wxPython in Action

NOEL RAPPIN
ROBIN DUNN

MANNING
Greenwich
(74° w. long.)
PART 1 INTRODUCTION TO WXPYTHON ........................................... 1

1 Welcome to wxPython 3
1.1 Getting started with wxPython 5
1.2 Creating the bare-minimum wxPython program 7
   Importing wxPython 9 • Working with applications
   and frames 11
1.3 Extending the bare-minimum
   wxPython program 12
1.4 Creating the final hello.py program 15
1.5 What can wxPython do? 17
1.6 Why choose wxPython? 19
   Python programmers 19 • wxWidget users 20 • New users 20
1.7 How wxPython works 21

The Python language 21 • The wxWidgets toolkit 22
Putting it together: the wxPython toolkit 25

1.8 Summary 27

2 Giving your wxPython program a solid foundation 29

2.1 What do I need to know about the required objects? 30

2.2 How do I create and use an application object? 31

Creating a wx.App subclass 31 • Understanding the application object lifecycle 34

2.3 How do I direct output from a wxPython program? 35

Redirecting output 35 • Modifying the default redirect behavior 37

2.4 How do I shut down my wxPython application? 38

Managing a normal shutdown 38 • Managing an emergency shutdown 39

2.5 How do I create and use the top-level window object? 39

Working with wx.Frame 40 • Working with wxPython IDs 42
Working with wx.Size and wx.Point 43 • Working with wx.Frame styles 44

2.6 How do I add objects and subwindows to a frame? 47

Adding widgets to a frame 47 • Adding a menubar, toolbar, or status bar to a frame 49

2.7 How can I use common dialogs? 51

2.8 What are some common errors with application objects and frames? 53

2.9 Summary 54

3 Working in an event-driven environment 56

3.1 What terminology do I need to understand events? 57

3.2 What is event-driven programming? 58

Coding event handlers 60 • Designing for event-driven programs 61 • Event triggers 62

3.3 How do I bind an event to a handler? 63

Working with the wxEvtHandler methods 65
3.4 How are events processed by wxPython? 68
   Understanding the event handling process 69 • Using the Skip() method 75
3.5 What other event properties are contained in the application object? 77
3.6 How can I create my own events? 77
   Defining a custom event for a custom widget 78
3.7 Summary 81

4 Making wxPython easier to handle with PyCrust 83
4.1 How do I interact with a wxPython program? 84
4.2 What are the useful features of PyCrust? 86
   Autocompletion 87 • Calltips and parameter defaults 88
   Syntax highlighting 89 • Python help 90 • Command recall 91 • Cut and paste 92 • Standard shell environment 93 • Dynamic updating 94
4.3 What do the PyCrust notebook tabs do? 95
   Namespace tab 95 • Display tab 97 • Calltip tab 97
   Session tab 98 • Dispatcher tab 98
4.4 How can I wrap PyCrust around my wxPython application? 99
4.5 What else is in the Py package? 104
   Working with the GUI programs 104 • Working with the support modules 105
4.6 How can I use modules from the Py package in my wxPython programs? 112
4.7 Summary 115

5 Creating your blueprint 116
5.1 How can refactoring help me improve my code? 117
   A refactoring example 118 • Starting to refactor 121
   More refactoring 122
5.2 How do I keep the Model and View separate in my program? 126
   What is a Model-View-Controller system? 126 • A wxPython model: PyGridTableBase 128 • A custom model 136
5.3 How do you unit-test a GUI program? 140
  The unittest module 140 • A unittest sample 141
  Testing user events 143

5.4 Summary 145

6 Working with the basic building blocks 146

6.1 Drawing to the screen 148
  How do I draw on the screen? 148

6.2 Adding window decorations 155
  How do I add and update a status bar? 155 • How do I include
  a submenu or checked menu? 158 • How do I include
  a toolbar? 161

6.3 Getting standard information 165
  How do I use standard file dialogs? 165 • How do I use a
  standard color picker? 169

6.4 Making the application look nice 170
  How do I lay out widgets? 170 • How do I build an
  about box? 178 • How do I build a splash screen? 180

6.5 Summary 181

PART 2 ESSENTIAL WXPYTHON................................. 183

7 Working with the basic controls 185

7.1 Displaying text 186
  How do I display static text? 186 • How can I get the user to
  enter text? 189 • How do I change the text without
  user input? 192 • How do I create a multi-line or styled
  text control? 193 • How do I create a font? 196 • Can I have
  styled text if my platform doesn’t support rich text? 197 • What if
  my text control doesn’t match my string? 198 • How do I respond
  to text events? 199

7.2 Working with buttons 199
  How do I make a button? 200 • How do I make a button with
  a bitmap? 201 • How do I create a toggle button? 202
  What’s a generic button, and why should I use one? 203
7.3 Entering and displaying numbers 205

How do I make a slider? 205 • How can I get those neat up/down arrow buttons? 208 • How can I make a progress bar? 210

7.4 Giving the user choices 211

How do I create a checkbox? 211 • How can I create a group of radio buttons? 212 • How can I create a list box? 216 • Can I combine a checkbox and a list box? 219 • What if I want a pull-down choice? 220 • Can I combine text entry and a list? 221

7.5 Summary 222

8 Putting widgets in frames 224

8.1 The life of a frame 225

How do I create a frame? 225 • What are some different frame styles? 227 • How do I create a frame with extra style information? 230 • What happens when I close a frame? 232

8.2 Using frames 234

What are the methods and properties of wx.Frame? 234

How do I find a subwidget of a frame? 237 • How do I create a frame with a scrollbar? 238

8.3 Alternative frame types 242

How do I create an MDI frame? 242 • What's a mini-frame and why would I use it? 244 • How do I make a non-rectangular frame? 245 • How can I drag a frame without a title bar? 248

8.4 Using splitter windows 250

Creating a splitter window 250 • A splitter example 251

Changing the appearance of the splitter 253 • Manipulating the splitter programmatically 254 • Responding to splitter events 255

8.5 Summary 256

9 Giving users choices with dialogs 258

9.1 Working with modal dialogs 259

How do I create a modal dialog? 259 • How do I create an alert box? 261 • How do I get short text from the user? 264
How can I display a list of choices in a dialog? 266 • How can I display progress? 267

9.2 Using standard dialogs 269

How can I use a file picker? 269 • How can I use a font picker? 273 • How can I use a color picker? 275
Can I allow the user to browse images? 277

9.3 Creating a wizard 278

9.4 Showing startup tips 281

9.5 Using validators to manage data in a dialog 282

How do I use a validator to ensure correct data? 282 • How do I use a validator to transfer data? 286 • How do I validate data as it is entered? 288

9.6 Summary 291

Creating and using wxPython menus 293

10.1 Creating Menus 294

How do I create a menu bar and attach it to a frame? 295
How do I create a menu and attach it to the menu bar? 295
How do I add items to a pull-down menu? 297 • How do I respond to a menu event? 301

10.2 Working with menu items 303

How do I find a specific menu item in a menu? 303 • How do I enable or disable a menu item? 306 • How do I associate a menu item with a keyboard shortcut? 307 • How do I create a toggle menu item with a checkbox or radio button? 311

10.3 Sprucing up your menus 313

How do I create a submenu? 313 • How do I create a pop-up menu? 315 • How can I create fancier menus? 317

10.4 Usability guidelines for menus 319

Keeping menus uniform in length 319 • Creating logical item groups 319

10.5 Summary 321

Placing widgets with sizers 323

11.1 What’s a sizer? 324

11.2 Basic sizers with the grid sizer 326

What is a grid sizer? 327 • How do you add or remove children from a sizer? 329 • How do sizers manage the size and
alignment of their children? 331 * Can I specify a minimum size for my sizer or its children? 334 * How do sizers manage the border around each child? 336

11.3 Using the other sizer types 337

What's a flex grid sizer? 337 * What's a grid bag sizer? 341
What's a box sizer? 345 * What's a static box sizer? 349

11.4 Can I see a real-world example of sizers in action? 350

11.5 Summary 354

**12 Manipulating basic graphical images** 356

12.1 Working with images 357


12.2 Dealing with device contexts 367

What is a device context, and how can I create one? 367
How do I draw to a device context? 371 * How do I draw images to the context? 376 * How can I draw text to the context? 379

12.3 Graphics manipulation 381

How do I manage the foreground drawing pen? 381
How do I manage the background drawing brush? 384
How can I manage logical and physical device coordinates? 385
What color names are predefined? 387

12.4 Summary 388

**PART 3 ADVANCED WXPYTHON** ........................................391

**13 Building list controls and managing items** 393

13.1 Building a list control 394

What is icon mode? 394 * What is small icon mode? 395
What is list mode? 396 * What is report mode? 397
How do I create a list control? 398

13.2 Managing items in a list 400

What is an image list and how do I add images to it? 400
How can I add and delete items from a list? 402

13.3 Responding to users 405

How can I respond to a user selection in a list? 405 * How can I respond to a user selection in a column header? 407
13.4 Editing and sorting list controls 411
   How can I edit a label? 411 • How can I sort my list? 413
   How can I learn more about list controls? 416
13.5 Creating a virtual list control 420
13.6 Summary 423

14 Coordinating the grid control 425
14.1 Creating your grid 426
   How do I create a simple grid? 426 • How do I create a grid with
   a grid table? 429
14.2 Working with your grid 432
   How do I add and delete rows, columns, and cells? 432
   How do I manage the row and column headers of a grid? 433
   How can I manage the size of grid elements? 436
   How can I manage which cells are selected or visible? 440
   How do I change the color or font of a grid cell? 442
14.3 Custom renderers and editors 445
   How do I use a custom cell renderer? 445 • How do I edit
   a cell? 449 • How do I use a custom cell editor? 450
14.4 Capturing user events 455
   How can I capture user mouse selections? 455 • How can I
   capture user keyboard navigation? 457
14.5 Summary 458

15 Climbing the tree control 460
15.1 Creating tree controls and adding items 461
   How do I add a root? 463 • How do I add more items
to the tree? 463 • How do I manage items? 464
15.2 What styles control the display of the tree control? 465
15.3 Sorting elements of a tree control 467
15.4 Controlling the image for each item 468
15.5 Navigating the tree programmatically 471
15.6 Managing the tree selection 472
15.7 Controlling which items are visible 473
15.8 Making a tree control user editable 477
15.9 Responding to other user events from a tree control 478
15.10 Using a tree list control 480
15.11 Summary 482

16 Incorporating HTML into your application 485
16.1 Displaying HTML 486
   How can I display HTML in a wxPython window? 486
   How can I display HTML from a file or URL? 488
16.2 Manipulating the HTML window 490
   How can I respond to a user click on an active link? 490
   How can I change an HTML window programmatically? 491
   How can I display the page title in a frame's title bar? 493
   How can I print an HTML page? 495
16.3 Extending the HTML window 496
   How does the HTML parser work? 496
   How can I add support for new tags? 498
   How can I support other file formats? 501
   How can I get a more fully featured HTML Widget? 502
16.4 Summary 503

17 The wxPython printing framework 504
17.1 How do I print in wxPython? 505
   Understanding the printout lifecycle 506
   Print framework in action 507
   Working with wx.Printout methods 511
17.2 How do I display the print dialog? 512
   Creating a print dialog 512
17.3 How do I display the page setup dialog? 515
   Creating a page setup dialog 515
   Working with page setup properties 516
17.4 How do I print something? 518
17.5 How can I perform a print preview? 519
17.6 Summary 520

18 Using other wxPython functionality 521
18.1 Putting objects on the clipboard 522
   Getting data in the clipboard 522
   Manipulating data in the clipboard 523
   Retrieving text data from the
18.2 Being the source of a drag and drop  527
  Dragging in action  529
18.3 Being the target of a drag and drop  530
  Using your drop target  531  • Dropping in action  533
18.4 Transferring custom objects  534
  Transferring a custom data object  534  • Retrieving a custom object  535  • Transferring an object in multiple formats  535
18.5 Setting timed events using wx.Timer  536
  Generating EVT_TIMER events  536  • Learning other timer uses  539
18.6 Creating a multithreaded wxPython application  539
  Working with the global function wx.CallAfter()  540
  Managing thread communication with the queue object  543
  Developing your own solution  543
18.7 Summary  544
index  545