

```
# Patrick McMillin
# PHYS 466 Dr. Doty
# 30 January 2018

# This script will define the Gaussian function, and then integrate it numerically.

import numpy as np

def main():
    Ten_Steps = Integrate_Trapezoid(Gaussian_Function, 0, 1, 0.1)
    print "The result when using 10 steps from 0 to 1 is:", Ten_Steps
    Fifty_Steps = Integrate_Trapezoid(Gaussian_Function, 0, 1, 0.02)
    print "The result when using 50 steps from 0 to 1 is:", Fifty_Steps

    Twenty_Steps = Integrate_Trapezoid(Gaussian_Function, -1, 1, 0.1)
    print "The result when using 20 steps from -1 to 1 is:", Twenty_Steps
    One_Hundred_Steps = Integrate_Trapezoid(Gaussian_Function, -1, 1, 0.02)
    print "The result when using 100 steps from -1 to 1 is:", One_Hundred_Steps

# Here is the Gaussian function we will integrate.
def Gaussian_Function(x):
    sigma = 1
    return (1/(np.sqrt(2*(np.pi)*(sigma**2))))*np.exp(-1*(x**2)/(2*(sigma**2)))

# Here is my homemade trapezoid rule integrator.
def Integrate_Trapezoid(f, Lower, Upper, Step_Size):
    Result = 0
    i = Lower+Step_Size
    while i < (Upper+Step_Size-0.01):
        Result += ((f(i-Step_Size)+f(i))/2)*Step_Size
        i += Step_Size
    return Result

main()
```

```
[PJM-MacBook~/Documents/PHYS_466>python Assignment_Submissions/Integrate_Gaussian.py
The result when using 10 steps from 0 to 1 is: 0.341143036536
The result when using 50 steps from 0 to 1 is: 0.34133668027
The result when using 20 steps from -1 to 1 is: 0.682286073072
The result when using 100 steps from -1 to 1 is: 0.68267336054
PJM-MacBook~/Documents/PHYS_466>
```