

```
1 import gc
2
3 class Balance:
4     radToDeg = 57.3 # radians to degrees, really just another scaling factor
5
6     def __init__(self, lMotor, rMotor, imu, dt):
7         self.pidL = lMotor
8         self.pidR = rMotor
9         self.imu = imu
10        self.dt = dt
11
12        # Working PID Constants
13        self.kp = 219
14        self.ki = 45
15
16        self.mkp = 0.045
17        self.mki = 0.5
18
19        # the actual setpoint (takes into account position feedback)
20        self.setPoint = 0.07
21        # the upward angle if at starting position (no position feedback)
22        self.basePoint = 0.07
23        self.balancing = False
24        self.count = 0
25        # integrator state
26        self.integ = 0
27
28        # set PI constants
29        def set_balance_pi(self, p, i):
30            self.kp = p
31            self.ki = i
32
33        def set_motor_pi(self, p, i):
34            self.mkp = p
35            self.mki = i
36
37        # for keeping track of how long it has been balancing
38        def increment_count(self):
39            self.count += 1
40
41        def do_balance(self):
42            angle = (self.imu.euler()[2] - 90) / self.radToDeg
43            print(angle)
44
45            # if relatively straight up
46            if (abs(angle) < 0.1):
47                # and has been held up for 3 seconds while not actively balancing
48                if (self.count > 3 and not self.balancing):
```