CIS 122

Now you're thinking with turtles!
Logistics

● Most homework received
  ○ Try to get them in on time
  ○ Only 3 late credits
  ○ Solutions will be posted when they're all in

● Not going to review homework
  ○ If you have questions, come ask me
Logistics

● Midterm next Monday
  ○ Recap on Thursday
  ○ Review session on Friday
  ○ Study guide is coming

● There will be a homework due this week
  ○ Shorter than usual
  ○ Only 2 real problems (and a bit of extra credit)

● Due Date
  ○ Scheduled for Sunday night
  ○ Would you prefer earlier?
Dueling Paradigms

• Different philosophies towards programming
  ○ Functional Programming
  ○ Imperative Programming
  ○ Object Oriented Programming
Dueling Paradigms

- Functional Programming
  - Functions exist to return values
  - Calling a function should not change the world

- No side effects
  - Reassigning variables
  - Printing information

- Idempotent
  - Calling function multiple times does not change result
def foo(x):
    x = x+1
    return x

a = foo(1)
a = foo(1)
a = foo(1)
a = foo(1)

print a
Dueling Paradigms

- Imperative Programming
  - Functions exist to do work
  - May or may not return useful information

- Functions can change the world
  - Variables may hold different values afterwards
  - May have printed out messages

- Non-idempotent
  - Calling function multiple times may yield different results
Dueling Paradigms

```
x = 0

def foo():
    x = x+1
    return x

a = foo()
a = foo()
a = foo()

print a
```
Turtle Graphics

- Graphical Output
  - Turtle Drawing
  - Imperative

- The turtle module contains line drawing functions
  - You control a "turtle"
  - Tell it to go forwards, backwards, left, right
  - Kind of like an etch-a-sketch

- Turtle functions don't return values
  - (well, technically, they return None)
  - They issue commands to the turtle
Turtle Graphics

- IDLE doesn't cooperate with turtle graphics
  - Need to open IDLE in a special mode
- Open the command prompt
  - Terminal for macs
- Enter the IDLE path followed by "-n"
  - C:\Python27\Lib\idlelib\idle.pyw -n (lab computers)
  - <somewhere else> -n (pc laptops)
  - idle -n (mac laptops)
- IDLE should start up with a special message
  - ==== No Subprocesses ==== 
Turtle Graphics

● First, import the turtle module
  ○ import turtle

● Now, you're ready to draw!
  ○ turtle.forward(dist)
  ○ turtle.backward(dist)
  ○ turtle.left(angle)
  ○ turtle.right(angle)

● And one more really useful function
  ○ turtle.reset()
What does this code do?

turtle.forward(100)
turtle.left(120)
turtle.forward(100)
turtle.left(120)
turtle.forward(100)
Turtle Graphics

● What does this code do?

    turtle.forward(100)
    turtle.left(120)
    turtle.forward(100)
    turtle.left(120)
    turtle.forward(100)

● Draws an equilateral triangle
  ○ Equilateral triangle has 60° angles
  ○ Why did we turn left 120°?
Turtle Graphics Practice

- Write code to draw this shape
  - Write it in a file
  - Start with `turtle.reset()`
Turtle Graphics Practice

- Writing out the same code is a pain
  - Programmers are lazy
  - Never do the same work twice

- Write a square function
  - square(length)
  - Draws a square with sides of the given length

- Use your square function to draw our shape again