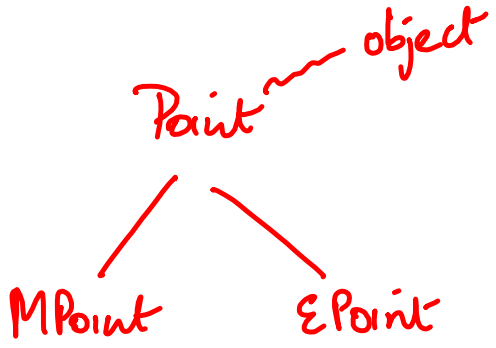
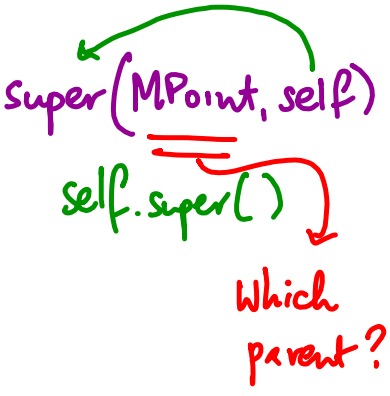
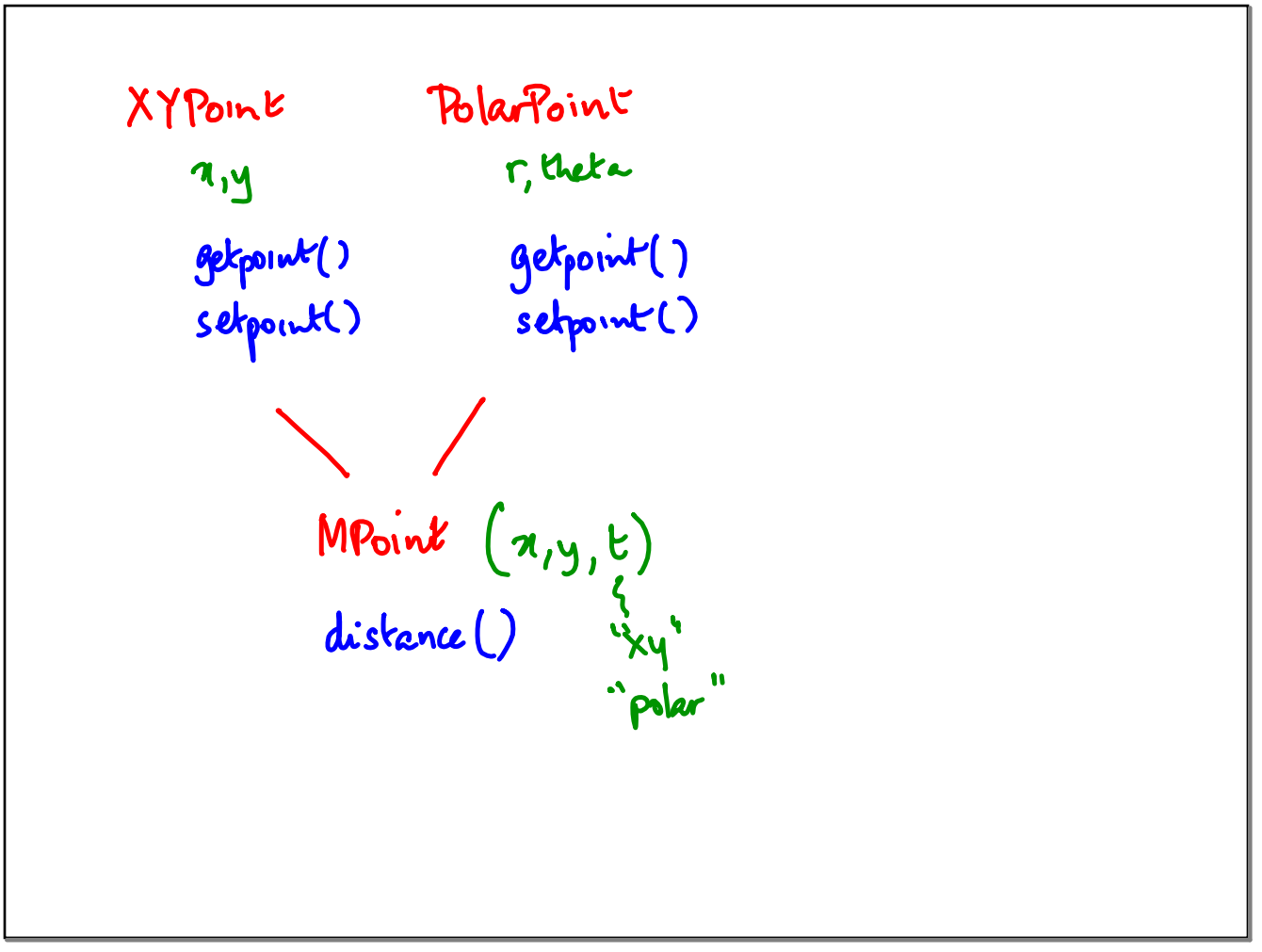
Inheritance ...

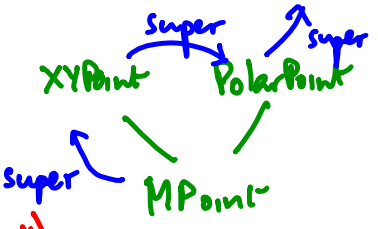
Old type class vs New type class



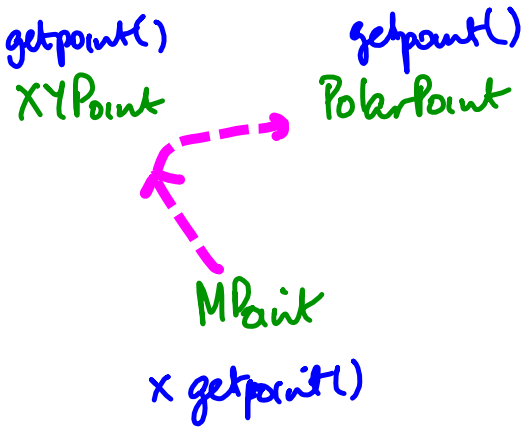



XYPoint
x,y
getpoint()
setpoint()
PolarPoint
r, theta
getpoint()
setpoint()
MPoint (x,y,t)
distance()
"xy"
"polar"

Multiple Inheritance




```
class MPoint(XYPoint, PolarPoint):
    def __init__(self, a=0, b=0, t="xy"):
        if t != "xy":
            t = "polar"
        self.ptype = t
        if self.ptype == "xy":
            super(MPoint, self).__init__(a,b)
        else
            super(XYPoint, self).__init__(a,b)
```



Need to check ptype & call the correct getpoint()

Even if an MPoint is set up using --init-- of PolarPoint, it remains of type MPoint

"Magic" functions

__init__ automatically called when object is created

__str__ implicitly called when str(o) is used "(3,4)"

__repr__ also returns a string "MPoint(3,4)"

__add__ p+q ~> calls p.__add__(q)