In [6]:
```
import scipy.integrate
import math

f1 = lambda x : math.sin(x)
f2 = lambda x : math.exp(-x**2)
#scipy.integrate.quad(f1, 0, 1)
scipy.integrate.quad(f2, 0, 1)
#scipy.integrate.quad(sin, -0.5, 0.5)
```

Out[6]:
```
(0.7468241328124271, 8.291413475940725e-15)
```

In [11]:
```
#Integrating polynomials
import scipy.integrate
import math
import numpy as np
p = np.poly1d([2, 5, 1])
#p(1)
P = np.polyint(p)
P
```

Out[11]:
```
poly1d([0.66666667, 2.5       , 1.        , 0.        ])
```

In [12]:
```
q = P(5) - P(1)
print(q)
```

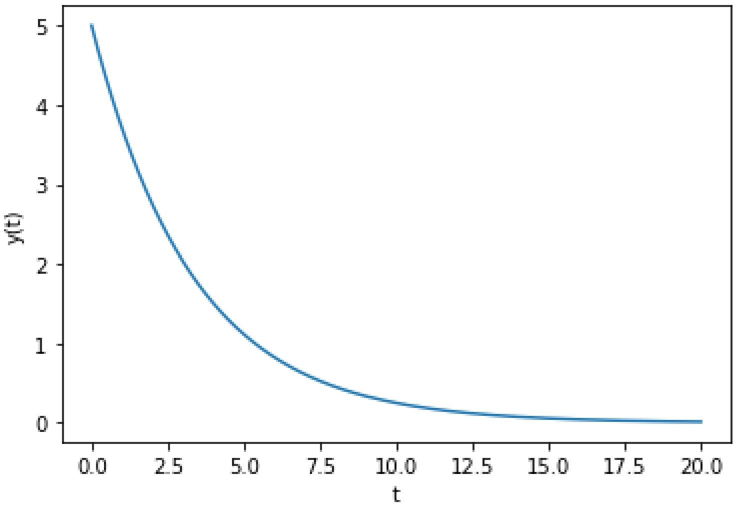
```
146.66666666666666
```

In [14]:
```
#Double integrals
import math
f = lambda x, y : 16*x*y
g = lambda x : 0
h = lambda y : math.sqrt(1-4*y**2)
scipy.integrate.dblquad(f, 0, 0.5, g, h)
```

Out[14]:
```
(0.5, 1.7092350012594845e-14)
```

In [16]:

```
#ODE
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
# fn that returns dy/dt
def f(y,t):
    k = 0.3
    dydt = -k*y
    return dydt
#initial condition
y0 = 5
#time points - start, end, divisions
t = np.linspace(0,20,1000)
#solve ODE
y = odeint(f,y0,t)
plt.plot(t,y)
plt.xlabel('t')
plt.ylabel('y(t)')
plt.show()
```



In [ ]:

In [ ]:

In [ ]: