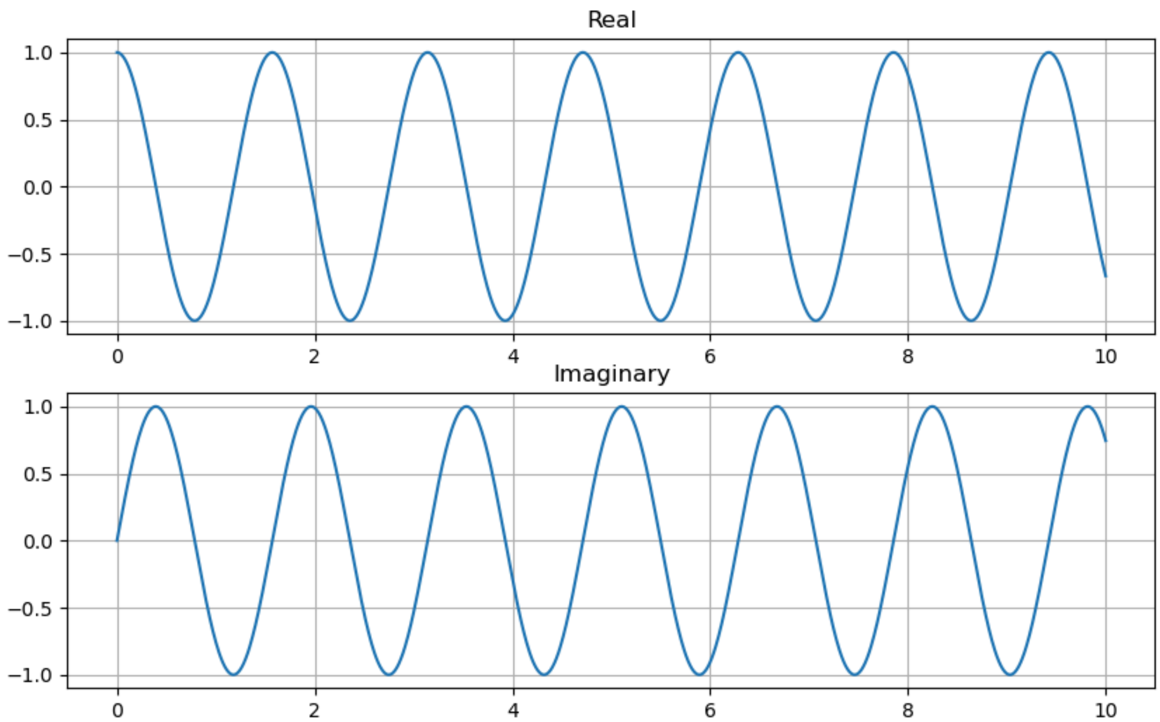
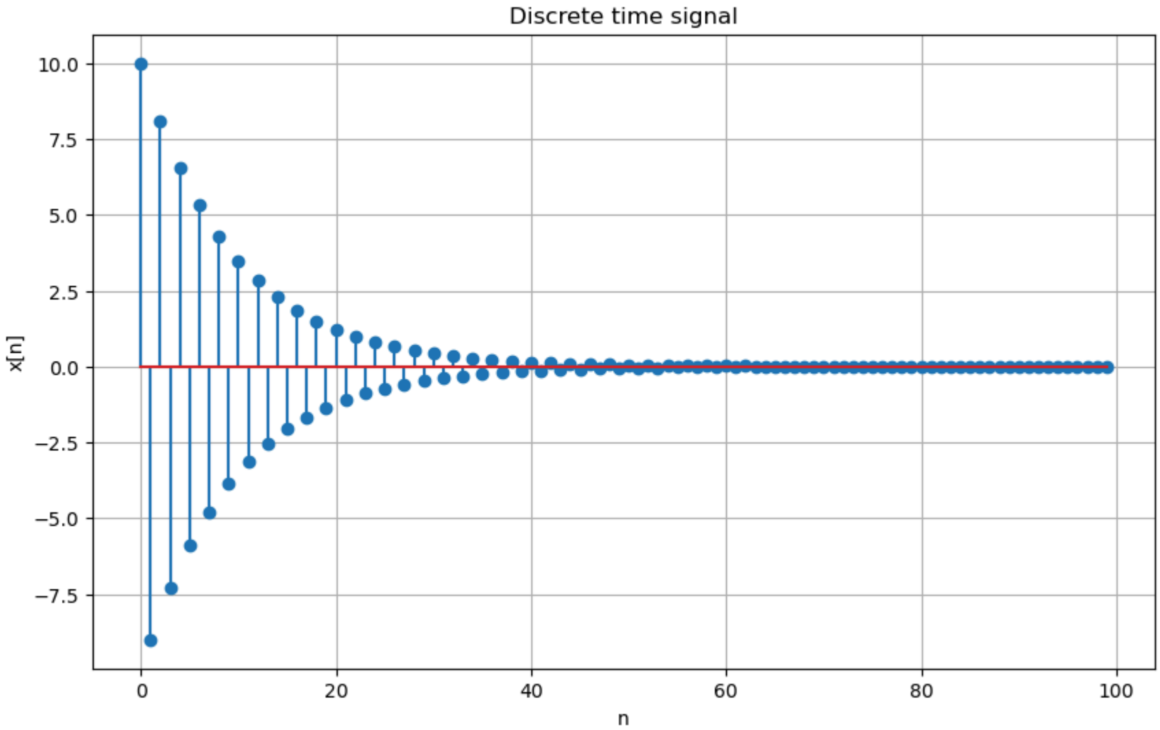
Task 1

In [25]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

# initialize variables
omega = 4
t = np.linspace(0, 10, 1000) # 1000 points in the range displays it more accurat
x_t = np.exp(1j * omega * t) # function x(t) = e^(jwt)

# set real and imaginary part for different graph axes
real_set = np.real(x_t)
imaginary_set = np.imag(x_t)

# set up the subplot and set the first as active
plt.figure(figsize=(10, 6))
plt.subplot(2, 1, 1)

# plot the first graph
plt.plot(t, real_set)
plt.title("Real")
plt.grid(True)

# set the second as active and plot the second graph
plt.subplot(2, 1, 2)
plt.plot(t, imaginary_set)
plt.title("Imaginary")
plt.grid(True)
plt.show()
```



Task 2

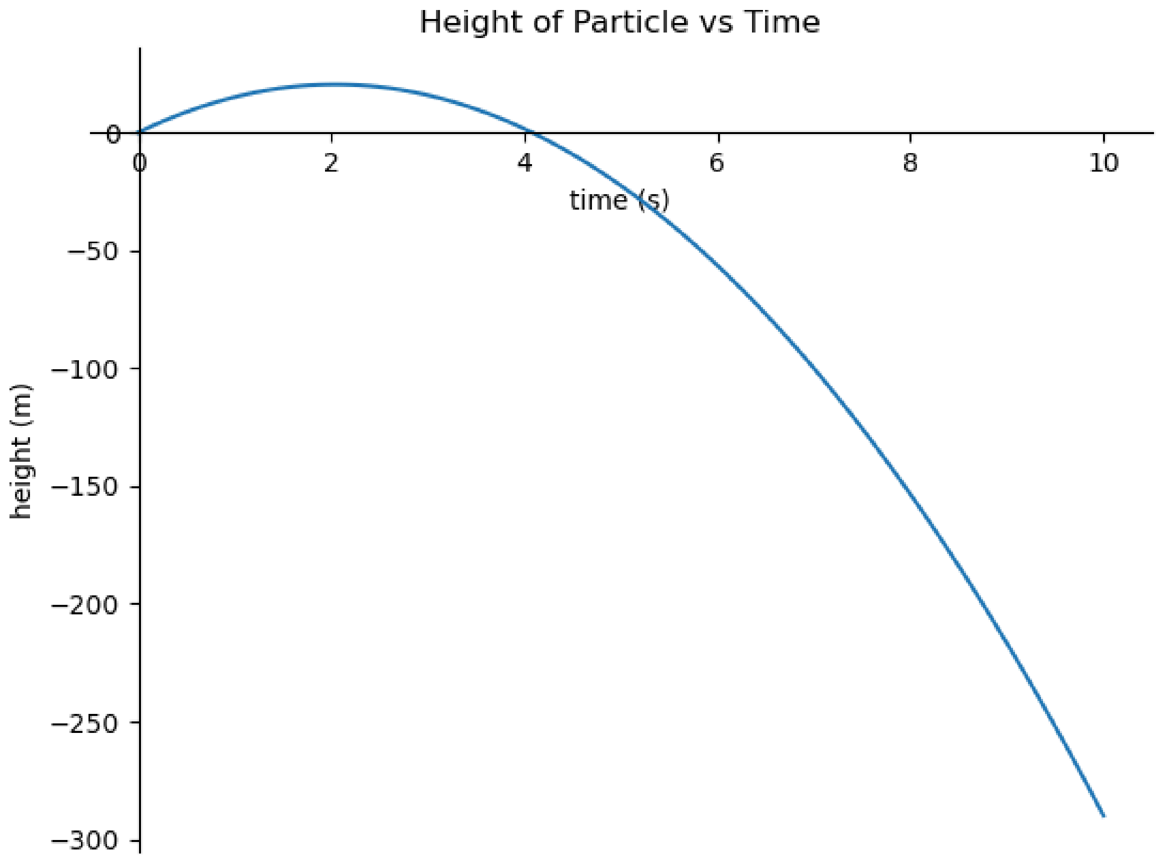
In [27]:
```
# setting up the numpy array
x = np.zeros(100)
x[0] = 10

# for loop to get other values of numpy array
for n in range(1,99):
    x[n] = -0.90*x[n-1]

plt.figure(figsize=(10, 6))
plt.stem(range(100), x)
plt.title("Discrete time signal")
plt.xlabel("n")
plt.ylabel("x[n]")
plt.grid(True)
```



Task 3

In [29]:
```
import sympy as sp

# Define symbolic variables
v_0, g, time = sp.symbols('v_0 g time')

# Define the expression for y(t)
y_t = v_0*time - 0.5*g*time**2

# Substitute numerical values for v_0 and g
y_t_subValue = y_t.subs({v_0: 20, g: 9.8})

# Plot the function over the range t = 0 to t = 10
sp.plot(y_t_subValue, (time, 0, 10),
```

```
        xlabel='time (s)',
        ylabel='height (m)',
        title='Height of Particle vs Time')
```



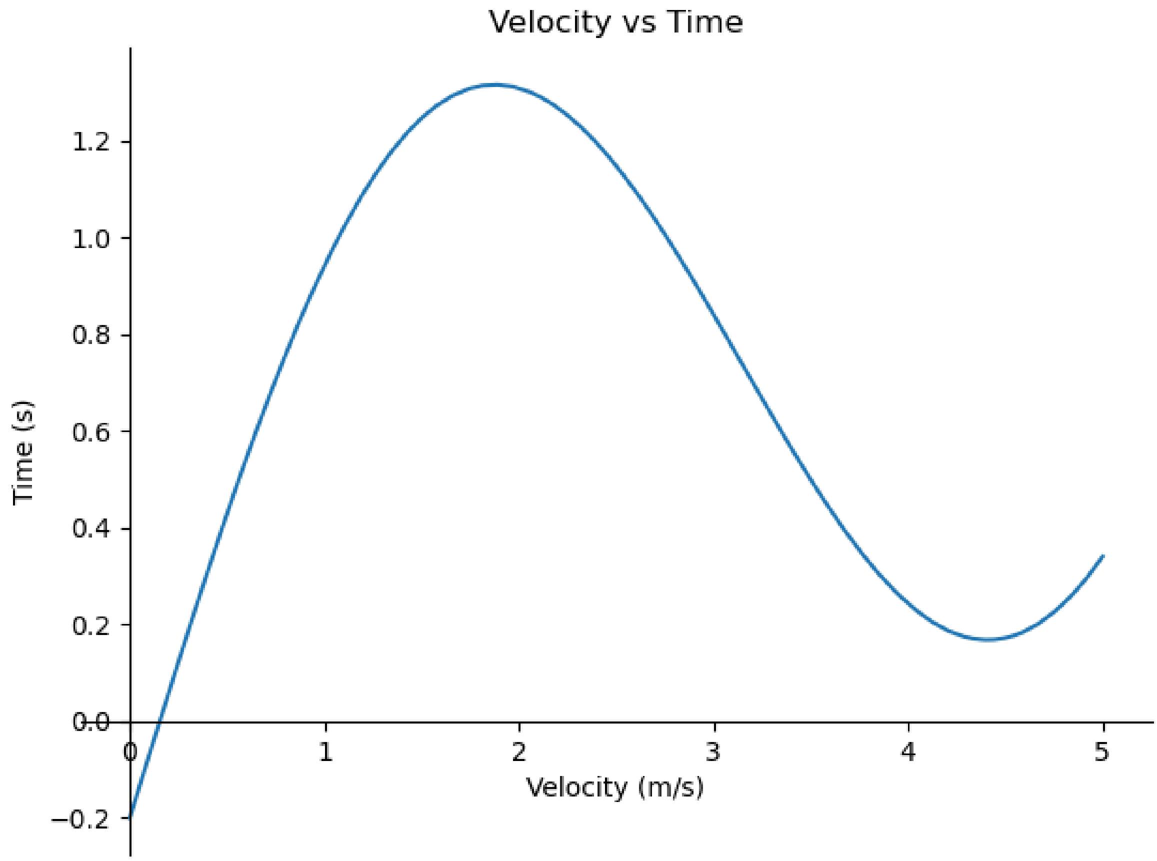
Out[29]: <sympy.plotting.plot.Plot at 0x2cf61d973b0>

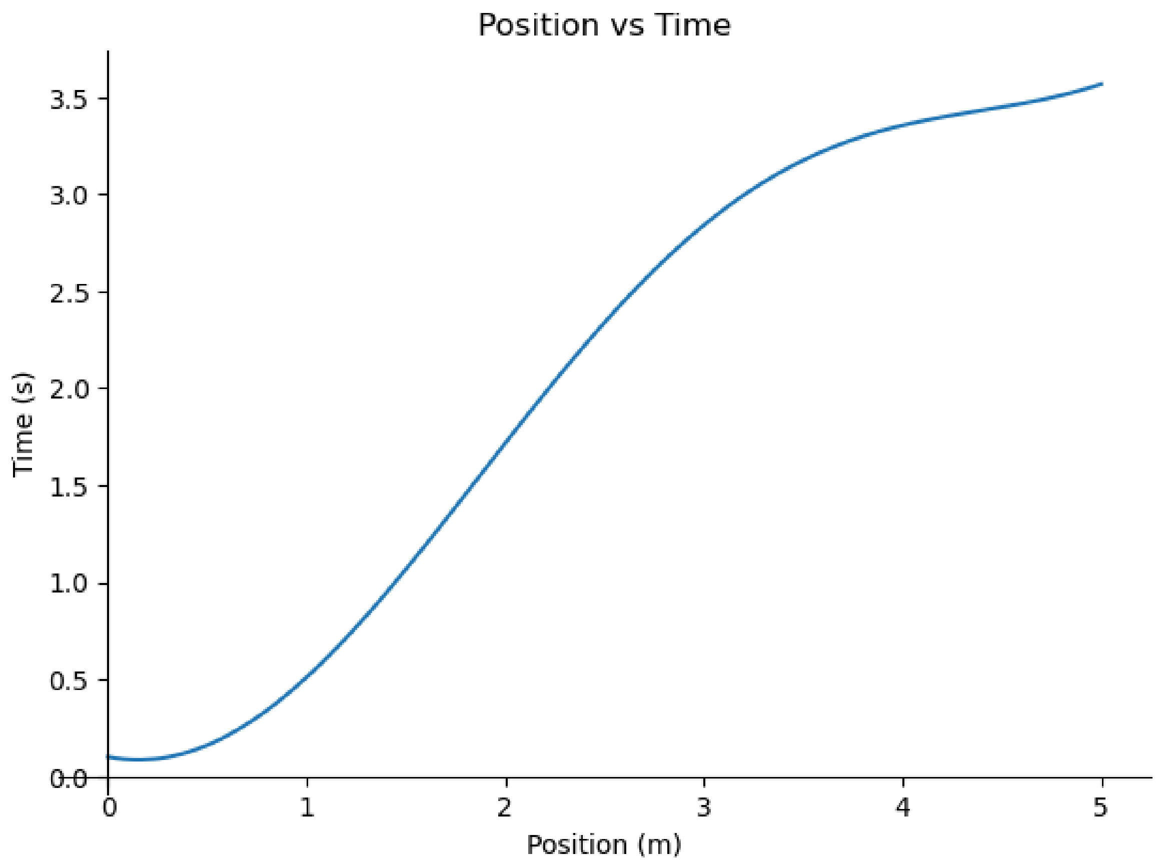
Task 4

In [31]:

```
# establish symbols
v_0, s_0, t = sp.symbols("v_0 s_0 t")
C1, C2 = sp.symbols("C1 C2")    # the constants of integration

a_t = 0.3 + sp.cos(t)

# velocity function
v_t = sp.integrate(a_t, t) + C1
v_t_sub = v_t.subs({C1: -0.2})

# position function
s_t = sp.integrate(v_t_sub, t) + C2
""" C2 = 1.1 because -cos(t) at t=0 sets they-value
at -1 so we need +1.1 to get the initial position of 0.1"""
s_t_sub = s_t.subs({C2: 1.1})

# plot the functions
sp.plot(v_t_sub, (t, 0, 5),
        title="Velocity vs Time",
        xlabel="Velocity (m/s)",
        ylabel="Time (s)")

sp.plot(s_t_sub, (t, 0, 5),
```

```
        title="Position vs Time",
        xlabel="Position (m)",
        ylabel="Time (s)")
```





Out[31]: <sympy.plotting.plot.Plot at 0x2cf61dc4ad0>