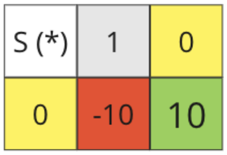
Learning rate = 0.1

Epsilon = 1 (this decays each timestep i.e. reduces slowly towards the value 0)

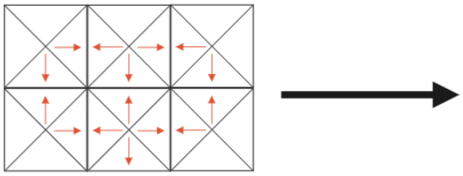
Discount Factor = 0.99

Environment



S (*)	1	0
0	-10	10

Initialise Q-Table (Q-Values) && 4 unique actions (up, right, down, left)

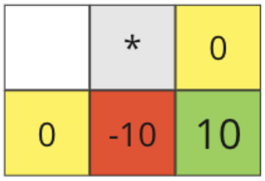


	UP	RIGHT	DOWN	LEFT
S	0	0	0	0
1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
-10	0	0	0	0
10	0	0	0	0

For now, **our Q-table is useless**; we need **to train our Q-function using the Q-Learning algorithm**

Training Timestep 1:

Choose an action using the Epsilon Greedy Strategy (Epsilon = 1, it will be decayed with time i.e. reduced) Because epsilon is big (= 1.0), We can take a random action.

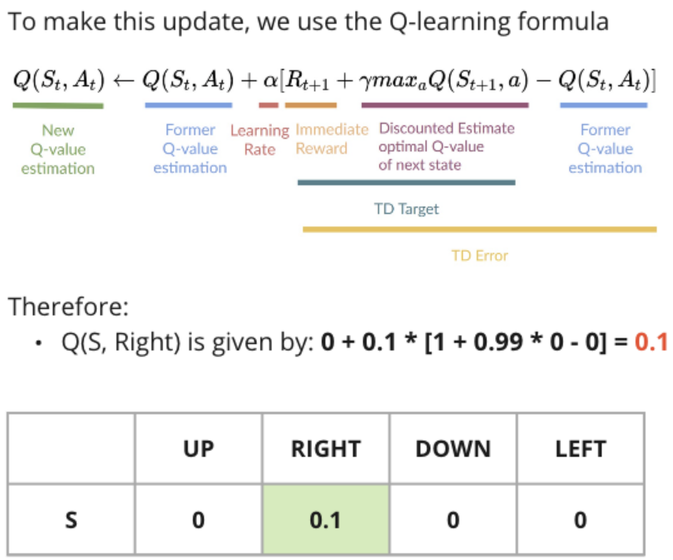


	*	0
0	-10	10

Moving to the right gets a reward of 1

We then need to update our q-value for this State - Action pair i.e. moving to the right while at the starting state.

	UP	RIGHT	DOWN	LEFT
S	0	0	0	0



To make this update, we use the Q-learning formula

$$\underbrace{Q(S_t,A_t)}_{\text{New Q-value estimation}} \leftarrow \underbrace{Q(S_t,A_t)}_{\text{Former Q-value estimation}} + \underbrace{\alpha}_{\text{Learning Rate}}[\underbrace{\underbrace{\underbrace{R_{t+1}}_{\text{Immediate Reward}} + \underbrace{\gamma max_a Q(S_{t+1},a)}_{\text{Discounted Estimate optimal Q-value of next state}}}_{\text{TD Target}} - \underbrace{Q(S_t,A_t)}_{\text{Former Q-value estimation}}}_{\text{TD Error}}]$$

Therefore:

- Q(S, Right) is given by: **0 + 0.1 * [1 + 0.99 * 0 - 0] = 0.1**

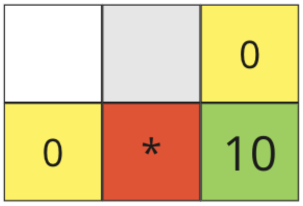
	UP	RIGHT	DOWN	LEFT
S	0	0.1	0	0

Eq. from above

$$\underset{\text{the former Q value}}{0} \; + \overset{\text{Learning rate}}{0.1} \; * [\underset{\text{the immediate reward}}{1} \; + \overset{\text{discount factor}}{0.99} \; * \underset{\substack{\text{the current max value for this sate.} \\ \text{since all values were init. to zeros,} \\ \text{the current max == 0}}}{0} \; - \overset{\text{the former Q value}}{0}$$

Training Timestep 2:

We first decay the epsilon slightly i.e. from 1 to 0.99
Because epsilon is still high (= .99), We can take another random action. e.g moving **down**



Moving to the down gets a reward of -10

We then need to update our q-value for this State - Action pair i.e. moving to the right while at the starting state.

	UP	**RIGHT**	**DOWN**	**LEFT**
S	**0**	**0.1**	**0**	**0**
1	**0**	**0**	**0**	**0**

Therefore:

- Q(1, Down) is given by: **0 + 0.1 * [-10 + 0.99 * 0 - 0] = -1**

	UP	**RIGHT**	**DOWN**	**LEFT**
S	**0**	**0.1**	**0**	**0**
1	**0**	**0**	**-1**	**0**