```
import random
from environment import Agent, Environment
from planner import RoutePlanner
from simulator import Simulator
import pygame

class LearningAgent(Agent):
    """An agent that learns to drive in the smartcab world."""

    def __init__(self, env):
        super(LearningAgent, self).__init__(env)  # sets self.env = env, state = None,
next_waypoint = None, and a default color
        self.color = 'red'  # override color
        self.planner = RoutePlanner(self.env, self)  # simple route planner to get
next_waypoint
        self.state = None
        # TODO: Initialize any additional variables here

    def reset(self, destination=None):
        self.planner.route_to(destination)
        # TODO: Prepare for a new trip; reset any variables here, if required

    def update(self, t):
        # Gather inputs
        self.next_waypoint = self.planner.next_waypoint()  # from route planner, also
displayed by simulator
        inputs = self.env.sense(self)
        deadline = self.env.get_deadline(self)

        # TODO: Update state
        self.state = self.next_waypoint

        # Done: Select action according to your policy
        action = [None, 'forward', 'left', 'right']

        q = [self.getQ(self.state, action) for a in self.actions]
        maxQ = max(q)
        count = q.count(maxQ)
        if count> 1:
            best = [i for i in range(len(self.actions)) if q[i] == maxQ]
            i = random.choice(best)
        else:
            i = q.index(maxQ)

            #http://stackoverflow.com/questions/306400/how-do-i-randomly-select-an-item-
from-a-list-using-python
```

```
        # Execute action and get reward
        reward = self.env.act(self, action)

        # TODO: Learn policy based on state, action, reward

        print "LearningAgent.update(): deadline = {}, inputs = {}, action = {}, reward =
{}".format(deadline, inputs, action, reward)  # [debug]


def run():
    """Run the agent for a finite number of trials."""

    # Set up environment and agent
    e = Environment()  # create environment (also adds some dummy traffic)
    a = e.create_agent(LearningAgent)  # create agent
    e.set_primary_agent(a, enforce_deadline=False)  # specify agent to track
    # NOTE: You can set enforce_deadline=False while debugging to allow longer trials

    # Now simulate it
    sim = Simulator(e, update_delay=0.5, display=True)  # create simulator (uses
pygame when display=True, if available)
    # NOTE: To speed up simulation, reduce update_delay and/or set display=False

    sim.run(n_trials=100)  # run for a specified number of trials
    # NOTE: To quit midway, press Esc or close pygame window, or hit Ctrl+C on the
command-line


if __name__ == '__main__':
    run()
```