Announcements

- HW7 grades emailed yesterday
- HW10 is posted
    - due Friday
- Last day of class (in 2 weeks) is (a review session
- Final Wednesday - (9am?)
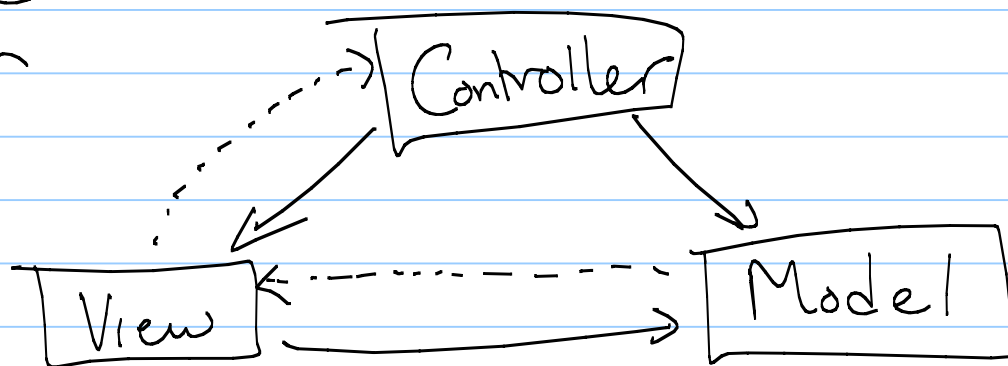
Designing Python GUIs: MVC

The Model—View—Control Pattern : 1979

the data
and business
rules part
of the
application

what the
user
sees

how the
user face
works

```
         ┌──────────────┐
     ╭─ ─>│ Controller   │ ─╮
     ┊    └──────────────┘  │
     ┊      ╱          ╲    ▼
┌────────┐╱              ╲ ┌────────┐
│  View  │◀- - - - - - - - │ Model  │
└────────┘ ─────────────▶  └────────┘
```

MVC in Tkinter

The model is often a completely separate class, which stores underlying information somehow

The view + control are combined: view is widgets + display, control is callback (so widgets call functions to model).

Example: Stop watch

2 classes in this:

(1) The Stopwatch class will store:

self. running — True/False
self. start time
self. elaptime

→ no Tkinter!

Functions: start ()
stop ()
reset ()
elapsed time ()

② The UserInterface class:

This will make the window
   + buttons.
- __init__ will set up the window.

- Button widgets will be set up
   to call the functions from
   the Stopwatch class.

Main script: Simply creates variables
   for 2 classes + starts the
   main loop.

See code

Homework

Similar setup: making a GUI addressbook.
The controller (like StopWatch) is
already written.
(You will _Not_ change this class!)

You will write a main (plus a few
classes) to give display - control.

Today: Work Day

Find a partner!
Get started on writing control/GUI.