Implementation: Iterative Policy Evaluation

The pseudocode for **iterative policy evaluation** can be found below.

Iterative Policy Evaluation

Input: MDP, policy π, small positive number θ
Output: $V \approx v_\pi$
Initialize V arbitrarily (e.g., $V(s) = 0$ for all $s \in \mathcal{S}^+$)
repeat
  $\Delta \leftarrow 0$
  for $s \in \mathcal{S}$ **do**
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma V(s'))$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
  end
until $\Delta < \theta$;
return V

Note that policy evaluation is guaranteed to converge to the state-value function v_π corresponding to a policy π, as long as $v_\pi(s)$ is finite for each state $s \in \mathcal{S}$. For a finite Markov decision process (MDP), this is guaranteed as long as either:

- $\gamma < 1$, or
- if the agent starts in any state $s \in \mathcal{S}$, it is guaranteed to eventually reach a terminal state if it follows policy π.

Please use the next concept to complete **Part 0: Explore FrozenLakeEnv** and **Part 1: Iterative Policy Evaluation** of `Dynamic_Programming.ipynb`. Remember to save your work!

If you'd like to reference the pseudocode while working on the notebook, you are encouraged to open this sheet in a new window.