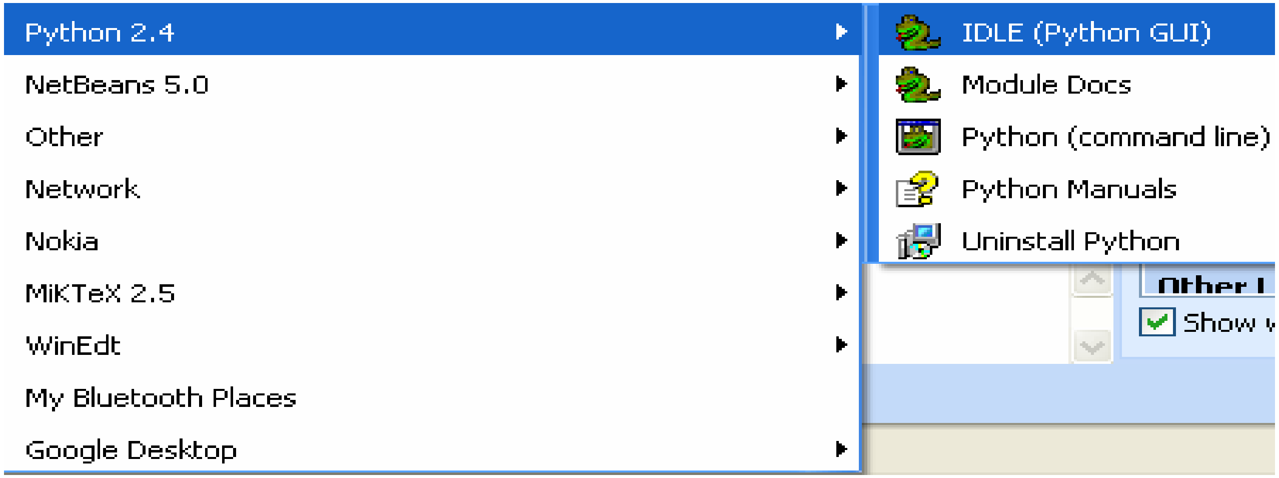
Mobile Phone Programming

Software Developer Kit (SDK)

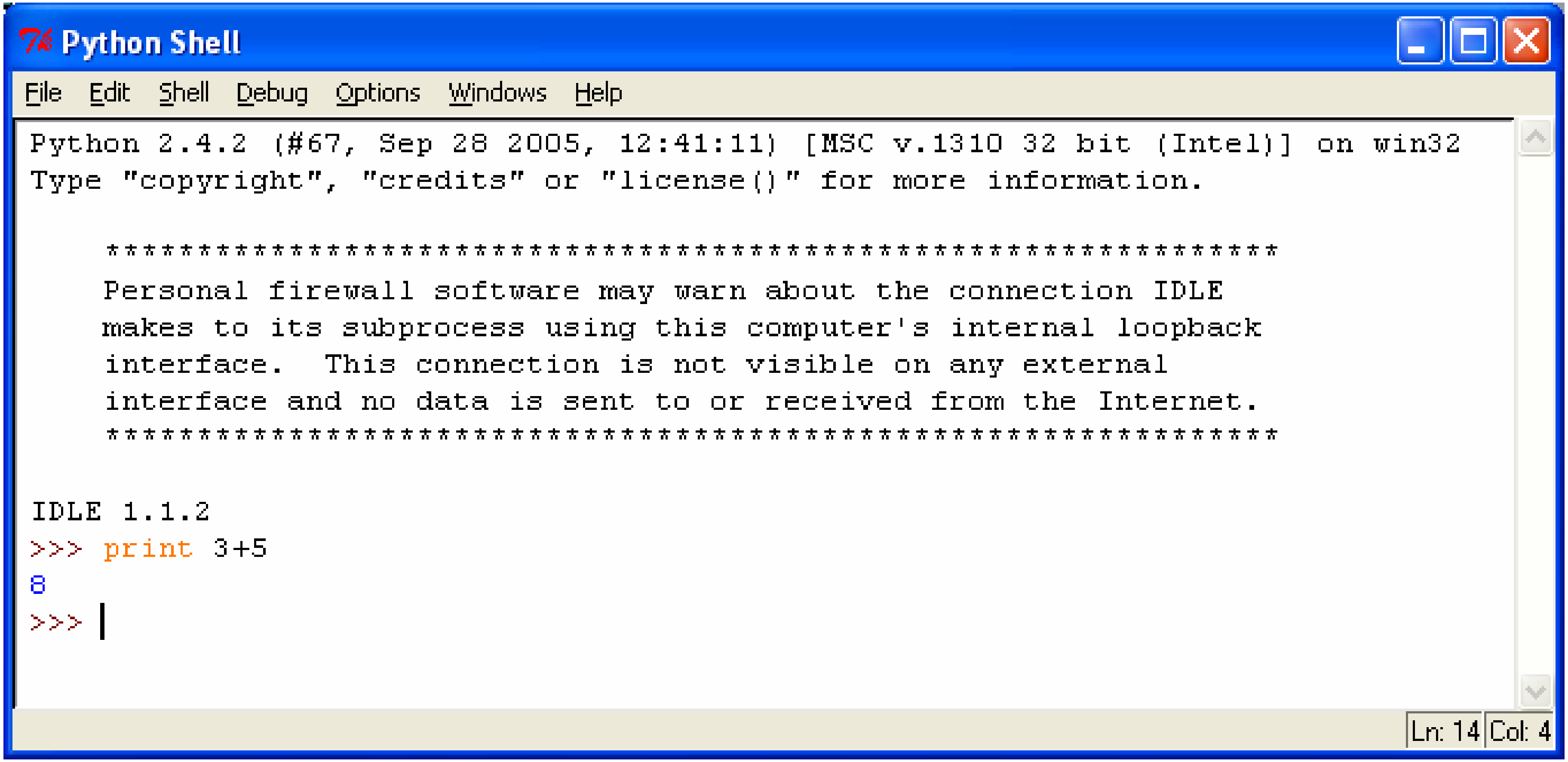
- Python 2.5
- S60 Platform SDK for Symbian OS
- Python For Series60
- Documentation

Python 2.5

- Download the file:
  - python-2.5.msi
  - Run it
- More info at http://www.python.org



Python shell



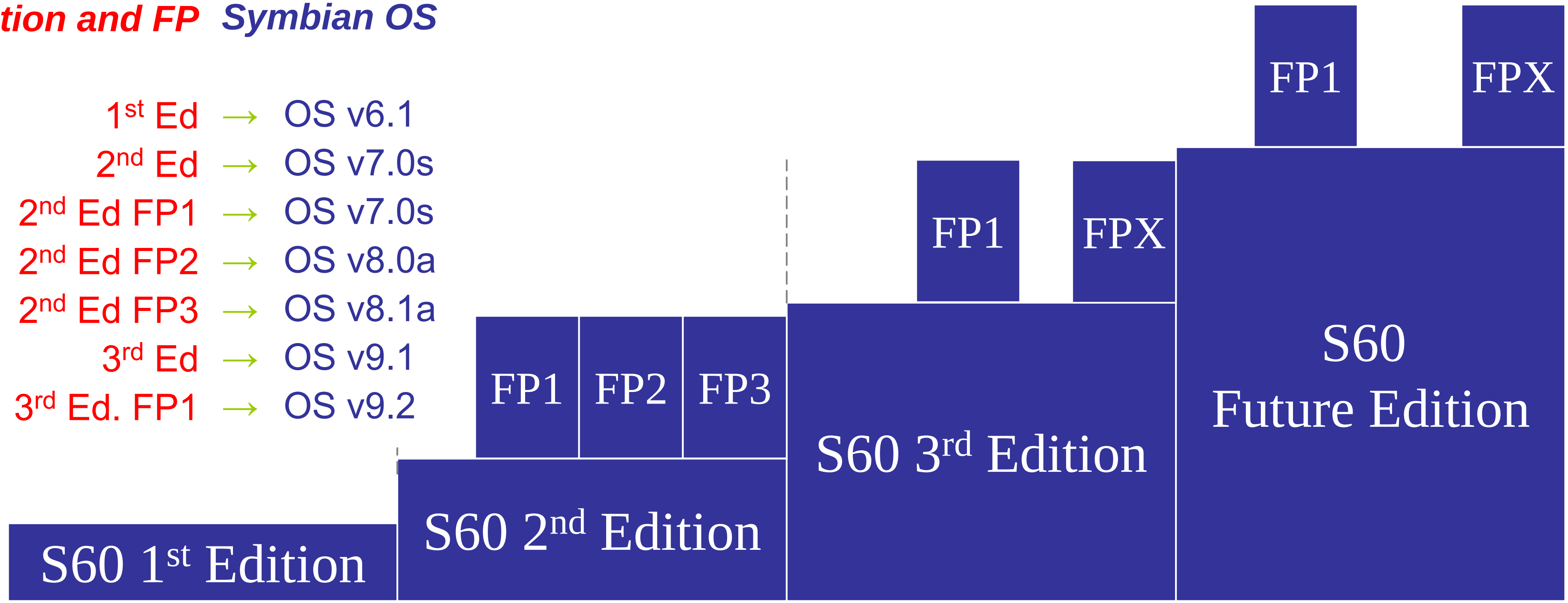
```
Python 2.4.2 (#67, Sep 28 2005, 12:41:11) [MSC v.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

    ****************************************************************
    Personal firewall software may warn about the connection IDLE
    makes to its subprocess using this computer's internal loopback
    interface.  This connection is not visible on any external
    interface and no data is sent to or received from the Internet.
    ****************************************************************

IDLE 1.1.2
>>> print 3+5
8
>>> 
```

- Different versions:
  - 1st Edition Feature Pack 1
  - 2nd Edition
    - Feature Pack 1
    - Feature Pack 2
    - Feature Pack 3
  - 3rd Edition
    - Feature Pack 1

Can be downloaded from: **www.forum.nokia.com**

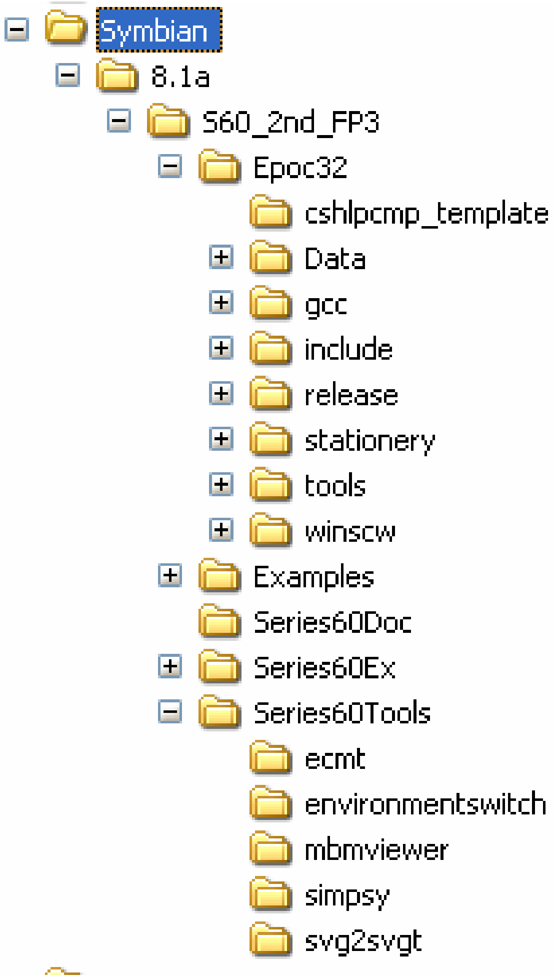
Mapping Series 60 and Symbian OS

Edition and FP		Symbian OS
1st Ed	→	OS v6.1
2nd Ed	→	OS v7.0s
2nd Ed FP1	→	OS v7.0s
2nd Ed FP2	→	OS v8.0a
2nd Ed FP3	→	OS v8.1a
3rd Ed	→	OS v9.1
3rd Ed. FP1	→	OS v9.2



S60 Platform SDKs for Symbian OS, for C++

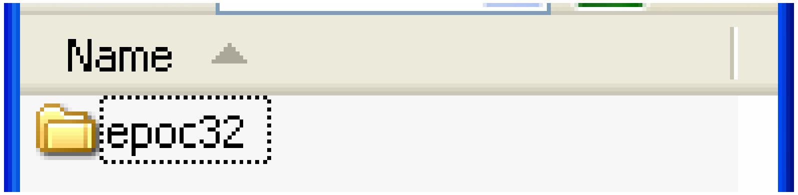
- How to install it:
  - Download the
    - s60_???_sdk_fp?.zip
  - Extract the .zip file
  - Run setup.exe



```
⊟ Symbian
  ⊟ 8.1a
    ⊟ S60_2nd_FP3
      ⊟ Epoc32
          cshlpcmp_template
        ⊞ Data
        ⊞ gcc
        ⊞ include
        ⊞ release
        ⊞ stationery
        ⊞ tools
        ⊞ winscw
      ⊞ Examples
        Series60Doc
      ⊞ Series60Ex
      ⊟ Series60Tools
          ecmt
          environmentswitch
          mbmviewer
          simpsy
          svg2svgt
```

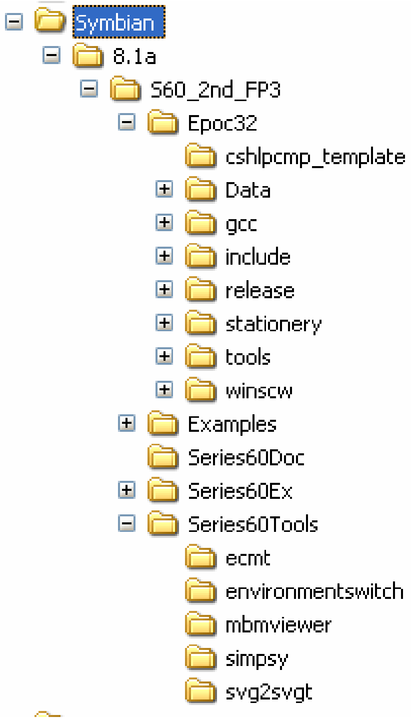
Python for S60 plug-in

- Download the file:
  - PythonForS60_1_3_18_SDK_???EdFP?.zip
  - Extract the .zip file
  - Replace the folder in the Symbian root



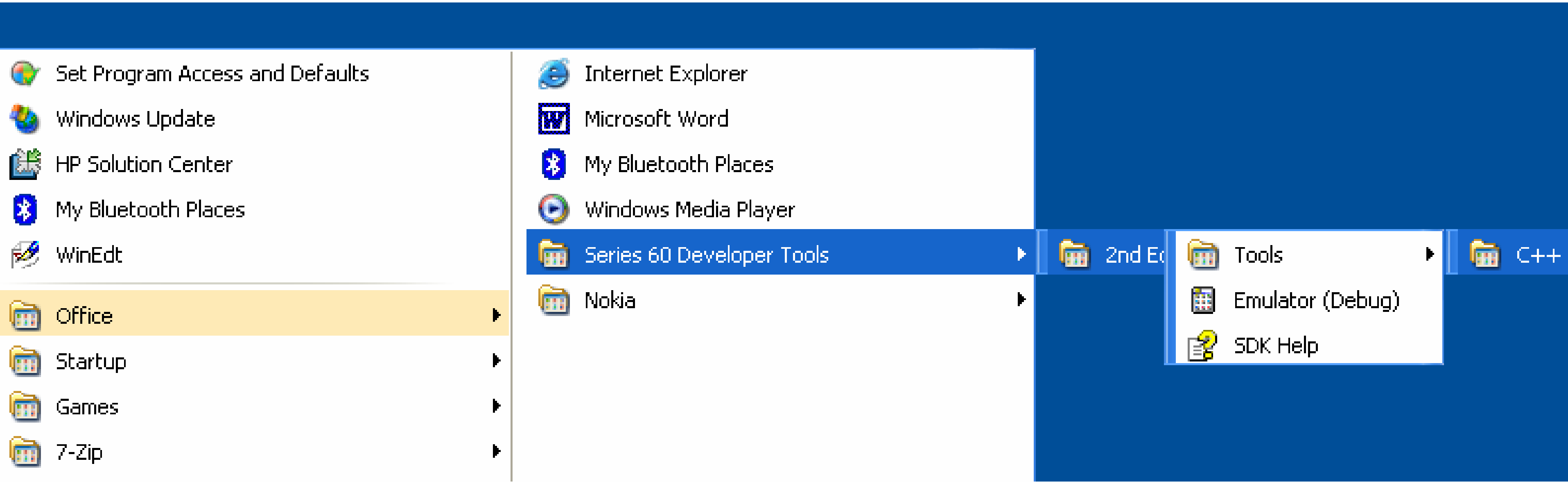
Name

epoc32



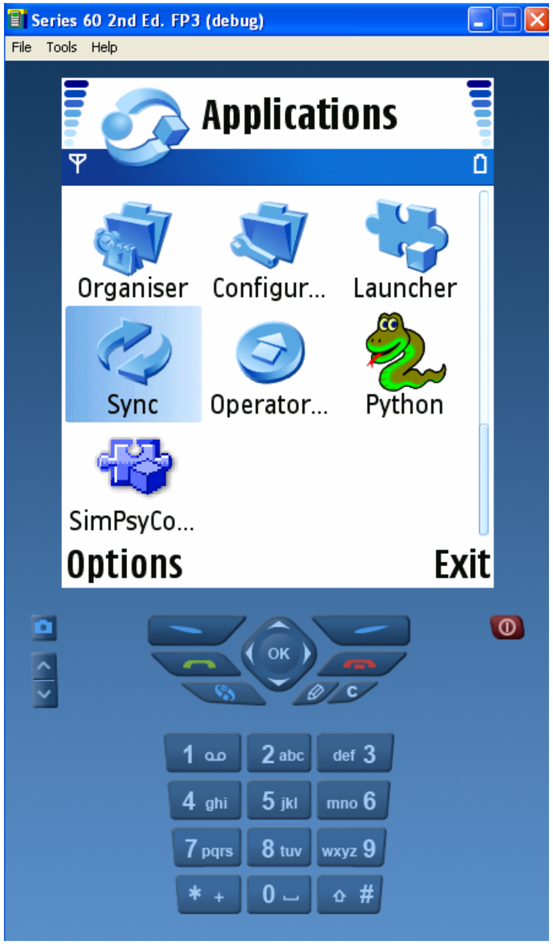
```
⊟ Symbian
  ⊟ 8.1a
    ⊟ S60_2nd_FP3
      ⊟ Epoc32
          cshlpcmp_template
        ⊞ Data
        ⊞ gcc
        ⊞ include
        ⊞ release
        ⊞ stationery
        ⊞ tools
        ⊞ winscw
      ⊞ Examples
        Series60Doc
      ⊞ Series60Ex
      ⊟ Series60Tools
          ecmt
          environmentswitch
          mbmviewer
          simpsy
          svg2svgt
```

Start the Emulator



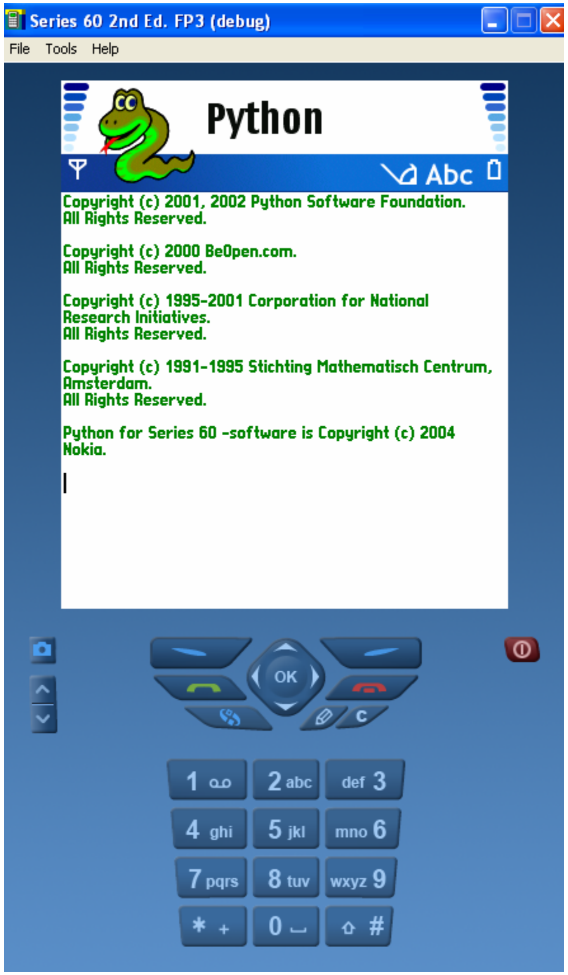
Python on the Emulator

- After installation the emulator should start with generic mobile
- On the emulated phone the snake is the Python symbol
- Click on the icon for Python



Python on the Emulator

- Now the python interpreter can be used to
  - Run a script
    - ball.py
    - snake.py
  - Interactive console
  - Bluetooth console



Copyright (c) 2001, 2002 Python Software Foundation.
All Rights Reserved.

Copyright (c) 2000 BeOpen.com.
All Rights Reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All Rights Reserved.

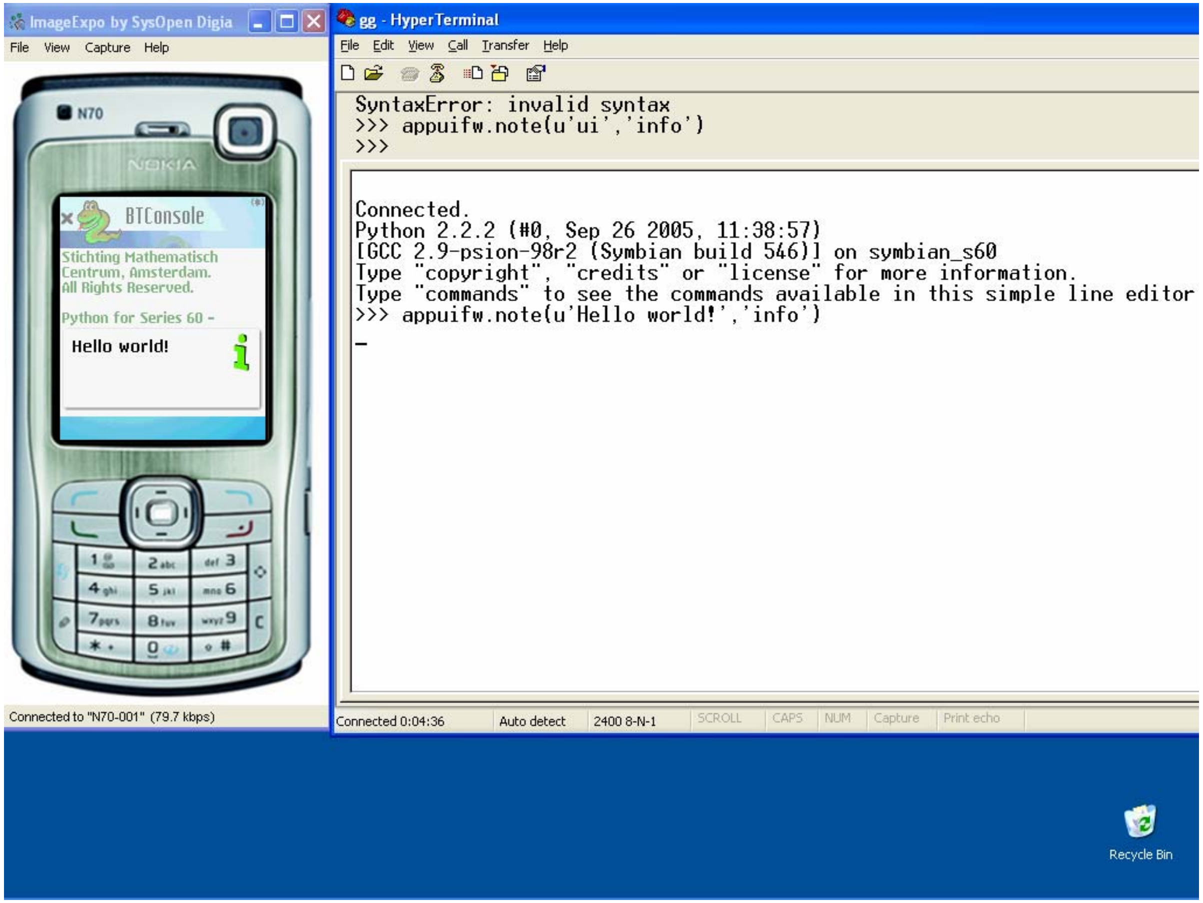
Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.

Python for Series 60 -software is Copyright (c) 2004 Nokia.

Python on the Mobile Phone

- Interactive console
  - Type your script directly on the mobile phones
  - Nice to try, but it takes time
- Bluetooth Console
  - Connect the mobile phone with your laptop using the Bluetooth connection
  - Use the HyperTerminal to type virtually on the mobile phone

Bluetooth Console



```
SyntaxError: invalid syntax
>>> appuifw.note(u'ui','info')
>>>
```

```
Connected.
Python 2.2.2 (#0, Sep 26 2005, 11:38:57)
[GCC 2.9-psion-98r2 (Symbian build 546)] on symbian_s60
Type "copyright", "credits" or "license" for more information.
Type "commands" to see the commands available in this simple line editor
>>> appuifw.note(u'Hello world!','info')
_
```

How can I write my Python script?

- Use any text editor and start your program
- First include the modules needed
- Define variables and functions
- Main part
- Save the text under *file.py*

What to do with my Python script?

- There are two main options:
  - Using it on the Mobile Phone
    - As a Python script
    - As a standalone version
  - Using it on the Emulator
    - Save the Python script under

      DRIVE:\Symbian\8.1a\S60_2nd_FP3\Epoc32\release\winscw\udeb\z\system\apps\python
    - Invoke the Emulator, start Python, and *Run script* from the *Options*

Installing a Python script on the phone

- One possible way to install a script on the phone:
  - Connect PC and phone via Bluetooth or Infrared
  - Right-click the file you want to install and send it to your phone
  - Allow the Python Installer to install the Python script
- During the installation process you have to choose whether the file has to be to be installed as a "Python script" (under Python execution environment) or as a "Python lib module" (library component). In this case you have to choose to install the file as a "Python script".
- After the installation the application can be started by invoking *Run script* under *Options*

Documentation

- In the file Documentation.zip you will find some pdf files with the API documentation and much more
- If you don’t find what you are looking for……

Useful links

- http://www.forum.nokia.com/
  - Discussion board
  - Devices specifications
- http://wiki.opensource.nokia.com/projects/Python_for_S60
- http://mobiledevices.kom.aau.dk/
- http://www.mobilenin.com/pys60/menu.htm

Mobile Phone Programming Course

Module 1

mobile phones

- Everybody should have installed now the Symbian SDK and Python on the PC
- The Bluetooth connection should work between PC and phone
- Maybe you did already the first trials with the emulator and/or the Bluetooth console.

Overview

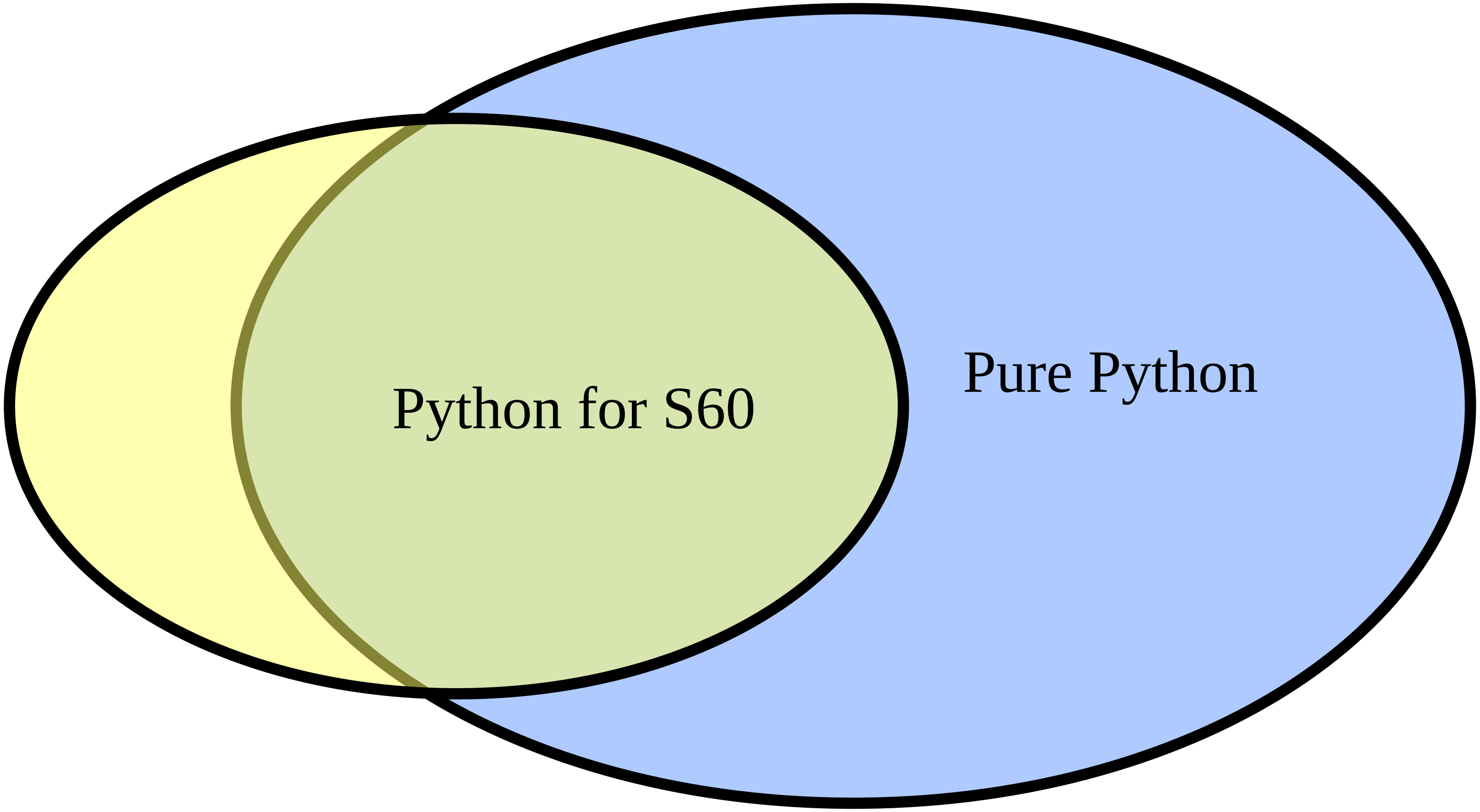
- Programming Python
  - Python in a Nutshell
    - Variables and Lists
    - Control Flows
  - Python for Series 60
    - Modules
- Bringing Python to the mobile phone

Python

- Powerful script language
- So far used to program desktop applications
- Lately a compact version of Python has been released to program on mobile phones:

  Python for Series 60
- The latest release is Pyton for S60 1.3.18

Pure Python and Python for S60



Python in a Nutshell

- Powerful script language
- We will not discuss all Python commands here, but refer to other sources
- Here we focus on those that we need later for the exercises

Lists

```
##########################################################
#Python examples for AAU Mobile Phone Programming        #
#(c) Mobile Phone Group                                  #
##########################################################
#
#set up list with strings
animal = ['dog','cat','bird']
print animal
#
#set up list with numbers
numbers = [1,2,5,6]
print numbers
print numbers[2]
#
#add entries
numbers[2:2] = [3,4]
print numbers
print numbers[2]
#
#remove enntries
numbers[4:6] = []
print numbers
```

```
['dog', 'cat', 'bird']
[1, 2, 5, 6]
5
[1, 2, 3, 4, 5, 6]
3
[1, 2, 3, 4]
```

More Lists

```
###################################################
#Python examples for AAU Mobile Phone Programming         #
#(c) Mobile Phone Group                                   #
###################################################
#
#set up list
numbers = [17,32,3,14]
print numbers
#add element at the end using the append() command
numbers.append(54)
print numbers
#add element at a given position using the insert() command
numbers.insert(2,22)
print numbers
#delete element (the first element that fits to the argument)
numbers.remove(22)
print numbers
#sort list
numbers.sort()
print numbers
#getting the length (number of elements) of the list
print len(numbers)
```

```
[17, 32, 3, 14]
[17, 32, 3, 14, 54]
[17, 32, 22, 3, 14,
   54]
[17, 32, 3, 14, 54]
[3, 14, 17, 32, 54]
5
```

if statement

```
##################################################
#Python examples for AAU Mobile Phone Programming #
#(c) Mobile Phone Group                           #
##################################################

#python asking ofr input
age = int(raw_input("Please enter your age"))
#evaluate the input
if age < 0:
    print 'Not born yet'
elif age == 0:
    print 'Just born'
elif age == 18:
    print 'You are eighteen!'
else:
    print 'Whatever, your are born'
```

```
Please enter your age
33
Whatever, your are born
```

for statement

```
#####################################################
#Python examples for AAU Mobile Phone Programming #
#(c) Mobile Phone Group                                            #
#####################################################

#set up list with strings
animal = ['dog','cat','bird']
for individual in animal:
    print individual
```

```
dog
cat
bird
```

while statement

```
#####################################################
#Python examples for AAU Mobile Phone Programming #
#(c) Mobile Phone Group                                         #
#####################################################

number =10
while (number != 0):
    print number
    number = number -1
```

```
10
9
8
7
6
5
4
3
2
1
```

Module

```
##################################################
#Python examples for AAU Mobile Phone Programming   #
#(c) Mobile Phone Group                              #
##################################################
#define the functions
def sum(a,b):
    print a+b

def diff(a,b):
    print a-b

###################################################
#Python examples for AAU Mobile Phone Programming    #
#(c) Mobile Phone Group                              #
###################################################

from myfunction import sum

sum(55,12)
```

```
67
```

Python Modules

- **Operating System Services and Information**
  - **e32** - A Symbian OS related services package
  - **sysinfo** - Access to system information
- **User Interface and Graphics**
  - **appuifw** - Interface to the S60 GUI framework
  - **graphics** - A graphics related services package
  - **camera** - Interface for taking photographs
  - **keycapture** - Interface for global capturing of key events
  - **topwindow** - Interface for creating windows that are shown on top of other applications
- **Audio and Communication Services**
  - **audio** - An audio related services package
  - **telephone** - Telephone services
  - **messaging** - A messaging services package
  - **inbox** - Interface to device inbox
  - **location** - GSM location information
  - **socket** - for socket communications
- **Data Management**
  - **contacts** - A contacts related services package
  - **calendar** - Access to calendar related services

e32 Module

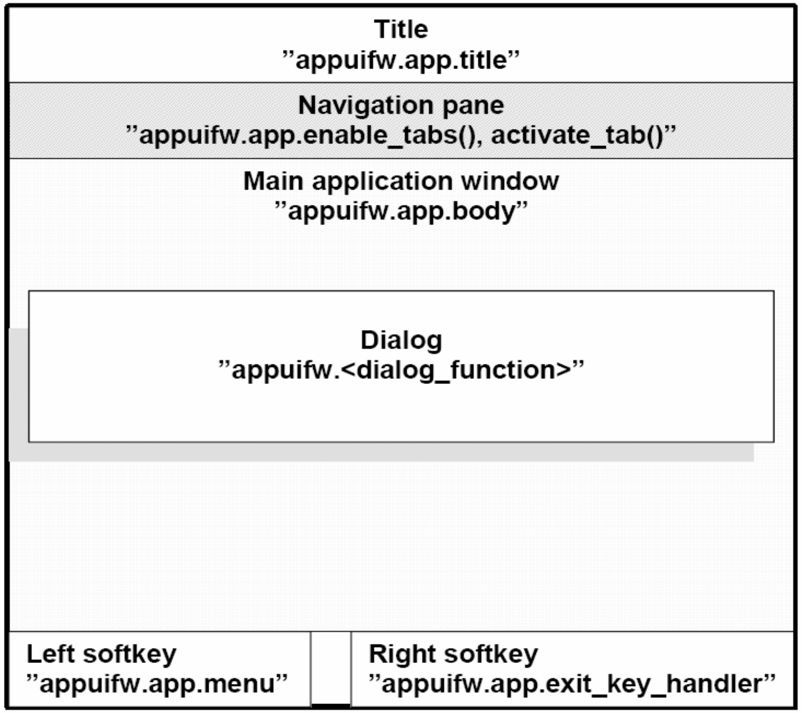
- ao_sleep
- drive_list
- file_copy
- in_emulator set_home_time
- pys60_version
- s60_version info
- start_exe
- start_server

sysinfo Module

- **active_profile**
  - Returns the current active profile as a string, which can be one of the following: ’general’,’silent’, ’meeting’, ’outdoor’, …
- **battery**
- **pixels**
- **free_drivespace**
- **imei**
- **total_ram**
- **free_ram**
- **total_rom**
- **ring_type**
  - Returns the current ringing type as a string: ’normal’, ’ascending’, ’ring once’, ’beep’, or ’silent’.
- **os_version**
- **signal_bars**
- **signal_dbm**
- **sw_version**

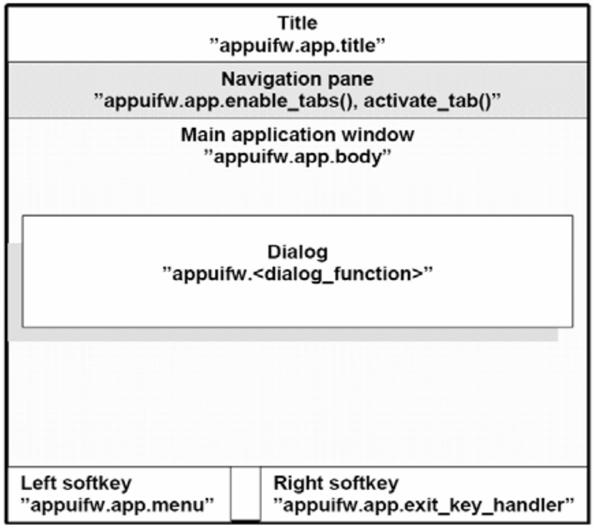
appuifw Module

- Establishes an interface to the Series 60 user interface (UI)
- Three different screen types such as:
  - Normal
  - Large
  - Full
- Variety of user interactions: *queries*



appuifw Module

- The menu can be made in different style with tabulators
- Same example given in the exercises later



Title
"appuifw.app.title"

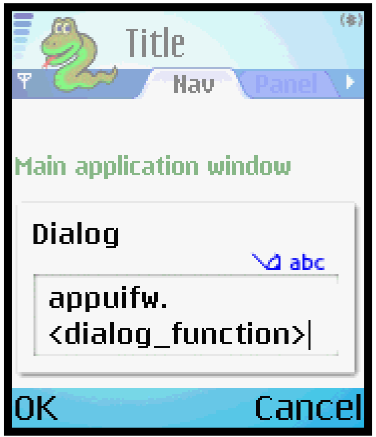
Navigation pane
"appuifw.app.enable_tabs(), activate_tab()"

Main application window
"appuifw.app.body"

Dialog
"appuifw.<dialog_function>"

Left softkey
"appuifw.app.menu"

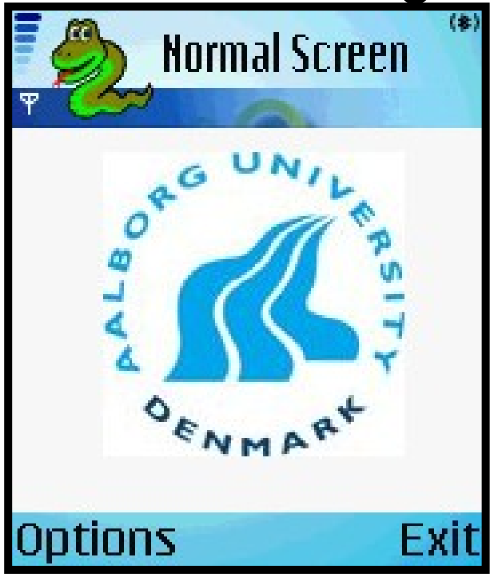
Right softkey
"appuifw.app.exit_key_handler"



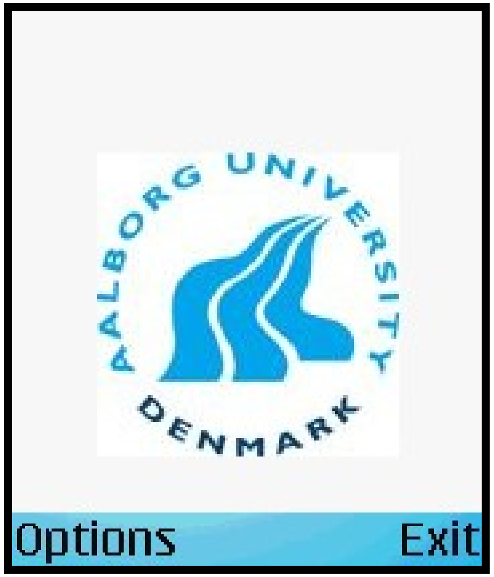
appuifw.
<dialog_function>

Screen Type

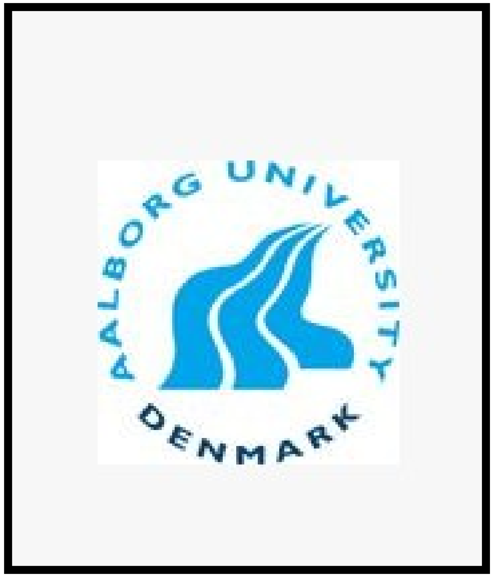
- Three different screen types such as:
  - Normal/Large/Full



normal



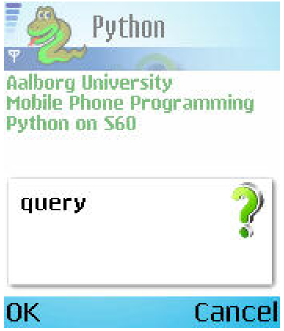
large

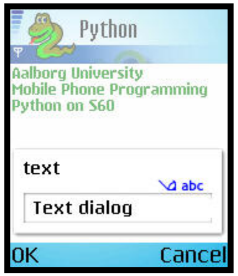


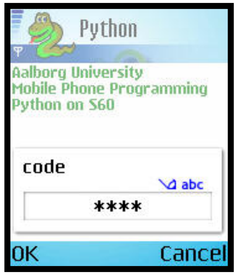
full

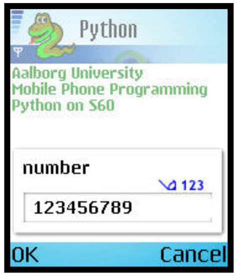
appuifw Module – dialog / query

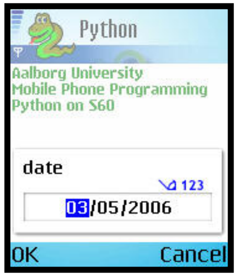
- Single field dialog as query
  - *text*
  - *code*
  - *number*
  - *date*
  - *time*
  - *query*
- User interaction requested

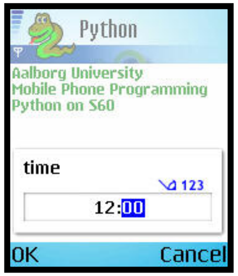

Python
Aalborg University
Mobile Phone Programming
Python on S60
query
OK Cancel

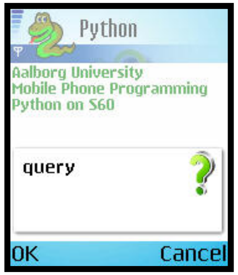

Python
Aalborg University
Mobile Phone Programming
Python on S60
text abc
Text dialog
OK Cancel


Python
Aalborg University
Mobile Phone Programming
Python on S60
code abc
****
OK Cancel


Python
Aalborg University
Mobile Phone Programming
Python on S60
number 123
123456789
OK Cancel

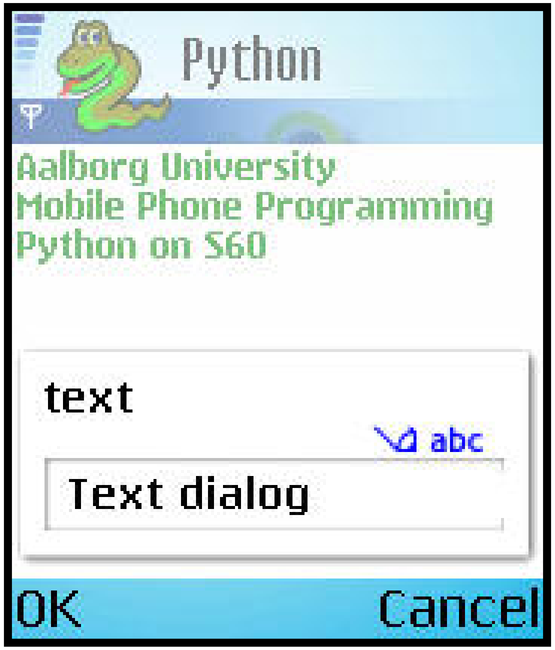

Python
Aalborg University
Mobile Phone Programming
Python on S60
date 123
03/05/2006
OK Cancel


Python
Aalborg University
Mobile Phone Programming
Python on S60
time 123
12:00
OK Cancel


Python
Aalborg University
Mobile Phone Programming
Python on S60
query
OK Cancel

Dialog Text

retrievedtext = appuifw.query(u“text", "text")



Python

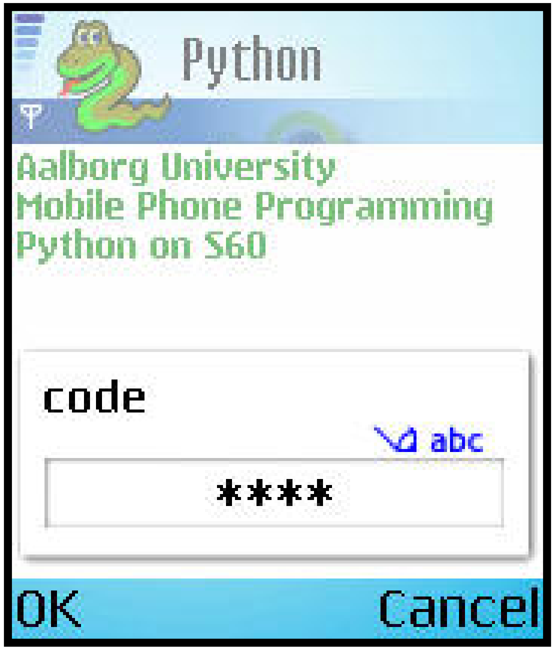
Aalborg University
Mobile Phone Programming
Python on S60

text

Text dialog

Dialog Code

retrievedcode = appuifw.query(u“code", "code")



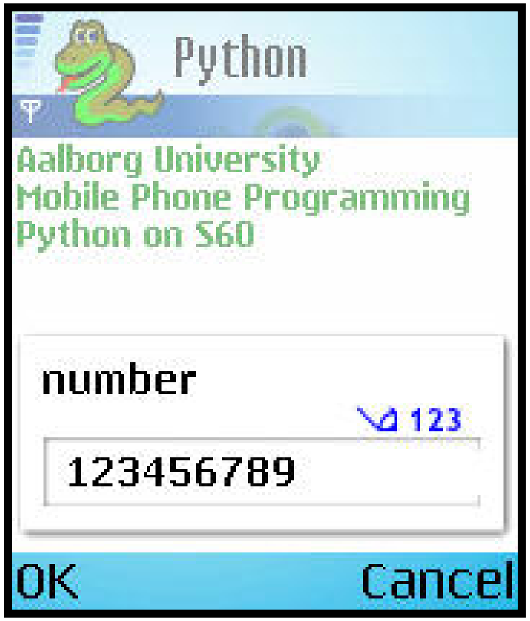
Aalborg University
Mobile Phone Programming
Python on S60

code

****

Dialog Number

retrievednumber = appuifw.query(u“number", “number")



Python

Aalborg University
Mobile Phone Programming
Python on S60

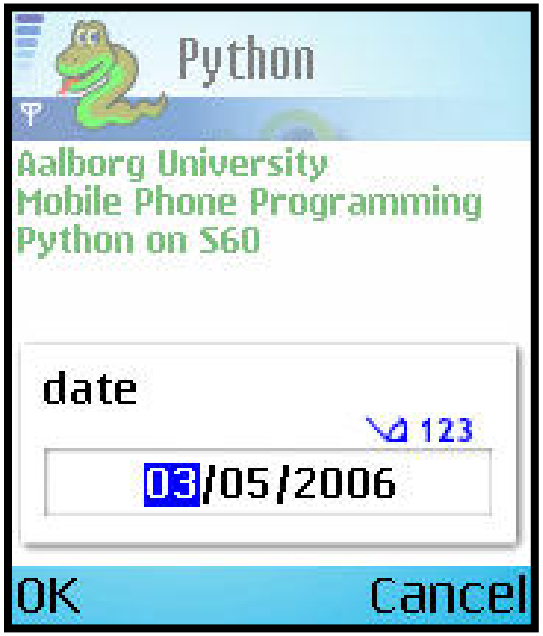
number

123

123456789

Dialog Date

retrieveddate = appuifw.query(u“date", “date")



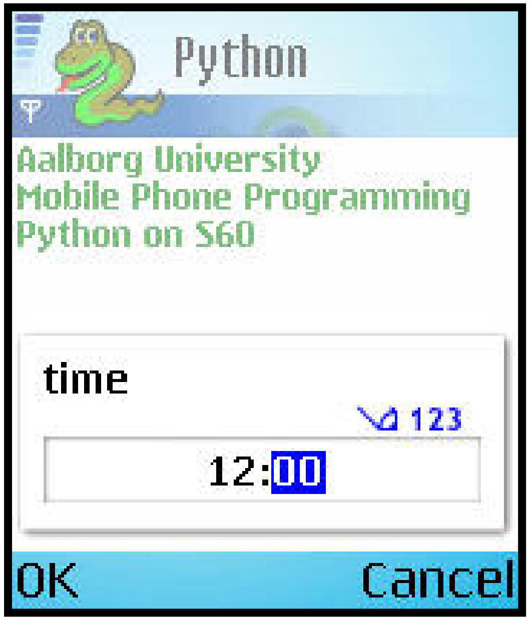
Aalborg University
Mobile Phone Programming
Python on S60

date

03/05/2006

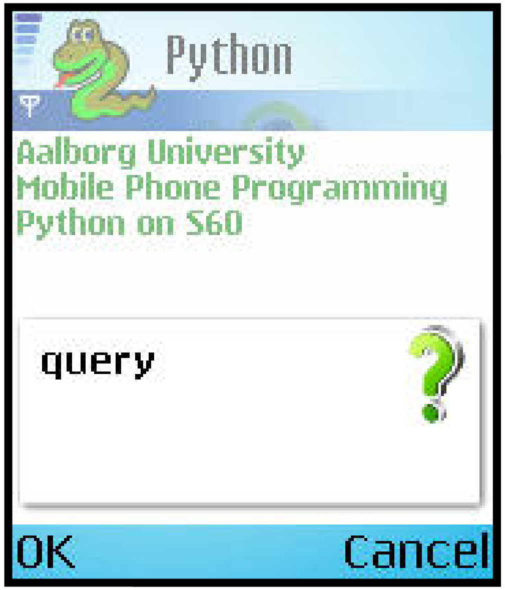
Dialog Time

```
retrievedtime = appuifw.query(u“time", "time")
```



Dialog Query

retrievedquery = appuifw.query(u“query", “query")



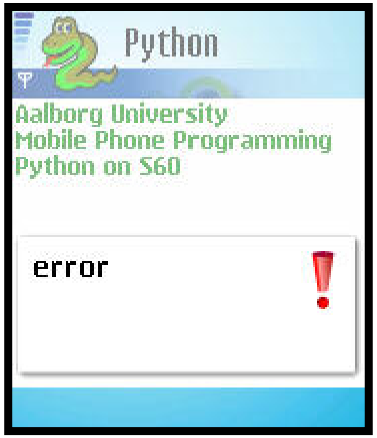
Python

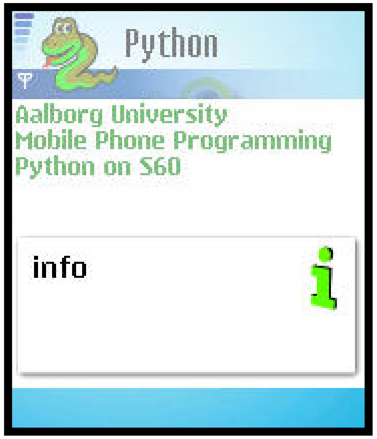
Aalborg University
Mobile Phone Programming
Python on S60

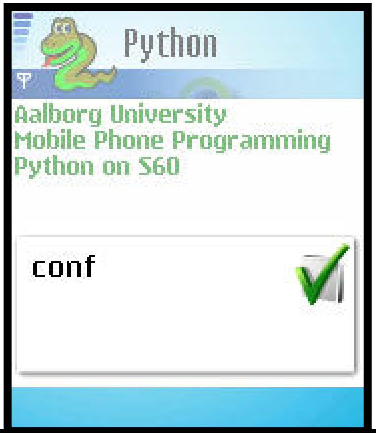
query

appuifw Module – dialog / note

- Information conveyed over phone display
  - *error*
  - *info*
  - *conf*
- No user interaction


Python
Aalborg University
Mobile Phone Programming
Python on S60
error


Python
Aalborg University
Mobile Phone Programming
Python on S60
info


Python
Aalborg University
Mobile Phone Programming
Python on S60
conf

Menu

- Useful for creating menu in your applications:
- Example:

```
menu=[(u”name”,funcName),(u”name1”),funcName1)]
```

messaging Module

- sms send(recipient, message)
  - Sends an SMS message with body text message (Unicode) to telephone number recipient (string).
- mms send(recipient, message, [attachment=None])
  - Note: Available from S60 3.0 onwards (inclusive).Sends an MMS message with body text message (Unicode) to telephone number recipient (string).
  - The optional parameter attachment is full path to e.g. image file attached to the message.

Location Module

- gsm location()
  - Retrieves GSM location information: Mobile Country Code, Mobile Network Code, Location Area Code, and Cell ID. A location area normally consists of several base stations. It is the area where the terminal can move without notifying the network about its exact position. mcc and mnc together form a unique identification number of the network into which the phone is logged.

camera Module

- All camera related functionality
  - Availabe hardware information
    - cameras_available()
    - image_modes()
    - flash_modes()
  - take_photo()
    - take_photo(mode,size,flash,zoom,exposure,white_balance, position)

audio Module

- Functionality to work with the audio device
  - play()
  - stop()
  - record()
  - close()
  - state()
- Note: Not that easy in the Bluetooth Console, so try it out with a script!

telephone Module

- Two function to use the phone capabilities
  - dial()
  - hang_up()

Socket module

- bt_discover([*address*])
  - Performs the Bluetooth device discovery
- bt_obex_discover([*address*])
  - Same as discover, but for discovery of OBEX class services on the chosen device.
- bt_obex_send_file(*address*, *channel*, *filename*)
  - Sends file *filename* (Unicode) wrapped into an OBEX object to remote *address*, *channel*.
- bt_obex_receive(*socket*, *filename*)
  - Receives a file as an OBEX object, unwraps and stores it into *filename* (Unicode). *socket* is a bound OBEX socket.

Just a small note …

- How to make all the screenshots … let Python help you

```
import graphics
import appuifw
querylist = [u'text',u'number',u'date',u'time',u'code',u'query']
for element in querylist:
    informationretrived = appuifw.query(element,element)
    image=graphics.screenshot()
    path=(u'c:\\Nokia\\Images\\'+element+u'Dialog.jpg')
    image.save(path)
```

skeleton

```
######  import all modules needed
import appuifw
import e32
###### set the screen size (in this example, to large)
appuifw.app.screen='large'
###### create a menu and the callback functions for the menu
def function_A():
    print "function_A"
def function_B():
    print "function_B"
appuifw.app.menu = [(u"Function A", function_A),(u"Function B", function_B)]
##### create and set an exit key handler
def exit_key_handler():
    app_lock.signal()
##### set the title
appuifw.app.title = u"New Title"
#####
app_lock = e32.Ao_lock()
appuifw.app.exit_key_handler = exit_key_handler
app_lock.wait()
```

Python Example

So lets get started!

Are you missing some functionality?

- First look up the API if it is in there, if not …
- No problem, you can make your own modules and functions
- More programming skills needed
- More on the fifth day of the course
- Python/Symbian C++

Exercise

- Get familiar with the API of Python S60 from Nokia document folder
- Play around with the Bluetooth console trying out each function of the appuifw module individually
- Write your own scripts and try them out:
  - Script 1: Making all query and note dialogs and save the screenshots
  - Script 2: Test all possible screen sizes and menu functions

Note: You can use the skeleton of the script included in the material

Exercise

- Script 3: Ask for a person’s name and find it in the contacts, afterwards dial his number
- Script 4: Shoot a photo and save it in that way that it can be found in the gallery
- Script 5: Shoot a photo and send it over Bluetooth
- Script 6: Record and play an audio file