In [9]:
```
import sympy as sp
```

In [10]:
```
x=sp.symbols('x')
```

In [11]:
```
exp=x**2+3*x+2
```

In [12]:
```
simp_exp=sp.simplify(exp)
```

In [13]:
```
simp_exp
```

Out[13]: $x^2 + 3x + 2$

In [14]:
```
y=sp.diff(exp,x)
```

In [15]:
```
y
```

Out[15]: $2x + 3$

In [16]:
```
import sympy as sp
x=sp.symbols('x')
exp=x**2+3*x+2
simp_exp=sp.simplify(exp)
simp_exp
y=sp.diff(exp,x)
y
z=sp.diff(y,x)
z
```

Out[16]: $2$

In [17]:
```
#Pandas
```

In [18]:
```
import pandas as pd
```

In [36]:
```
df=pd.DataFrame({'A':[1,2,3],'B':[4,5,6]})
df
```

Out[36]:

| | A | B |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |

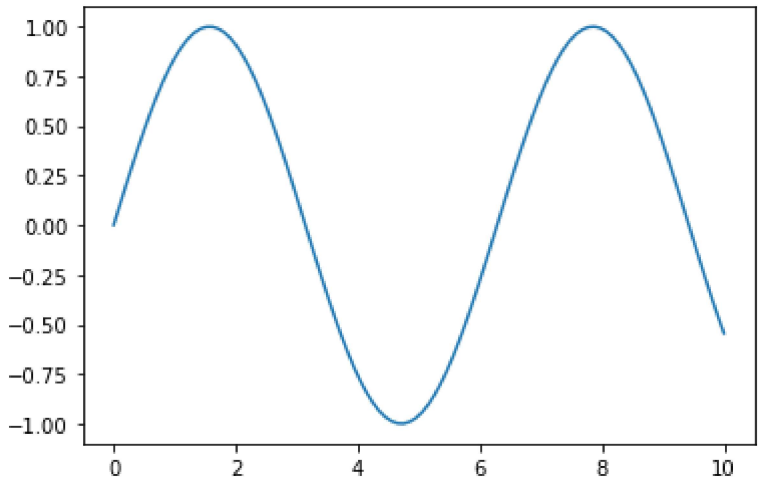
In [20]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [21]:
```
x=np.linspace(0,10,100)
```

In [22]:
```
y=np.sin(x)
```

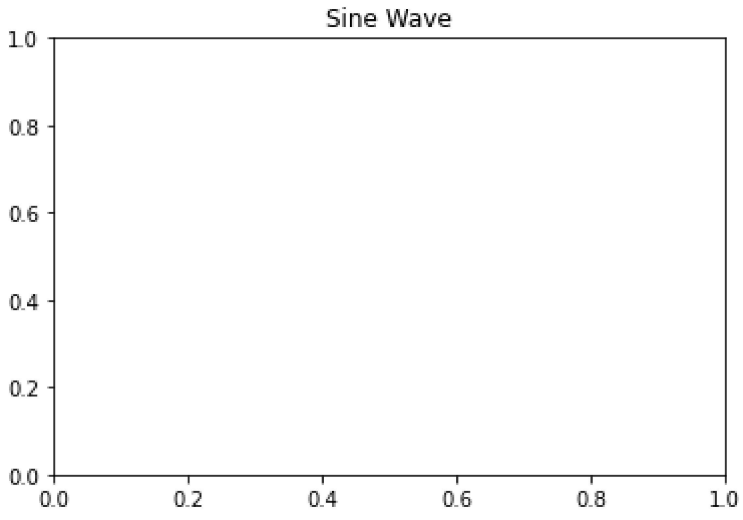
In [23]:
```
plt.plot(x,y)
```

Out[23]:
```
[<matplotlib.lines.Line2D at 0x1b16da86520>]
```




In [24]:
```
plt.title('Sine Wave')
```

Out[24]:
```
Text(0.5, 1.0, 'Sine Wave')
```




In [25]:
```
plt.show()
```

In [ ]:
```

```

In [26]:
```
import statsmodels.api as sm
```

In [28]:
```
model=sm.OLS(y,x).fit()
```

In [29]:
```
model
```

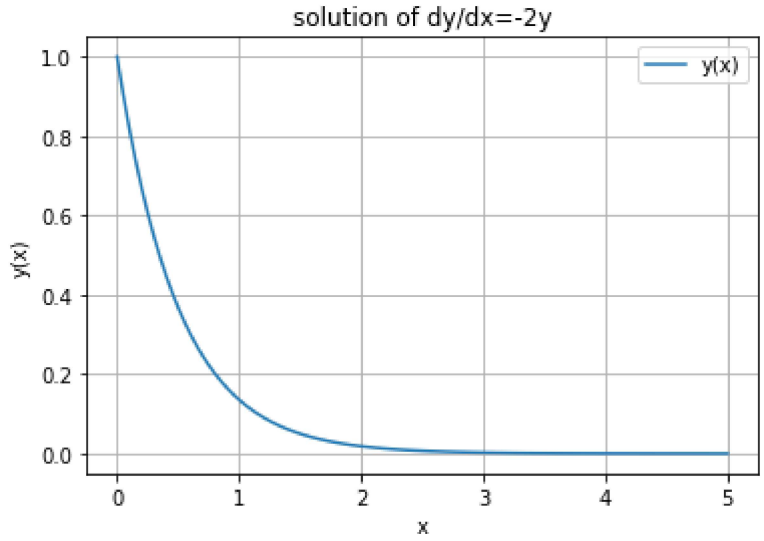
Out[29]:
```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x1b16da6761
0>
```

In [30]:
```
print(model.summary())
```

```
                                 OLS Regression Results
==============================================================================
==========
Dep. Variable:                      y   R-squared (uncentered):
0.035
Model:                            OLS   Adj. R-squared (uncentered):
0.026
Method:                 Least Squares   F-statistic:
3.625
Date:                Sat, 14 Dec 2024   Prob (F-statistic):
0.0598
Time:                        10:55:47   Log-Likelihood:
-102.76
No. Observations:                 100   AIC:
207.5
Df Residuals:                      99   BIC:
210.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
x1             0.0224      0.012      1.904      0.060      -0.001       0.04
6
==============================================================================
=
Omnibus:                       47.242   Durbin-Watson:                   0.01
2
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                8.73
8
Skew:                          -0.356   Prob(JB):                       0.012
7
Kurtosis:                       1.739   Cond. No.                         1.0
0
==============================================================================
=

Notes:
[1] R² is computed without centering (uncentered) since the model does not co
ntain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [33]:
```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
def dydx(y,x):
    return -2*y
y0=1
x=np.linspace(0,5,100)
y=odeint(dydx,y0,x)
plt.plot(x,y,label="y(x)")
plt.title("solution of dy/dx=-2y")
plt.xlabel("x")
plt.ylabel("y(x)")
plt.legend()
plt.grid()
```




In [ ]: