

```
1  from pybricks import ev3devices
2  from pybricks.parameters import Direction, Stop
3  import pybricks.tools
4
5
6  # Wrapper class for motor to allow for run to stop without quitting program
7  class Motor():
8      def __init__(self, port, config,
9                  positive_direction=Direction.CLOCKWISE,
10                 gears=None):
11          self.config = config
12          self.m = ev3devices.Motor(port, positive_direction=positive_direction,
13                                   gears=gears)
14          self.control = self.m.control
15
16      def angle(self):
17          return self.m.angle()
18
19      def speed(self):
20          return self.m.speed()
21
22      def reset_angle(self, angle):
23          self.m.reset_angle(angle)
24
25      def hold(self):
26          self.m.hold()
27
28      def run(self, speed):
29          self.m.run(speed)
30
31      def run_time(self, speed, time, then=Stop.HOLD, wait=True):
32          self.m.run_time(speed, time, then, False)
33          while wait and self.control.done() == False and self.config.state.getState() != 3:
34              pybricks.tools.wait(10)
35          if then == Stop.HOLD:
36              self.m.hold()
37          elif then == Stop.BRAKE:
38              self.m.brake()
39
40      def run_angle(self, speed, rotation_angle, then=Stop.HOLD, wait=True):
41          self.m.run_angle(speed, rotation_angle, then, False)
42          while wait and self.control.done() == False and self.config.state.getState() != 3:
43              pybricks.tools.wait(10)
44          if then == Stop.HOLD:
45              self.m.hold()
46          elif then == Stop.BRAKE:
47              self.m.brake()
48
49      def run_target(self, speed, target_angle, then=Stop.HOLD, wait=True):
50          self.m.run_target(speed, target_angle, then, False)
51          while wait and self.control.done() == False and self.config.state.getState() != 3:
52              pybricks.tools.wait(10)
53          if then == Stop.HOLD:
54              self.m.hold()
55          elif then == Stop.BRAKE:
56              self.m.brake()
57
58      def run_until_stalled(self, speed, then=Stop.COAST, duty_limit=None):
59          self.m.run_until_stalled(speed, then, duty_limit)
60          if then == Stop.HOLD:
61              self.m.hold()
62          elif then == Stop.BRAKE:
63              self.m.brake()
```

```
64
65     def track_target(self, target_angle):
66         self.m.track_target(target_angle)
67
68     def dc(self, duty):
69         self.m.dc(duty)
70
71     def stop(self):
72         self.m.stop()
73
```