CS 100: Roadmap to Computing
Fall 2014
Lecture 02: Fun With Turtles

Python Data Types (3): Turtle Graphics

- Objects and Classes
- Python Standard Library
- Turtle Graphics

Python Standard Library

- In addition to built-in data types (number, Boolean, string, list) Python has a large library of other data types – the Python Standard Library

- You can find a list of these 'modules' here

  Help -> Python docs -> Global Module Index

- Each module is a file of Python code that contains definitions of one (or more) data type(s), functions (methods) that you can perform on objects of that type, and possibly data (like the value of pi)

Quick task: Look up three modules that have names that interest you and see what data types and functions they contain.

Hint: you might try random, urllib or pickle, for example

Share the one you like best with the person sitting next to you.

Turtle Graphics Module

- Find and open the documentation for the turtle module
- Look up turtle info in this document as we discuss turtle graphics
- Turtle graphics are a simple but powerful way to draw things on a coordinate plane
- To get started, import the turtle module.
- The Python Standard Library is contained in the standard distribution, but you must import any module that you want to use

```
>>> import turtle
```

Turtle Graphics Module

- Turtle graphics are a simple but powerful way to draw things on a coordinate plane, using a drawing pen (a turtle)
- Open the documentation for the turtle module, and refer to it as we discuss turtle graphics
- The Python Standard Library is contained in the standard distribution, but you must import any module in it before you can use it
- To get started, import the turtle module (below)

```
>>> import turtle
>>>
```

Turtle Graphics Module

- The turtle module defines a some new classes of graphical things
- Once you've imported the turtle module, you can create a graphics screen and a turtle (a whimsical name for a drawing pen), using their constructors
- Note: The constructor syntax is

  variableName = moduleName.ClassName()

Screen constructor

```
>>> import turtle
>>> aScreen = turtle.Screen()
>>> shelly = turtle.Turtle()
```

Turtle constructor

Moving a Turtle

- A turtle has a position and an orientation on a graphics screen
- Change shelly’s position with a forward (or back) statement
- Change shelly’s orientation with a right or left statement

```
>>> import turtle
>>> aScreen = turtle.Screen()
>>> shelly = turtle.Turtle()
>>> shelly.forward(100)
>>> shelly.right(90)
```

A Fancier Turtle

- A turtle also has color and width attributes that you can change
- A method (function) that applies to a particular object uses the dot operator, with the syntax

objectName.method(parameterList)

```
>>> shelly.color('blue')
>>> shelly.width(10)
```

A Fat Blue Triangle

- Save this example as a Python file and run it

```
blueT = Turtle()
blueT.color('blue')
blueT.width(10)
blueT.forward(100)
blueT.right(120)
blueT.forward(100)
blueT.right(120)
blueT.forward(100)
blueT.right(120)
```

Some turtle methods

Usage	**Explanation**
forward() | bk()	move the turtle
right() | left()	rotate the turtle
circle()	lstdraw a circle
up() | down()	raise/lower the drawing pen
goto()	move to x, y coordinate
setheading()	set turtle orientation
showturtle() | hideturtle()	set turtle visibility
color()	set drawing color
width()	set line width

What we have learned

- A Python module is a file that Python code – usually defining one or more related new types of things ('classes').
- Each class has a constructor method to create new objects in that class
- Each class will have methods (functions) for doing things with objects of that type. A method is invoked using the dot ('.') operator.
- By convention, a class name is capitalized, object and method names are lower case