

Programación Científica 1

Taller 4: Funciones e interface.

Profesores: Gilles Pieffet

1. En este taller nos vamos a conectar al servidor del grupo Sistemas Complejos para entrar y correr los programas de Python. Para conectarse entre en el terminal el comando siguiente:

```
$ ssh -Y -o ServerAliveInterval=300 -o  
ConnectTimeout=90 -l bioquimstud? 186.28.225.59
```

donde el ? corresponde a su posición en la lista de estudiantes:

```
1: ANDREA  
2: TATIANA  
3: PALOMA  
4: CARLOS A  
5: JOEL  
6: CARLOS E  
7: KAREN  
8: LUISA
```

y la clave para conectarse es su código.

2. Una vez que está conectado, entre en la carpeta swampy y en adentro del interpretador python/ipython escriba el código siguiente:

```
from TurtleWorld import *  
  
world = TurtleWorld()  
bob = Turtle()  
print bob
```

Aparece un tortuga ('Bob') que se puede mover con las funciones *fd* (forward/adelante), *bk* (backward/atras), *lt* (left turn/giro izquierda) y *rt* (right turn/giro derecho). Como la tortuga tiene lápiz, puede dibujar en la ventana cuando se mueva. Entre el siguiente código para verla en acción:

```
fd(bob, 100)  
lt(bob)  
fd(bob, 100)
```

3. Ahora modifique el programa para dibujar un cuadrado. Cuando lo tiene dibujado, guarde en un archivo "mypolygon.py" todas las instrucciones que le permitieron dibujar el cuadrado (puede experimentar y dibujar otras formas también pero por lo menos tiene que tener las instrucciones para dibujar el cuadrado). Para que la ventana no se cierre automáticamente cuando se acabe el programa, puede utilizar la función siguiente al fin del programa:

```
wait_for_user()
```

4. Re-escribe el programa que corresponde al dibujo del cuadrado, pero esta vez utilizando un bucle *for* para reemplazar instrucciones idénticas.

En los ejercicios siguientes, piense en el interés de cada una de las tareas que tiene que realizar.

5. Escribe una función llamada *cuadrado* que recibe una tortuga como argumento *t*. La función debe utilizar la tortuga para dibujar un cuadrado. Escribe una llamada de función que pasa a *bob* como argumento de *cuadrado* y corre el programa.
6. Agregue otro argumento llamado *longitud* a la función *cuadrado*. Modifique el cuerpo de la función de tal manera que la longitud de los lados sea *longitud* y cambia la llamada de función para tener en cuenta este segundo argumento. Corre el programa otra vez y luego comprueballo con varios valores de *longitud*.
7. Las funciones *lt* y *rt* hacen giros de 90° por defecto, pero se les puede dar un segundo argumento para indicar la cantidad de grados. Por ejemplo:

```
lt(bob, 45)
```

gira *bob* de 45° a la izquierda.

Escribe una función *poligono* basada en *cuadrado* que recibe un argumento extra *n* que corresponde al número de lados del polígono. Modifique el cuerpo de la función de tal manera que dibuje un polígono regular con *n* lados.

Pista: los ángulos externos de un polígono regular de *n* lados son de $360/n$ grados.

Nota: si *bob* se mueve de manera muy lenta, lo puede acelerar cambiando *bob.delay* que corresponde al tiempo entre movimientos en segundos. Un valor de 0,01 debería ser suficiente (*bob.delay* = 0.01).

8. Escribe una función llamada *circulo* que recibe una tortuga *t* y un radio *r* como argumentos y dibuja un círculo aproximado llamando a *poligono* con una longitud y un número de lados apropiados. Comprueba su función con varios valores de *r*.

Pista: encuentra el perímetro de un círculo y asegúrese que $longitud * n = perimetro$.