

# 1-import du package

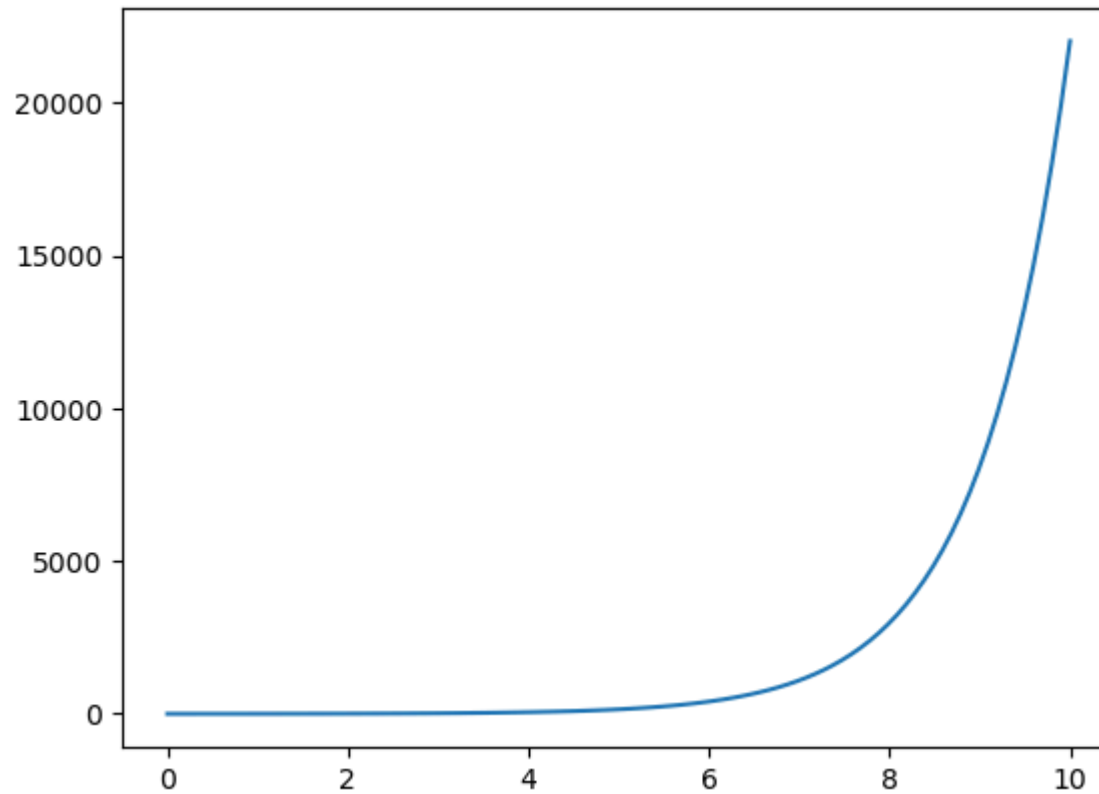
```
In [2]: import matplotlib.pyplot as plt  
import numpy as np
```

## 2-Tracer des graphiques simples

```
In [14]: # generer des donnees  
x=np.linspace(0,10,100)  
y=np.exp(x)
```

```
In [15]: # tracer un graphique simple  
plt.plot(x,y)
```

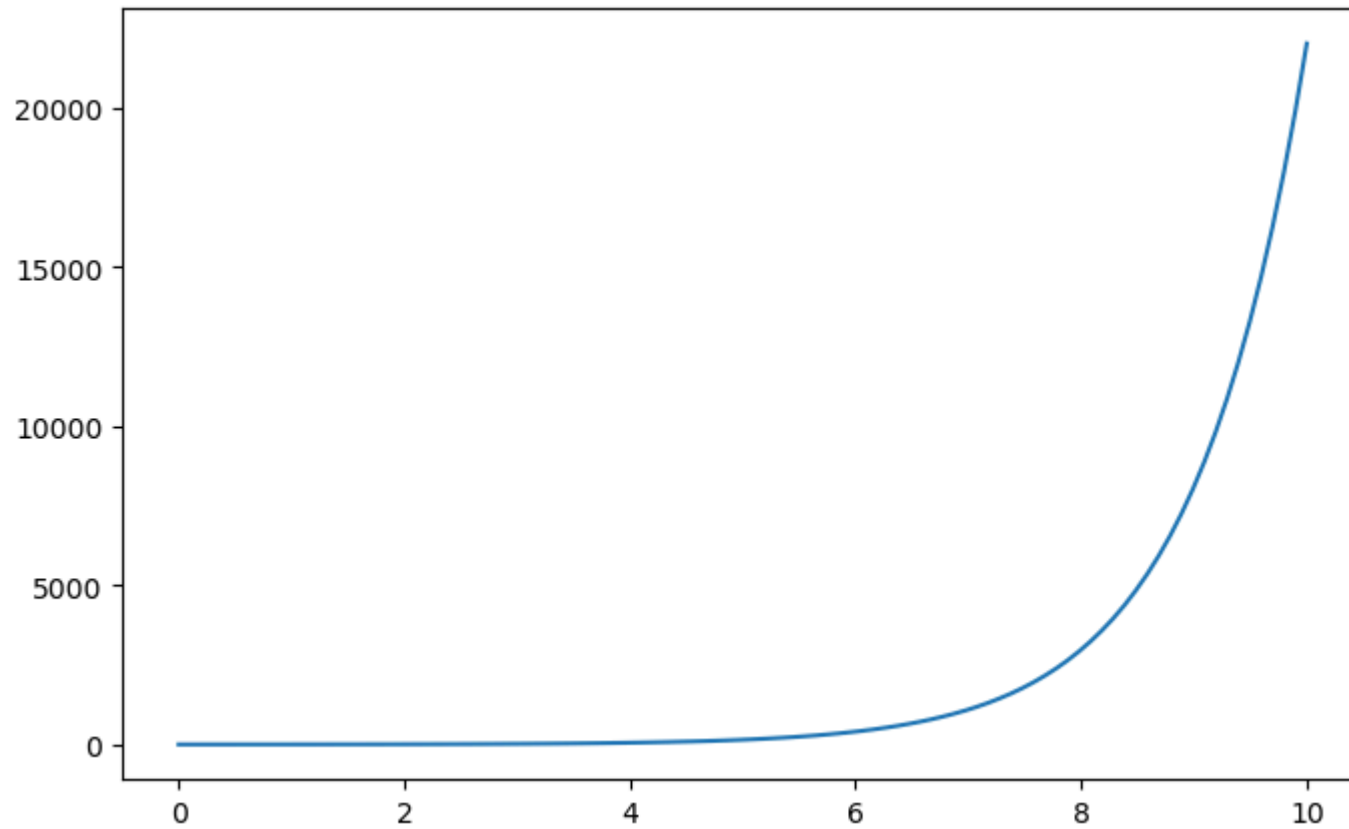
```
Out[15]: [<matplotlib.lines.Line2D at 0x154c0f37390>]
```



### 3-personnaliser un graphique

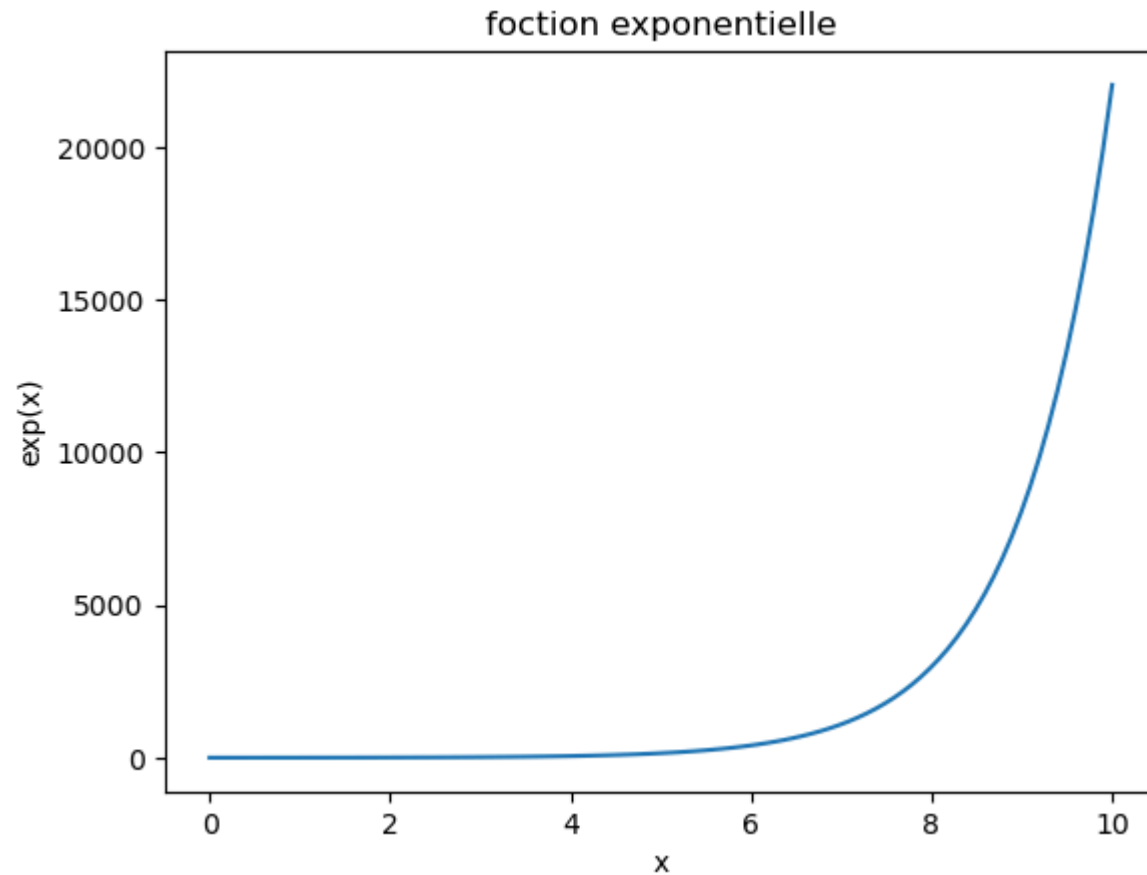
```
In [21]: # changer la taille d'un graphique avec la largeur et la hauteur en pouce  
plt.figure(figsize=(8,5))  
plt.plot(x,y)
```

```
Out[21]: [<matplotlib.lines.Line2D at 0x154c3c98750>]
```



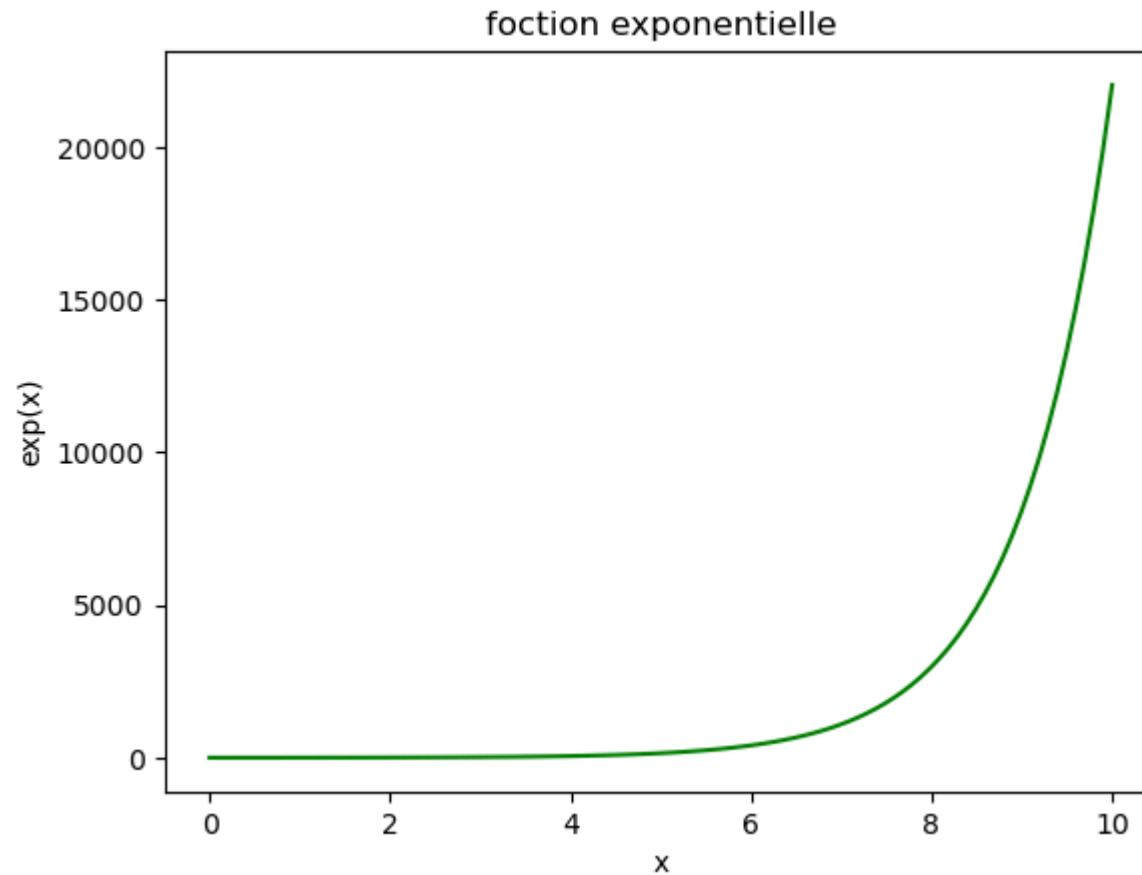
```
In [22]: # titres de la figure et des axes
plt.plot(x,y)
plt.title('fonction exponentielle') # ajouter un titre a la figure
plt.xlabel('x') # ajouter un nom a l'axe des abscisses
plt.ylabel('exp(x)') # ajouter un nom a l'axe des ordonnees
```

```
Out[22]: Text(0, 0.5, 'exp(x)')
```



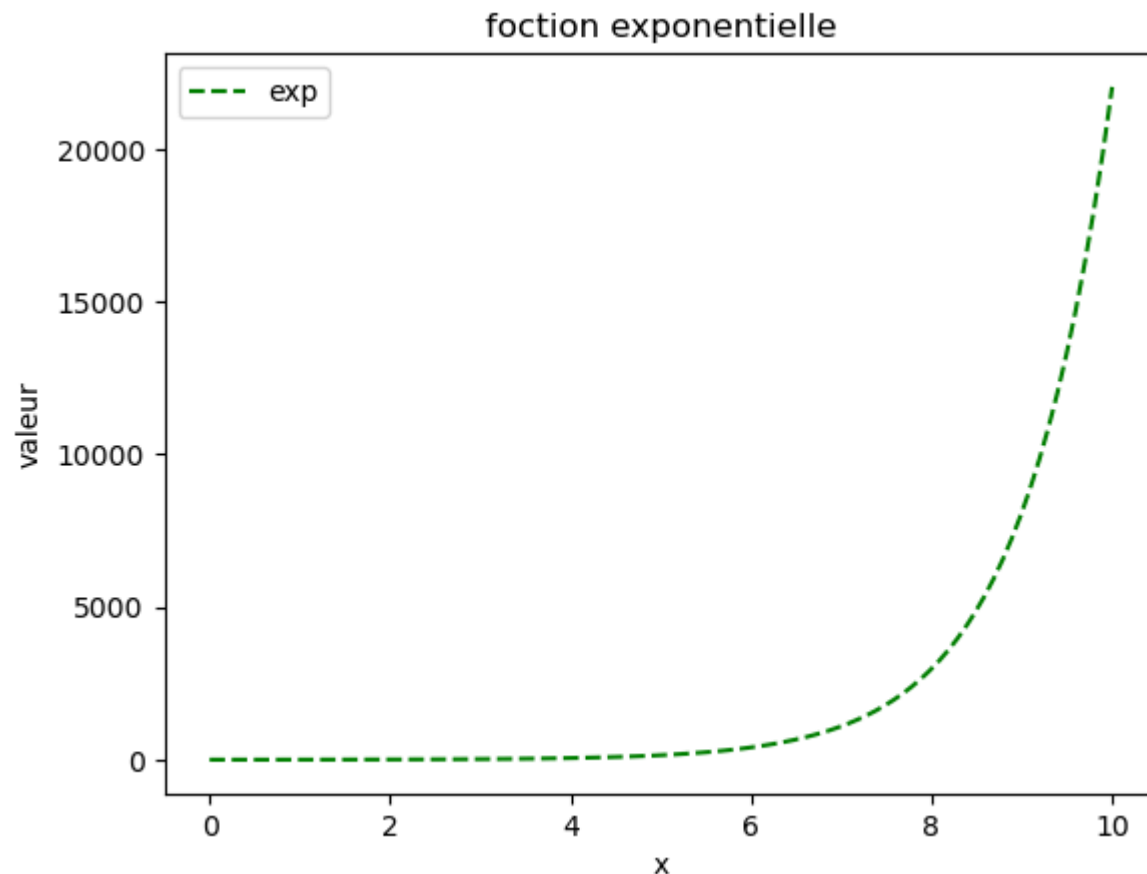
```
In [25]: # changer la couleur d'une courbe (c='color')
plt.plot(x,y, c='green')
plt.title('foction exponentielle') # ajouter un titre a la figure
plt.xlabel('x') # ajouter un nom a l'axe des abscisses
plt.ylabel('exp(x)') # ajouter un nom a l'axe des ordonnees
```

```
Out[25]: Text(0, 0.5, 'exp(x)')
```



```
In [27]: # changer le type de ligne et ajouter une legende
plt.plot(x,y, c='green', ls='--',label='exp')
plt.title('foction exponentielle') # ajouter un titre a la figure
plt.xlabel('x') # ajouter un nom a l'axe des abscisses
plt.ylabel('valeur') # ajouter un nom a l'axe des ordonnees
plt.legend()
```

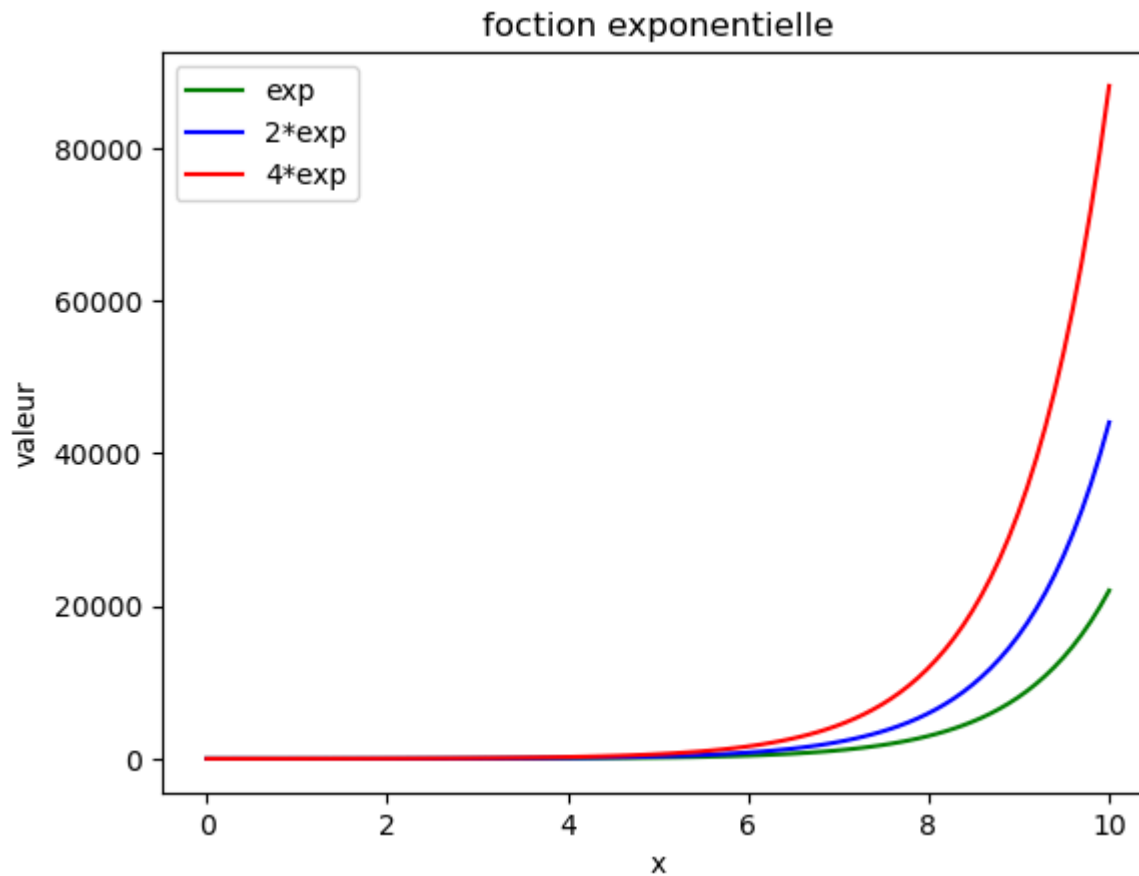
```
Out[27]: <matplotlib.legend.Legend at 0x154c8471590>
```



## 4-faire plusieurs courbes sur un meme grahique

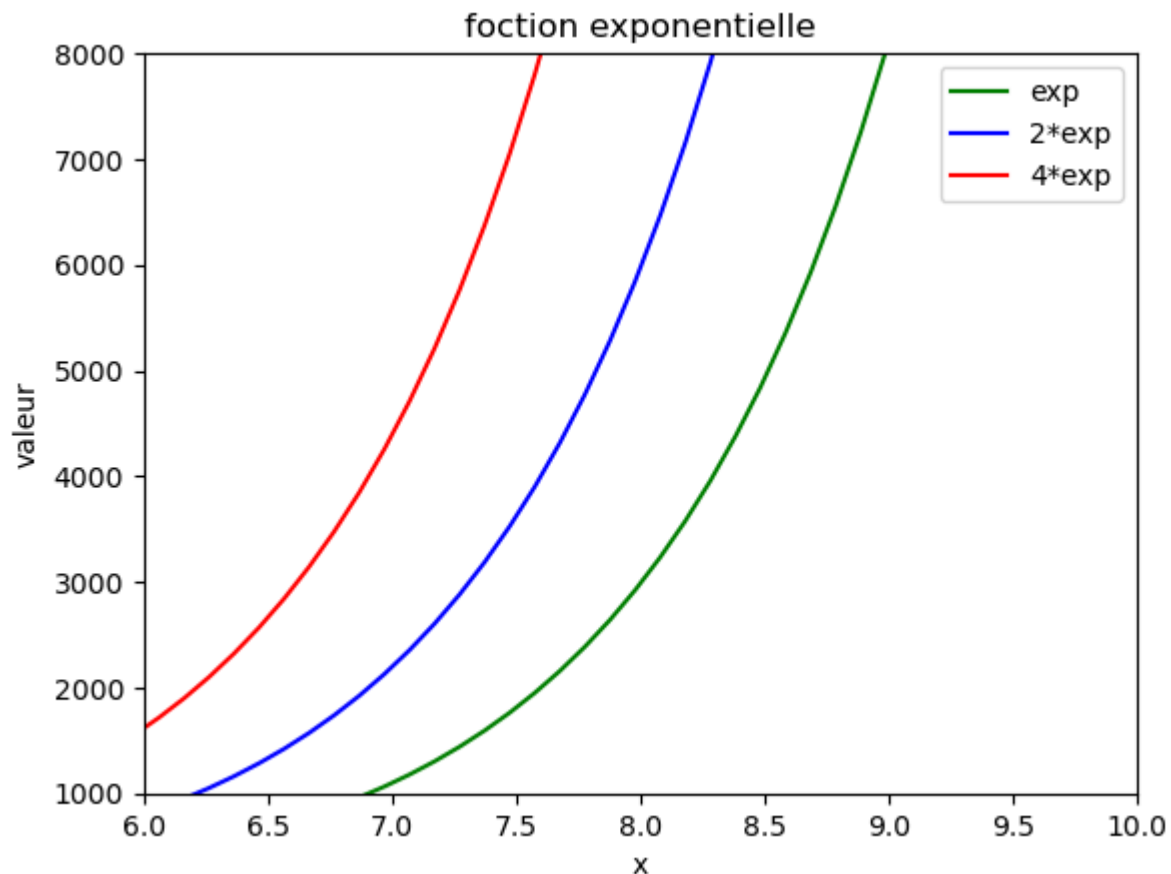
```
In [32]: # creer un graphique avec plusieurs courbes
z=y*2
f=z*2
plt.plot(x,y, c='green', ls='--',label='exp')
plt.plot(x,z, c='blue', ls='--',label='2*exp')
plt.plot(x,f, c='red', ls='--',label='4*exp')
plt.title('foction exponentielle') # ajouter un titre a la figure
plt.xlabel('x') # ajouter un nom a l'axe des abscisses
plt.ylabel('valeur') # ajouter un nom a l'axe des ordonnees
plt.legend()
```

Out[32]: <matplotlib.legend.Legend at 0x154c8b77010>



```
In [34]: # changer la limite des axes (faire le focus)
plt.plot(x,y, c='green', ls='-', label='exp')
plt.plot(x,z, c='blue', ls='-', label='2*exp')
plt.plot(x,f, c='red', ls='-', label='4*exp')
plt.xlim((6,10))
plt.ylim((1000,8000))
plt.title('foction exponentielle') # ajouter un titre a la figure
plt.xlabel('x') # ajouter un nom a l'axe des abscisses
plt.ylabel('valeur') # ajouter un nom a l'axe des ordonnees
plt.legend()
```

Out[34]: <matplotlib.legend.Legend at 0x154c3cec950>

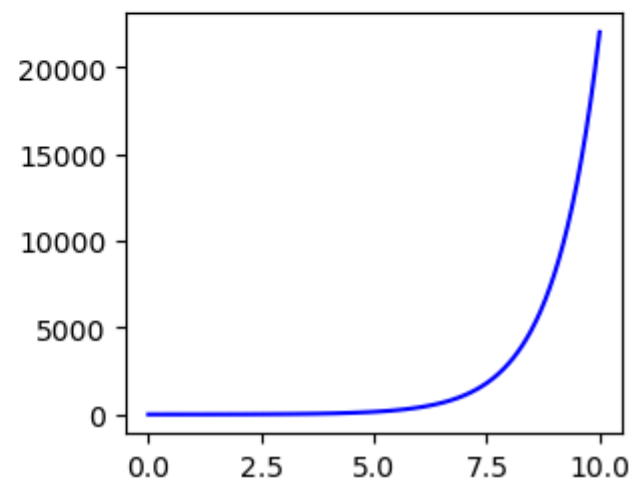
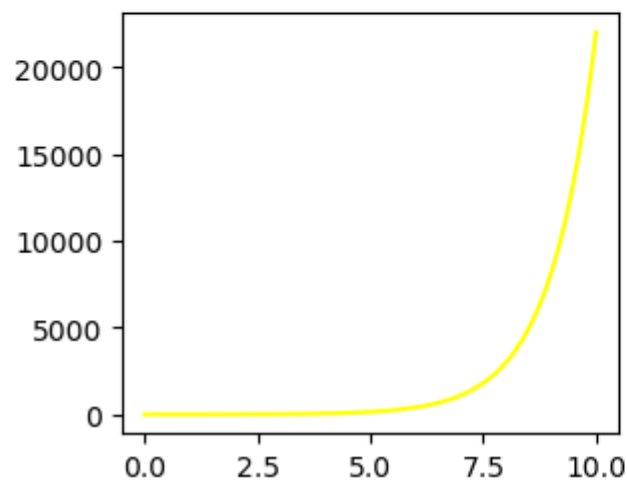
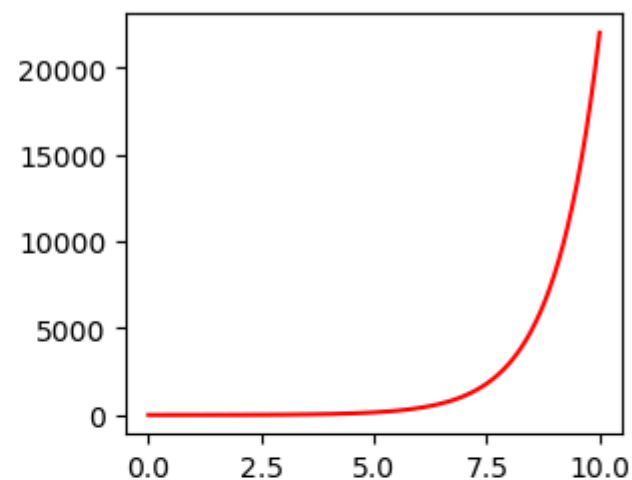
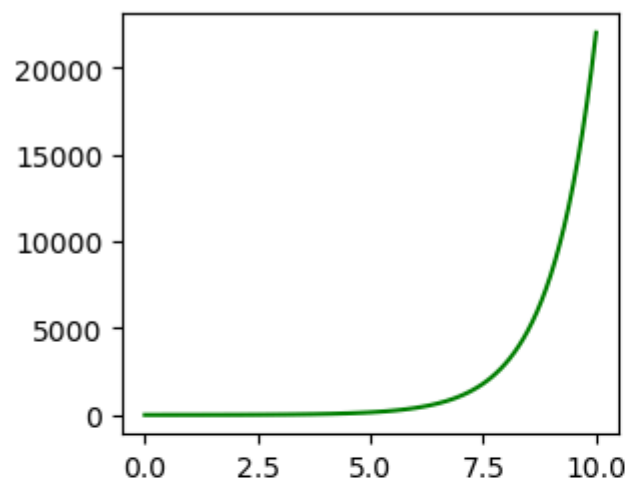


## 5-faire plusieurs graphiques sur une meme figure

```
In [42]: # avec subplot
plt.figure(figsize=(8,6))
plt.subplot(2,2,1)
plt.plot(x,y, c='green', ls='--',label='exp')
plt.subplot(2,2,2)
plt.plot(x,y, c='red', ls='--',label='exp')
plt.subplot(2,2,3)
plt.plot(x,y, c='yellow', ls='--',label='exp')
plt.subplot(2,2,4)
```



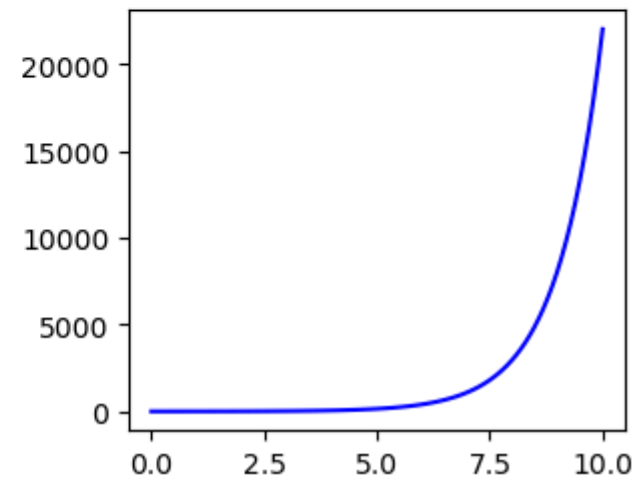
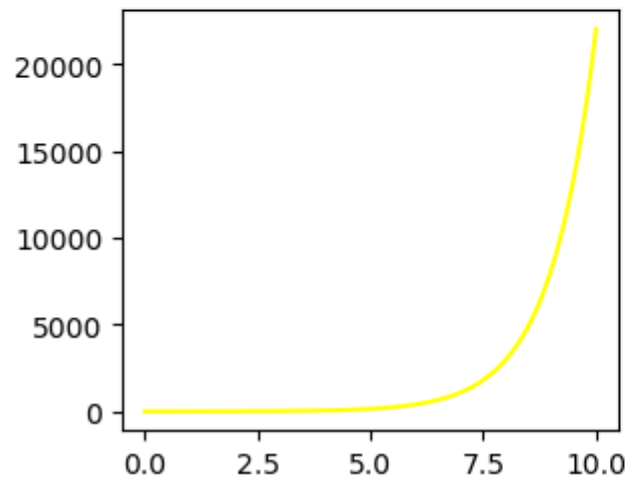
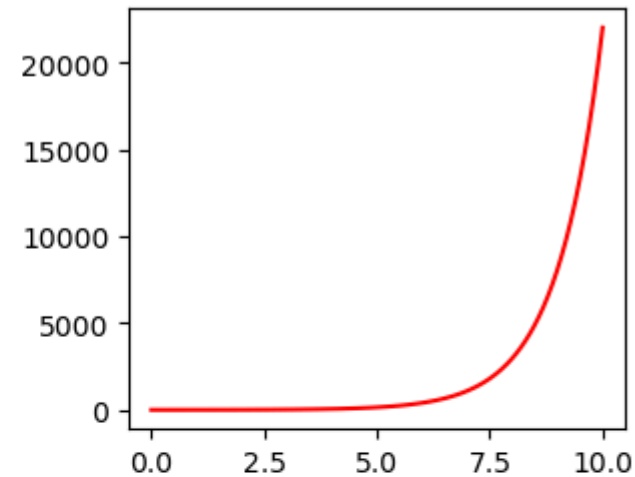
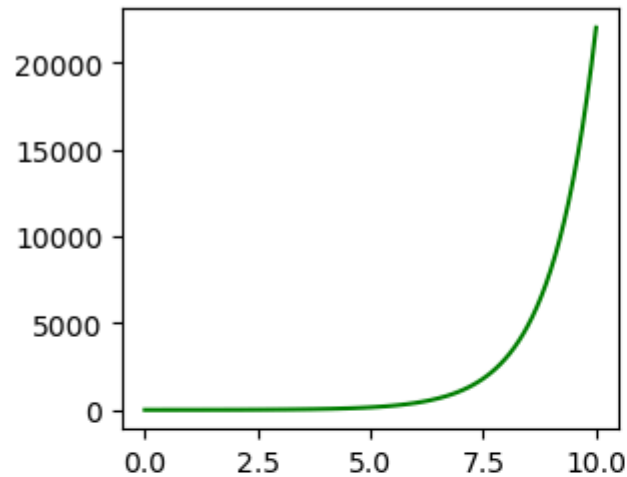
```
plt.plot(x,y, c='blue', ls='-',label='exp')
plt.subplots_adjust(wspace=0.5)
```



## 6-sauvegarder les graphiques

```
In [44]: plt.figure(figsize=(8,6))
plt.subplot(2,2,1)
plt.plot(x,y, c='green', ls='-',label='exp')
plt.subplot(2,2,2)
plt.plot(x,y, c='red', ls='-',label='exp')
```

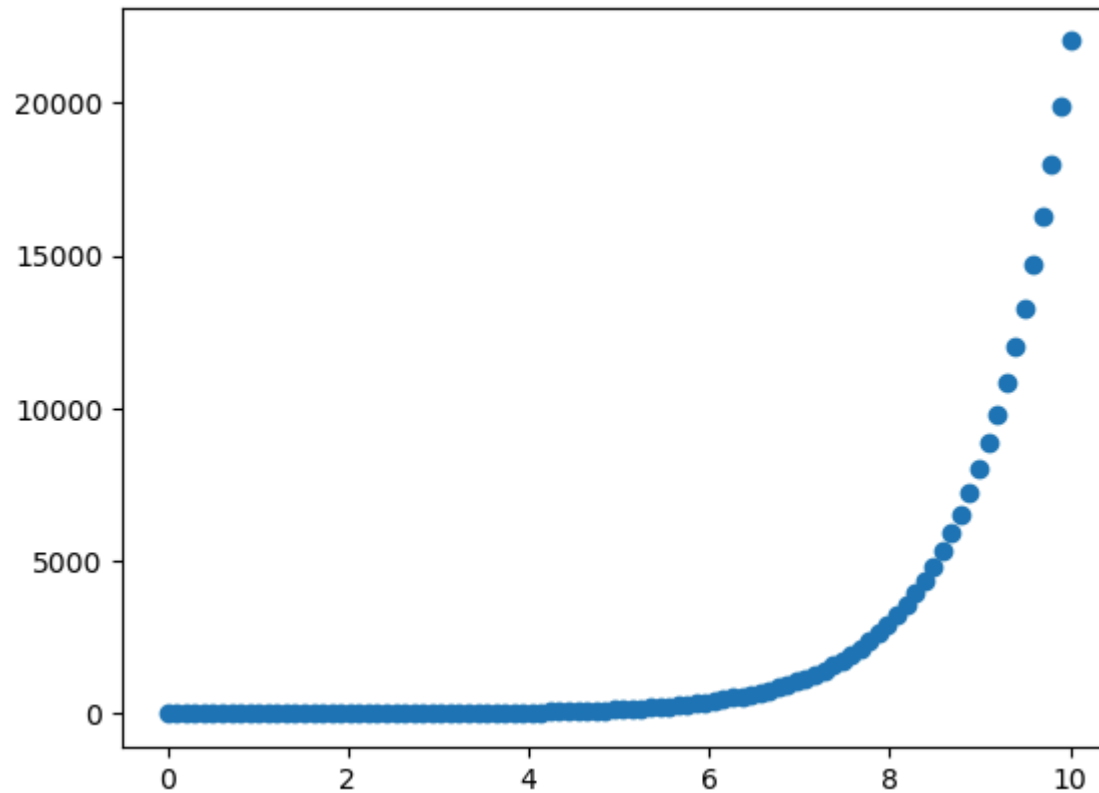
```
plt.subplot(2,2,3)
plt.plot(x,y, c='yellow', ls='-',label='exp')
plt.subplot(2,2,4)
plt.plot(x,y, c='blue', ls='-',label='exp')
plt.subplots_adjust(wspace=0.5)
plt.savefig('figure.png')
```



## 7-quelques graphiques utiles pour le machine learning

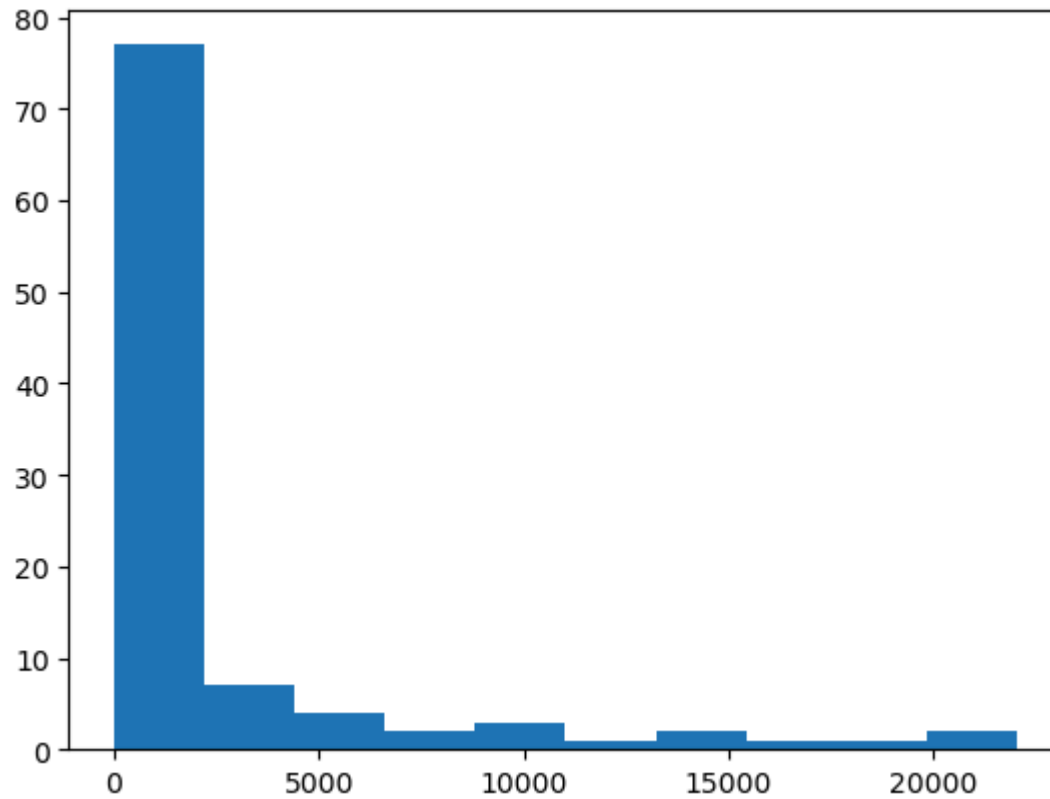
```
In [45]: # nuage de point  
plt.scatter(x,y)
```

```
Out[45]: <matplotlib.collections.PathCollection at 0x154cafde050>
```



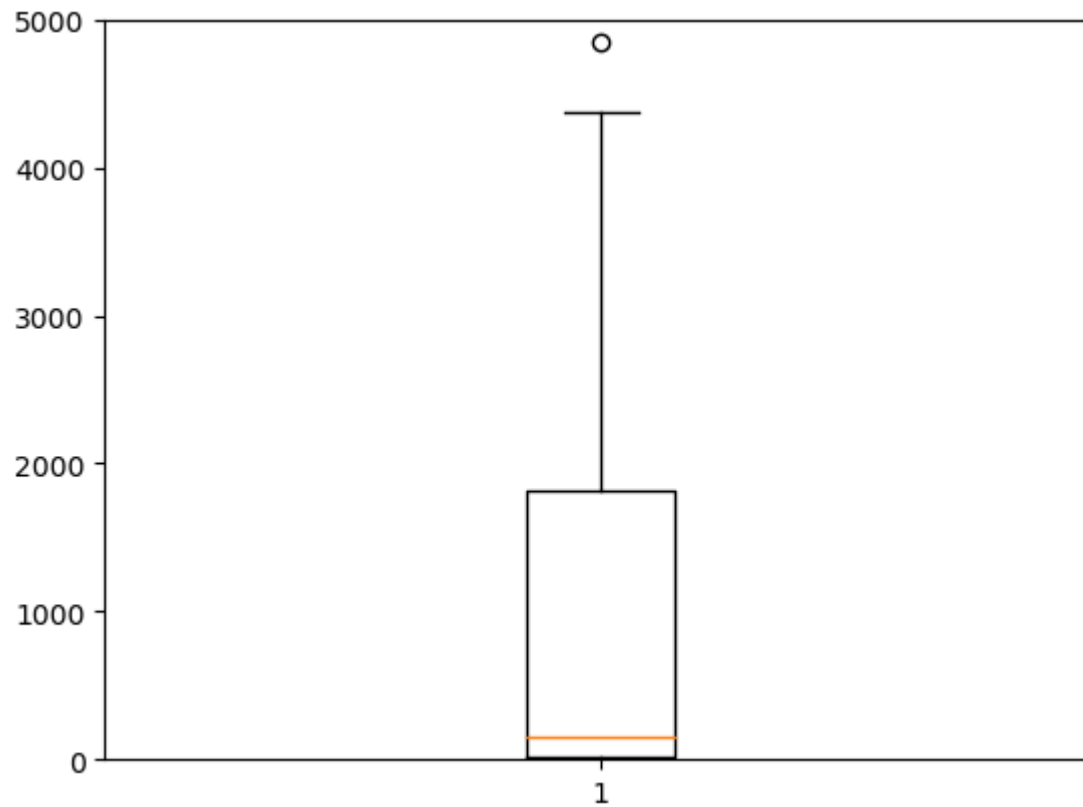
```
In [49]: # histogramme  
plt.hist(y)
```

```
Out[49]: (array([77., 7., 4., 2., 3., 1., 2., 1., 1., 2.]),  
array([1.00000000e+00, 2.20354658e+03, 4.40609316e+03, 6.60863974e+03,  
8.81118632e+03, 1.10137329e+04, 1.32162795e+04, 1.54188261e+04,  
1.76213726e+04, 1.98239192e+04, 2.20264658e+04])),  
<BarContainer object of 10 artists>)
```



```
In [55]: # boxplot  
plt.boxplot(y)  
plt.ylim((0,5000))
```

```
Out[55]: (0.0, 5000.0)
```



In [ ]: