

## La loi d'Ohm (version linregress et avec fonctions)

I (mA)	0	25	50	75	100	125
U (V)	0	1,8	3,3	5,2	6,8	8,5

tableau.png

In [1]:

```
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
# création de la fonction modelisation
# modélisation par une droite d'équation
# y=ax+b (polynôme de degré 1)

def modelisation(x,y):
    slope, intercept, r_value, p_value, std_error = stats.linregress(x,y)
    ymodel = slope*x+intercept
    print ('U= {0:.1f}'.format(slope),'x I')
    print ('Le coefficient de corrélation r vaut {0:.4f}'.format(r_value))
    print('Les valeurs de la tension modélisée sont',ymodel)
    return (ymodel)
```

In [3]:

```
# fonction permettant de tracer le graphique avec les points
# expérimentaux et la courbe obtenue après modélisation

def courbemodelisee (x,y,ymodel) :
    fig = plt.figure(figsize=(12,10))
    plt.plot(x,y,'r+',label='U=f(I)')
    plt.plot(x,ymodel,'b',label='modèle linéaire')
    plt.legend()
    plt.xlabel("intensité I (A)")
    plt.ylabel("tension U (V)")
    plt.grid()
    plt.title("Caractéristique Intensité-Tension dun "
              "dipôle ohmique")
    plt.show()
```

In [4]:

```
# tableaux numpy obligatoires à cause de l'opération vectorisée
# permettant de créer Umodel

I=np.array([0,25e-3,50e-3,75e-3,100e-3,125e-3])
U=np.array([0,1.8,3.3,5.2,6.8,8.5])
```

```
Umodel=modelisation(I,U)
courbemodelisee(I,U,Umodel)
```

$U = 67.9 \times I$

Le coefficient de corrélation  $r$  vaut 0.9997

Les valeurs de la tension modélisée sont [0.02380952 1.72095238 3.41809524 5.1152381  
6.81238095 8.50952381]

