# [RL] [Fundamentals of Reinforcement Learning](https://www.coursera.org/learn/fundamentals-of-reinforcement-learning/home/welcome) [Week 4](https://www.coursera.org/learn/fundamentals-of-reinforcement-learning/home/week/4)

### What is policy evaluation?

Evaluate how good the policy is by evaluating its value functions.

With inputs: policy, find value functions.

### What is control?

Control the policy to maximize value/rewards.

With inputs: dynamic probabilities and value functions, find optimal policy.

### What is dynamic programming?

A way to compute for the optimal policy and value functions which uses bellman equations.

A substitute to the linear equations solver.

### What is iterative policy evaluation?

Evaluate the value functions of a policy iteratively.

### How is iterative policy evaluation done?

Through bellman optimality equation updates. We update your previous value functions according to the newly computed value functions based off of bellman optimality equation.

We can have 2 arrays of value functions: one for the previous and one for the new. We compute the new value functions based on the previous value functions. After computing it for all states, we then update by overwriting the previous value functions with the new ones. This update of all states is called a sweep. When the change between the previous and new value functions becomes sufficiently small, we stop our updates. We conclude that we have reached the optimal/true value function for that policy.

### What is a sweep?

Updating all previous value functions with the newly computed value functions.

### What is the policy improvement theorem?

If act greedily with respect to the value functions of a given policy pi, then our new policy pi-prime will be strictly better than the original policy pi. (or equal if policy pi is already optimal.)

### What is policy iteration?

Repeated policy evaluation and improvement until you reach the optimal policy.

From any start, policy pi, it will reach/converge to an optimal policy.

Starting Policy -> Evaluate -> Improve -> Evaluate -> Improve -> Optimal Policy.

1. Evaluate a policy's value functions.
2. Improve policy by acting greedily with respect to the value functions.
   1. New policy is strictly better than previous one.
   2. But by changing the policy, you no longer know the current policy's value functions.
3. Evaluate the new/current policy's value functions
4. Improve policy. Repeat.
5. When there is no more improvement, then it has converged.

### Bellman equation vs Bellman optimality equation

Bellman equation => we can rewrite our value functions recursively.

Bellman optimality equation => we can rewrite our value functions recursively and in terms of itself (i.e. without policy). This is only possible because the optimal policy assumes that you're taking the max/acting greedily. We replace the policy with max/argmax, enabling us to rewrite it in terms of value functions only. Use bellman optimality equation as our update rule and continuously update our value functions.

### Argmax vs Max

Max = returns maximum value.

Argmax = returns the parameter at which value is maximum.