

PROPERTIES

INSTANCE VARIABLES

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