Here's a simple wxPython program. It creates a window with one text box that displays the position of the mouse pointer. Counting white space, it's about 20 lines long.

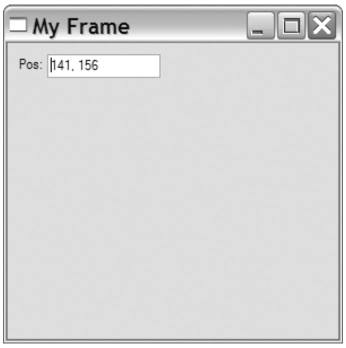
Listing 1.1 A working wxPython program in a mere 20 lines

```
#!/bin/env python
import wx
class MyFrame(wx.Frame):

    def __init__(self):
        wx.Frame.__init__(self, None, -1, "My Frame", size=(300, 300))
        panel = wx.Panel(self, -1)
        panel.Bind(wx.EVT_MOTION,  self.OnMove)
        wx.StaticText(panel, -1, "Pos:", pos=(10, 12))
        self.posCtrl = wx.TextCtrl(panel, -1, "", pos=(40, 10))

    def OnMove(self, event):
        pos = event.GetPosition()
        self.posCtrl.SetValue("%s, %s" % (pos.x, pos.y))

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



Pos: 141, 156

Figure 1.1 Our first wxPython program, showing the position of the mouse

What can we say about the program in listing 1.1? It's very short, for one thing. Admittedly, it doesn't do a whole lot, but still, creating a window, populating it, getting it to respond to mouse events—that's not bad for 20 lines. It's not an exaggeration to say this example could easily be three or four times longer in some, more caffeinated, programming languages. Figure 1.1 shows the running program.

The code sample is quite readable. Even if you don't know the details of Python or wxPython, if you have any experience with interface programming you likely have a sense of what words like `Frame`, `__init__`, `EVT_MOTION`, `TextCtrl`, and `MainLoop` mean. The indentation might seem a bit weird if you aren't used to Python (where are all those closing braces, anyway?), and you probably don't know what all the arguments mean (what's with those `-1`s?), but