Reinforcement learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment. The goal is to learn a strategy, called a policy, that maximizes the cumulative reward the agent receives over time. Here's how it generally works:

1. **Agent and Environment**: In reinforcement learning, you have an agent (e.g., a robot or a software program) and an environment (the world or context in which the agent operates). The environment presents states to the agent, and the agent takes actions in response.

2. **States, Actions, and Rewards**: The environment is typically modeled as a set of states, and the agent can perform different actions in each state. After taking an action, the agent receives a reward, which is a numerical value given based on the desirability of the outcome. The agent also transitions to a new state following the rules of the environment.

3. **Policy**: The policy is a strategy used by the agent to decide which action to take in a given state. It can be deterministic (a direct mapping of state to action) or stochastic (providing probabilities for each action).

4. **Learning Goal**: The ultimate goal of the agent is to maximize the total reward it receives in the long run. This often involves trade-offs between exploring new actions to find more rewarding outcomes and exploiting known actions that already yield high rewards.

5. **Value Functions**: To achieve its goal, the agent often learns value functions which estimate how good it is to be in a given state or to perform a particular action in a state. There are two main types of value functions:

    - **State Value Function (V)**: Estimates the expected reward from being in a given state and following a particular policy thereafter.
    - **Action Value Function (Q)**: Estimates the expected reward from taking a certain action in a given state, then following a policy.

6. **Learning Process**: The agent improves its policy by continuously updating it based on the rewards received and the value functions updated. This is often done using methods like Q-learning, where the agent updates the Q-values based on the reward received and the maximum future reward possible from the next state.

7. **Exploration vs. Exploitation**: A key challenge in RL is balancing exploration (trying new, untested actions) and exploitation (using actions known to yield high rewards). Algorithms like ε-greedy are commonly used to manage this balance, where ε represents the probability of choosing a random action (exploration) versus the best-known action (exploitation).

Reinforcement learning is widely used in various applications, such as robotics, gaming, autonomous driving, and in areas where sequential decision-making is critical under uncertainty.