some notes

(REINFORCE)

- After finishing playing the game, you'll have a full traj of all states
e.g. from s_0 to s_100000, assuming this is how long it took to finish the game.

- Having the full set of states, actions & rewards, rewards can be used to compute
"Commodity of Returns". Adding all the rewards, weighted by the coefficients defined
by the discount gamma, an estimate for cumulative return G will be obtained.
This is somehow an estimate of Q-fn.

- Using the policy grad forumla, we can compute an est of grad of expected reward
wrt policy, given one single session.
- Having some NN, you would not only compute grad wrt policy, but also wrt net weigths.

- While training -in general- you will probably get diff states; some states might
be difficult while others are easier. Hence, some states will have more weights than others.
Normally in other methods, the agent will get higher score for achieving better in those
difficult situations, however with REINFORCE the opposite will occur.

(BASELINE)

- This is what it means. We aim to reward not just the Q-fn as it is, but the "advantage".
i.e. how good the agent performs to what it usually does.

In case the agent takes a remarkably better action, the advantage will be higher.
On the other hand, in simpler tasks where the agent performs changes a little bit,
e.g. going from 100 to 90. Thus causing the Q-value to be smaller than expected.
The advantage will be -ve in this scenario.

(Duct tape zone)

- Applying policy-based methods in practice, you might end up with alg that completely
abandons one action in at least a subset of situations. Thus, you will no longer receive
samples representing this particular action.
Usually dropping actions like this is NOT very wise thing to do.
- One way around is by introducing entropy.

(A3C)

- It uses learned state values (critic) as a baseline for policy grad (actor)

- In general, sample data is NOT iid. However what on-policy method do to deal with such problem is to play independent replicas of the env; taking actions by sampling independently from policy. This will lead to diff traj of policies, more or less independent.

- Training Critic in Advantage Actor Critic:
A critic predicts V(s), we min [r + gamma * const * (V(s_)) - V(s)]**2

(Policy-based Methods)

- In general, they aim to parameterize the action-picking policy, by max expected returns.

N.B. Policy Grad is a grad of expected reward wrt policy parameters.
It's NOT grad of policy parameters!

(Partial Trajectories)

- The following methods learn from partial traj:
 - Advantage Actor-Critic
 - Q-learning
 - SARSA

(Using Q-learning)

- Unlike REINFORCE, Q-learning can be trained much more efficiently with experience replay.
N.B. technically, REINFORCE can be modified for experience replay with importance sampling, but that's harder to implement and less efficient.
- Unlike REINFORCE, Q-learning can be trained on partial experience (e.g. s,a,r,s_)