In [2]:
```
#pip install matplotlib numpy pandas
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
FILEPATH="./jeu_de_donnée/mont-bus-travel.csv"
HORIZON=np.random.randint(1,100)
FEATURE="#Passengers"
SEP=","
TITLE="Evolution des personnes dans le transport en commun"
```

In [4]:
```
df=pd.read_csv(FILEPATH,sep=SEP)
```

In [5]:
```
df.head()
```

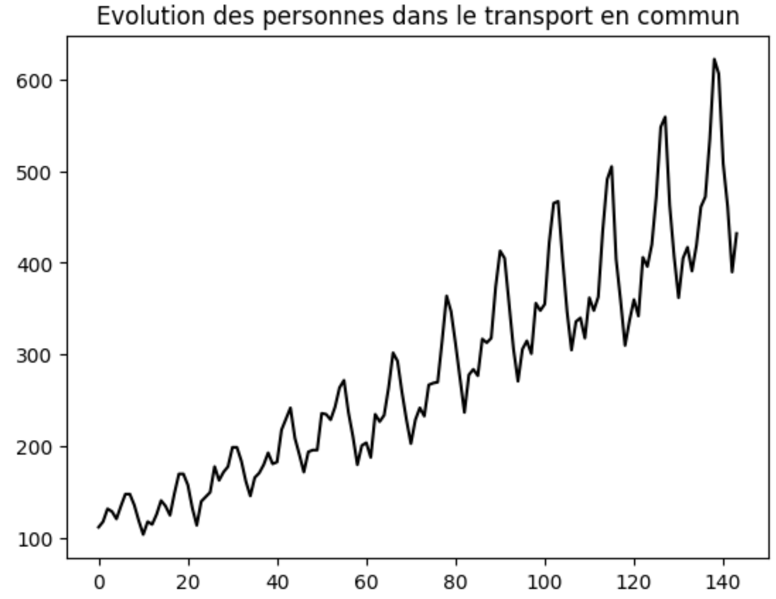
Out[5]:

	Month	#Passengers
0	1949-01	112
1	1949-02	118
2	1949-03	132
3	1949-04	129
4	1949-05	121

In [6]:
```
all_alpha=[0.001,0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 0.9]
```

In [7]:
```
plt.plot(df[FEATURE], c="black", ls='-')
plt.title(TITLE)
```

Out[7]:
```
Text(0.5, 1.0, 'Evolution des personnes dans le transport en commun')
```



In [8]:
```
def predict_simple_expo_lissage(data, alpha, taille, horizon=1):
    results=[]
    for i in range(taille-horizon):
        if i==0:
            results.append((1-alpha)*data[i])
        else:
            tmp=(1-alpha)*data[i]+alpha*results[i-1]
            results.append(tmp)
    return results
```

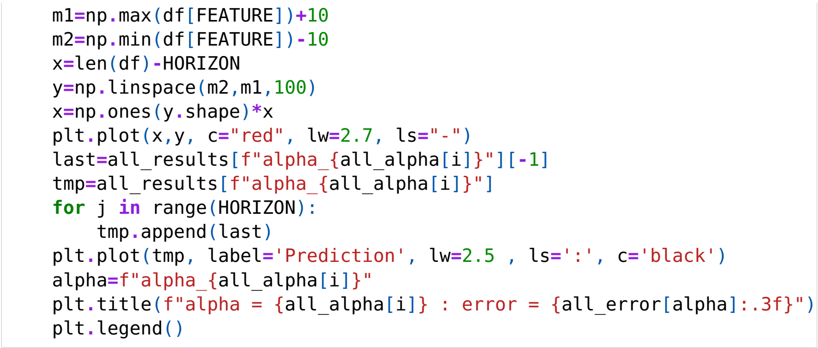
In [9]:
```
alpha=all_alpha[-1]
res=predict_simple_expo_lissage(df[FEATURE], alpha, len(df[FEATURE]))
```

In [10]:
```
all_results={}
for alpha in all_alpha:
    all_results[f"alpha_{alpha}"]=predict_simple_expo_lissage(df[FEATURE]
```

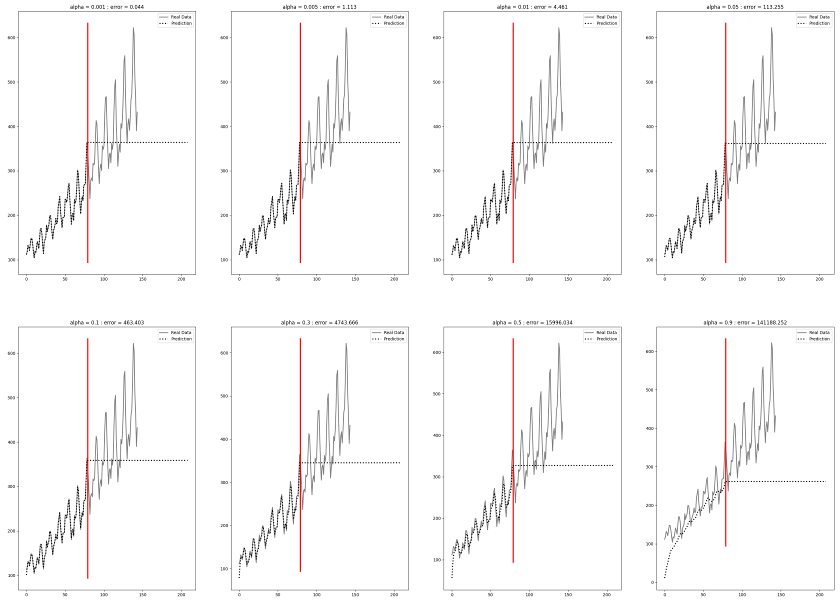
In [11]:
```
def sum_square_error(real, predic):
    result=real-predic
    result=result**2
    return np.sum(result)
```

In [12]:
```
all_error={}
for key in all_results.keys():
    real=np.array(df[FEATURE])
    predic=np.array(all_results[key])
    all_error[key]=sum_square_error(real[:real.shape[0]-HORIZON],predic)
```

In [14]:
```
plt.figure(figsize=[35, 25])
for i in range(len(all_alpha)):
    plt.subplot(2, 4, i+1)
    plt.plot(df[FEATURE] , label='Real Data', c='gray', lw=2)
```



```
    m1=np.max(df[FEATURE])+10
    m2=np.min(df[FEATURE])-10
    x=len(df)-HORIZON
    y=np.linspace(m2,m1,100)
    x=np.ones(y.shape)*x
    plt.plot(x,y, c="red", lw=2.7, ls="-")
    last=all_results[f"alpha_{all_alpha[i]}"][-1]
    tmp=all_results[f"alpha_{all_alpha[i]}"]
    for j in range(HORIZON):
        tmp.append(last)
    plt.plot(tmp, label='Prediction', lw=2.5 , ls=':', c='black')
    alpha=f"alpha_{all_alpha[i]}"
    plt.title(f"alpha = {all_alpha[i]} : error = {all_error[alpha]:.3f}")
    plt.legend()
```



alpha = 0.001 : error = 0.044

alpha = 0.005 : error = 1.113

alpha = 0.01 : error = 4.461

alpha = 0.05 : error = 113.255

alpha = 0.1 : error = 463.403

alpha = 0.3 : error = 4743.666

alpha = 0.5 : error = 15996.034

alpha = 0.9 : error = 141188.252