Lab 3 Notes

- when we create an object and if we want to print it we see lines unreadable somehow, so in python there is a predefined function called `__str()__` It's what gets displayed when you print an object or use `str()` on it. The output is meant to be human-readable, giving a general description of the object.

  Calling print(object) ⇒ Will implicitly call **str**()

  ```
  def __str__(self):
      return f"A Rocket positioned at ({self.x},{self.y})"
  # when you print(my_rocket) it will print that line
  ```

- `__repr__()` is primarily used during development, especially for **debugging** and **testing**. Its purpose is to provide a precise, unambiguous representation of an object that can help developers understand the state of the object and potentially recreate it.

  Calling just the object ⇒ Will implicitly call **repr**()

  ```
  def __repr__(self):
          return f"Rocket({self.x},{self.y})"
  ```

- In Python, the `__eq__()` method is used to define how the equality operator (`==`) behaves for instances of a class. By default, `==` checks if two objects are the same instance (i.e., they refer to the same memory address). However, you can override `__eq__()` to check if two objects are considered equal based on their attributes.