

Ficha de Avaliação Final	
Curso:	UFCD 10793
UFCD/Módulo/Temática:	UFCD 10793 - Fundamentos de Python
Ação:	10793_02/AT & 10793_05/N
Formador/a:	Sandra Liliana Meira de Oliveira
Data:	março de 2025
Nome do Formando/a:	

Resolve os seguintes exercícios recorrendo ao Jupyter Notebook.

Jupyter Widgets são São controlos gráficos que permitem uma interatividade com código, gráficos e operações, sem que o utilizador precise usar código explicitamente.

Em particular: "**sliders**", botões, caixas de entrada de texto, e alguns selecionadores de listas e calendários.

É possível também criar widgets customizados.

Podes encontrar a lista de widgets no seguinte endereço:

[Widget List – Jupyter Widgets 8.0.7 documentation \(ipywidgets.readthedocs.io\)](https://ipywidgets.readthedocs.io/en/8.0.7/widget/List.html)

E exemplos de interação neste:

[Using Interact – Jupyter Widgets 8.0.7 documentation \(ipywidgets.readthedocs.io\)](https://ipywidgets.readthedocs.io/en/8.0.7/interact.html)

1. Explora a biblioteca escrevendo e executando o seguinte código.

```
import ipywidgets as widgets
widgets.IntSlider()

from IPython.display import display
w = widgets.IntSlider()
display(w)

w.value = 100

widgets.Text(value='Hello World!', disabled=True)

slider = widgets.FloatSlider()
```

```
value=7.5,
min=5.0,
max=10.0,
step=0.1,
description='Input:',
)

# Create text box to hold slider value
text = widgets.FloatText(description='Value')

# Link slider value and text box value
widgets.link((slider, 'value'), (text, 'value'))

# Put them in a vertical box
widgets.VBox([slider, text])

slider = widgets.FloatSlider(
    value=7.5,
    min=5.0,
    max=10.0,
    step=0.1,
    description='Input:',
)

# Create non-editable text area to display square of value
square_display = widgets.HTML(description="Square: ",
value='{}'.format(slider.value**2))

# Create function to update square_display's value when slider changes
def update_square_display(change):
    square_display.value = '{}'.format(change.new**2)

slider.observe(update_square_display, names='value')

# Put them in a vertical box
widgets.VBox([slider, square_display])
```

2. É possível utilizar os widgets para interagir com o nosso código usando a função ***interact***.
Reproduz o seguinte código para observares a mesma em utilização

A função `interact` (`ipywidgets.interact`) cria automaticamente um interface de utilizador (UI).

No nível mais básico, 'interact' gera automaticamente controlos UI para parâmetros da função e, em seguida, chama a função com esses argumentos.

```
def f(x):  
    return 3 * x  
interact(f, x=10);  
interact(f, x=True);  
interact(f, x='Hi there!');  
  
@widgets.interact(x=True, y=1.0)  
def g(x, y):  
    return (x, y)  
  
import matplotlib.pyplot as plt  
import numpy as np  
  
def f(m, b):  
    fig = plt.figure()  
    plt.clf()  
    plt.grid()  
    x = np.linspace(-10, 10, num=1000)  
    plt.plot(x, m * x + b)  
    plt.ylim(-5, 5)  
    plt.show()  
  
interact(f, m=(-2.0, 2.0), b=(-3, 3, 0.5))
```

3. Reproduz o seguinte num jupyter notebook. Coloca cada exemplo numa secção distinta.

```
import ipywidgets as widgets  
from ipywidgets import HBox, VBox  
import numpy as np  
import matplotlib.pyplot as plt  
from IPython.display import display  
%matplotlib inline  
  
#exemplo1  
@widgets.interact  
def f(x=5):  
    print(x)  
  
#exemplo2  
  
#exemplo3
```

```
@widgets.interact(x=(0, 5))
def f(x=5):
    print(x)

#exemplo4

@widgets.interact_manual(
    color=['blue', 'red', 'green'], lw=(1., 10.))
def plot(freq=1., color='blue', lw=2, grid=True):
    t = np.linspace(-1., +1., 1000)
    fig, ax = plt.subplots(1, 1, figsize=(8, 6))
    ax.plot(t, np.sin(2 * np.pi * freq * t),
            lw=lw, color=color)
    ax.grid(grid)

#exemplo5

freq_slider = widgets.FloatSlider(
    value=2.,
    min=1.,
    max=10.0,
    step=0.1,
    description='Frequency:',
    readout_format='.1f',
)
freq_slider

#exemplo6

range_slider = widgets.FloatRangeSlider(
    value=[-1., +1.],
    min=-5., max=+5., step=0.1,
    description='xlim:',
    readout_format='.1f',
)
range_slider

#exemplo7

grid_button = widgets.ToggleButton(
    value=False,
    description='Grid',
    icon='check'
```

```
)  
grid_button  
  
#exemplo8  
  
color_buttons = widgets.ToggleButtons(  
    options=['blue', 'red', 'green'],  
    description='Color:',  
)  
color_buttons  
  
#exemplo9  
  
title_textbox = widgets.Text(  
    value='Hello World',  
    description='Title:',  
)  
title_textbox  
  
#exemplo10  
  
color_picker = widgets.ColorPicker(  
    concise=True,  
    description='Background color:',  
    value='#efefef',  
)  
color_picker  
  
#exemplo11  
  
button = widgets.Button(  
    description='Plot',  
)  
button  
  
#exemplo12  
def plot2(b=None):  
    xlim = range_slider.value  
    freq = freq_slider.value  
    grid = grid_button.value  
    color = color_buttons.value  
    title = title_textbox.value  
    bgcolor = color_picker.value  
  
    t = np.linspace(xlim[0], xlim[1], 1000)  
    f, ax = plt.subplots(1, 1, figsize=(8, 6))
```

```
ax.plot(t, np.sin(2 * np.pi * freq * t),
        color=color)
ax.grid(grid)

@button.on_click
def plot_on_click(b):
    out.clear_output(wait=True)
    with out:
        plot2()
        plt.show()

tab1 = VBox(children=[freq_slider,
                     range_slider,
                     ])
tab2 = VBox(children=[color_buttons,
                     HBox(children=[title_textbox,
                                     color_picker,
                                     grid_button]),
                     ])

out = widgets.Output()
tab = widgets.Tab(children=[tab1, tab2])
tab.set_title(0, 'plot')
tab.set_title(1, 'styling')
VBox(children=[tab, button, out])

#exemplo12
```

4. Cria uma função em Python, que dado o valor do raio, introduzido pelo utilizador com recurso a widgets, devolva o Volume de uma esfera ($V = \frac{4}{3} \pi r^3$).

alínea	Cotação (escala 1-20)
1	4
2	4
3	6
4	6