```
import numpy as np
import matplotlib.pyplot as plt
from math import log2
import csv
from src.interpolation import Interpolation
from src.minimization import Minimization
from src.source import draw
import matplotlib

name = "./out2.csv"
xa, xb = 0, 1
ya, yb = 0, 1
shape = np.array([[xa, ya], [xa, yb], [xb, yb], [xb, ya]])

matplotlib.use(’TkAgg’)


def print_errors():
    dicts = []
    for i in range(2, 5):  # 1, 9
        print(i)
        inter = Minimization(2**i, shape)  # Interpolation(2**i, shape)
        dicts.append({"n": 2**i,
                      "function error": inter.error(is_grad=False),
                      "gradient error": inter.error(is_grad=True)})

    with open(name, mode="w") as csv_file:
        fieldnames = ["n", "function error", "gradient error"]
        writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(dicts)


def draw_errors():
    n = [2**x for x in range(1, 9)]
    err = [Interpolation(i, shape).error() for i in n]
    err_grad = [Interpolation(i, shape).error(True) for i in n]

    f = np.vectorize(log2)
    def f1(x): return -2*x - 3
    def f2(x): return - x - 2
    f1 = np.vectorize(f1)
    f2 = np.vectorize(f2)
    z = range(1, 9)

    fig, (ax1, ax2) = plt.subplots(1, 2)
    x, y = n, err
    ax1.plot(f(x), f(y), label=’func_err’)
    ax1.plot(z, f1(z), ’ro’, label=’y = -2x - 3’)
    x, y = n, err_grad
    ax2.plot(f(x), f(y), label=’grad_err’)
    ax2.plot(z, f2(z), ’ro’, label=’y = -x - 2’)

    ax1.set_title("func_error(num_of_nodes)")
    ax1.set_xlabel("log(nodes)")
    ax1.set_ylabel("log(error)")
    ax1.grid()
    ax1.legend()

    ax2.set_title("grad_error(num_of_nodes)")
    ax2.set_xlabel("log(nodes)")
    ax2.set_ylabel("log(gradient error)")
    ax2.grid()
    ax2.legend()

    # plt.savefig("pic/Figure_3.png")
    plt.show()


def test(n):
    inter = Minimization(n, shape)
    print("Minimization error: ", inter.error(False))


def test2(n):
    inter = Interpolation(n, shape)
    print("Interpolation error: ", inter.error(False))


if __name__ == "__main__":
    n = 10
    test(n)
    test2(n)
    # print_errors()
    # draw_errors()
```