

Gymnasium (essentially gym)

- A project that provides an API for single agent RL environments
- Four key functions: `make`, `Env.reset()`, `Env.step()`, and `Env.render()`
- `Env` is a python class representing a markov decision process
- Allows creation of an initial state, transition/move to new states, and being able to visualize the environment.
- Wrapper can be used to augment/modify the environment

To initialize an environment, use the `make()` function. An example (derived from original documentation)

```
...  
import gymnasium as gym  
env = gym.make('CartPole-v1')  
...
```

Now, an `env` function is returned.

`pprint_registry()` shows all creatable environments

Code from the original documentation; creates a single-episode run of lunar lander:

```
import gymnasium as gym  
  
env = gym.make("LunarLander-v3", render_mode="human")  
observation, info = env.reset()  
  
episode_over = False  
while not episode_over:  
    action = env.action_space.sample() # agent policy that uses the observation and info  
    observation, reward, terminated, truncated, info = env.step(action)  
  
    episode_over = terminated or truncated  
  
env.close()
```

End code.

An environment is first created with `make()`. Render mode specifies how to make the environment. The specified `LunarLander` environment is made.

Using `Env.reset()` on the environment gets the first environment observation with some additional info.

We want to continue the agent environment loop till the environment ends, which we don't know when that will happen. `Episode_over` is a variable that determines when to stop the environment.

The agent performs an action in this environment; `Env.step()` is what executes the specified step. This action becomes random with `env.action_space.sample()`. This is like having a robot press a specified button on a controller to play a video game. From taking that action, there is a new environment and a reward for taking that action. (positive for doing something good, negative for doing something bad). This whole paragraph details a timestamp.