# Exercise: Q-Learning (Tabular and Value Function Approximation)

Link to github classroom: `https://classroom.github.com/a/ZtU6Nv_I`
Complete the exercises regarding Q-Learning and record your observations. How do the hyperparameters change the outcome of your experiments?

1. **Tabular Q-Learning**
   Implement the Q-Learning update step in *q_learning_tabular.py* and try different state discretizations (`BINS`) and learning rates (`LEARNING_RATE`). How does the number of states and learning rate affect the training of the RL algorithm?

2. **Q-Learning with Linear Value Function Approximation**
   Implement Q-Learning with Linear Value Function Approximation. First create *make_Q* that takes an environment as input and creates a PyTorch Model. Then implement the value function training step in *q_learning_vfa.py* using the `Q` module and the `optimizer`. How does the training differ from the tabular case? How sensitive is the algorithm to the weight initialization?

   Update the hyperparameters and the model to achieve a mean reward of more than 50 for the CartPole environment.

   For the open questions, please write your answers in 'answers.txt'. We will grade those manually.