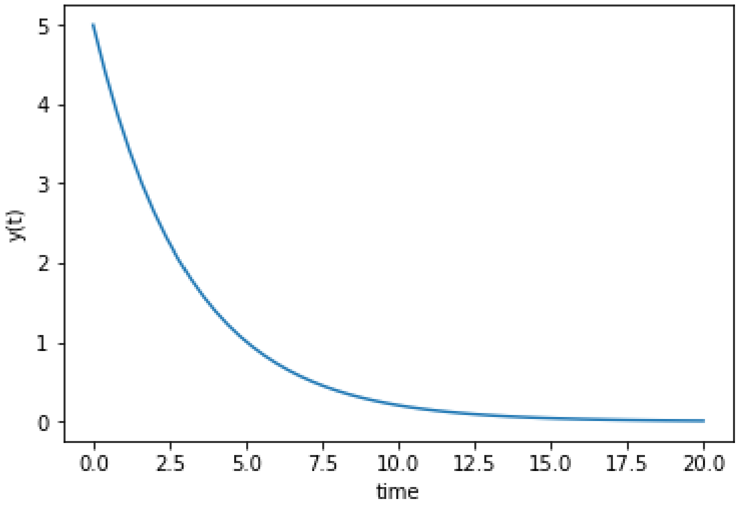
In [9]:

```
import numpy as np
import matplotlib.pyplot as plt

def ForwardEuler(f, U0, T, n):
    # Solve y'=f(y,t), y(0)=U0, with n steps until t=T.
    t = np.zeros(n+1)
    y = np.zeros(n+1)  # y[k] is the solution at time t[k]
    y[0] = U0
    t[0] = 0
    dt = T/float(n)

  # y[1] = y[0] + dt*f(y[0], t[0])         # pentru two-step
  # for k in range(1, n):

    for k in range(n):
        t[k+1] = t[k] + dt
        y[k+1] = y[k] + dt*f(y[k], t[k])
    return y, t

def f(y,t):
    k = 0.3
    dydt = -k * y
    return dydt

y, t = ForwardEuler(f, U0=5, T=20, n=50)

plt.plot(t,y)
plt.xlabel('time')
plt.ylabel('y(t)')
plt.show()
```



In [ ]: