

untitled9

October 12, 2023

```
[2]: #      1      Python
from random import random as r
import random

random.seed(random.randint(-10000, 10000))

def calc_pi(x0, y0, r0, expNmb):
    '''
        PI
    '''
    positive_res = 0
    r02 = r0**2
    x_max, x_min = x0 - r0, x0 + r0
    y_max, y_min = y0 - r0, y0 + r0
    for k in range(expNmb):
        px, py = r(), r()
        xp = (x_max - x_min) * px + x_min
        yp = (y_max - y_min) * py + y_min
        if (xp-x0)**2 + (yp-y0)**2 < r02:
            positive_res += 1
    return 4 * positive_res / expNmb
```

```
[3]: '''
    2
    1
    ExpNmb = 10**4 (x0 = 1, y0 = 2, r0 = 5).
'''
print(calc_pi(1, 2, 5, 10**4))

'''
    ExpNmb = 104, 105, 106, 107, 108
    seria_1,
    seria_2,
    seria_3, seria_4, seria_5
'''
def get_serial(seria_number):
```

```

exp_nmb = 10**4
x0, y0, r0 = 1, 2, 5
print(f"Seria number {seria_number} start...")
res_list = []
while exp_nmb <= 10**6:
    pi = calc_pi(x0, y0, r0, exp_nmb)
    res_list.append(pi)
    exp_nmb = exp_nmb * 10
print(f"Seria number {seria_number} end.")
return res_list

seria_1 = get_serial("1")
seria_2 = get_serial("2")
seria_3 = get_serial("3")
seria_4 = get_serial("4")
seria_5 = get_serial("5")
print(f"seria_1 = {seria_1}")
print(f"seria_1 = {seria_2}")
print(f"seria_1 = {seria_3}")
print(f"seria_1 = {seria_4}")
print(f"seria_1 = {seria_5}")

```

```

3.162
Seria number 1 start...
Seria number 1 end.
Seria number 2 start...
Seria number 2 end.
Seria number 3 start...
Seria number 3 end.
Seria number 4 start...
Seria number 4 end.
Seria number 5 start...
Seria number 5 end.
seria_1 = [3.112, 3.1374, 3.140584]
seria_1 = [3.1588, 3.14332, 3.144952]
seria_1 = [3.1308, 3.14688, 3.145756]
seria_1 = [3.1388, 3.13624, 3.139972]
seria_1 = [3.1652, 3.14324, 3.14138]

```

```

[5]: '''
      3

      math      pi.      ,

      '''
seria_total = [seria_1, seria_2, seria_3, seria_4, seria_5]
from math import pi

```

```

seria_fault_single = []
for i in range(2):
    fault = []
    for j in range(2):
        fault.append(round(abs(seria_total[i][j] - pi)/pi, 10))
    print(f"Fault {i+1}: {fault}")
    seria_fault_single.append(fault)

seria_fault_total = []
for i in range(2):
    total_fault = 0
    for j in range(2):
        total_fault += seria_fault_single[j][i]
    total_fault = round(total_fault / 5, 10)
    seria_fault_total.append(total_fault)
print("Total fault: ", seria_fault_total)

'''

'''

```

Fault 1: [0.0094196342, 0.0013345631]
 Fault 2: [0.0054772685, 0.0005498314]
 Total fault: [0.0029793805, 0.0003768789]