Announcements

- Course Evaluations
  - Very important! They're used by:
    - Future students in choosing courses
    - Faculty for improving the course
    - University for evaluating faculty
- For CSC108, the evaluations will only be done online (not on paper).
- Please start thinking about what you'll say.

Classes

Using objects

- You have used many kinds of objects that are built in to Python: int, str, list, dict, etc.

```
total = 0           # Create a new int object.
name = “Priya”    # Create a new str object.
```

- And others that are part of modules you had to import, such as media.

```
import media
p = media.get_pixel(pic, 3, 92)   # Returns a Pixel object.
```

Defining new types of objects

- Consider a calendar program. One of the basic things you'd need to keep track of is events.
- What should we keep track of for a single event?
- We could define individual variables for each of these, but we'd need to do that for every event. Messy.
- Or we could define a new type of object to bundle these together.
- We have int, str, dict, list, etc.
  Now we can have Event!

Encapsulation

- You've used the media library a lot.
- How is a pixel represented? What variables store the information about it? What types are they?
- In order to use a pixel, **we don't need to know** these internal details. We just need to know how to call the methods and functions that work with pixels.
- How do we find out? dir and help.
(Here you see the value of a good docstring.)

- So the programmer who uses pixels can be blissfully ignorant of the details.
- This is good! Why?
- It goes the other way too: The programmer who writes/updates/debugs/modifies/extends the pixel code **doesn't need to know** how others are using it.
- This is good too! Why?
- Encapsulation: keeping data, and the code that uses it, in one place and hiding the details.
- A simple but powerful idea. It helps us manage complicated code.

Summary

Underscore methods

- What do the underscores signify?

Constructors

- What is the name of every constructor?
- What does a constructor do?
- When does a constructor "happen"?
- What if you don't define your own?

Summary

self

- What is it and what does it refer to?

`__str__`

- When is it called?
- What if you don't define your own?

`__cmp__`

- When is it called?
- What if you don't define your own?