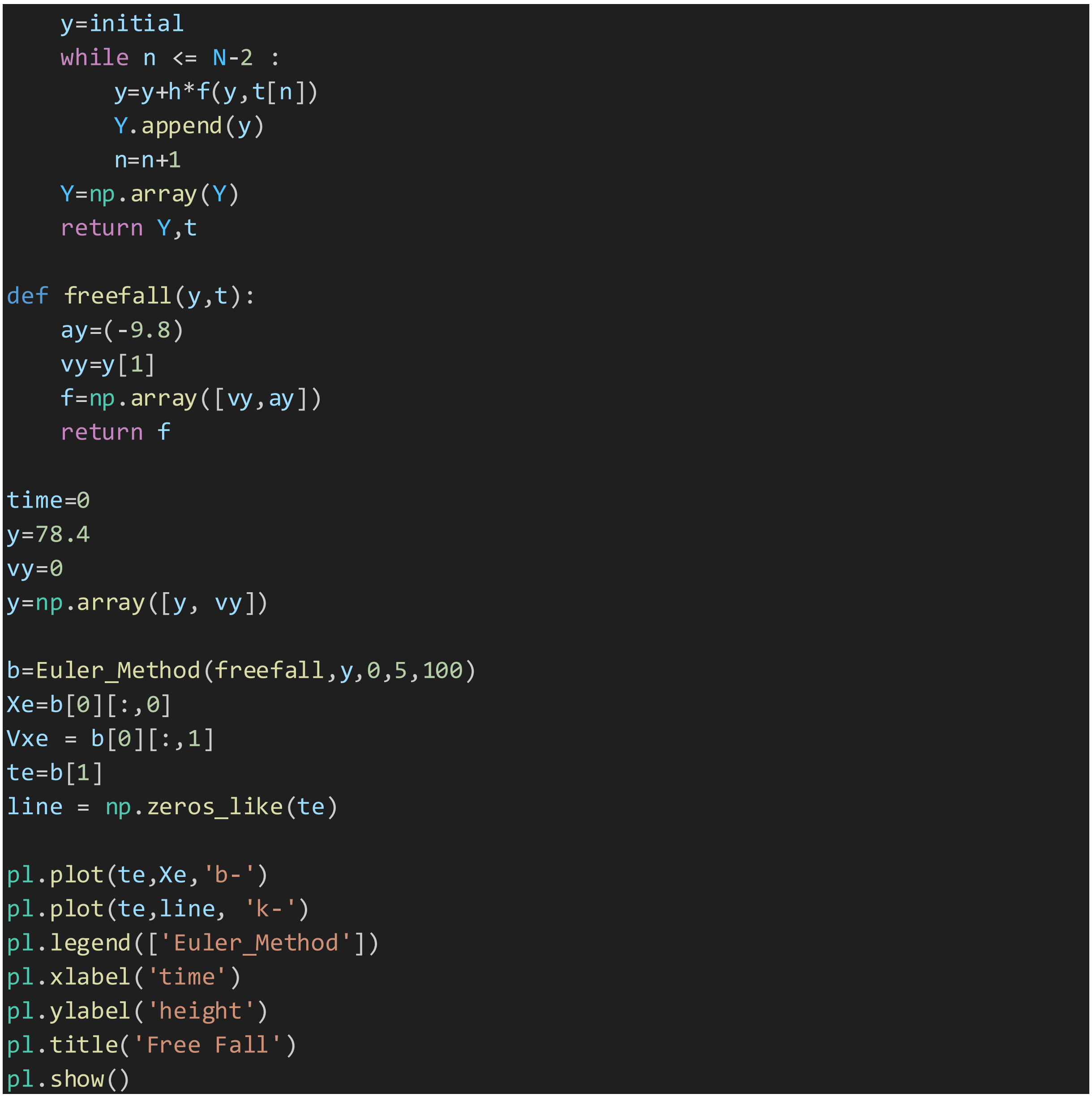
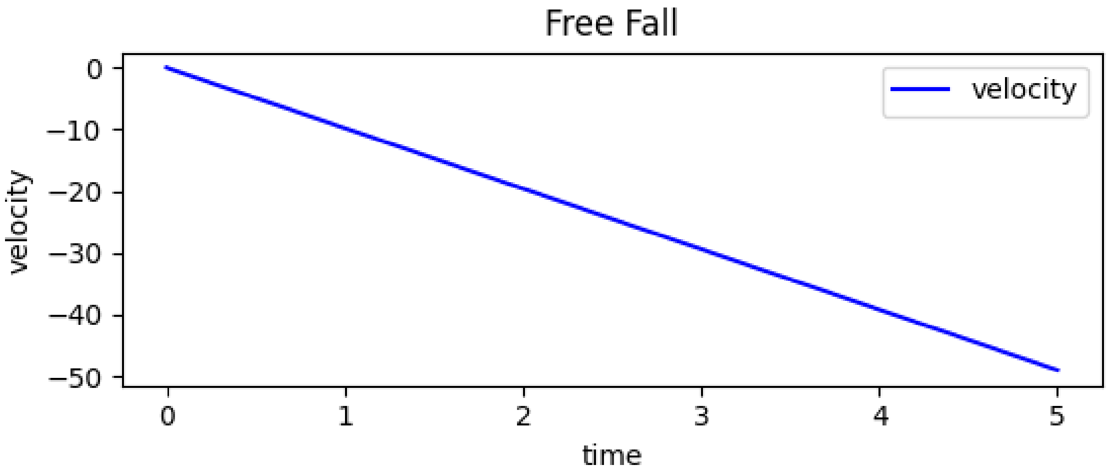
```python
import numpy as np
import pylab as pl
# 1
listA = [1,2,3,4]
listB = [5,6,7,8]

arrayA = np.array([1,2,3,4])
arrayB = np.array([4,5,6,7])

print("list + list : ", listA + listB) # [1, 2, 3, 4, 5, 6, 7, 8]
print("list x list : ", listA * listB) # TypeError
print("array + array : ", arrayA + arrayB) # [ 5  7  9 11]
print("array x array : ", arrayA * arrayB) # [ 4 10 18 28]

# 2
print ("list(range(10)) : ", list(range(10))) # [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print ("np.arange(2,10,1) : ", np.arange(2,10,1)) # [2 3 4 5 6 7 8 9]
print ("np.linspace(2,3,10) : ", np.linspace(2,3,10)) # [2.         2.11111111
2.22222222 2.33333333 2.44444444 2.555555562.66666667 2.77777778 2.88888889
3.        ]
print ("np.logspace(0,3,4) : ", np.logspace(0,3,4)) # [   1.   10.  100.
1000.]
print ("np.zeros([2,2]) : ", np.zeros([2,2])) # [[0. 0.]
                                            #  [0. 0.]]
print ("np.ones([2,2]) : ", np.ones([2,2])) # [[1. 1.]
                                          #  [1. 1.]]



# 3

# 3 - 1 (해를 알고있는 상황)
t = np.linspace(0,5,51)
vy = - 9.8 * t
pl.subplot(2,1,1)
pl.plot(t,vy,'b-')
pl.legend(['velocity'])
pl.xlabel('time')
pl.ylabel('velocity')
pl.title('Free Fall')
pl.show()

# 3 - 2 (오일러 방법 사용)
def Euler_Method(f,initial,start,stop,step) :
    t=np.linspace(start,stop,step)
    h=t[1]-t[0]
    Y=[initial]
    N=len(t)
    n=0
```



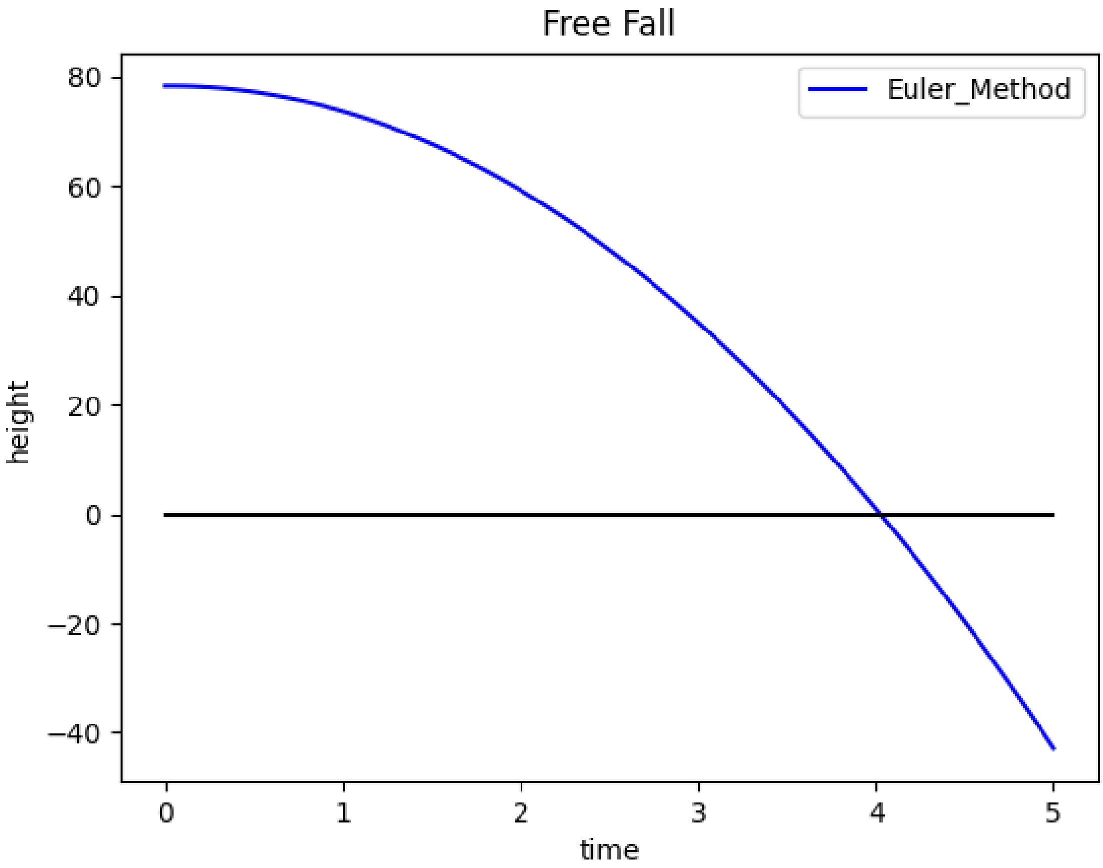
```
    y=initial
    while n <= N-2 :
        y=y+h*f(y,t[n])
        Y.append(y)
        n=n+1
    Y=np.array(Y)
    return Y,t

def freefall(y,t):
    ay=(-9.8)
    vy=y[1]
    f=np.array([vy,ay])
    return f

time=0
y=78.4
vy=0
y=np.array([y, vy])

b=Euler_Method(freefall,y,0,5,100)
Xe=b[0][:,0]
Vxe = b[0][:,1]
te=b[1]
line = np.zeros_like(te)

pl.plot(te,Xe,'b-')
pl.plot(te,line, 'k-')
pl.legend(['Euler_Method'])
pl.xlabel('time')
pl.ylabel('height')
pl.title('Free Fall')
pl.show()
```

그래프 1.



Free Fall
velocity
velocity
0
−10
−20
−30
−40
−50
0
1
2
3
4
5
time

그래프 2.



Free Fall
Euler_Method
height
80
60
40
20
0
−20
−40
0
1
2
3
4
5
time