File 1

from xmlrpc.server import SimpleXMLRPCServer  
from xmlrpc.server import SimpleXMLRPCRequestHandler  
  
  
class FactorialServer:  
 def calculate\_factorial(self, n):  
 if n < 0:  
 raise ValueError("Input must be a non-negative integer.")  
 result = 1  
 for i in range(1, n + 1):  
 result \*= i  
 return result  
  
  
class RequestHandler(SimpleXMLRPCRequestHandler):  
 rpc\_paths = ('/RPC2',)  
  
  
# Create an XML-RPC server  
with SimpleXMLRPCServer(('localhost', 8000), requestHandler=RequestHandler) as server:  
 server.register\_introspection\_functions()  
  
 # Register the FactorialServer class  
 factorial\_server = FactorialServer()  
 server.register\_instance(factorial\_server)  
  
 print("FactorialServer is ready to accept requests.")  
  
 # Register a range of factorial calculation method  
 for i in range(1, 101):  
 server.register\_function(factorial\_server.calculate\_factorial, f'factorial\_{i}')  
  
 server.serve\_forever()

File 2

import xmlrpc.client  
import random  
  
# Create an XML-RPC client  
with xmlrpc.client.ServerProxy("http://localhost:8000/RPC2") as proxy:  
 try:  
 # Generate a random integer between 1 and 20  
 input\_value = random.randint(1, 20)  
 result = proxy.calculate\_factorial(input\_value)  
 print(f"Factorial of {input\_value} is: {result}")  
 except Exception as e:  
 print(f"Error: {e}")

output:



