Introduction to Programming using PYTHON

Session 4

N. D. Mendes
ndm@algos.inesc-id.pt

INESC-ID

October 16, 2006

Part I

Defining Functions

Defining Functions

```
def polar(x,y):
  import math
  r = (x ** 2 + y ** 2) ** .5
  a = math.atan2(x,y)
  return r, a
```

Defining Functions

```
def polar(x,y):
  import math
  r = (x ** 2 + y ** 2) ** .5
  a = math.atan2(x,y)
  return r, a
```

```
polar(1,1)      polar(x=1,y=1)   polar(y=1,x=1)
polar(x=1, 1)   polar(1,y=1)     polar(1,x=1)
```

Defining Functions

```
def compose_mail(greeting, body, signature="The Management"):
  return greeting + "\n\n" + body + "\n\n" + signature
```

Defining Functions

```
def compose_mail(greeting, body, signature="The Management"):
  return greeting + "\n\n" + body + "\n\n" + signature
```

```
compose_mail('Hi', 'This is an important message')
compose_mail(body='This is not an important message', greeting='hello', signature='me')
```

Defining Functions

```
def print_record(name, age, *p, **d):
  print "Name:", name
  print "Age:", age
  if p:
   print "More Info 1:", p
  if d:
   print "More Info 2:", d
```

Defining Functions

```
def print_record(name, age, *p, **d):
  print "Name:", name
  print "Age:", age
  if p:
   print "More Info 1:", p
  if d:
   print "More Info 2:", d
```

```
print_record('Simon', 25)
print_record('Simon', 25, 'a nice guy')
print_record('Simon', 25, education_level='BSc', address='Main Street, 2')
```

Part II

The `map` function

The `map` function

```
def shift(c):
  return chr((ord(c)+1) % 256)

map(shift,'abcdefg')
```

The `map` function

```
map(polar, range(10), range(10))
```

Part III

Repetitions

Repetitions
The `while` loop

```
ans = raw_input("How old are you?")

while not ans.isdigit():
  print "Wrong answer!"
  ans = raw_input("How old are you?  [or q(uit)] ")
  if ans.lower()[0] == 'q':
   break
```

Repetitions
The `while` loop

```
ans = raw_input("How old are you?")

while not ans.isdigit():
  print "Wrong answer!"
  ans = raw_input("How old are you?  [or q(uit)] ")
  if ans.lower()[0] == 'q':
   break
else:
  print "So, you are %d years old" % int(ans)
```

```
import random

for i in range(10):
  list = []
  lottery = random.randint(1,10)
  if lottery == 1:
   break
  list.append(lottery)
```

```
import random

for i in range(10):
  list = []
  lottery = random.randint(1,10)
  if lottery == 1:
   break
  list.append(lottery)
else:
  print list
```

```
wizards = {'Saruman' : 'the white', 'Gandalf':
'the grey', 'Radagast' : 'the brown' }

for w, c in wizards.iteritems():
  print w, c
```

```
for i, v in enumerate(['one', 'two', 'three']):
  print i, v
```

```
questions = ['name?', 'age?', 'Phd thesis?']
answers = ['Simon', 25, 'miRNA']

for q, a in zip(questions,answers):
  print q, a
```

```
for i in reversed(xrange(100)):
  print i
```

```
cities = ['London', 'Madrid', 'Lisbon', 'Paris',
'Lyon']

for c in sorted(cities):
  print c
```

Part IV

Exercise 1

Exercise

Write a function named `grep` which receives a Boolean function and a sequence and returns a list containing all elements of the given sequence which make the Boolean function evaluate to `True`

Exercise

Write a function named `grep` which receives a Boolean function and a sequence and returns a list containing all elements of the given sequence which make the Boolean function evaluate to `True`

```
def grep(function, sequence):
  out = []
  for e in sequence:
   if function(e):
    out.append(e)
  return out
```

Part V

Nested data structures

Nested data structures
A few warnings

```
l1 = [1, [2, 3], 4]
l2 = l1[:]

l1 is l2
l1[0] is l2[0]
l1[1] is l2[1]
```

Nested data structures
A few warnings

```
l1 = [1, [2, 3], 4]
l2 = l1[:]

l1 is l2
l1[0] is l2[0]
l1[1] is l2[1]

l1 = l1 * 3
l1[1][0] = 10
```

Nested data structures
A few warnings

```
l1 = [1, [2, 3], 4]
l2 = l1[:]

l1 is l2
l1[0] is l2[0]
l1[1] is l2[1]

l1 = l1 * 3
l1[1][0] = 10

import copy

l3 = copy.deepcopy(l1)
l1 is l3
l1[0] is l3[0]
l1[1] is l3[1]
```

Part VI

Files

Files

```
filename = 'myfile'

file = open(filename)
entire_content = file.read()

file.close()
```

Files

```
filename = 'myfile'

file = open(filename)
all_lines = file.readlines()

file.close()
```

Files

```
filename = 'myfile'
file = open(filename)

line = file.readline()
number_lines = 0
total_chars = 0

while line:
  number_lines += 1
  total_chars += len(line.strip())

file.close()
```

Files

```
filename = 'myfile'
file = open(filename)

number_lines = 0
total_chars = 0

for line in file.xreadlines():
  number_lines += 1
  total_chars += len(line.strip())

file.close()
```

Files

```
filename = 'myfile'
file = open(filename)
outfile = open(filename + "_out", "w")

line_number = 0

for line in file.xreadlines():
  line_number += 1
  print >>outfile "%d:  %s" % (line_number, line.strip())

file.close()
outfile.close()
```

For the next session

- From the manual
  - Read chapters 13, 14, 15, 16
- Continue working of Series 1