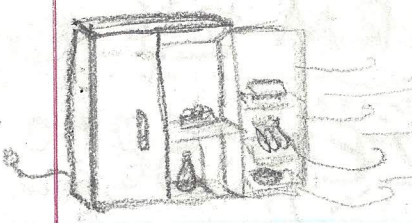
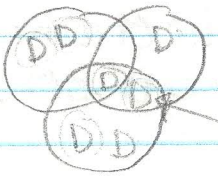


FrozenSets!

Crista Falk
3/13/17
A2

SETS can't "contain" mutable objects, BUT sets are mutable themselves.

```
>>> x = set([1, 2, 3])
>>> x.add(4)
>>> x
set([1, 2, 3, 4])
```



: intersection

FROZENSETS - like sets, except these are immutable (can't be modified)

```
>>> F = frozenset(['a', 'b', 'c'])
```

✗ F.add(4) → ERROR: 'frozenset' obj has no attribute 'add'

SIMPLIFIED NOTATION

- can use curly braces to define sets instead of built-in func

```
>>> mySet = {"woah", "easy", 42}
>>> mySet
set(["woah", "easy", 42])
```

SET OPERATIONS:

- add(element) - adds element if immutable, not in already
- clear() - all elements removed from set
- copy() - creates a shallow copy, gets returned
  - set X = set Y creates a POINTER to same OS
  - set X = setY.copy() is what you want!
- difference() - returns diff of 2 or more sets (as a new set)
  - x.difference(y) or x.dif...(y).dif...(y)
  - could use operator x-y
- difference_update() - basically x = x-y
- discard(el) - removes 'el' element from set if there
- remove(el) - a KeyError will be thrown if el not in set
- intersection(s)
  - x.intersection(y) returns elements in both x & y

- isdisjoint() - True if 2 sets have a null intersection
- issubset() - checks if set is a proper subset of a set
- issuperset() - a but superset
- pop() - removes & returns an arbitrary set element