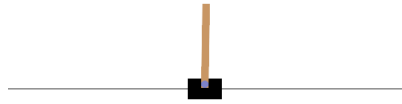




# Cart Pole



This environment is part of the Classic Control environments which contains general information about the environment.

Action Space	<code>Discrete(2)</code>
Observation Space	<code>Box([-4.8 -inf -0.41887903 -inf], [4.8 inf 0.41887903 inf], (4,), float32)</code>
import	<code>gymnasium.make("CartPole-v1")</code>

## Description

This environment corresponds to the version of the cart-pole problem described by Barto, Sutton, and Anderson in ["Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problem"](#). A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.

## Action Space

The action is a `ndarray` with shape `(1,)` which can take values `{0, 1}` indicating the direction of the fixed force the cart is pushed with.

- 0: Push cart to the left
- 1: Push cart to the right

**Note:** The velocity that is reduced or increased by the applied force is not fixed and it depends on the angle the pole is pointing. The center of gravity of the pole varies the amount of energy needed to move the cart underneath it

v1.0.0 (latest)



The observation is a `ndarray` with shape `(4,)` with the values corresponding to the following positions and velocities:

Num	Observation	Min	Max
0	Cart Position	-4.8	4.8
1	Cart Velocity	-Inf	Inf
2	Pole Angle	$\sim -0.418$ rad ( $-24^\circ$ )	$\sim 0.418$ rad ( $24^\circ$ )
3	Pole Angular Velocity	-Inf	Inf

**Note:** While the ranges above denote the possible values for observation space of each element, it is not reflective of the allowed values of the state space in an untruncated episode.

Particularly:

- The cart x-position (index 0) can take values between `(-4.8, 4.8)`, but the episode terminates if the cart leaves the `(-2.4, 2.4)` range.
- The pole angle can be observed between `(-0.418, 0.418)` radians (or  $\pm 24^\circ$ ), but the episode terminates if the pole angle is not in the range `(-0.2095, 0.2095)` (or  $\pm 12^\circ$ )

## Rewards

Since the goal is to keep the pole upright for as long as possible, by default, a reward of `+1` is given for every step taken, including the termination step. The default reward threshold is 500 for v1 and 200 for v0 due to the time limit on the environment.

If `sutton_barto_reward=True`, then a reward of `0` is awarded for every non-terminating step and `-1` for the terminating step. As a result, the reward threshold is 0 for v0 and v1.

## Starting State

All observations are assigned a uniformly random value in `(-0.05, 0.05)`

## Episode End

The episode ends if any one of the following occurs:

- Termination: Pole Angle is greater than  $\pm 12^\circ$
- Termination: Cart Position is greater than  $\pm 2.4$  (center of the cart reaches the edge of the display)
- Truncation: Episode length is greater than 500 (200 for v0)

v1.0.0 (latest)



Cartpole only has `render_mode` as a keyword for `gymnasium.make`. On reset, the `options` parameter allows the user to change the bounds used to determine the new random state.

```
>>> import gymnasium as gym
>>> env = gym.make("CartPole-v1", render_mode="rgb_array")
>>> env
<TimeLimit<OrderEnforcing<PassiveEnvChecker<CartPoleEnv<CartPole-v1>>>>>
>>> env.reset(seed=123, options={"low": -0.1, "high": 0.1}) # default low=-0.05, high=0.05
(array([ 0.03647037, -0.0892358 , -0.05592803, -0.06312564], dtype=float32), {})
```

Parameter	Type	Default	Description
<code>sutton_barto_reward</code>	<b>bool</b>	<code>False</code>	If <code>True</code> the reward function matches the original sutton barto implementation

## Vectorized environment

To increase steps per seconds, users can use a custom vector environment or with an environment vectorizer.

```
>>> import gymnasium as gym
>>> envs = gym.make_vec("CartPole-v1", num_envs=3, vectorization_mode="vector_entry_point")
>>> envs
CartPoleVectorEnv(CartPole-v1, num_envs=3)
>>> envs = gym.make_vec("CartPole-v1", num_envs=3, vectorization_mode="sync")
>>> envs
SyncVectorEnv(CartPole-v1, num_envs=3)
```

## Version History

- v1: `max_time_steps` raised to 500.
  - In Gymnasium `1.0.0a2` the `sutton_barto_reward` argument was added (related [GitHub issue](#))
- v0: Initial versions release.

