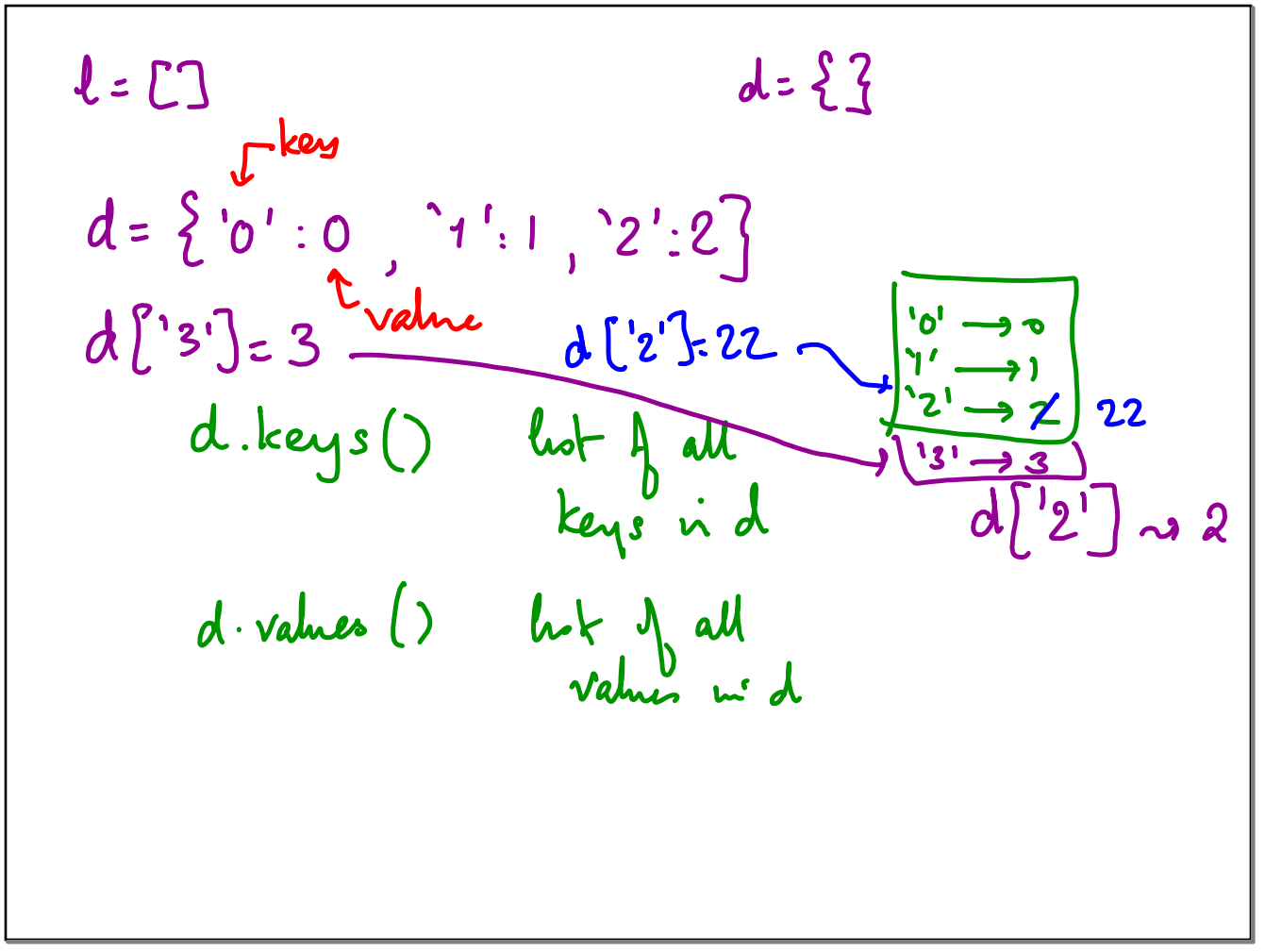
Classes & Objects

List `l = [3,4,5]`

`x = ["hello", 2.4]`

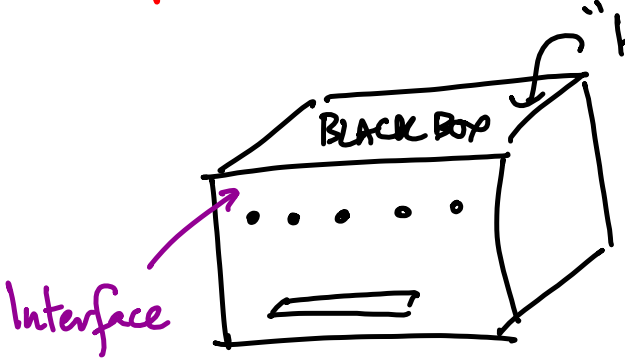
`0 1`

Dictionary `{key1 : value1, key2 : value2, ... }`


l = []
d = {}
key
d = {'0' : 0 , '1' : 1 , '2' : 2}
value
d['3'] = 3
d['2'] = 22
'0' → 0
'1' → 1
'2' → 2
22
'3' → 3
d['2'] ⇝ 2
d.keys() list of all keys in d
d.values() list of all values in d

Classes, Objects

Abstract data type



"Axioms" that interconnect the allowed operations & their return values

ADT

Implementing an ADT

Internal storage

How to execute functions

Given a template / blueprint of ADT

Represent points in 2D

Operations such as:

- Distance between 2 pts ⇝ distance(p1, p2)
- Translate a point
- Distance from origin

}

p1.distance(p2)

or

p2.distance(p1)

In python:

Use class to define template for an ADT

Instantiate classes as separate objects

```
class Point:
    ≡
    ≡
    ≡
    def setvalue(--)
        ≡ Update internal
          state
```

```
p1 = Point()
p2 = Point()
  :
p1.setvalue(6,7)
```

```
class Point:

    def setvalue(self,a,b):
        self.x = a
        self.y = b

        self.r = sqrt(a**2 + b**2)
        self.theta = arctan(b/a)
```

p1.setvalue(6,7)

p1.x = 6
p1.y = 7

Won't work if we switch to r, θ repr.

Can't I set up a point when I create it?

```
p1 = Point(6,7)
```

```
class Point:
    def __init__(self,a,b):
        self.x = a
        self.y = b
```