Unit 1

- What is RL
  - big picture:
    - *Agent(AI)* will *Learn* from *Environment* by *Interacting* with Environment and receive *Rewards* (neg/pos) as *Feedback* for performing *Actions*
    - ```
      while not done:
          Environment -> observation -> Agent
      Agent -> action -> Environment
      Environment -> reward, new observation, done -> Agent
      ```
  - example:

```
boy learning to play video game
    - boy is Agent,
      screen is observation
      buttons is set of possible actions
      game is environment
      rewards is points

    - boy observes screen (sees avatar, coin, squid)
    - boy presses right button
    - game reacts by updating screen -> avatar touches coin, rewards point
        - boy learns getting coin rewards +1
        - boy presses button again and tough
        - game reacts by updating screen -> avatar touches squid and dies, rewardi
        - boy learns touching squid rewards dead, and ends game
```

  - formal definition:
    - RL is **FRAMEWORK** for solving **CONTROL TASKS** (aka decision problems)
    - by building **AGENTS**
    - AGENTS that **LEARN** from the **ENVIRONMENT**
      - LEARN by **INTERACTING** with ENVIRONMENT
      - LEARN thru **TRIAL and ERROR**
```

            - LEARN by receiving **REWARDS**(NEGATIVE/POSITIVE) as **FEEDBACK**

- RL Framework
    - RL Process
        - Environment -> (state St, reward Rt) -> Agent -> action *At* -> Environment -> (state *St+1*, reward *Rt+1*) –> Agent (loop)
        - IOW: *S0 -> A0 -> R1,S1 -> A1 -> R2, S2 …. Rn, Sn -> An -> Rn+1, Sn+1 ->…*
        - Agent's GOAL: Maximize its **CUMULATIVE REWARD** aka **EXPECTED RETURN**
        - WHY IS THIS(Maximization of EXPECTED RETURN) the Agent's goal? b/c RL is based the REWARD HYPOTHESIS
    - the central idea of RL: The reward hypothesis
        - ALL GOALS can be described as the MAXIMIZATION of EXPECTED RETURN
            - Example?
        - TO HAVE BEST BEHAVIOR, MAXIMIZE the EXPECTED CUMULATIVE REWARD
            - are EXPECTED CUMULATIVE REWARD same as EXPECTED RETURN? ==> YES!
    - ANOTHER NAME for RL Process: MARKOV DECISION PROCESS (MDP)
    - Markov property AGENT only needs CURRENT STATE to decide what ACTION to TAKE - CONTRAST with NEEDING HISTORY of ALL STATES and ACTIONS they took before
    - Observations/States Space Observations/States - information our AGENT gets from the ENVIRONMENT. - Example: videogame, -> Observation/State is a Screenshot (AKA Frame)
    - Action Space
    - Rewards and the discounting
    - Task Types
- Exploration/Exploitation Tradeoff
- Solving RL Problems: 2 main approaches
    - The Policy PI
    - Policy based Methods
    - Value based Methods
- "Deep" in Deep RL

```
z(x) = \\sum \\limits _{i} ^{n} w_{i}x_{i} + b = w \\cdot x + b \n
```