

# Manual de uso

## - Funcionamento do código:

Nessa parte do código é que definimos as massas, e comprimento das cordas que as ligam, e a gravidade. Na parte do “t” (tempo) está definido no primeiro parâmetro quando a simulação começa, no segundo quando ela termina, e no terceiro quantas atualizações serão feitas nesse intervalo. O usuário pode mudar a vontade esses parâmetros de acordo com o que deseja e como deseja simular o movimento

```
[ ] t = np.linspace(0, 60, 1000) ### espaço de tempo (de 0 a 60 segundos, 1000 atualizações)
    g = 9.81
    m1= 1000
    m2=1001
    L1 = 1
    L2 = 1
```

Aqui, poderão ser selecionadas pelo usuário os parâmetros de y0, que são as condições iniciais das equações diferenciais. Os parâmetros são, da esquerda pra direita:  $\theta_1, \frac{d\theta_1}{dt}, \theta_2, \frac{d\theta_2}{dt}$

```
▶ matriz = odeint(iniciais, y0=[8, 0, 5, 0], t=t, args=(g,m1,m2,L1,L2))
matriz
```

Selecionadas as condições iniciais, o código está pronto para funcionar.

## - Uso do código e gif:

Depois de selecionadas as condições iniciais, para rodar o código é só rodá-lo por completo de cima para baixo. Para o gif daremos uma breve instrução.

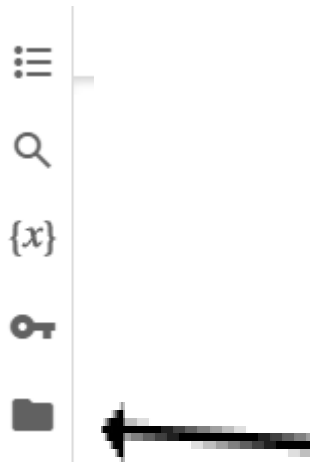
Para selecionar o nome com que o gif será salvo, deverá alterar o nome em vermelho indicado pela seta, nesse caso selecionamos o nome “pendulo.gif” , o nome deve acabar com “.gif”

```
def gif(i):  
    ln1.set_data([0, x1[i], x2[i]], [0, y1[i], y2[i]])  
  
fig, ax = plt.subplots(1,1, figsize=(8,8))  
ln1, = plt.plot([], [], 'ro--', lw=3, markersize=8)  
ax.set_ylim(-5,5)  
ax.set_xlim(-5,5)  
ani = animation.FuncAnimation(fig, gif, frames=1000, interval=50)  
ani.save('pendulo.gif',writer='pillow',fps=25)
```

T

Após selecionado o nome, deve rodar essa parte do código, isso pode demorar um pouco. Após concluído verá que apareceu apenas uma imagem em sua tela, para ter acesso ao gif, deve:

(1) Clicar em arquivos



(2) Após isso só baixar o arquivo com o nome que você escolheu para o gif