

# jupyter

April 8, 2020

## Sumário

- 1 Introdução
- 2 Get Jupyter Notebook filename
- 3 Export
- 4 GitHub
- 5 Requirements
- 6 Erros
- 7 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 apenas campos determinados; - Incluir apenas as células que tenham determinada *tag*; - Incluir apenas as células de *markdown*; - Excluir 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>

```
[51]: %%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>

```
[52]: ipynb_filename
```

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

```
[53]: #%%run '../codes/files/get_jupyternotebook_name.py'
```

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

```
[55]: %%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>

```
[56]: ipynb_pathname
```

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

### 3 Export

```
[57]: %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.



```

1 import nbstripout
----> 2 nbstripout --help
3 get_ipython().system('nbstripout --install --attributes .
gitattributes')

```

TypeError: bad operand type for unary -: '\_Helper'

Criei uma

```
[64]: %run '../codes/git/update_github.py'
git_full('/home/michel/Geodata/SourceCode/package_jupyter', '.', 'Atualizando')
```

```

b'' b''
b'[master ec7ec9a] Atualizando\n 10 files changed, 1128 insertions(+), 347
deletions(-)\n delete mode 100644 Teste.ipynb\n create mode 100644 Teste.md\n
delete mode 100644 Teste.pdf\n rewrite docs/jupyter.pdf (95%)\n create mode
100644 environment.yml\n create mode 100644 requirements.txt\n' b''
b'' b'To github.com:michelmetran/package_jupyter.git\n   cbc3f02..ec7ec9a
master -> master\n'
Done!!

```

## 5 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*.

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

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

Tentei usar também o package `pipreqs`, porém ele não funciona em *Jupyter Notebook*. Descobri ainda que o comando `conda env export > environment.yml` pode auxiliar na criação destes parâmetros.

## 6 Erros

Em uma tentativa de exportar o *Jupyter Notebook* para PDF tive problemas. O arquivo não era exportado e apresentava a seguinte mensagem de erro: - *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>.*

Para solucionar, descobri que é necessário instalar, no Linux, alguns pacotes de aplicativos com os seguintes comandos:

```

sudo apt-get install texlive-xetex texlive-fonts-recommended
texlive-generic-recommended # Versão Compacta

```

```
sudo apt-get install texlive-full # Versão Completa
```

## 7 Referências

### Jupyter Notebook Extensions

A partir do post [28 Jupyter Notebook Tips, Tricks, and Shortcuts](#)

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