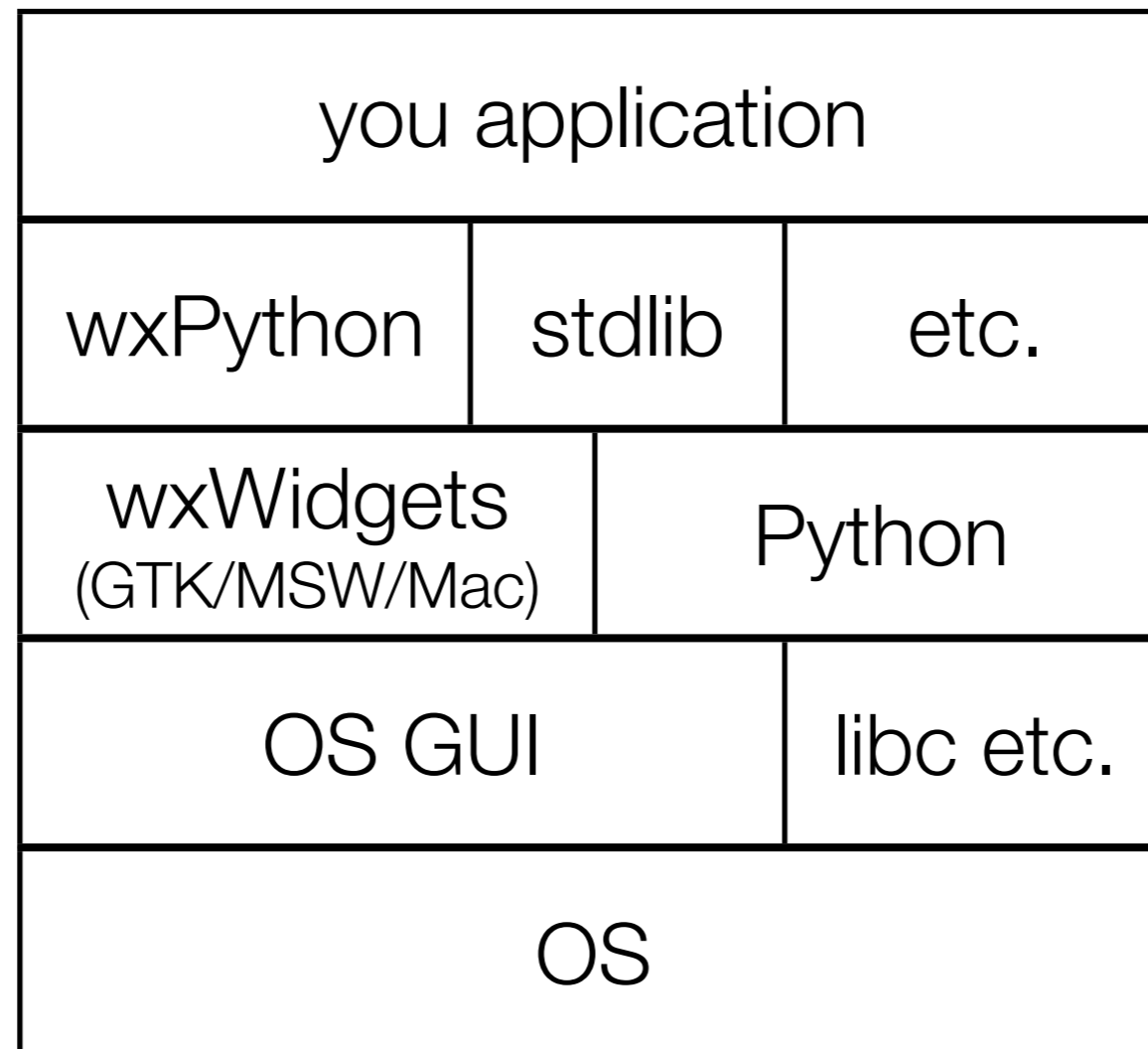
Noel Rappin and Robin Dunn



wxPython Demo

- ดาวนั้โหลดมาลงเพิ่มเติมได้จาก
 - <http://www.wxpython.org/download.php>

Architecture of a wx App



Why (not) use a GUI designer?

- Boa Constructor
- wxGlade

Common Widgets

- windows
- labels
- text fields
- pushbuttons
- radio buttons
- check boxes
- pick lists
- menus

Hello World

```
import wx

class App(wx.App):
    def OnInit(self):
        frame = wx.Frame(parent=None,
                          title="Hello World!")
        frame.Show()
        return True

app = App(redirect=False)
app.MainLoop()
```

Hello World 2

```
import wx

class App(wx.App):
    def OnInit(self):
        frame = wx.Frame(parent=None,
                          title="Hello World!")
        text = wx.StaticText(frame, -1, 'Howdy',
                              style=wx.ALIGN_CENTER)
        #text = wx.StaticText(frame, label='Howdy',
                              # style=wx.ACTION_CENTRE)
        font = wx.Font(72, wx.ROMAN, wx.ITALIC, wx.BOLD)
        text.SetFont(font)
        print 'in App.Oninit'
        frame.Show()
        return True

print 'toplevel code'
app = App(redirect=False)
app.MainLoop()
```


Hello World 3

```
#!/usr/bin/env python
import wx

class App(wx.App):
    def OnInit(self):
        self.frame = HelloFrame()
        self.frame.Show()
        self.SetTopWindow(self.frame)
        return True

class HelloFrame(wx.Frame):
    def __init__(self, parent=None, id=1, title="Hello World!"):
        wx.Frame.__init__(self, parent=parent, id=id, title=title)
        text = wx.StaticText(self, label='Howdy', style=wx.ALIGN_CENTER)
        font = wx.Font(72, wx.ROMAN, wx.ITALIC, wx.BOLD)
        text.SetFont(font)

def main():
    app = App(redirect=False)
    app.MainLoop()

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

Task 0

```
class App(wx.App):
    def OnInit(self):
        self.frame = TaskFrame()
        self.frame.Show()
        self.SetTopWindow(self.frame)
        return True

class TaskFrame(wx.Frame):
    def __init__(self, parent=None, id=-1, title='New Task'):
        wx.Frame.__init__(self, parent=parent, id=id, title=title)
        data_control = wx.DatePickerCtrl(self,
            style=wx.DP_DROPDOWN | wx.DP_SHOWCENTURY, pos=(10, 80))
        text = wx.StaticText(self, label='Howdy',
            style=wx.ALIGN_CENTER)
        font = wx.Font(72, wx.ROMAN, wx.ITALIC, wx.BOLD)
        text.SetFont(font)

def main():
    app = App(redirect=False)
    app.MainLoop()

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

Sizers

- GridSizer
- FlexGridSizer
- GridBagSizer
- BoxSizer
- StaticBoxSizer

Task 1 (BoxSizer)

```
class TaskFrame(wx.Frame):
    def __init__(self, parent=None, id=-1, title='New Task'):
        wx.Frame.__init__(self, parent=parent, id=id, title=title)
        sizer1 = wx.BoxSizer(orient=wx.VERTICAL)

        data_control = wx.DatePickerCtrl(self,
            style=wx.DP_DROPDOWN | wx.DP_SHOWCENTURY, pos=(10, 80))
        sizer1.Add(data_control, 1, wx.EXPAND | wx.ALL, border=5)
        text = wx.StaticText(self, label='Howdy',
            style=wx.ALIGN_CENTER)
        font = wx.Font(72, wx.ROMAN, wx.ITALIC, wx.BOLD)
        text.SetFont(font)
        sizer1.Add(text, 1, wx.EXPAND | wx.ALL, border=5)
        self.SetSizer(sizer1)
```

Task 2 (Even Handlers)

```
class TaskFrame(wx.Frame):
    def __init__(self, parent=None, id=-1, title='New Task'):
        wx.Frame.__init__(self, parent=parent, id=id, title=title)
        self.tasks = []
        sizer1 = wx.BoxSizer(orient=wx.VERTICAL)
        self.date_control = wx.DatePickerCtrl(self,
            style=wx.DP_DROPDOWN | wx.DP_SHOWCENTURY, pos=(10, 80))
        sizer1.Add(self.date_control, 0, wx.EXPAND | wx.ALL, border=5)

        self.desc = wx.TextCtrl(self, value='Howdy',
            style=wx.TE_PASSWORD)
        sizer1.Add(self.desc, 1, wx.EXPAND | wx.ALL, border=5)
        self.button = wx.Button(self, label='Create')
        self.button.Bind(wx.EVT_BUTTON, self.create_task)
        sizer1.Add(self.button, 0, wx.ALL, border=5)
        self.SetSizer(sizer1)

    def create_task(self, event):
        wx_date = self.date_control.Value
        date = datetime.date(wx_date.Year, wx_date.Month+1, wx_date.Day)
        desc = self.desc.Value
        self.tasks.append((date, desc))
```

Task 3 (Nested Sizers)

```
class TaskFrame(wx.Frame):
    def __init__(self, parent=None, id=-1, title='New Task'):
        wx.Frame.__init__(self, parent=parent, id=id, title=title)
        self.tasks = []
        sizer1 = wx.BoxSizer(orient=wx.VERTICAL)
        self.date_control = wx.DatePickerCtrl(self,
            style=wx.DP_DROPDOWN | wx.DP_SHOWCENTURY, pos=(10, 80))
        sizer1.Add(self.date_control, 0, wx.EXPAND | wx.ALL, border=5)

        self.desc = wx.TextCtrl(self, value='Howdy', style=wx.TE_CENTER)
        sizer1.Add(self.desc, 1, wx.EXPAND | wx.ALL, border=5)
        self.create_button = wx.Button(self, label='Create')
        self.create_button.Bind(wx.EVT_BUTTON, self.create_task)

        self.display_button = wx.Button(self, label='Display')
        self.display_button.Bind(wx.EVT_BUTTON, self.display_tasks)

        sizer2 = wx.BoxSizer(orient=wx.HORIZONTAL)
        sizer2.Add(self.create_button, 1, wx.EXPAND | wx.ALL, border=3)
        sizer2.Add(self.display_button, 1, wx.EXPAND | wx.ALL, border=3)

        sizer1.Add(sizer2, 0, wx.ALL | wx.ALIGN_RIGHT, border=5)
        self.SetSizer(sizer1)
```

Task 4 (Display Tasks)

```
class TaskFrame(wx.Frame):
    headers = ('Date', 'Description')
    ...
    def display_tasks(self, event):
        task_list = TaskList(self.headers, self.tasks)
        task_list.Show()

class TaskList(wx.Frame):
    def __init__(self, headers, data, parent=None, id=-1, title='Tasks'):
        wx.Frame.__init__(self, parent=parent, id=id, title=title)
        self.list = wx.ListCtrl(self,
                                style=wx.LC_REPORT | wx.LC_HRULES | wx.LC_VRULES)
        self.set_headers(headers)
        self.set_data(data)

    def set_headers(self, headers):
        for i, header in enumerate(headers):
            self.list.InsertColumn(i, header)

    def set_data(self, data):
        self.list.DeleteAllItems()
        for index, record in enumerate(data):
            retval = self.list.InsertStringItem(index=self.list.GetItemCount(),
                                                label=str(record[0]))
            self.list.SetStringItem(retval, col=1, label=record[1])
```

Task 5 (Popup menu)

```
class TaskList(wx.Frame):
    def __init__(self, headers, data, parent=None, id=-1, title='Tasks'):
        ...
        self.list.Bind(wx.EVT_RIGHT_UP, self.on_right_click)
        self.list.Bind(wx.EVT_COMMAND_RIGHT_CLICK, self.on_right_click)

    def on_right_click(self, event):
        if not hasattr(self, "popup_edit_id"):
            self.popup_edit_id = wx.NewId()
            self.popup_delete_id = wx.NewId()
            self.Bind(wx.EVT_MENU, self.edit, id=self.popup_edit_id)
            self.Bind(wx.EVT_MENU, self.delete, id=self.popup_delete_id)
            selected_item = self.list.GetNextItem(-1, wx.LIST_NEXT_ALL,
                                                wx.LIST_STATE_SELECTED)

            if selected_item != -1:
                menu = wx.Menu()
                menu.Append(self.popup_edit_id, "Edit")
                menu.Append(self.popup_delete_id, "Delete")
                self.PopupMenu(menu)
                menu.Destroy()

        def delete(self, event):
            selected_item = self.list.GetNextItem(-1, wx.LIST_NEXT_ALL,
                                                wx.LIST_STATE_SELECTED)

            self.list.DeleteItem(selected_item)
```


Task 6 (Dialog)

```
class TaskList(wx.Frame):
    ...
    def edit(self, event):
        selected_item = self.list.GetNextItem(-1,
            wx.LIST_NEXT_ALL, wx.LIST_STATE_SELECTED)
        dlg = wx.TextEntryDialog(self, 'Description',
            'Edit Description')
        dlg.SetValue(tasks[selected_item][1])

        if dlg.ShowModal() == wx.ID_OK:
            tasks[selected_item][1] = dlg.GetValue()
            self.list.SetStringItem(selected_item, col=1,
                label=dlg.GetValue())

        dlg.Destroy()
```

Task 7 (Custom dialog)
