1 Turtle

The turtle module provides some simple graphics functionality. The Turtle function takes no arguments and returns a new turtle object.

A turtle object has several attributes:

- position: the x,y location on the screen
- heading: the direction the turtle is facing
- color, fillcolor: the color the turtle will draw and use to fill in drawn shapes
- pen position: the pen can be up or down

Here are some of the methods available on turtle objects:

Name	Arguments	Description
forward	distance	Moves the turtle forward
backward	distance	Moves the turtle backward
left	angle	Turns the turtle counterclockwise
right	angle	Turns the turtle clockwise
up	none	Lifts up the pen
down	none	Puts the pen down
color	color name (string)	Changes the pen color
fillcolor	color name (string)	Changes the color used to fill a polygon
heading	none	Returns the current heading
xcor	none	Returns the current x position
ycor	none	Returns the current y position
goto	x, y	Moves the turtle to position x, y
begin_fill	none	Marks the start of a polygon
end_fill	none	Closes the polygon and fills it with fillcolor
setheading	angle	Sets the orientation of the turtle to given angle

A complete listing is here: http://docs.python.org/2/library/turtle.html#turtle-methods.

This example draws a backwards L.

```
>>> import turtle
>>> franklin = turtle.Turtle()
>>> franklin.forward(100)
>>> franklin.left(90)
>>> franklin.forward(300)
>>> franklin.ycor()
300.0
>>> franklin.heading()
90.0
```

1 Importing from modules

Many, many functions are available in python. Some functions like `max` are builtin and available all of the time. Other functions are organized into *modules* and must be *imported.* A **module** is a file containing a collection of related functions and possibly variables.

To import a module, we use the import statement. For example, we can import the `math` module.

```
>>> import math
```

To access the functions (and variables) inside a module, we use the **dot operator**:

```
>>> math.pi
3.141592653589793
>>> math.sqrt(100)
10.0
```

More information about the functions and variables in the `math` can be found here: `http://docs.python.org/2/library/math.html`.

2 Objects

Every object in python has three properties: (1) an identity, (2) a type, and (3) a value, or possibly a set of values called attributes. The identity of the object is some number that is guaranteed to be unique: no two objects in your program will have the same identity. In python the identity is simply the memory address where the object is stored.

Everything in python is an object. Consider `x = 4`. The variable `x` refers to an object. The value of the object is 4. Its type is `int`. Its identity is its memory address. In fact, the builtin function `id` will tell it to you:

```
>>> x = 4
>>> id(x)
4298181232
```

Some objects have not just a single value but a collection of attributes. For example, a turtle object has several attributes: position, heading, color, etc.

Some types of objects have **methods**. Methods are like functions that are attached to the object. Methods allow us to ask the object to do something.

Some objects are **mutable**, which means it is possible to change their value/attributes. A turtle object is mutable because you can move the turtle and change its position. Some objects are **immutable**. An `int` is immutable: the number 4 is a distinct object and cannot be changed into another number. (The statement `x = x + 1` changes which object `x` refers to but it does not change the object.)

3 Turtle

Examples done in class.

- Drawing a triangle based on user input.

```
import turtle, math
height = int(raw_input("What is your triangle's height? "))
base = int(raw_input("What is your triangle's base? "))

franklin = turtle.Turtle()
franklin.color('red')

franklin.begin_fill()

franklin.forward(base)
radians = math.atan(float(height)/base)
degrees = math.degrees(radians)
turn_angle = 180 - degrees
franklin.left(turn_angle)
hypo_length = math.sqrt(base**2 + height**2)
franklin.forward(hypo_length)
franklin.left(180 - (180 - (90 + degrees)))
franklin.forward(height)

franklin.end_fill()

turtle.done() # IMPORTANT: gotta remember to do this at the end
```

- What is the output of this program?

```
import turtle
t1 = turtle.Turtle()
t2 = turtle.Turtle()
t2.color('red')
t1.up()
t1.forward(100)
t1.right(90)
t1.down()
t2.left(90)
t1.forward(25)
t2.forward(t1.xcor())
turtle.done() # IMPORTANT: keep window open until you close it
              #(or click anywhere on it)
```