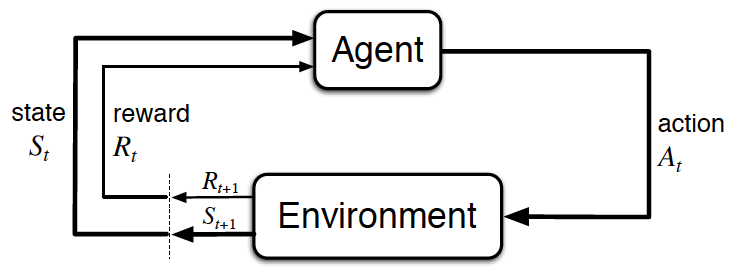
**Summary**

[[](https://classroom.udacity.com/nanodegrees/nd893/parts/8f607726-757e-4ef5-8b64-f2368755b89a/modules/a85374fa-6a60-425b-a480-85b211c5bd5d/lessons/fd08905e-9026-42ad-ae35-b2d315323e5d/concepts/ee28399b-f809-4e2b-936b-5a88d7297899)](https://classroom.udacity.com/nanodegrees/nd893/parts/8f607726-757e-4ef5-8b64-f2368755b89a/modules/a85374fa-6a60-425b-a480-85b211c5bd5d/lessons/fd08905e-9026-42ad-ae35-b2d315323e5d/concepts/ee28399b-f809-4e2b-936b-5a88d7297899)

[The agent-environment interaction in reinforcement learning. (Source: Sutton and Barto, 2017)](https://classroom.udacity.com/nanodegrees/nd893/parts/8f607726-757e-4ef5-8b64-f2368755b89a/modules/a85374fa-6a60-425b-a480-85b211c5bd5d/lessons/fd08905e-9026-42ad-ae35-b2d315323e5d/concepts/ee28399b-f809-4e2b-936b-5a88d7297899)

**The Setting, Revisited**

* The reinforcement learning (RL) framework is characterized by an **agent** learning to interact with its **environment**.
* At each time step, the agent receives the environment's **state** (*the environment presents a situation to the agent)*, and the agent must choose an appropriate **action** in response. One time step later, the agent receives a **reward** (*the environment indicates whether the agent has responded appropriately to the state*) and a new **state**.
* All agents have the goal to maximize expected **cumulative reward**, or the expected sum of rewards attained over all time steps.

**Episodic vs. Continuing Tasks**

* A **task** is an instance of the reinforcement learning (RL) problem.
* **Continuing tasks** are tasks that continue forever, without end.
* **Episodic tasks** are tasks with a well-defined starting and ending point.
  + In this case, we refer to a complete sequence of interaction, from start to finish, as an **episode**.
  + Episodic tasks come to an end whenever the agent reaches a **terminal state**.

**The Reward Hypothesis**

* **Reward Hypothesis**: All goals can be framed as the maximization of (expected) cumulative reward.

**Goals and Rewards**

* (Please see **Part 1** and **Part 2** to review an example of how to specify the reward signal in a real-world problem.)

**Cumulative Reward**

* The **return at time step***Gt*​:=*Rt*+1​+*Rt*+2​+*Rt*+3​+…
* The agent selects actions with the goal of maximizing expected (discounted) return. (*Note: discounting is covered in the next concept.*)

**Discounted Return**

* The **discounted return at time step***Gt*​:=*Rt*+1​+*γRt*+2​+*γ*2*Rt*+3​+….
* The **discount rate***γ* is something that you set, to refine the goal that you have the agent.
  + It must satisfy 0≤*γ*≤1.
  + If *γ*=0, the agent only cares about the most immediate reward.
  + If *γ*=1, the return is not discounted.
  + For larger values of *γ*, the agent cares more about the distant future. Smaller values of *γ* result in more extreme discounting, where - in the most extreme case - agent only cares about the most immediate reward.

**MDPs and One-Step Dynamics**

* The **state space**S is the set of all (*nonterminal*) states.
* In episodic tasks, we use S+ to refer to the set of all states, including terminal states.
* The **action space** A is the set of possible actions. (Alternatively, A(*s*) refers to the set of possible actions available in state *s*∈S.)
* (Please see **Part 2** to review how to specify the reward signal in the recycling robot example.)
* The **one-step dynamics** of the environment determine how the environment decides the state and reward at every time step. The dynamics can be defined by specifying p



* A **(finite) Markov Decision Process (MDP)** is defined by:
  + a (finite) set of states  S (or S+, in the case of an episodic task)
  + a (finite) set of actions A
  + a set of rewards R
  + the one-step dynamics of the environment
  + the discount rate *γ*∈[0,1]