

```

# -*- coding: utf-8 -*-
"""
Created on Sun Mar 27 10:40:41 2016

@author: Dom
"""

import numpy as np
import matplotlib.pyplot as plt
#import matplotlib.animation as animation
from matplotlib.widgets import Slider,Button

Q0=2    #valeur par défaut du facteur de qualité
Hmax=3  #valeur par défaut du gain maximal

#x=np.logspace(0.01,100.0,0.001)
x=np.linspace(0.01,100,3000)

def H(x,Q,H0):
    return H0/(1+1j*Q*(x-1/x))

def GdB(x,Q,H0):
    return 20.*np.log10(abs(H(x,Q,H0)))

# autre solution, avec la fonction angle de numpy
def phase(x,Q,H0):
    return np.angle(H(x,Q,H0),deg=True)

#fig1=plt.figure(figsize=(10,7))
fig,ax=plt.subplots()
#plt.subplots()
plt.subplots_adjust(bottom=0.25,wspace=0.2) #left=0.25,

#plt.subplots_adjust()

texte_Q = u'Q = %.0f'    # affichage de Q
affiche_Q = ax.text (0.05, 0.7, '', transform = ax.transAxes)

#plt.subplot(211)
plt.subplot(121)
# Bode du gain par défaut
l,=plt.semilogx(x,GdB(x,Q0,Hmax),lw=2,color='red')
plt.ylim(-50,15)
plt.grid()
affiche_Q.set_text(texte_Q%(Q0))

#plt.subplot(212)
plt.subplot(122)
# Bode du déphasage par défaut
lphi,=plt.semilogx(x,phase(x,Q0,Hmax),lw=2,color='red')
#plt.ylim(-95,95)
plt.grid()
affiche_Q.set_text(texte_Q%(Q0))

#boîtes contenant les contrôles dynamiques
axcolor='yellow'
axQ=plt.axes([0.15,0.1,0.6,0.03],axisbg=axcolor)

```

```

axH0=plt.axes([0.15,0.15,0.6,0.03],axisbg=axcolor)

sQ=Slider(axQ,'$Q$',0.1,10, valinit=Q0)
sH0=Slider(axH0,'$H_0$',0.2,5, valinit=Hmax) #gain maximal

#action lorsque Le contrôle dynamique est modifié
def update(val):
    Q=sQ.val
    Hm=sH0.val
    l.set_ydata(GdB(x,Q,Hm))
    lphi.set_ydata(phase(x,Q,Hm))
    fig.canvas.draw_idle()
    affiche_Q.set_text (texte_Q%(Q))
sQ.on_changed(update)
sH0.on_changed(update)

#remise à la valeur par défaut
resetax=plt.axes([0.7,0.015,0.1,0.04])
button=Button(resetax,'Reset',color='grey',hovercolor='0.675')
def reset(event):
    sQ.reset()
    sH0.reset()
button.on_clicked(reset)

#titre général
plt.suptitle("""diagramme de Bode""")
plt.show()

```