

# MacPython

Jack Jansen

Centrum voor Wiskunde en Informatica

Jack.Jansen@cwi.nl

# About this talk

- Introduction
- Past – MacPython-OS9
- Present – MacPython-OSX
- **Rant on Open Source software**
- Future – World Domination

# About me

Active Python user since 1992,  
“responsible” for MacPython since  
1995, when Guido got rid of his  
Mac.

Work at CWI on multimedia  
research, active in SMIL standard,  
did SMIL editor/player in Python.

# History: before OSX

- Main distribution: binary
- included IDE by Just van Rossum:
  - editor, debugger, profiler, etc
- Applets: mini-applications that take the place of scripts
- true standalone applications

# MacOS technologies

- MacOS Toolbox a.k.a. Carbon:
  - Windows, Dialogs, Menus, Events, Fonts, Resources, etc
- QuickTime
- Open Scripting Architecture (“AppleScript”)
- 90% automatic with bgen

# Unix emulation

Matthias Neeracher's GUSI provides sockets, unix-like file descriptors, select, Pthreads and more

It is difficult to overstate its importance!

# Batteries Included!

- Python Imaging Library
- Numerical Python
- Img
- Tkinter with embedded Tcl/Tk

# Why include batteries?

- Mac users don't have compilers
  - ... and if they had they wouldn't know how to use them
    - ... and GUSI makes life even more difficult
- Mac users expect software to “just work”



# Community

- Very small number of active developers: between 2 and 25
- Active user community: about 300?
- Users: 10K downloads per release
  - where are they? what are they doing??!?

# MacPython-OSX

- Based on perfectly normal unix Python 2.3
  - but: use MacOSX framework
- Keeps IDE, Carbon, Quicktime, OSA, applets
- Cocoa support through PyObjC
- Batteries no longer included

# MacOSX frameworks

- Packaging construct to allow easy install/removal
- Framework is really “shared library on steroids”:
  - library, resources, header files, documentation, auxiliary programs

# Why use framework?

- Easy uninstall: remove framework and application only
- Some location independence while still allowing sharing
- Tcl/Tk does it too:-)

# Batteries not included!

- Many more useful packages for OSX than OS9
  - Often more solutions: Tkinter versus wxPython
  - too much work
- Many more developers who contribute
- Enter PackageManager!

# Package Manager

- “Package Install Manager for Python” was voted down:-(
- Central concept: single responsible person(s), the scapegoat
- On-line database with packages that are tested and tried
  - both source and binary

- Does not require developer tools for end user

- Can update itself

  - ... except for its GUI:-)

- Allows decentralization

- Diminishes rush when 2.X comes out

- A PEP will follow

# Distribution

- Binary installer is essential for Mac
  - no dependency on Developer Tools!
  - no Terminal window, please!
- But do cater for developers too
  - and try to allow for a shallow learning curve...





# Ⓢ Open Source

- theory

*you may download source*

*you may rebuild it yourself*

*you may even fix bugs*





# Ⓢ Open Source

- practice

*you will download source*

*you shall rebuild it yourself*

*you must even fix bugs*





- *Python core is pretty good:*
  - *few dependencies on third party packages*
  - *build process tested and tried*
- *Extensions can be bad (no offense meant!)*
  - *Small number of developers*
  - *long time between releases*



# Future - the easy bits

- Get rid of clunkyness
  - Solution: PyObjC
- Make GUI development easier
  - Solution: PyObjC
- Does not have to wait for 2.4

# World Domination - part one

- Python is better than AppleScript
  - so let's make it as user-friendly
- Python is better than RealBasic or Hypercard
  - So let's come up with a decent entry-level GUI builder

- Python is better than ObjC

- so show this to the ObjC community

- Python is great for CGI

- so flatten the learning curve

# World domination - Part two

- Mac is better than Unix
  - so make MacPython better than `/usr/local/bin/python`
    - for unix-heads, that is!
- Mac is better than Windows
  - ... aww, forget about them:-)

# World domination - Finale

- Draw attention
  - Reviews, magazine articles
- Easy access for testing the waters
  - Inclusion on CDs
  - Run from CD without installation



# But seriously:

- For end-users beat the ease of use of good shrink-wrapped software
- think of YOURSELF! This presentation was finished 4 hours after first contact with Keynote
  - Try that with  $\text{LaT}_e\text{X}$ : -)
- But don't get in the way of the developers...

42