Outline of today's session

- Introductions (again): students, assistants and instructor
- Questions?
  - Reading quizzes? Homework problems?
- Counted loops: *RANGE* expressions in *FOR* statements
- Using objects and classes
  - Constructing
  - Using methods
  - Using instance variables
- Lots of exercises!

```
def main():
    """ Calls a function (chaos). """
    chaos()

def chaos():
    """
    Computes and prints a chaotic sequence of numbers,
    as a function of a number input from the user.
    """
    print('This function illustrates a (possibly) chaotic sequence.')
    input_string = input('Enter a number between 0 and 1: ')
    x = float(input_string)

    for k in range(10):
        x = 3.9 * x * (1 - x) # Try constants other than 3.9 if you like
        print(k, ':', x)

    print('Examine the sequence of numbers printed in the right column.')
    print('Does it appear chaotic?')
```

Questions:

- On the reading?
- On homework?
- On the code you studied?
- On anything?

RANGE expressions in FOR statements

- **SVN ~ Checkout** today's project: **`Session03_LoopsAndUsingObjects`**
- Study m1 with your instructor
  - Ask questions! Be curious!
- Then do m2

Do these NOW. If you are still working on Session02 modules, come back to those later today

What are objects?

- ***Traditional*** view, in languages like C
  - Data types are *passive*
    - They have values
    - There are operations that act on the data types
      - The data type itself cannot do anything
- ***Object-oriented*** view, in languages like Python (and most other modern languages)
  - Have *objects*, which are *active* data types. Objects:
    - ***Know stuff – they contain data***
      - The data that an object holds are its ***instance variables*** (aka ***fields***)
    - ***Can do stuff – they can initiate operations***
      - The operations that an object can do are its ***methods***

How do you *use* objects?

Recall that objects:
- Know stuff (***instance variables***, aka ***fields***)
- Can do stuff (***methods***)

- To ***construct*** an object:

```
win = zg.GraphWin()
point1 = zg.Point(500, 450)
line = zg.Line(point1, zg.Point(30, 40))
circle = zg.Circle(point1, 100)
```

Constructor:
- Call it like a function, using the name of the ***class***
- Style: Class names begin with an ***uppercase*** letter
- The constructor ***allocates space*** for the object and does whatever ***initialization*** the class specifies

Quiz

- To *ask an object to do something*, i.e. to apply its ***methods*** to it:

```
point1.draw(window)
line.move(45, -60)
x = point1.getX()
center = circle.getCenter()
```

Method call:
- Use the ***dot notation:***

`Who.Does_What(With_What)`

Just like a function call, except that the method has access to the object invoking the method.
So the object is an ***implicit argument*** to the method call

- To reference what the object knows (its ***instance variables***, aka ***fields***):

```
point1.x    circle.p1    circle.p2
```

Instance variable (aka *field*) reference:
- Use the ***dot notation*** but ***without parentheses*** `Who.Has_What`

Using Objects

1. Study and work through m3 with your instructor
   - Ask questions! Be curious!
2. Do the first problem in m4
3. Return to Session02 modules you have not yet completed (if any)
4. Complete the modules from Session03 now and for homework