Calculator\_2.0 Checkpoint

class Calculator :

def sum(a,b) :

return a+b

def sub(a,b):

return a-b

def mul(a,b) :

return a\*b

def div(a,b) :

if b != 0 :

return a/b

else :

raise ValueError("Division impossible !")

operations = {

"+":sum,

"-":sub,

"\*":mul,

"/":div

}

def \_\_int\_\_(self, opreations) :

self.operations = operations

def add\_operation(self, signe, operation) :

self.operations[signe] = operation

def calculate(self, x, signe\_op, y) :

if not (isinstance(x, (int, float)) and isinstance(y, (int, float))) :

raise ValueError("Les deux valeures doivent etre des nombres !\n")

elif signe\_op not in self.operations :

raise ValueError("Mauvais choix d'opération !\n")

else :

signe\_trouve = self.operations[signe\_op]

resultat = signe\_trouve(x, y)

print(f"{x} {signe\_op} {y} = {resultat}\n")

import math

def exp(a) :

return math.exp(a)

def sqrt(a) :

return math.sqrt(a)

def log(a,b) :

return math.log(a, b)

calculator = Calculator()

calculator.add\_operation("^", lambda x, y: exp(x))

calculator.add\_operation("sqrt", lambda x, y: sqrt(x))

calculator.add\_operation("log", log)

#for key, value in calculator.operations.items() :

#print(f"{key} : {value.\_\_name\_\_}")

def main() :

continuer = True

while continuer :

nombre1 = float(input("Veuillez entrer un nombre\n"))

signe\_operation = input("Veuillez entrer le signe de l'opération à exécuter\n")

nombre2 = float(input("Veuillez entrer un second nombre\n"))

calculator.calculate(nombre1,signe\_operation,nombre2)

choix = input("Voulez-vous continuer ?\n")

if choix == "oui" :

continuer = True

elif choix == "non" :

continuer = False

else :

print("choix invalide !")

continuer = False

main()