

# Ereignissteuerung

Ereignissteuerung  
und  
Ereignisbehandlung

# Ereignissteuerung

- Ereignissteuerung ist eines der wichtigsten Paradigmen der OO Modellierung und Programmierung.
- In unseren bisherigen Projekten wird ein App-Objekt (abgeleitet von wx.App) erzeugt
- und dann davon die MainLoop gestartet.

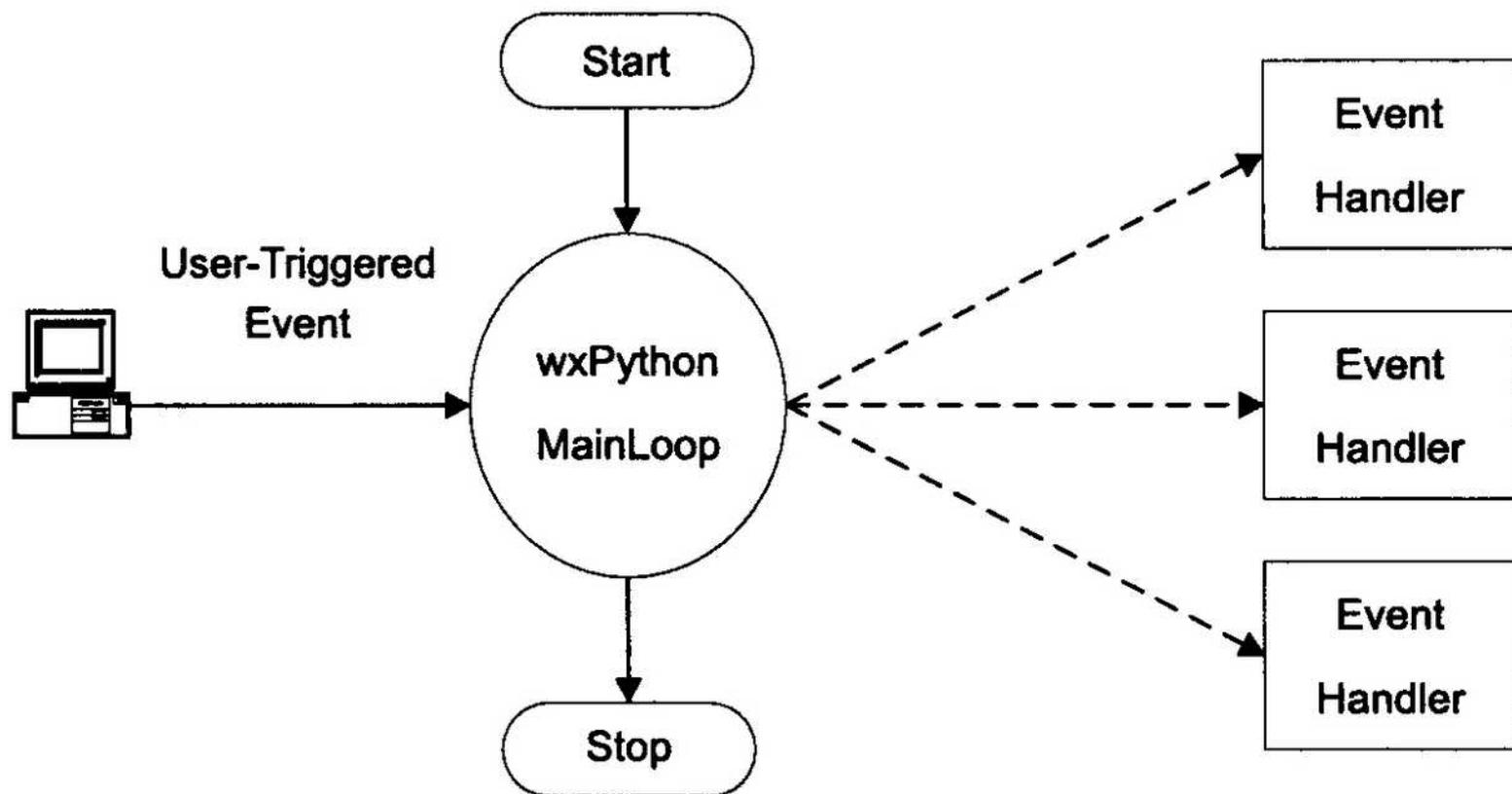
```
if __name__ == '__main__':  
    app = RaumplanerApp(redirect=False)  
    app.MainLoop()
```

# Ereignissteuerung

- Nun wartet das Programm auf Ereignisse, die es betreffen.
- Solche Ereignisse betrafen bisher hauptsächlich Eingaben in der PyShell, die wir mit ENTER abgeschlossen haben.
- Mit dem Einsatz einer grafischen Benutzeroberfläche (Gui) ersetzen wir die Interaktionen über die PyShell durch neue Interaktionen, die von der Gui ausgehen sollen.

# Ereignissteuerung

- Bild aus dem Buch von Robin Dunn



# Ereignissteuerung

- Damit ein eingebauter Button oder ein Menüpunkt (*beispielsweise*) eine Ereignisbehandlung auslösen kann, brauchen wir zwei Schritte
  - die Definition einer Methode zur Behandlung des Ereignisses
  - eine Verknüpfung dieser Methode mit dem Ereignis

# Ereignissteuerung

- Im Beispiel eines „Beenden“-Buttons:

Im Konstruktor:

```
button = wx.Button(panel, 1003, "Beenden")  
self.Bind(wx.EVT_BUTTON,  
           self.OnBeenden,  
           button)
```

die Methode:

```
def OnBeenden(self, event):  
    self.Close(True)
```

# Ereignissteuerung

Kein EVA !

- „***event-driven programming***“ ist ein grundsätzlich anderes Konzept als das traditionelle „***EVA-Prinzip***“ (***Eingabe-Verarbeitung-Ausgabe***)
  - bei dem ein Anwender einen Bearbeitungsprozess anstößt und
  - dann darauf wartet, dass dieser Bearbeitungsprozess fertig ist und
  - er eine Ausgabe erhält.

# Ereignissteuerung

Warten bei der  
Ereignis-gesteuerten Programmierung:

- Das Programm reagiert auf Ereignisse, die für es relevant sind
  - es reagiert dann mit einer Bearbeitung,
  - die schließlich zu einer Ausgabe führen *kann*.
- Tritt kein Ereignis auf, wartet das Programm.
  - Das geschieht für den Benutzer scheinbar ohne zu arbeiten.

# Ereignissteuerung

## Inaktivität?

- Während das Programm scheinbar ohne Aktivität ist, laufen im Hintergrund Prozesse ab, die man bildlich mit „*Augen und Ohren offen halten*“ bezeichnen könnte:
- Das Programm muss periodisch prüfen, ob eines der relevanten Ereignisse auftritt (aufgetreten ist).

# Ereignissteuerung

Für diesen Vorgang gibt es zwei Varianten, von denen die erste durch wxPython eingesetzt wird:

- Alle auftretenden Ereignisse werden in eine programmweite Warteschlange gestellt und von dort aus nacheinander deren Verarbeitung angestoßen.
- Alternativ fragt das Programm mögliche Quellen von Ereignissen periodisch ab, ob sie etwas zu melden haben.