The implementation of the Q-learning algorithm was pretty straight-forward and followed the formulas and procedures discussed in class.
The only adjustments have been made inside of the QTable class as required. For the representation of the table, I decided to use a dictionary which stores unique combinations of the x and y coordinates representing the state and an action, with the Q-value as the dictionary value.
The get_q method simply returns the value of the state and action from the dictionary, or 0 if the keys do not exist. On the other hand, the get_q_row returns the maximum Q-value and the respective action resulting in it by iterating over all dictionary entries of the state coordinates.
The set_q method simply updates the value of the state and action dictionary key.
In the learn_episod method, we are initializing a random state, and them applying the formula for Q-learning updating until we reach the end-state. At the end of each iteration, the resulting state becomes the new state to be updated in the next iteration. Finally, the learn method simply iterates the learn_episodes method over the desired number of episodes.
The __str__ method creates a new string to accurately display the Q-table board as described in the problem specification.
For the sake of testing, I have also added a play method to see how our agent solves the maze after Q-learning of 100 episodes, and the results were exactly as expected, with our agent choosing the optimal path.
As for testing, the code was ran a number of times to check that the resulting Q-table board was similar to that provided in the examples, which was confirmed. The final board is included below:

UP

----	----	----	----	----	----	----
2.24	----	----	----	3.75	3.41	----
1.98	----	----	----	4.77	4.47	----
1.76	----	0.78	----	----	----	----
1.56	----	0.90	----	----	----	----

RIGHT

2.57	3.00	3.60	4.33	3.95	2.22	----
----	----	----	----	5.29	-9.91	----
----	----	----	----	7.66	9.11	----
----	----	----	----	----	----	----
1.21	1.05	0.89	0.78	0.67	0.59	----

DOWN

1.91	----	----	----	5.25	5.29	-9.66
1.73	----	----	----	6.28	7.06	----
1.55	----	0.89	----	----	----	----
1.35	----	1.03	----	----	----	----
----	----	----	----	----	----	----

LEFT

----	2.19	2.54	2.88	3.20	3.56	3.33
----	----	----	----	----	3.71	----
----	----	----	----	----	5.81	----
----	----	----	----	----	----	----
----	1.38	1.20	1.04	0.90	0.78	0.68