Properties

Instance Variables

- start with no underscores
  - just an instance variable with no restrictions
  - accessible externally

Instance Variables

- start with a single _
  - indicates meant for internal use
  - not enforced
  - convention
- start with two _ (e.g. __)
  - triggers name mangling
  - enforced

Other languages

- many have public, private and protected
- python just has conventions
- getters/setters carry over from other languages

Getters/Setters - Advantages

- allows changing internals without changing external interactions
- add extra checks on values

Getters/Setters - Downsides

- not particularly Pythonic
- looks messier/more boiler plate

property

- makes methods that look like attributes
- allows writing code to use direct member access
- allows modifying getting/setting without changing interface

property -- implementation

- define functions `_set_varname`, `_get_varname`
- make property with

  ```
  varname = property(_get_varname, _set_varname)
  ```

- can add any logic we want in setting/getting

property - option 2 -- decoarator

```
@property
def varname(self):
    # getter code
    return self._varname

@varname.setter
def varname(self, varname):
    # other check code
    self._varname = varname
```