

Mobiltelefonprogrammierung mit Python



Python für die S60 Plattform

Dr. Philippe Morath
pm@philippemorath.info

Python Stammtisch, 08. Mai 2007

Inhalt

Mobiltelefone & S60 Plattform

Mobiltelefonprogrammierung

Python S60 auf dem Mobiltelefon

Python S60 im Emulator

Einfache Beispiele

PyS60 API

Zusammenfassung

S60 Plattform (1)

Betriebssystem Symbian, lizenziert an



Nokia



Samsung



LG



lenovo

Über 100 Millionen S60-Mobiltelefone ausgeliefert

Nokia-Pressemitteilung vom 25. April 2007

57 Geräte, davon 28 basierend auf S60
3rd Edition

S60 Plattform (2)

1st Edition (Symbian OS 6.1)

2nd Edition (Symbian OS 7.0): Java MIDP 2

3rd Edition (Symbian OS 9.1): Keine
Binärkompatibilität zu 1st und 2nd Edition

Mobiltelefonprogrammierung

- C++:** Performant, einzige Möglichkeit, die ganze Hardware anzusprechen, geringe Portabilität
- Java:** Sehr weit verbreitet und standardisiert, umfangreiche API
- Python:** Einfach und elegant, bietet oft mehr Hardwarezugriff als Java, eingeschränkt auf S60, grosse Downloads

Python auf S60

<http://opensource.nokia.com/projects/pythonfors60/>

Basiert auf Python 2.2.2

Erste Veröffentlichung in 2005

Apache Lizenz 2.0 und Python Lizenz

Installation PyS60 auf Mobiltelefon

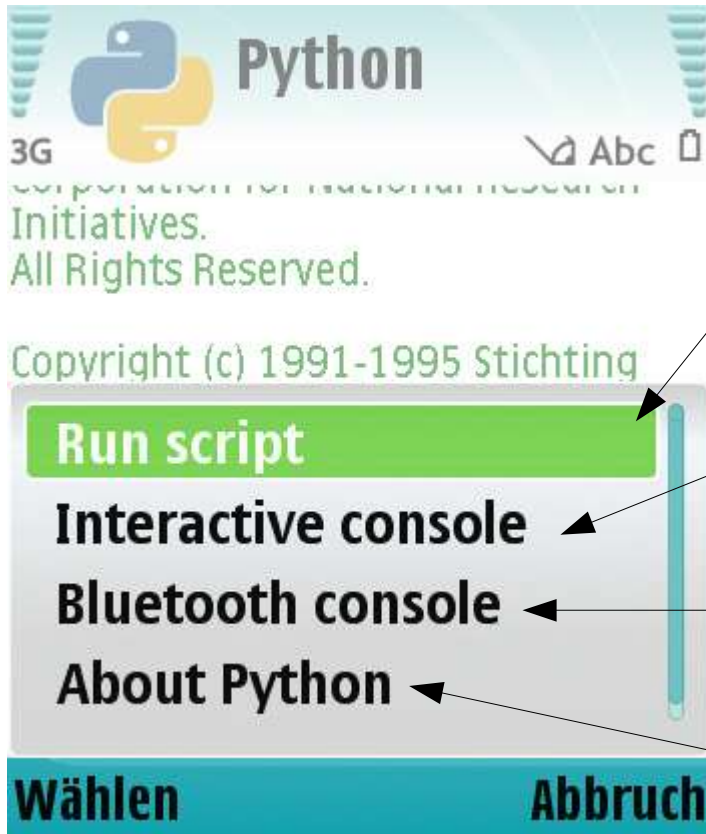
Wir betrachten nur S60 3rd Ed (1st und 2nd analog)

PythonForS60_1_3_20_3rdEd_selfsigned.SIS installieren

PythonScriptShell_1_3_20_3rdEd_selfsigned.SIS installieren



PyS60 auf Mobiltelefon (1)



Installierte Skripte ausführen

Python-Konsole

Remote Konsole

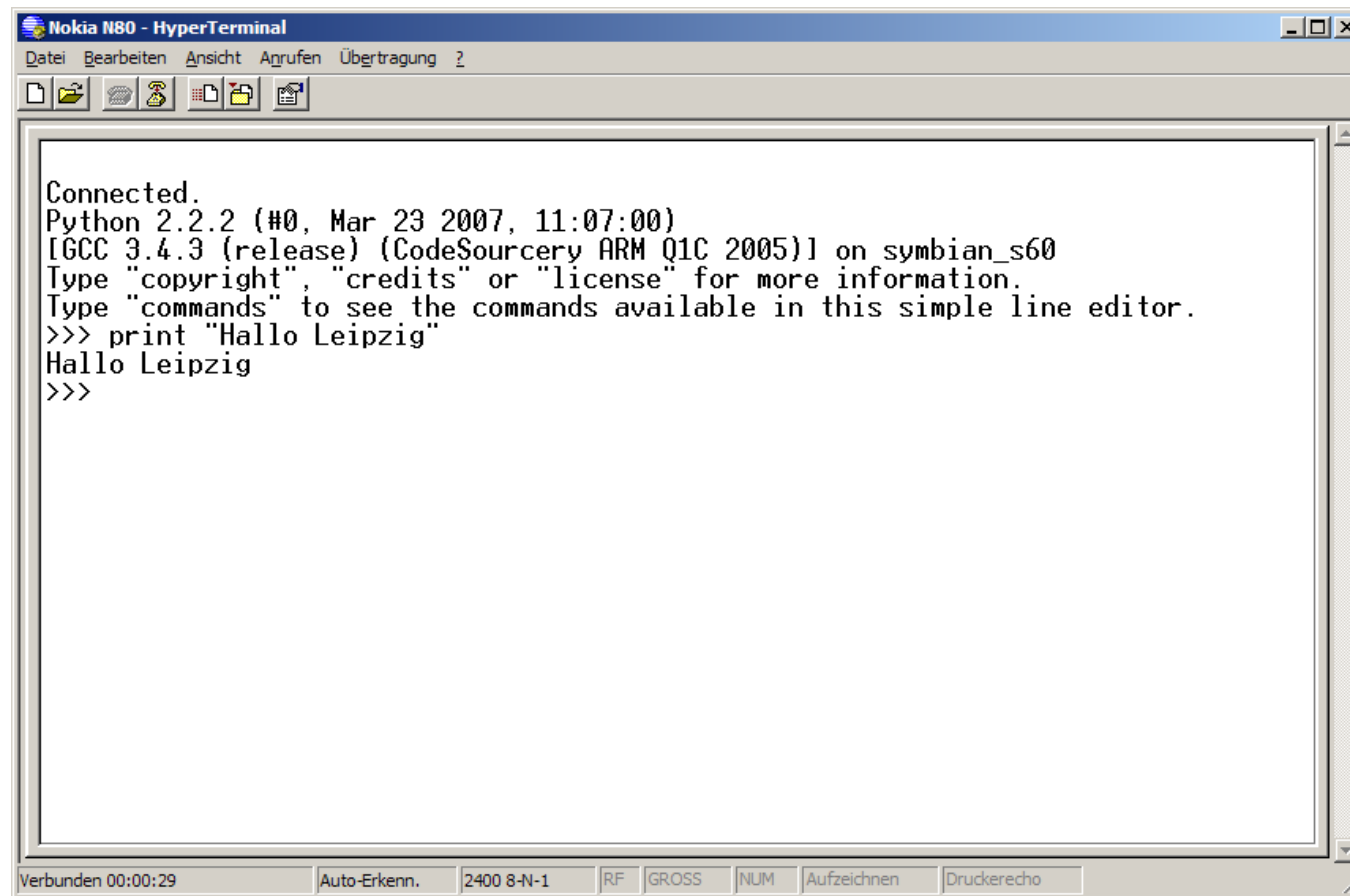
Versionshinweis

PyS60 auf Mobiltelefon (2)



PyS60 auf Mobiltelefon (3)

Zugriff mit HyperTerminal



The image shows a screenshot of a HyperTerminal window titled "Nokia N80 - HyperTerminal". The window has a menu bar with "Datei", "Bearbeiten", "Ansicht", "Anrufen", and "Übertragung". Below the menu bar is a toolbar with icons for file operations. The main text area contains the following text:

```
Connected.  
Python 2.2.2 (#0, Mar 23 2007, 11:07:00)  
[GCC 3.4.3 (release) (CodeSourcery ARM Q1C 2005)] on symbian_s60  
Type "copyright", "credits" or "license" for more information.  
Type "commands" to see the commands available in this simple line editor.  
>>> print "Hallo Leipzig"  
Hallo Leipzig  
>>>
```

At the bottom of the window, there is a status bar with the following information: "Verbunden 00:00:29", "Auto-Erkenn.", "2400 8-N-1", "RF", "GROSS", "NUM", "Aufzeichnen", and "Druckerecho".

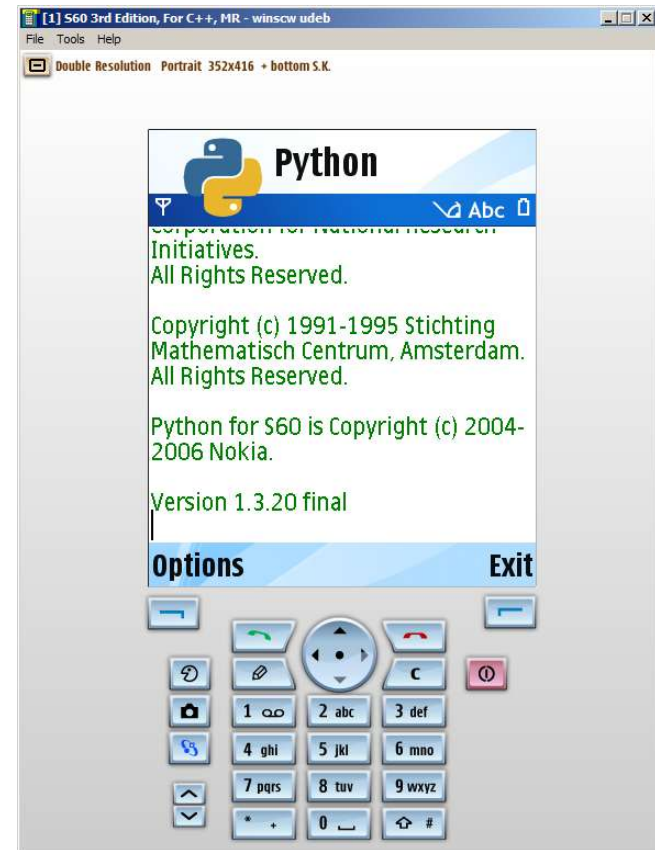
Installation PyS60 im Emulator (1)

- Java runtime \geq 1.4.2_02 von <http://java.sun.com/downloads/index.html>
- ActivePerl \geq 5.6.1 von <http://www.activestate.com/Products/ActivePerl>
- SDK für S60 3rd Edition Maintenance von <http://forum.nokia.com> (S60-SDK-0616-3.0-mr.3.749.zip)
- Um Java zu ermöglichen,
edp.java.version.start=1.4.1,...,6.0.0 in
...\\Symbian\\9.1\\S60_3rd_MR\\Epoc32\\tools\\ecm
t\\config\\config.properties eintragen

Installation PyS60 im Emulator (2)

- PythonForS60_1_3_20_SDK_3rdEd_selfsigned .zip nach ...\Symbian\9.1\S40_3rd_MR auspacken

Benutzung PyS60 im Emulator



Einfache Beispiele (1)

```
import appuifw
```

```
appuifw.note(u"Hallo Leipzig", "info")
```



Einfache Beispiele (2)

```
# SMS versenden
```

```
import messaging
```

```
sms_send("01776392596",u"Hallo Leipzig per SMS")
```

```
# MMS versenden
```

```
import messaging
```

```
mms_send("01776392596",u"Hallo Leipzig per MMS",  
u"e:\\picture.jpg")
```

Einfache Beispiele (3)

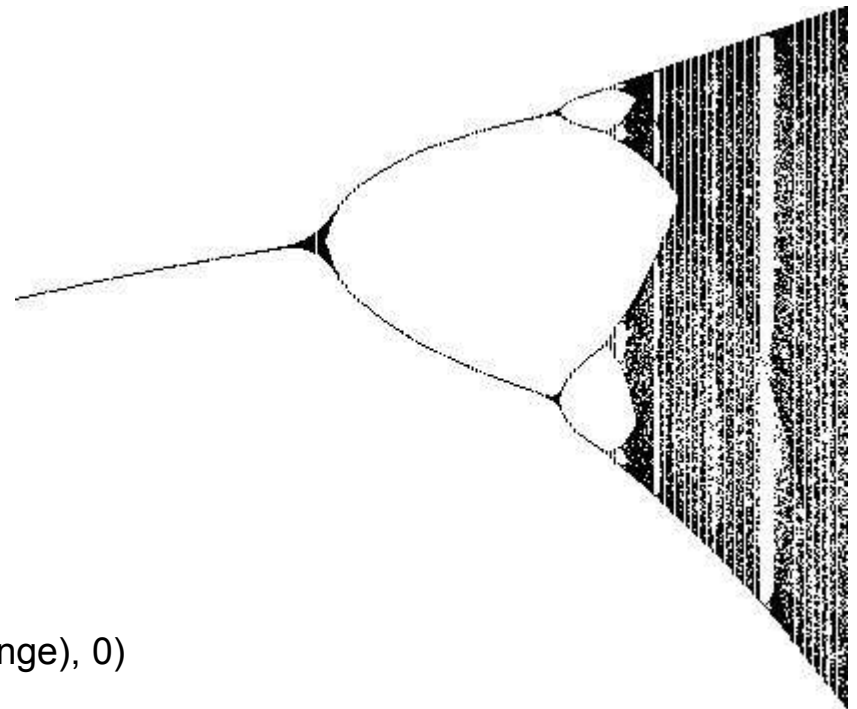
```
from appuifw import *
import e32

app.screen = 'full'
app.body = canvas = Canvas()
width, height = canvas.size

r_min = 2.4
r_max = 4.0
r_range = r_max - r_min

for rcount in range(height):
    r = (r_min + rcount * r_range / height)
    xn = 0.25
    for i in range(50):
        xnext = xn * r * (1.0 - xn)
        xn = xnext
    for i in range(width):
        xnext = xn * r * (1.0 - xn)
        xn = xnext
        canvas.point((xnext * width, (r - r_min) * height / r_range), 0)
        e32.ao_yield()

lock = e32.ao_lock()
app.exit_key_handler = lock.signal
lock.wait()
```



Python API (1)

Untermenge der Python Standard Library

Erweiterungen:

e32 Schnittstellen zu Symbian

appuifw S60 User Interface

Dynamisch Ladbare Erweiterungen:

graphics, e32db, messaging, inbox, location,
sysinfo, camera, audio, telephone, calendar,
contacts

Python API (2)

- Nur eine Untermenge des SDK wird auf das Mobiltelefon installiert, weitere Module können bei Bedarf kopiert werden
- Viele Python 2.2.2 Erweiterungen laufen auch ohne Anpassungen auf dem Mobiltelefon

Module von Python S60 (1)

sysinfo

Aktives Profil, Batterie, Displaygroesse, Speicherplatz, IMEI, Betriebssystemversion, SW-Version, Signalstärke

graphics

Screenshot, Bilder, Grafikprimitive (Linien, Polygone, Punkte)

camera

Kameraparameter, Fotoaufnahme

keycapture

Key Ereignisse

topwindow

Fenster, die über anderen Anwendungen angezeigt werden

Module von Python S60 (2)

gles & glcanvas

OpenGL

audio

Aufnahme, Wiedergabe, Lautstärke, Positionierung innerhalb der Stücke

telephone

Anrufen, Auflegen

messaging

SMS, MMS

inbox

Lesen von SMS

location

GSM Standort

Module von Python S60 (3)

contacts

Kontakte, Gruppen, Vcard

calendar

Ereignisse, Geburtstage, Aufgaben, Wiederholungen

e32db & e32dbm

Zugriff auf Symbian native DB (SQL-Untermenge)

Weiterführende Themen

Verteilung von Applikationen als SIS-Archive

Signieren von Applikationen

Pys60 mit C++ erweitern

Weitere Informationen: Webseiten

Projektseite von Python

www.python.org

S60 Plattform

www.s60.com

Pys60 Projektseite

<http://sourceforge.net/projects/pys60/>

PyS60 Wiki

http://wiki.opensource.nokia.com/projects/Python_for_S60

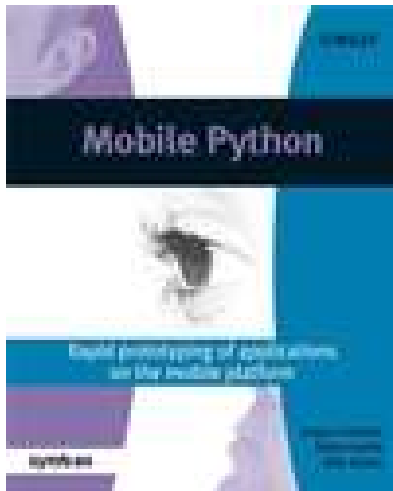
PyS60 Tutorial

www.mobilenin.com/pys60/menu.htm

PyS60 Diskussionsforum

<http://discussion.forum.nokia.com/forum/forumdisplay.php?f=102>

Weitere Informationen: Literatur



Mobile Python:
Rapid prototyping of applications on the
mobile platform
Jürgen Scheible, Ville Tuulos, Jukka
Laurila

Verlag: Wiley

ISBN: 978-0-470-51505-1

200 Seiten

Erscheint 2007

Vielen Dank für die Aufmerksamkeit

Fragen?