

An illustration of a city at night, viewed from a rooftop. In the foreground, a man is sitting at a desk, working on a laptop. To his right, a cat is visible. In the background, a city skyline is visible with various buildings and a water tower. The scene is lit with a warm, orange glow, suggesting sunset or sunrise. The overall style is a flat, modern illustration.

1ª Rodada de Workshop dos Quarenteners

PYTHON (DIA 2)

Sumário

1. Jupyter-Notebook
2. Exemplo da Aula Anterior
3. Numpy
4. Matplotlib
5. Aplicações
 1. Processamento básico de Imagens
 2. Análise de sinais do PSIM e modelagem de sistemas lineares

Jupyter-Notebook

Elementos do *dashboard*:

1. Abas como o estado de cada um dos Notebooks;
2. Botões para a criação de novos notebooks.



Jupyter-Notebook

Elementos do *dashboard*:

1. Abas como o estado de cada um dos Notebooks;
2. Botões para a criação de novos notebooks.

The screenshot displays the Jupyter dashboard interface. At the top, there are tabs for 'Files', 'Running', and 'Clusters'. Below these, a section titled 'Select items to perform actions on them.' shows a file browser view with a list of files: 'install_jupyter_and_open_it.cmd.cmd' and 'Jupyter_Notebooks_quickstart.docx'. To the right of this list, there are buttons for 'Upload', 'New', and a refresh icon. A red box highlights this area. Below the file list, there are buttons for 'Quit' and 'Logout'. A red arrow points from the 'New' button in the lower section to the 'New' button in the upper section. The lower section also shows a table with columns 'Name', 'Last Modified', and 'File size', containing two rows of data: 'poucos segundos atrás' with '311 B' and '346 kB'.

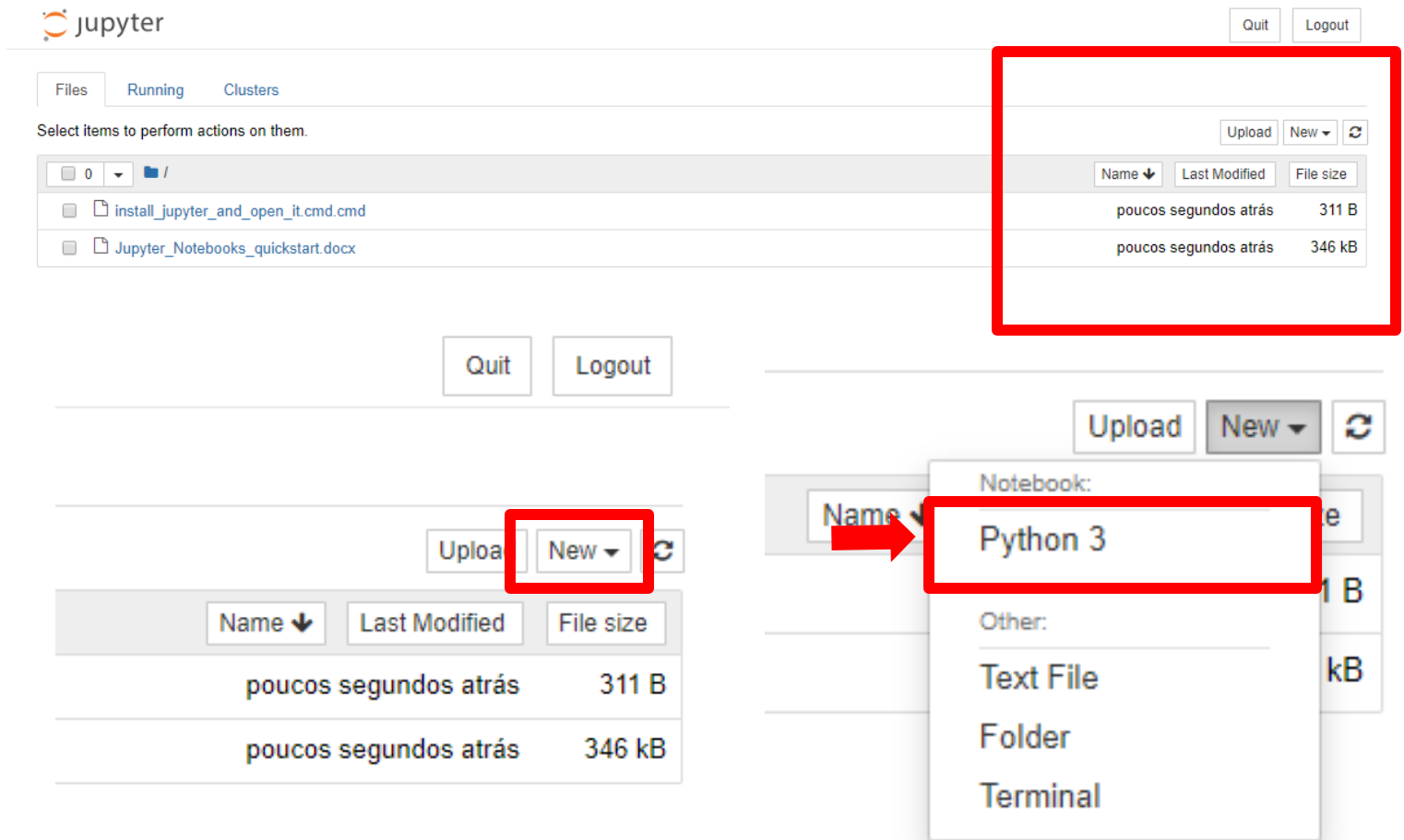
Name	Last Modified	File size
poucos segundos atrás		311 B
poucos segundos atrás		346 kB

Jupyter-Notebook

Elementos do *dashboard*:

1. Abas como o estado de cada um dos Notebooks;
2. Botões para a criação de novos notebooks.

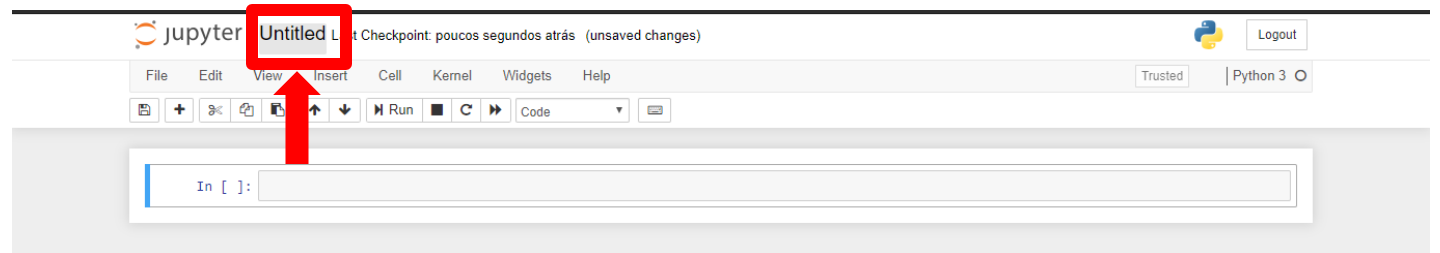
1



Jupyter-Notebook

Elementos do *Jupyter-Notebook*:

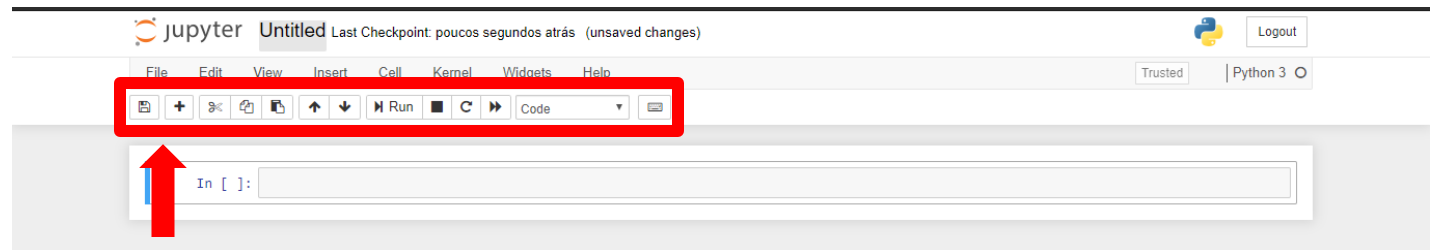
1. Nome do script



Jupyter-Notebook

Elementos do *Jupyter-Notebook*:

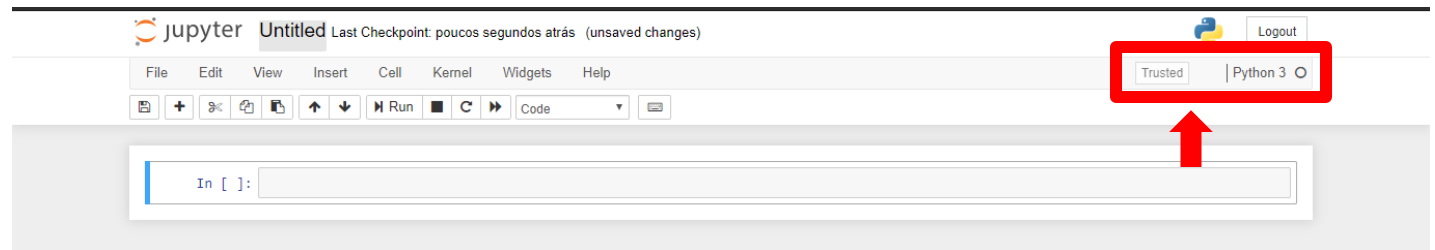
1. Nome do script
2. Botões de execução



Jupyter-Notebook

Elementos do *Jupyter-Notebook*:

1. Nome do script
2. Botões de execução
3. Flags



Jupyter-Notebook

Elementos do *Jupyter-Notebook*:

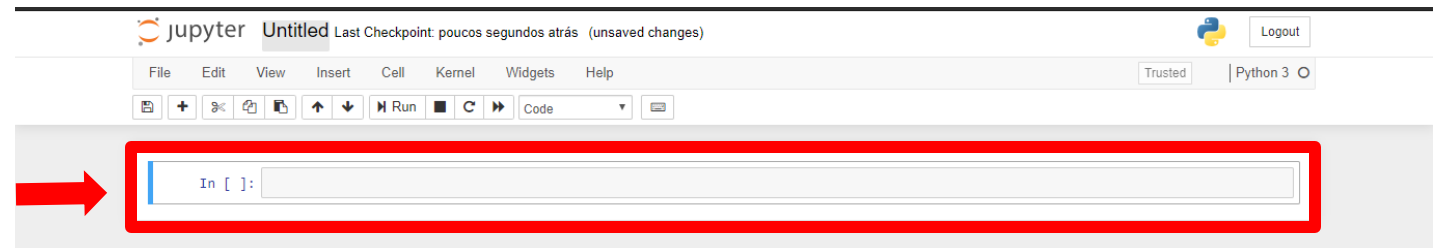
1. Nome do script

2. Botões de execução

3. Flags

4. Célula

1. Ctrl+Shift+- : Divide uma célula em duas
2. Shift+Enter: Executa e cria uma nova célula
3. Ctrl+Enter: Somente executa a célula
4. Esc+m: Muda a célula para configuração Markdown
5. Esc+y: Muda a célula para configuração Python



Exemplo da Aula Anterior

Exercício 4: Entrada de notas

Fazer um programa que entre com notas de um aluno de forma continua até quando for inserida uma nota -1 . Nisto o programa deve parar de ler as notas e retornar a média do aluno.



Numpy

- Array
- Matrizes

- Soma de arrays
- Multiplicação de arrays

Numpy

Exercício 5: Comparar as formas de calcular uma equação

Resolver um equação utilizando *numpy.array* e lista e comparar o tempo necessário para cada um dos casos.

Numpy

Exercício 6: Comparar as formas de calcular uma equação

~~Resolver um equação utilizando `numpy.array` e lista e comparar o tempo necessário para cada um dos casos.~~

Obter o erro numérico entre o array e a lista.

Matplotlib

- Plot function
- Imshow e Imread

Matplotlib

Exercício 7: Comparar as formas de calcular uma equação usando plot

Pegar a lista e o array criado no exercício anterior e comparar num plot para verificar o resultado numérico.

Aplicações: Processamento de Imagens

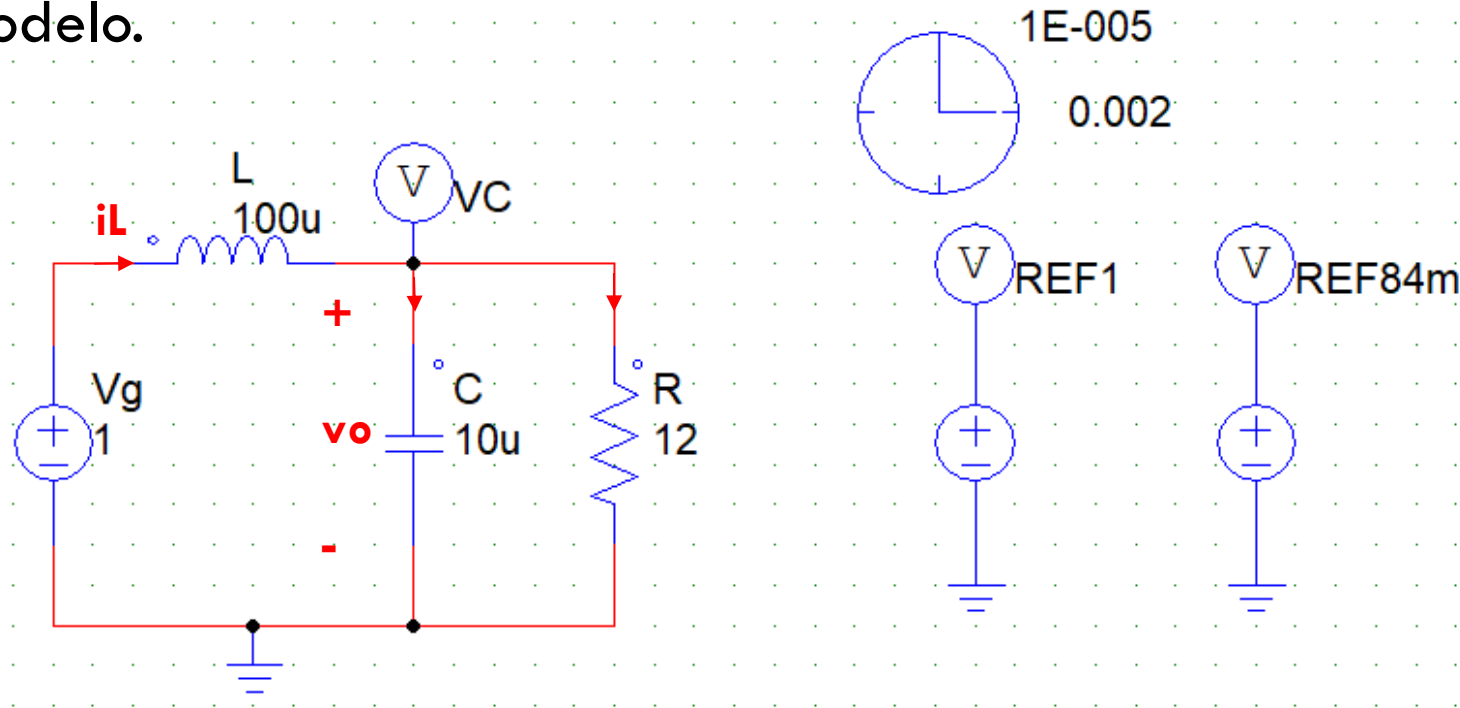
Exercício 8: Transformar uma imagem de colorida para preto e branco

Baixar a imagem da *lenna.png* no [Link](#) e processá-la para transformar em preto e branco.

Aplicações: Modelagem de Sistemas Lineares

Exercício 9: Montar no PSIM e comparar com o modelo

Montar um circuito no PSIM, simulá-lo e importar as ondas para o Python e comparar com um modelo.



Aplicações: Modelagem de Sistemas Lineares

Exercício 9: Montar no PSIM e comparar com o modelo

Montar um circuito no PSIM, simulá-lo e importar as ondas para o Python e comparar com um modelo.

$$\begin{bmatrix} i'_L \\ v'_O \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ v_O \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_g$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_L \\ v_O \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} V_g$$

R	12 Ω
L	100μH = 100e-6H
C	10μF = 10e-6F