Equality

And how it's not as simple as you think

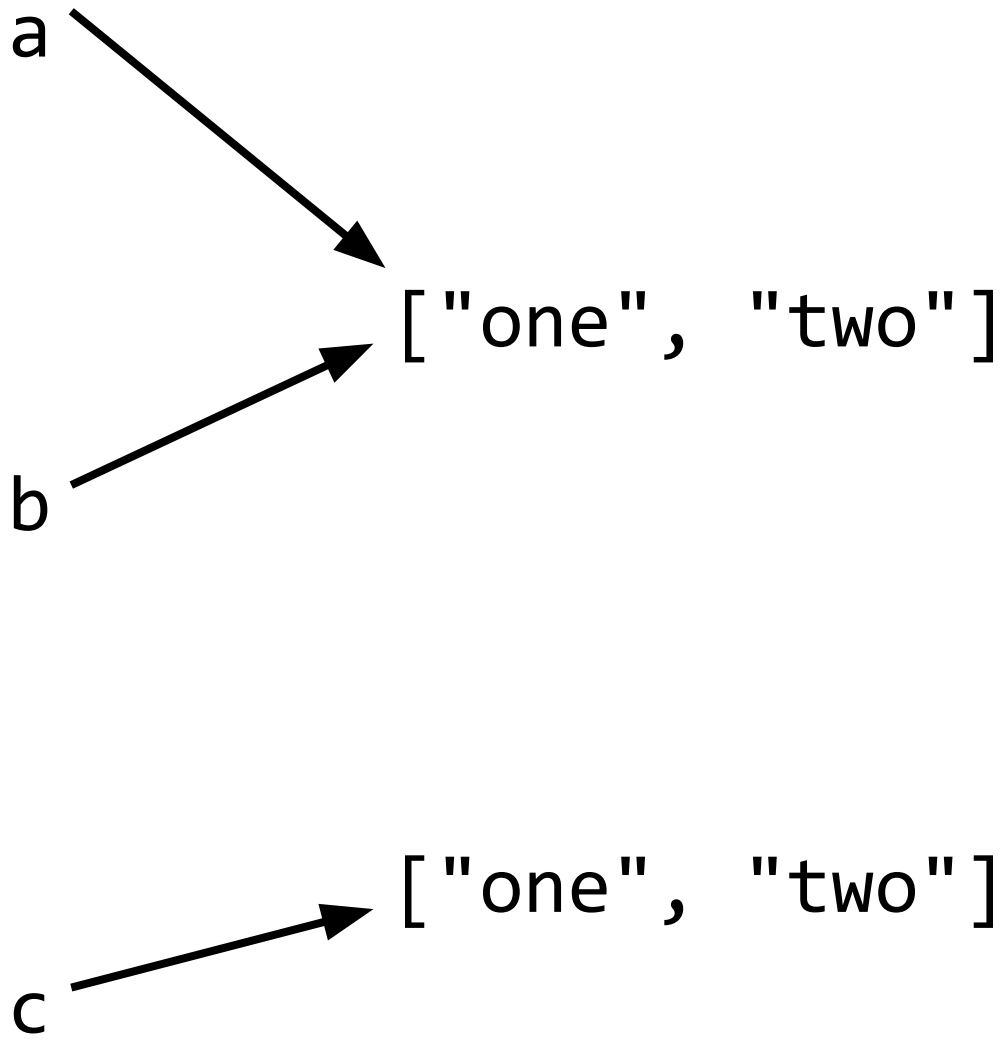
Actually this is still a Computer Science topic

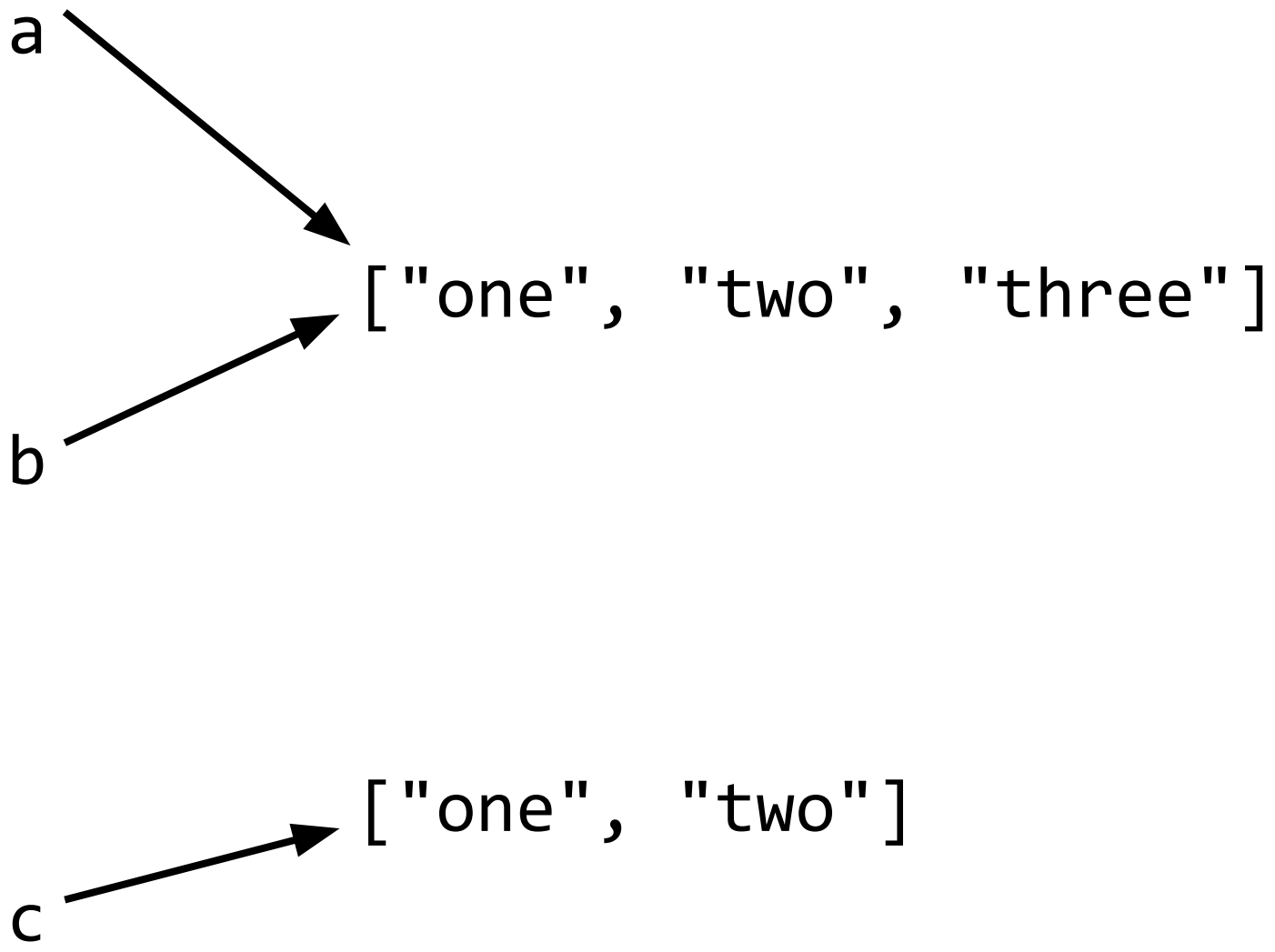
Try these...

```
>>> a = 1
>>> 1 == a
>>> a == 1
>>> a == 5
>>> b = a
>>> a == b
>>> b += 1
>>> a == b
```

Try these...

```
>>> a = ["one", "two"]
>>> a == ["one", "two"]
>>> b = a
>>> c = ["one", "two"]
>>> a == b
>>> a == c
>>> b == c
>>> b.append("three")
>>> a == b
>>> a == c
```


a
["one", "two"]
b
["one", "two"]
c


a
["one", "two", "three"]
b
["one", "two"]
c

Two kinds of equality

- **Value equality** -- Do the two values **mean** the same thing?
  - Value equality is represented in Python with the == operator.
- **Reference identity** -- Are the two sides **actually** the same thing?
  - Reference identity is represented in Python with the is operator

That depends on what the meaning of 'is' is.

```
>>> a = ["one", "two"]
>>> a is ["one", "two"]
>>> b = a
>>> c = ["one", "two"]
>>> a is b
>>> a is c
>>> b is c
>>> b.append("three")
>>> a is b
>>> a is c
```

What I want...

- I have some complicated structure in the variable `a`.
- I want to get something in `b` so that `b == a`
- But I don't want `b is a`.
- `b['thing'] = 'stuff'` shouldn't affect `a` at all.

Copying for Independence

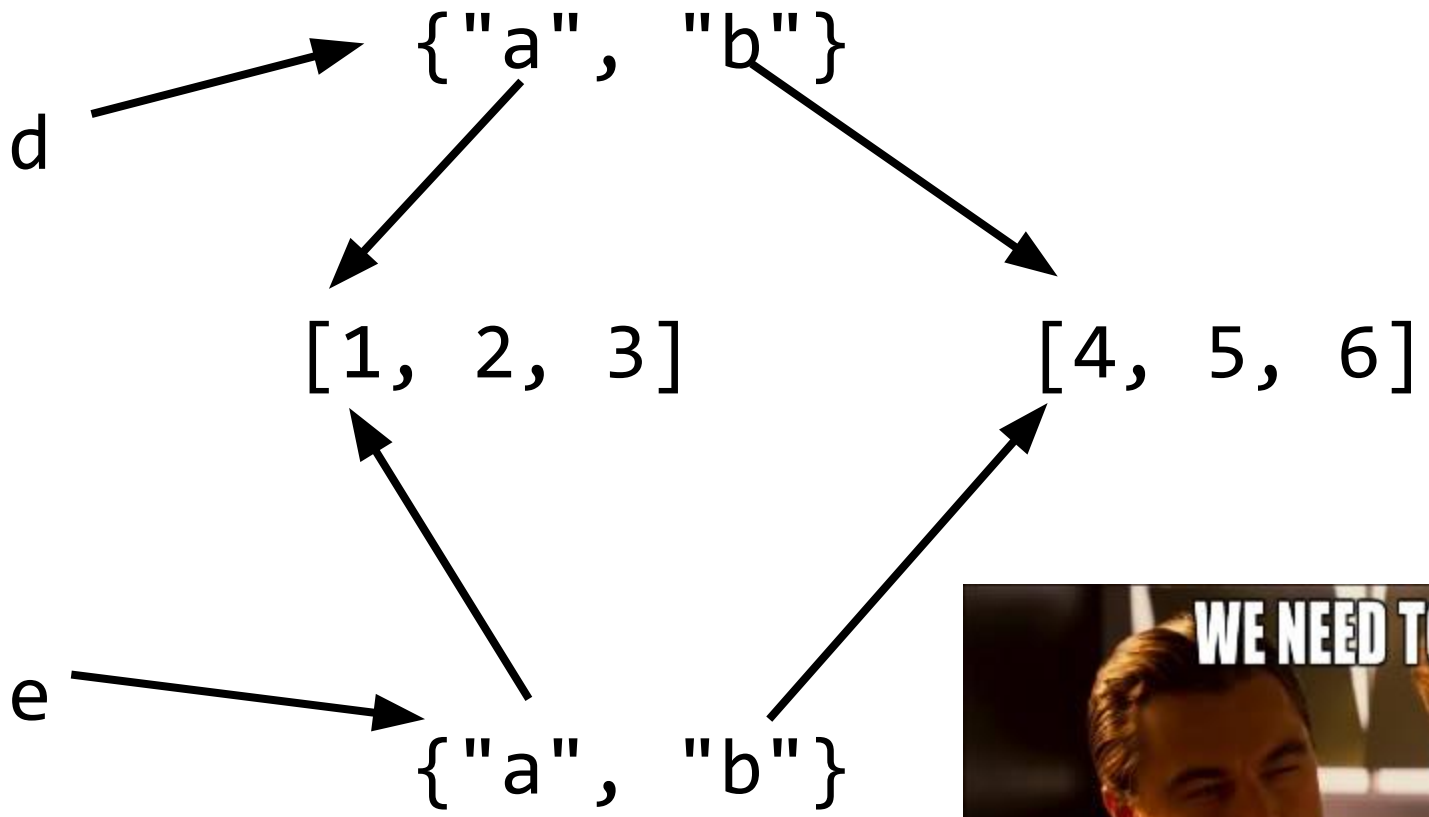
```
>>> import copy
>>> a = {"dictionary" : 10,
         "of" : 2,
         "lengths" : 7}
>>> b = copy.copy(a)
>>> b == a
>>> b is a
>>> b["word"] = 4
>>> b == a
```

But copying is complicated too

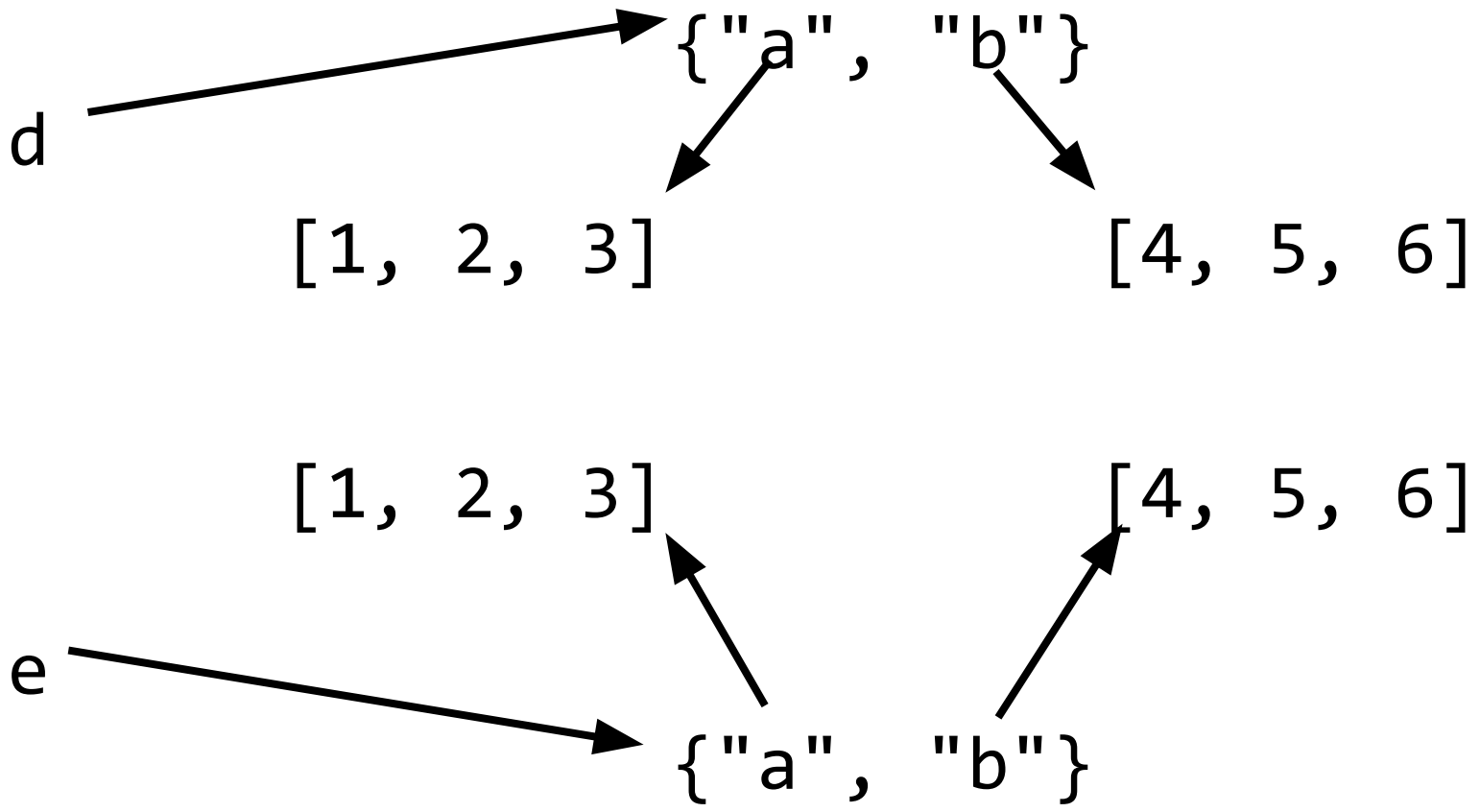
```
>>> d = {"a" : [1, 2, 3],
         "b" : [4, 5, 6]}
>>> e = copy.copy(d)
>>> d["b"]
  [4, 5, 6]
>>> d["b"].append(7)
>>> d["b"]
  [4, 5, 6, 7]
>>> e["b"]
  [4, 5, 6, 7]
```

What happened?

`copy.copy()` is a <u>shallow</u> copy.



`copy.deepcopy()` is a deep copy.



copy() vs. deepcopy()

```
>>> d = {"a":[1, 2, 3], "b":[4, 5, 6]}
>>> e = copy.copy(d)
>>> f = copy.deepcopy(d)
>>> d is e
  False
>>> d is f
  False
>>> d["a"] is e["a"]
  True
>>> d["a"] is f["a"]
  False
```

How can you use this in Scrabble?

- This week we'll be validating moves
- Part of validating a move: is the board valid after you make the move?
- ... but you don't want to mess up the board that someone gave your function
- ... so you copy it.