```
import gym
envs = gym.envs.registry.values()
total_envs = len(envs)
print(f"Total number of environments: {total_envs}")
```

```
Total number of environments: 44
/usr/local/lib/python3.11/dist-packages/gym/envs/registration.py:421: UserWarning: WARN: The `registry.all` method is deprecated. P
  logger.warn(
```

```
import gym
envs = gym.envs.registry.values()
env_names = sorted([env_spec.id for env_spec in envs])
for name in env_names:
    print(name)
```

```
Acrobot-v1
Ant-v2
Ant-v3
Ant-v4
BipedalWalker-v3
BipedalWalkerHardcore-v3
Blackjack-v1
CarRacing-v2
CartPole-v0
CartPole-v1
CliffWalking-v0
FrozenLake-v1
FrozenLake8x8-v1
HalfCheetah-v2
HalfCheetah-v3
HalfCheetah-v4
Hopper-v2
Hopper-v3
Hopper-v4
Humanoid-v2
Humanoid-v3
Humanoid-v4
HumanoidStandup-v2
HumanoidStandup-v4
InvertedDoublePendulum-v2
InvertedDoublePendulum-v4
InvertedPendulum-v2
InvertedPendulum-v4
LunarLander-v2
LunarLanderContinuous-v2
MountainCar-v0
MountainCarContinuous-v0
Pendulum-v1
Pusher-v2
Pusher-v4
Reacher-v2
Reacher-v4
Swimmer-v2
Swimmer-v3
Swimmer-v4
Taxi-v3
Walker2d-v2
Walker2d-v3
Walker2d-v4
```

```
import gym

env = gym.make("CartPole-v1")
print(f"Action space: {env.action_space}")
print(f"Observation space: {env.observation_space}")
```

```
Action space: Discrete(2)
Observation space: Box([-4.8000002e+00 -3.4028235e+38 -4.1887903e-01 -3.4028235e+38], [4.8000002e+00 3.4028235e+38 4.1887903e-01 3.4
/usr/local/lib/python3.11/dist-packages/gym/core.py:317: DeprecationWarning: WARN: Initializing wrapper in old step API which retur
  deprecation(
/usr/local/lib/python3.11/dist-packages/gym/wrappers/step_api_compatibility.py:39: DeprecationWarning: WARN: Initializing environme
  deprecation(
```

```
print("State: The state is a 4-dimensional vector representing the environment's current situation:")
print("- Cart Position: The horizontal position of the cart on the track.")
print("- Cart Velocity: The current velocity of the cart.")
print("- Pole Angle: The angle of the pole with respect to the vertical position.")
print("- Pole Angular Velocity: The rate at which the pole's angle is changing.")
print("\nAction: There are two discrete actions available to the agent:")
print("- 0: Push the cart to the left.")
print("- 1: Push the cart to the right.")
print("\nTransition: The transition is deterministic. Given the current state and an action, the environment will deterministically move
print("\nReward: A reward of +1 is given for every timestep that the pole remains upright. The episode ends if the pole's angle exceeds
```

```
State: The state is a 4-dimensional vector representing the environment's current situation:
- Cart Position: The horizontal position of the cart on the track.
- Cart Velocity: The current velocity of the cart.
- Pole Angle: The angle of the pole with respect to the vertical position.
- Pole Angular Velocity: The rate at which the pole's angle is changing.

Action: There are two discrete actions available to the agent:
- 0: Push the cart to the left.
- 1: Push the cart to the right.

Transition: The transition is deterministic. Given the current state and an action, the environment will deterministically move to a

Reward: A reward of +1 is given for every timestep that the pole remains upright. The episode ends if the pole's angle exceeds a cer
```

```
env = gym.make("FrozenLake-v1")
print(f"Action space: {env.action_space}")
print(f"Observation space: {env.observation_space}")
```

```
Action space: Discrete(4)
Observation space: Discrete(16)
/usr/local/lib/python3.11/dist-packages/gym/core.py:317: DeprecationWarning: WARN: Initializing wrapper in old step API which retur
  deprecation(
/usr/local/lib/python3.11/dist-packages/gym/wrappers/step_api_compatibility.py:39: DeprecationWarning: WARN: Initializing environme
  deprecation(
```

```
print("State: The state represents the agent's current position on the 4x4 grid. Each cell in the grid is a discrete state. The grid co
print("\nAction: There are four discrete actions available to the agent:")
print("- 0: Move Left")
print("- 1: Move Down")
print("- 2: Move Right")
print("- 3: Move Up")
print("\nTransition: The transition is stochastic. When the agent attempts to move in a specific direction, there is a chance that they
print("\nReward: The agent receives the following rewards:")
print("- +1: For reaching the Goal (G).")
print("- 0: For stepping on a Frozen (F) cell or a Start (S) cell.")
print("- 0: For falling into a Hole (H). The episode terminates.")
```

```
State: The state represents the agent's current position on the 4x4 grid. Each cell in the grid is a discrete state. The grid conta

Action: There are four discrete actions available to the agent:
- 0: Move Left
- 1: Move Down
- 2: Move Right
- 3: Move Up

Transition: The transition is stochastic. When the agent attempts to move in a specific direction, there is a chance that they will

Reward: The agent receives the following rewards:
- +1: For reaching the Goal (G).
- 0: For stepping on a Frozen (F) cell or a Start (S) cell.
- 0: For falling into a Hole (H). The episode terminates.
```

```
env = gym.make("MountainCar-v0")
print(f"Action space: {env.action_space}")
print(f"Observation space: {env.observation_space}")
```

```
Action space: Discrete(3)
Observation space: Box([-1.2  -0.07], [0.6  0.07], (2,), float32)
/usr/local/lib/python3.11/dist-packages/gym/core.py:317: DeprecationWarning: WARN: Initializing wrapper in old step API which retur
  deprecation(
/usr/local/lib/python3.11/dist-packages/gym/wrappers/step_api_compatibility.py:39: DeprecationWarning: WARN: Initializing environme
  deprecation(
```

```
print("State: The state is a 2-dimensional continuous vector:")
print("- Car Position: The horizontal position of the car, ranging from -1.2 to 0.6.")
print("- Car Velocity: The velocity of the car, ranging from -0.07 to 0.07.")
print("\nAction: There are three discrete actions available to the agent:")
print("- 0: Push the car to the left.")
print("- 1: Do nothing.")
print("- 2: Push the car to the right.")
print("\nTransition: At each timestep, the car's position and velocity are updated based on the chosen action and the force of gravity.
print("\nReward: A reward of -1 is given for each timestep. The episode terminates and a reward of 0 is given if the car reaches the go
```

```
State: The state is a 2-dimensional continuous vector:
- Car Position: The horizontal position of the car, ranging from -1.2 to 0.6.
- Car Velocity: The velocity of the car, ranging from -0.07 to 0.07.

Action: There are three discrete actions available to the agent:
- 0: Push the car to the left.
- 1: Do nothing.
- 2: Push the car to the right.

Transition: At each timestep, the car's position and velocity are updated based on the chosen action and the force of gravity. The v

Reward: A reward of -1 is given for each timestep. The episode terminates and a reward of 0 is given if the car reaches the goal pos
```

```
env = gym.make("Blackjack-v1")
print(f"Action space: {env.action_space}")
print(f"Observation space: {env.observation_space}")
```

```
    Action space: Discrete(2)
    Observation space: Tuple(Discrete(32), Discrete(11), Discrete(2))
    /usr/local/lib/python3.11/dist-packages/gym/core.py:317: DeprecationWarning: WARN: Initializing wrapper in old step API which returi
      deprecation(
    /usr/local/lib/python3.11/dist-packages/gym/wrappers/step_api_compatibility.py:39: DeprecationWarning: WARN: Initializing environmei
      deprecation(
```

```
print("State: The state is a tuple with three elements:")
print("- Player's Current Sum: The sum of the player's cards (4-31).")
print("- Dealer's Showing Card: The value of the dealer's face-up card (1-10).")
print("- Usable Ace: Whether the player has an ace that can be counted as 11 without busting (0 for no, 1 for yes).")
print("\nAction: There are two discrete actions available to the player:")
print("- 0: Stick (stop taking cards).")
print("- 1: Hit (take another card).")
print("\nTransition: The transition depends on the player's action:")
print("- If the player 'hits', a card is drawn from the deck. The player's sum is updated. If the sum exceeds 21, the player busts, and
print("- If the player 'sticks', the dealer plays their hand according to a fixed strategy (hit until sum is 17 or more). The episode tl
print("\nReward: The reward is given at the end of the episode:")
print("- +1: For winning against the dealer.")
print("- -1: For losing to the dealer (including busting).")
print("- 0: For a draw (push).")
print("- +1.5: For winning with a blackjack (an initial hand of an ace and a 10-value card).")
```

```
    State: The state is a tuple with three elements:
    - Player's Current Sum: The sum of the player's cards (4-31).
    - Dealer's Showing Card: The value of the dealer's face-up card (1-10).
    - Usable Ace: Whether the player has an ace that can be counted as 11 without busting (0 for no, 1 for yes).

    Action: There are two discrete actions available to the player:
    - 0: Stick (stop taking cards).
    - 1: Hit (take another card).

    Transition: The transition depends on the player's action:
    - If the player 'hits', a card is drawn from the deck. The player's sum is updated. If the sum exceeds 21, the player busts, and the
    - If the player 'sticks', the dealer plays their hand according to a fixed strategy (hit until sum is 17 or more). The episode then

    Reward: The reward is given at the end of the episode:
    - +1: For winning against the dealer.
    - -1: For losing to the dealer (including busting).
    - 0: For a draw (push).
    - +1.5: For winning with a blackjack (an initial hand of an ace and a 10-value card).
```

```
env = gym.make("Taxi-v3")
print(f"Action space: {env.action_space}")
print(f"Observation space: {env.observation_space}")
```

```
    Action space: Discrete(6)
    Observation space: Discrete(500)
    /usr/local/lib/python3.11/dist-packages/gym/core.py:317: DeprecationWarning: WARN: Initializing wrapper in old step API which returi
      deprecation(
    /usr/local/lib/python3.11/dist-packages/gym/wrappers/step_api_compatibility.py:39: DeprecationWarning: WARN: Initializing environmei
      deprecation(
```

```
print("State: The state is a single integer from 0 to 499, representing a combination of the taxi's location, the passenger's location, a
print("- Taxi's Location: A 5x5 grid, giving 25 possible row and column locations for the taxi.")
print("- Passenger's Location: 5 possible locations. 4 of these are designated pickup/dropoff locations (R, G, Y, B), and the 5th indicat
print("- Destination Location: 4 designated dropoff locations (R, G, Y, B).")
print("\nAction: There are 6 discrete actions available to the agent:")
print("- 0: Move South")
print("- 1: Move North")
print("- 2: Move East")
print("- 3: Move West")
print("- 4: Pickup Passenger")
print("- 5: Dropoff Passenger")
print("\nTransition: The transitions are deterministic. Moving south, north, east, or west changes the taxi's location if it does not hit
print("\nReward: The rewards are structured to encourage efficient and correct task completion:")
print("- +20: For successfully dropping off the passenger at the correct destination.")
print("- -10: For an illegal 'pickup' or 'dropoff' action.")
print("- -1: For each step taken. This incentivizes the agent to find the shortest path.")
```

```
State: The state is a single integer from 0 to 499, representing a combination of the taxi's location, the passenger's location, anc
- Taxi's Location: A 5x5 grid, giving 25 possible row and column locations for the taxi.
- Passenger's Location: 5 possible locations. 4 of these are designated pickup/dropoff locations (R, G, Y, B), and the 5th indicate:
- Destination Location: 4 designated dropoff locations (R, G, Y, B).

Action: There are 6 discrete actions available to the agent:
- 0: Move South
- 1: Move North
- 2: Move East
- 3: Move West
- 4: Pickup Passenger
- 5: Dropoff Passenger

Transition: The transitions are deterministic. Moving south, north, east, or west changes the taxi's location if it does not hit a w

Reward: The rewards are structured to encourage efficient and correct task completion:
- +20: For successfully dropping off the passenger at the correct destination.
- -10: For an illegal 'pickup' or 'dropoff' action.
- -1: For each step taken. This incentivizes the agent to find the shortest path.
```

```
env = gym.make("CliffWalking-v0")
print(f"Action space: {env.action_space}")
print(f"Observation space: {env.observation_space}")
```

```
Action space: Discrete(4)
Observation space: Discrete(48)
/usr/local/lib/python3.11/dist-packages/gym/core.py:317: DeprecationWarning: WARN: Initializing wrapper in old step API which returı
  deprecation(
/usr/local/lib/python3.11/dist-packages/gym/wrappers/step_api_compatibility.py:39: DeprecationWarning: WARN: Initializing environmeı
  deprecation(
```

```
print("State: The state represents the agent's position on a 4x12 grid. The states are numbered from 0 to 47. The top-left corner is stat
print("\nAction: There are four discrete actions available to the agent, represented by integers:")
print("- 0: Up")
print("- 1: Right")
print("- 2: Down")
print("- 3: Left")
print("\nTransition: The transitions are deterministic. If the agent chooses an action that would move it into a valid grid cell, it move
print("\nReward: The reward structure is as follows:")
print("- -1: For each step taken on a non-cliff, non-goal cell.")
print("- -100: For stepping on a cliff cell. This also sends the agent back to the starting state (state 36).")
print("- -1: For reaching the goal state (state 47). The episode terminates upon reaching the goal.")
```

```
State: The state represents the agent's position on a 4x12 grid. The states are numbered from 0 to 47. The top-left corner is state

Action: There are four discrete actions available to the agent, represented by integers:
- 0: Up
- 1: Right
- 2: Down
- 3: Left

Transition: The transitions are deterministic. If the agent chooses an action that would move it into a valid grid cell, it moves tc

Reward: The reward structure is as follows:
- -1: For each step taken on a non-cliff, non-goal cell.
- -100: For stepping on a cliff cell. This also sends the agent back to the starting state (state 36).
- -1: For reaching the goal state (state 47). The episode terminates upon reaching the goal.
```