In [2]:
```
1 import numpy as np
```

In [3]:
```
1 import pandas as pd
```

In [4]:
```
1 import matplotlib.pyplot as plt
```

In [5]:
```
1 from sklearn.linear_model import LinearRegression
```

In [6]:
```
1 import os
```

In [7]:
```
1 folder_path = r'C:/Users/fpate/Desktop/AIT-580'
2 os.chdir(folder_path)
```

In [8]:
```
1 os.getcwd()
```

Out[8]:
```
'C:\\Users\\fpate\\Desktop\\AIT-580'
```

In [9]:
```
1 data = pd.read_csv('co2_annmean_gl.csv')
```

In [10]:
```
1 data
```

Out[10]:

	year	mean	unc
0	1979	336.85	0.10
1	1980	338.91	0.07
2	1981	340.11	0.08
3	1982	340.86	0.03
4	1983	342.53	0.05
5	1984	344.07	0.07
6	1985	345.54	0.07
7	1986	346.97	0.07
8	1987	348.68	0.09
9	1988	351.16	0.07
10	1989	352.78	0.06
11	1990	354.05	0.07
12	1991	355.39	0.06
13	1992	356.09	0.06
14	1993	356.83	0.07
15	1994	358.33	0.07
16	1995	360.17	0.05
17	1996	361.93	0.04
18	1997	363.05	0.05
19	1998	365.70	0.04
20	1999	367.80	0.07
21	2000	368.96	0.06
22	2001	370.57	0.05
23	2002	372.59	0.04
24	2003	375.15	0.04
25	2004	376.95	0.06
26	2005	378.98	0.05
27	2006	381.15	0.05
28	2007	382.90	0.05
29	2008	385.02	0.05
30	2009	386.50	0.04
31	2010	388.76	0.06
32	2011	390.63	0.06
33	2012	392.65	0.07

	year	mean	unc
34	2013	395.40	0.06
35	2014	397.34	0.05
36	2015	399.65	0.05
37	2016	403.06	0.07
38	2017	405.22	0.07
39	2018	407.61	0.07
40	2019	410.07	0.08
41	2020	412.44	0.06
42	2021	414.71	0.06

In [11]:
```
#Question 1:- Using the co2_annmean_gl.csv file, plot the change in globa
#Does the change appear to be increasing in a linear pattern?
```

In [12]:
```
plt.figure(figsize=(10, 60))
```

Out[12]:
```
<Figure size 720x4320 with 0 Axes>
```


```
<Figure size 720x4320 with 0 Axes>
```


In [13]:
```
x=data['year']
```

In [14]:
```
1 x
```

Out[14]:
```
0     1979
1     1980
2     1981
3     1982
4     1983
5     1984
6     1985
7     1986
8     1987
9     1988
10    1989
11    1990
12    1991
13    1992
14    1993
15    1994
16    1995
17    1996
18    1997
19    1998
20    1999
21    2000
22    2001
23    2002
24    2003
25    2004
26    2005
27    2006
28    2007
29    2008
30    2009
31    2010
32    2011
33    2012
34    2013
35    2014
36    2015
37    2016
38    2017
39    2018
40    2019
41    2020
42    2021
Name: year, dtype: int64
```

In [15]:

```
year=x[1:]
year
```

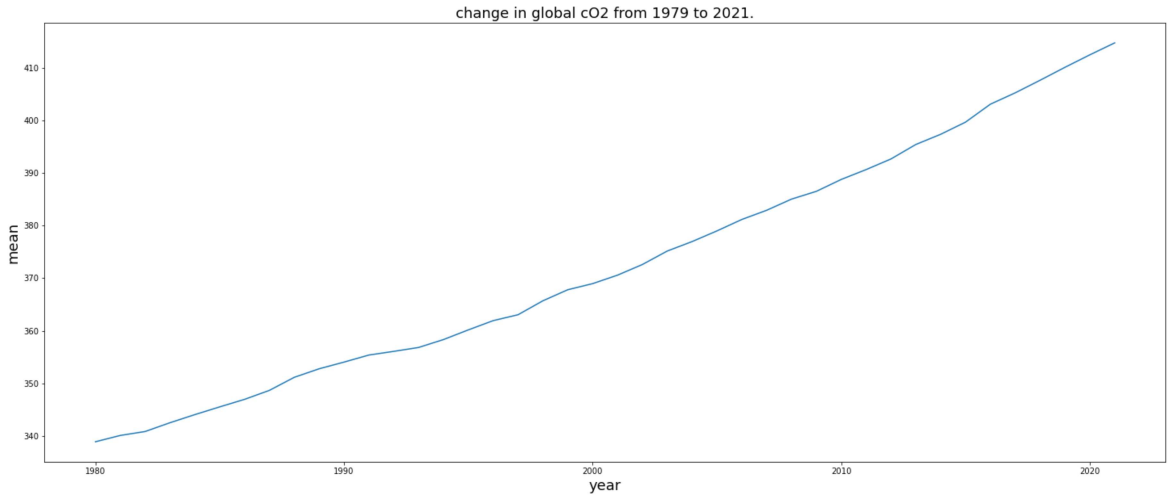
Out[15]:

```
1     1980
2     1981
3     1982
4     1983
5     1984
6     1985
7     1986
8     1987
9     1988
10    1989
11    1990
12    1991
13    1992
14    1993
15    1994
16    1995
17    1996
18    1997
19    1998
20    1999
21    2000
22    2001
23    2002
24    2003
25    2004
26    2005
27    2006
28    2007
29    2008
30    2009
31    2010
32    2011
33    2012
34    2013
35    2014
36    2015
37    2016
38    2017
39    2018
40    2019
41    2020
42    2021
Name: year, dtype: int64
```

In [16]:

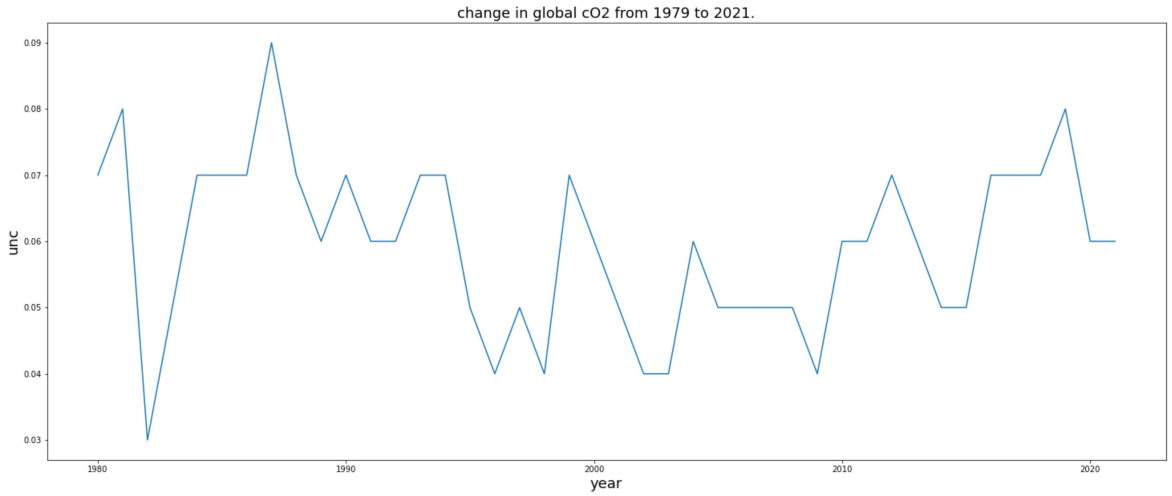
```
#x = range(len(data['year']))
fig = plt.figure(figsize=(25,10))
plt.plot(year, data['mean'][1:])
#plt.xticks(x, data['year'])
plt.title("change in global cO2 from 1979 to 2021.",fontsize=18)
plt.xlabel('year',fontsize=18)
plt.ylabel('mean',fontsize=18)
#plt.show()
#plt.savefig('pic.png')
```

Out[16]: Text(0, 0.5, 'mean')



In [17]:
```
fig = plt.figure(figsize=(25,10))
plt.plot(year, data['unc'][1:])
#plt.xticks(x, data['year'])
plt.title("change in global cO2 from 1979 to 2021.",fontsize=18)
plt.xlabel('year',fontsize=18)
plt.ylabel('unc',fontsize=18)
#plt.show()
#plt.savefig('pic.png')
```

Out[17]: Text(0, 0.5, 'unc')



In [18]:
```
# Yes it seems to be increasing in linear pattern for year and mean grapl
#In the first figure of year Vs mean, the graph tends to show the linear
# Vs unc, graph tends to show the non-linear graph.
```

In [19]:
```
# Question -2 Create and interpret a linear model showing the relationsh
```

In [20]:
```
x = data[['year']]
y = data['mean']
```

In [21]:
```
y[0]
```

Out[21]: 336.85

In [22]:
```
model = LinearRegression()
clf = model.fit(x,y)
prediction = model.predict(x)
```

In [23]:
```
print('Coefficient: ',clf.coef_)
print('Y intercept: ',clf.intercept_)
```

```
Coefficient:  [1.82260797]
Y intercept:  -3273.4924584717605
```

In [24]:

```
predictions = model.predict(x)
for i in range (len(predictions)):
    print('Actual:', y[i], 'predicted:',predictions[i])
```

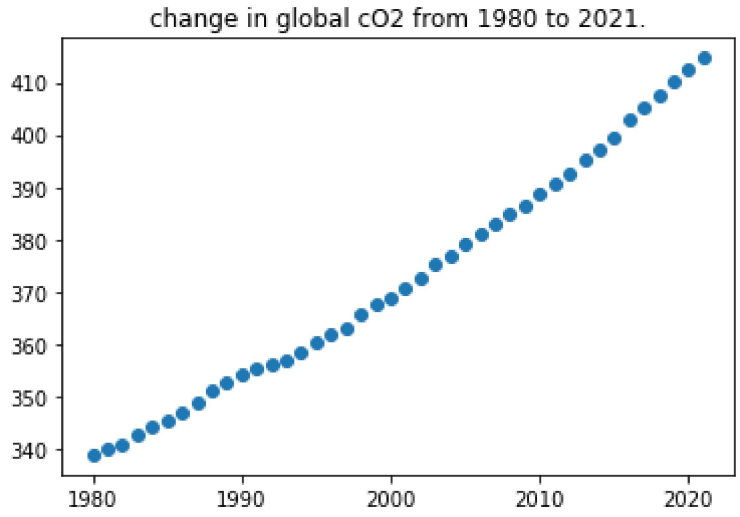
```
Actual: 336.85 predicted: 333.4487209302324
Actual: 338.91 predicted: 335.2713289036542
Actual: 340.11 predicted: 337.09393687707643
Actual: 340.86 predicted: 338.9165448504982
Actual: 342.53 predicted: 340.73915282392
Actual: 344.07 predicted: 342.56176079734223
Actual: 345.54 predicted: 344.384368770764
Actual: 346.97 predicted: 346.2069767441858
Actual: 348.68 predicted: 348.02958471760803
Actual: 351.16 predicted: 349.8521926910298
Actual: 352.78 predicted: 351.6748006644516
Actual: 354.05 predicted: 353.49740863787383
Actual: 355.39 predicted: 355.3200166112956
Actual: 356.09 predicted: 357.1426245847174
Actual: 356.83 predicted: 358.9652325581392
Actual: 358.33 predicted: 360.7878405315614
Actual: 360.17 predicted: 362.6104485049832
Actual: 361.93 predicted: 364.433056478405
Actual: 363.05 predicted: 366.2556644518272
Actual: 365.7 predicted: 368.078272425249
Actual: 367.8 predicted: 369.9008803986708
Actual: 368.96 predicted: 371.723488372093
Actual: 370.57 predicted: 373.5460963455148
Actual: 372.59 predicted: 375.3687043189366
Actual: 375.15 predicted: 377.1913122923588
Actual: 376.95 predicted: 379.0139202657806
Actual: 378.98 predicted: 380.8365282392024
Actual: 381.15 predicted: 382.6591362126246
Actual: 382.9 predicted: 384.4817441860464
Actual: 385.02 predicted: 386.3043521594682
Actual: 386.5 predicted: 388.1269601328904
Actual: 388.76 predicted: 389.9495681063122
Actual: 390.63 predicted: 391.772176079734
Actual: 392.65 predicted: 393.5947840531562
Actual: 395.4 predicted: 395.417392026578
Actual: 397.34 predicted: 397.2399999999998
Actual: 399.65 predicted: 399.06260797342156
Actual: 403.06 predicted: 400.8852159468438
Actual: 405.22 predicted: 402.7078239202656
Actual: 407.61 predicted: 404.53043189368736
Actual: 410.07 predicted: 406.3530398671096
Actual: 412.44 predicted: 408.1756478405314
Actual: 414.71 predicted: 409.99825581395316
```

In [25]:
```
from sklearn.metrics import mean_squared_error

y_pred = predictions
y_true = y
mean_squared_error(y_true, y_pred)
```

Out[25]: 5.043724117283534

In [26]:
```
plt.scatter(year, data['mean'][1:])
#plt.plot(x,predictions,'red')
plt.title("change in global cO2 from 1980 to 2021.",fontsize=12)
plt.show()
```



In [27]:
```
#Question 3:- Using your model, estimate and interpret the predicted mea
#future (2025, 2030, 2040, 2050, 2100). Are your predictions reasonable?
```

In [28]:
```
z = np.array([[2025],[2030],[2040],[2050],[2100]])
```

In [29]:
```
predictions = model.predict(z)
```

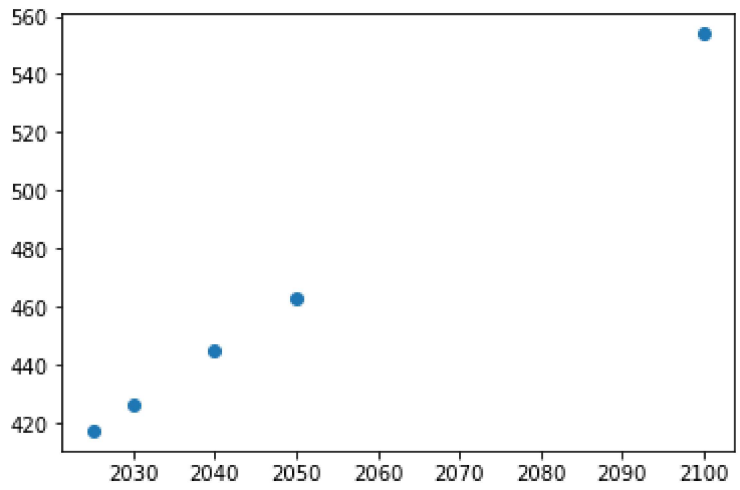
```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarnin
g: X does not have valid feature names, but LinearRegression was fitted wit
h feature names
  warnings.warn(
```

In [30]:
```
for i in range (len(predictions)):
    print('Actual:', z[i], 'predicted:',predictions[i])
```

```
Actual: [2025] predicted: 417.2886877076412
Actual: [2030] predicted: 426.40172757475057
Actual: [2040] predicted: 444.62780730896975
Actual: [2050] predicted: 462.8538870431894
Actual: [2100] predicted: 553.9842857142858
```

In [31]:
```
1 plt.scatter(z, predictions)
```

Out[31]: <matplotlib.collections.PathCollection at 0x1c9b71bb640>



In [32]:
```
1 # Question 4:- Explain how your model's accuracy can be improved.
```

In [33]:
```
1  #Model's mean squared error is 5.043724117283534, which shows that
2  #our model is not that accurate. Although it follows
3  #the linear pattern, its MSE value is approximately 5 and needs
4  #to be less and more accurate by increasing the values
5  # of mean and unc which are predicted. MSE of 5% provides the
6  #proof of linear trend of the data which is acceptable for
7  #linear Regression but for more accurate prediction either
8  #this error should me minimized based on data or
9  #different regression should be used to fit curve instead
10 #of lineBy doing so, our model will contain MSE value to be
11 #less i.e. approximately 1.03 or 1.2 something rather than
12 #5 or 6. The lower will be the value of MSE, more accurate
13 #values the model will be able to predict.
```