```
check_registry = CheckRegistry()
code_ex = CodeExercise(
    code=sin,
    check_registry=check_registry,
)
check_registry.add_check(
    code_ex,
    asserts=[
        assert_type, # checks if same type
        assert_shape, # checks if same shape
        assert_numpy_allclose, # checks if allclose
    inputs_parameters=[{"x": np.asarray([0., np.pi, 2*np.pi])}],
    outputs_references=[(np.asarray([0., 0., 0.]),)]
)
def assert_2pi_periodic():
    if not np.allclose(code_ex.code(0), code_ex.code(2*np.pi)):
        return "Function is not periodic."
    return "" # empty strings means it passes
check_registry.add_check(
    code_ex,
    asserts=[
        assert_2pi_periodic,
    ]
check_registry.add_check(
   code_ex,
    asserts=assert_numpy_allclose,
    fingerprint = lambda x: np.sum(x),
    inputs_parameters=[{"x": np.asarray([0., np.pi, 2*np.pi])}],
    outputs_references=[(0.,),],
  1 def sin(x):
  2
           import numpy as np
  3
           return np.cos(x) # Whoops! Wrong function
     Check Code
 Check failed
 assert_type passed for input
      array([0.
                        , 3.14159265, 6.28318531])
 assert_shape passed for input
                        , 3.14159265, 6.28318531])
       array([0.
 assert_numpy_allclose failed for
     x: array([0.
                          , 3.14159265, 6.28318531])
   > output 0: [ 1. -1. 1.]
   Output is not close to reference absolute difference is 3.0,
 relative difference is 3.0.
 Check was successful
 assert_2pi_periodic passed
 Check failed
 assert_numpy_allclose failed for
    x: array([0.
                          , 3.14159265, 6.28318531])
```