

Projet1 – TP2 : More advanced Python

Marin buttongeret, Bogdan Pasca, Kevin Perrot

{ Bogdan.Pasca } @ens-lyon.fr

September 27, 2010

Objects and Types

1. Ask the user to type a Python object. (That means numbers can be typed directly but strings must be enclosed in quotes.) Use `input` instead of `raw_input`. Check the type of the user-input in the following manner : include the line `import types` at the beginning of your script. Then compare the type of the user-input to the objects `types.IntType`, `types.FloatType` and `types.StringType`. Print "The input was an integer", "the input was a real number", "the input was a string", respectively.

Control-flow structures

1. Print all multiples of 17 that are smaller than 100. Use the `range` function in the following manner : `range(start, end, step)` where "start" is the starting value of the counter, "end" is the end value and "step" is the amount by which the counter is increased each time.
2. Print the elements of a list by iterating over them for `element in list:`. Modify the program so the user is asked for each element of the list if he wants to remove it or not. What is the problem encountered ? Give two solutions to solve this.

Dictionaries

Dictionaries are like lists whose indices are arbitrary numbers and strings. They can also be thought of as "database tables" with two columns (a "key" and a "value" column).

Creation : `dic={"me":17, "you":42, "he":13}`

Altering : `dic["me"]=16`

Appending : `dic["us"]=144`

Delete : `del dic["me"]`

Clear : `dic.clear()`

1. Experiment with dictionaries. Iterate over the keys, modify elements and play with different datatypes.

Regular expressions

Python uses a two fold process for regular expressions. First, the regular expression is compiled : `kw = re.compile(r"[a-zA-Z]")` then the search for the keyword is executed on the input string : `keyword.search("123123y21983asd")`.

1. Write the Python code to validate a date. Use regular expressions.
2. Write a regular expression that finds html tags in a file and prints them.
3. Use regular expressions to emulate the functionality of scanf.
4. Use regular expressions to emulate the global replacement option of a text editor. Use `kw.sub("replaceWith",line)`.

MISC Exercises

1. Execute the following code :

```
>>> a, b, c = 1, 1, 1
>>> while c<49:
    print c, " : ", b, type(b)
    a, b, c = b, a+b, c+1
```

What is your conclusion ?

Turtle, a Python module



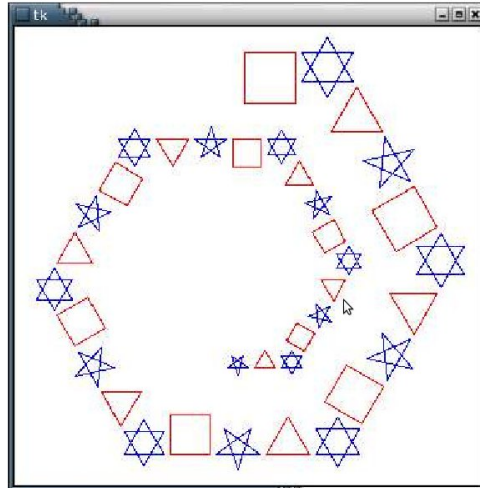
Check-out some of the turtle commands :

```
from turtle import *
forward(120)
left(90)
color('red')
forward(80)
```

For more go to : <http://docs.python.org/library/turtle.html>
Use loops to describe some star-based shapes :

```
>>> reset()
>>> a = 0
>>> while a <12:
    a = a +1
    forward(150)
    left(150)
```

1. Import the turtle module so that you are able to do simple drawings. Draw a series of equilateral triangles having different colors. First define a function `triangle` able to draw a triangle of a certain color. Use the function to draw the triangle at different places on the canvas, changing the color each time.
2. Define the corresponding functions and the loop allowing to draw :



The GUI flavor

We will use the Tkinter GUI (graphical user interface) widget set for Python. Execute the following Python script. Try to understand all the lines.

```
from Tkinter import *
window1 = Tk()
text1 = Label(window1, text='Hello World !', fg='red')
text1.pack()
button1 = Button(window1, text='Bye Bye', command = window1.destroy)
button1.pack()
window1.mainloop()
```

The next exercise will actually draw something :

```

from Tkinter import *
from random import randrange

def drawline():
    global x1, y1, x2, y2, color
    can1.create_line(x1,y1,x2,y2,width=2,fill=color)
    y2, y1 = y2+10, y1-10

def changecolor():
    global color
    pal=['purple','cyan','maroon','green','red','blue','orange','yellow']
    c = randrange(8)
    color = pal[c]

#----- Main -----
x1, y1, x2, y2 = 10, 190, 190, 10
color = 'dark green'
window1 = Tk()
can1 = Canvas(window1, bg='dark grey',height=200,width=200)
can1.pack(side=LEFT)
button1 = Button(window1,text='Bye Bye',command=window1.quit)
button1.pack(side=BOTTOM)
button2 = Button(window1,text='Draw a line',command=drawline)
button2.pack()
button3 = Button(window1,text='Change Color',command=changecolor)
button3.pack()
window1.mainloop()
window1.destroy()

```

1. Start from the above program and experiment with new shapes.
2. Modify the program so that you draw an rectangle which you can move around the canvas using 4 buttons.
3. Write a program which draws a window with a canvas and a button, as shown in the figure below. The larger green circles are the car semaphores and the smaller red circles are the pedestrian ones. Change the color of the semaphores using them button.

