

# jupyter

April 8, 2020

Sumário

- 1 Introdução
- 2 Get Jupyter Notebook filename
- 3 Export
- 4 GitHub
- 5 Outros
- 6 Requirements
- 7 Erros
- 8 Referências

## 1 Introdução

O *Jupyter Notebook* é a maneira que optei para escrever os códigos na linguagem *Python*, visto que além de rodar os códigos, é possível: 1. Documentar os *scripts*, escrevendo o significado e objetivo de cada conjunto de comandos; 2. Atualizar os meus repositórios na plataforma **GitHub**; 3. Trabalhar com uma diversidade de opções de exportação do arquivo em formatos diversos, adaptados até mesmo para as simples leitura, como PDFs e Markdowns.

É no processo de exportação dos arquivos que eu me ative nessa publicação, pois um dos objetivos de longo prazo que busco é exportar relatórios padronizados, para distribuição geral e irrestrita, ou seja, quero algo que não seja inteligível apenas por pessoas que conhecem de programação.

Para isso foram aqui apresentados uma diversidade de opções para exportação de um arquivo *.ipynb*, sendo possível incluir com campos determinados, tais como: - Apenas as células que tenham determinada *tag*; - Apenas as células de *markdown*; - Excluindo as células de *outputs*.

```
[1]: %run '../codes/files/create_folders.py'

create_folders('', folders=['docs'])
```

Directory "docs" already exists...

## 2 Get *Jupyter Notebook* filename

<https://stackoverflow.com/questions/12544056/how-do-i-get-the-current-ipython-jupyter-notebook-name>

```
[2]: %%javascript

var kernel = IPython.notebook.kernel;
var body = document.body, attribs = body.attributes;
var command = 'ipynb_filename = ' + '"' + attribs['data-notebook-name'].value + '"';
kernel.execute(command);
```

<IPython.core.display.Javascript object>

```
[3]: ipynb_filename
```

```
[3]: 'jupyter.ipynb'
```

```
[4]: #!run '../codes/files/get_jupyternotebook_name.py'
```

```
[5]: #get_jupyternotebook_name()
```

```
[6]: %%javascript

var kernel = IPython.notebook.kernel;
var nb = IPython.notebook;
var command = 'ipynb_pathname = ' + nb.base_url + nb.notebook_path + '"';
kernel.execute(command);
```

<IPython.core.display.Javascript object>

```
[7]: ipynb_pathname
```

```
[7]: '/package_jupyter/jupyter.ipynb'
```

## 3 Export

```
[8]: %run '../codes/files/export_jupyter.py'

export_jupyter(ipynb_filename, 'docs', ['html', 'markdown', 'pdf', 'python'],
↪False)
```

Arquivo jupyter.ipynb exportado corretamente para o formato html sem usar prefixo da data.

Arquivo jupyter.ipynb exportado corretamente para o formato markdown sem usar prefixo da data.

Arquivo jupyter.ipynb exportado corretamente para o formato pdf sem usar prefixo da data.

Arquivo jupyter.ipynb exportado corretamente para o formato python sem usar prefixo da data.

```
[50]: #import os
      #os.system('jupyter-nbconvert --to python jupyter.ipynb')
```

```
[12]: #--stdout
      #--TemplateExporter.exclude_input_prompt=True'
```

```
[13]: !jupyter-nbconvert jupyter.ipynb\
      --to markdown\
      --TagRemovePreprocessor.enabled=True\
      --ClearOutputPreprocessor.enabled=True\
      --TemplateExporter.exclude_markdown=False\
      --TemplateExporter.exclude_code_cell=True\
      --TemplateExporter.exclude_output=True\
      --TemplateExporter.exclude_raw=False\
      --TemplateExporter.exclude_input_prompt=True\
      --TagRemovePreprocessor.remove_cell_tags='["remove_cell"]'\
      --output Teste
```

[NbConvertApp] Converting notebook jupyter.ipynb to markdown

[NbConvertApp] Writing 1710 bytes to Teste.md

Quero saber como exportar somente algumas células. <https://stackoverflow.com/questions/57701538/jupyter-notebook-nbconvert-without-magic-commands-w-o-markdown>

## 4 GitHub

```
[48]: import os
      os.system('nbstripout --install --attributes .gitattributes')
```

```
[48]: 0
```

```
[46]: !nbstripout --install --attributes .gitattributes
```

```
[49]: %run '../codes/git/update_github.py'
```

```
[107]: git_full('/home/michel/Geodata/SourceCode/package_jupyter', '.', 'Atualizando')
```

```
b'' b''
b'[master cbc3f02] Atualizando\n 7 files changed, 389 insertions(+), 319
deletions(-)\n create mode 100644 Teste.ipynb\n create mode 100644 Teste.pdf\n
rewrite docs/jupyter.pdf (94%)\n' b''
b'' b'To github.com:michelmetran/package_jupyter.git\n    cb45215..cbc3f02
```

```
master -> master\n'  
Done!!
```

## 5 Outros

A partir do post [How to Git Jupyter Notebooks the Right Way](#), compreendi que é a partir da instalação do package *nbstripout* que é possível fazer o git de arquivos *ipynb*.

## 6 Requirements

O comando `pip freeze` é o mais difundido na internet para se obter os *requirements.txt*, ou seja, o arquivo com o qual é possível indicar quais os *packages* necessários para rodar um determinado *script*.

```
[42]: pip freeze > requirements.txt
```

Note: you may need to restart the kernel to use updated packages.

Tentei usar também o `pipreqs`, porém ele não funciona em Jupyter Notebook.

```
[45]: import pipreqs
```

eee

```
[39]: conda env export > environment.yml
```

Note: you may need to restart the kernel to use updated packages.

```
[26]: import os  
      cwd = os.getcwd()  
      py = os.path.join(cwd, 'docs')  
      py
```

```
[26]: '/media/Geodata/SourceCode/package_jupyter/docs'
```

## 7 Erros

Em uma tentativa de exportar para pdf, tive problemas. Não exportava.

Descobri que é necessário ter o `''sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-generic-recommended''` ou `''sudo apt-get install texlive-full''`

*nbconvert failed: xelatex not found on PATH, if you have not installed xelatex you may need to do so. Find further instructions at <https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex>.*

## 8 Referências

A partir do post *Jupyter Notebook Extensions*.

A partir do post *28 Jupyter Notebook Tips, Tricks, and Shortcuts*,

A partir do post [Link do Folium](#)

[ ]: