The link between an optimal Value and policy

In an optimal Q-Table we also have an optimal policy because we know what is the best action to take in each state.

$$\pi^*(s) = \arg\max_a Q^*(s, a)$$

But, in the beggining, our Q-table is useless since it gives arbitrary values.

As the agent explores the environment and we update the Q-table, it will give us better and better approximations to the optimal policy

Q-Learning algorithm

Input: policy π, num-episodes, α, GLIE $\{\epsilon_i\}$
Output: value function Q

```
Initialize Q-table (all 0's)

for i <- 1 to num-episodes do
    ε <- ε_i
    Observe S0
    t <- 0

    repeat
        Choose action A_t using policy derived from Q (ε-greedy)
        Take action A_t and observe R_t+1 , S_t+1
        Q(S_t, A_t) <- Q(S_t, A_t) + α(...) (update state-action pair)
        t <- t+1
    until S_t is terminal
end
return Q
```