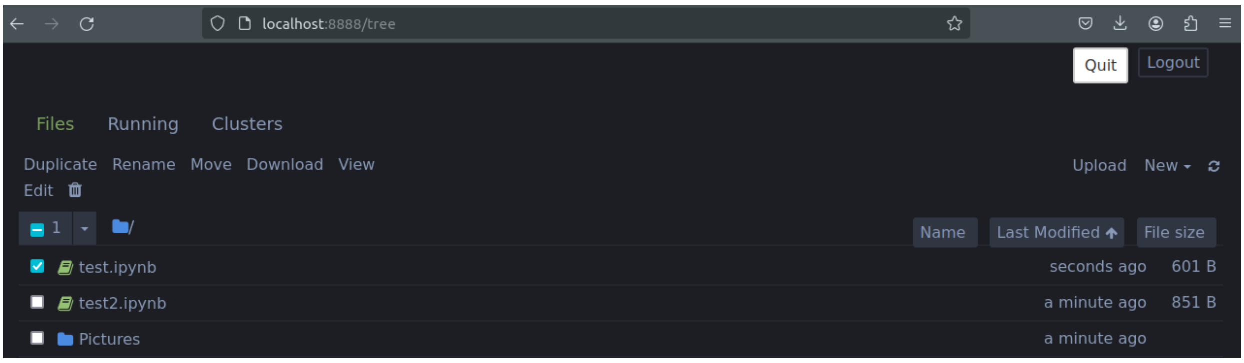
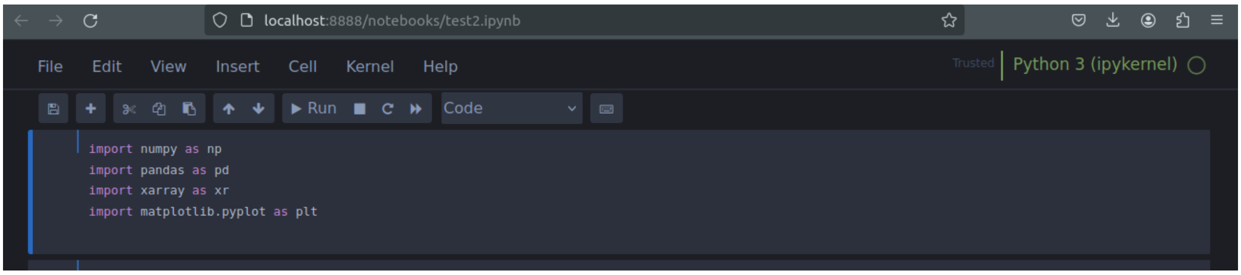
Home › python-classes

Python: A Guide to Customizing Themes in Jupyter

by **Akash** - May 05, 2024



	Name	Last Modified	File size
☑	test.ipynb	seconds ago	601 B
□	test2.ipynb	a minute ago	851 B
□	Pictures	a minute ago	



```
import numpy as np
import pandas as pd
import xarray as xr
import matplotlib.pyplot as plt
```

Hey there, Folks!

It's been a while, hasn't it?

Today, I'm excited to share a neat trick that'll make your Jupyter Lab or notebook experience even better: customizing themes.

Ever felt like your eyes were getting tired from staring at bright screens for too long? Trust me, you're not alone. That's where custom themes come in handy. They're not just easy on the eyes, especially in low-light settings, but they also give your workspace a geeky-cool vibe.

Now, we all have our preferences when it comes to themes. While some swear by the classic dark theme, others might prefer something a bit more colorful. Well, good news! I'm here to show you how to switch up your Jupyter themes to suit your style.

And guess what? It's all thanks to a fantastic third-party extension called "jupyterthemes," brought to you by the awesome Kyle Dunovan. You can find more about it on his GitHub page (https://github.com/dunovank/jupyter-themes).

Oh, and if you've missed out on any of my previous Python blogs, don't worry! You can catch up on all the fun (https://aireenproject.blogspot.com/search/label/python-classes).

Let's dive in and give your Jupyter environment a fresh new look!

Step 1: Install *jupyterthemes*

Before we get started, make sure you have the following:

- ***Python 3****: This theme has been tested on Python 3.8, but it should work with any Python 3 version from Python 3.4 onwards.*
- ***Jupyter****: You'll need either Miniconda or Anaconda installed.*
- ***Matplotlib****: If you haven't already installed Matplotlib, you can do so using the following command:*

> pip install matplotlib

Once you've checked off all the prerequisites mentioned above, you're ready to install *jupyterthemes*. Simply proceed to the terminal and run the following command:

> *pip install jupyterthemes*

If you prefer using Conda over pip for installation, you can do so by executing the following command in your terminal:

> conda install -c conda-forge jupyterthemes

Step 2: Check available themes

To list the available themes, simply use the following command in your terminal:

```
jt -l
```

This will show you the following themes as below;

```
Available Themes:
chesterish
grade3
gruvboxd
gruvboxl
monokai
oceans16
onedork
solarizedd
solarizedl
```

Step 3: Choose a theme

Now it's time to explore the themes and find the one that suits your style. You can easily do this by using the following command in your terminal:

```
jt -t chesterish
```

If you ever want to revert back to the original Jupyter theme, simply type the following command in your terminal:

```
jt -r
```

Step 4: Customizing your selected theme

Once you've set a theme using the "jt -t" command, the customization options don't end there! There's a whole array of options you can explore to tweak your Jupyter environment to your liking. To see all the available customization options, simply use the command:

command:

> jt -t

This will show you the options as below:

```
usage: jt [-h] [-l] [-t THEME] [-f MONOFONT] [-fs MONOSIZE] [-nf NBFONT]
        [-nfs NBFONTSIZE] [-tf TCFONT] [-tfs TCFONTSIZE] [-dfs DFFONTSIZE]
        [-ofs OUTFONTSIZE] [-mathfs MATHFONTSIZE] [-m MARGINS]
        [-cursw CURSORWIDTH] [-cursc CURSORCOLOR] [-cellw CELLWIDTH]
        [-lineh LINEHEIGHT] [-altp] [-altmd] [-altout] [-P] [-T] [-N] [-kl]
        [-vim] [-r] [-dfonts]
```

The description of these are below;

cl options	**arg**	
Usage help	-h	
List Themes	-l	
Theme Name to Install	-t	
Code Font	-f	
Code Font-Size	-fs	
Notebook Font	-nf	
Notebook Font Size	-nfs	
Text/MD Cell Font	-tf	
Text/MD Cell Fontsize	-tfs	
Pandas DF Fontsize	-dfs	
Output Area Fontsize	-ofs	
Mathjax Fontsize (%)	-mathfs	
Intro Page Margins	-m	
Cell Width	-cellw	
Line Height	-lineh	
Cursor Width	-cursw	
Cursor Color	-cursc	

Alt Prompt Layout	-altp	
Alt Markdown BG Color	-altmd	
Alt Output BG Color	-altout	
Style Vim NBExt*	-vim	
Toolbar Visible	-T	
Name & Logo Visible	-N	
Kernel Logo Visible	-kl	
Reset Default Theme	-r	
Force Default Fonts	-dfonts	

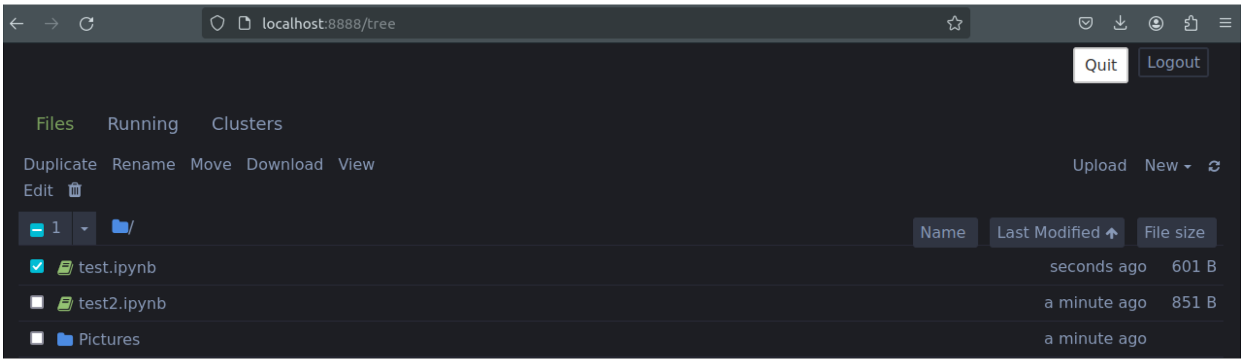
For example, if you're looking to adjust the font size of your selected theme, you can easily do so using the following command in your terminal:

```
jt -t onedork -fs 14
```

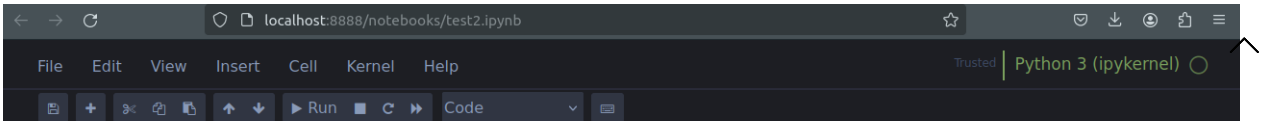
My favourite dark mode setting is below

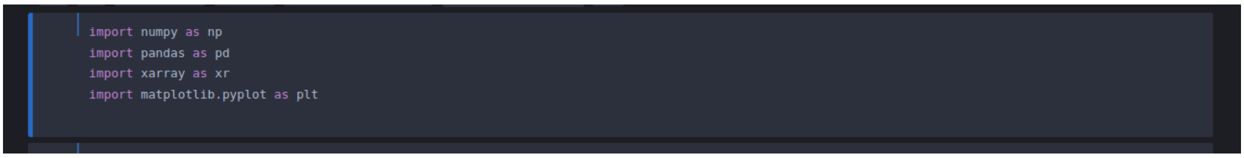
```
jt -t onedork -fs 96 -altp -tfs 12 -nfs 120 -cursc o -cursw 5 -cellw 96% -T -altmd  -kl -T -N
```

It will look like this;



☑ test.ipynb seconds ago 601 B
□ test2.ipynb a minute ago 851 B
□ Pictures a minute ago



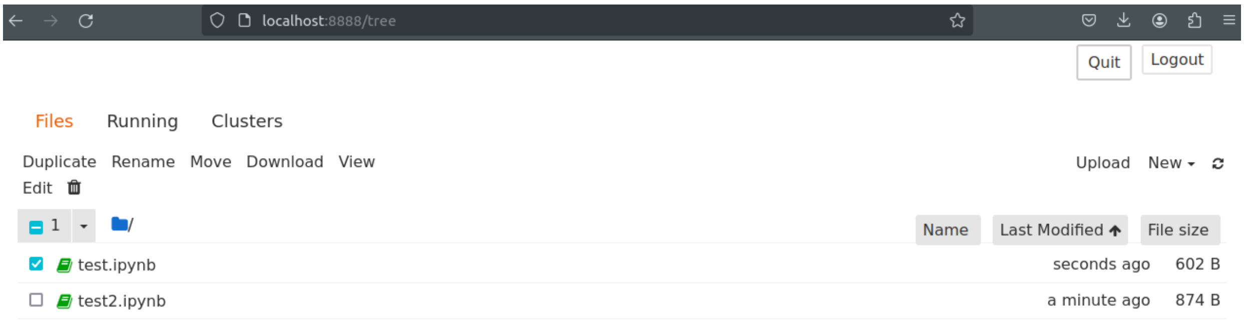


```
import numpy as np
import pandas as pd
import xarray as xr
import matplotlib.pyplot as plt
```

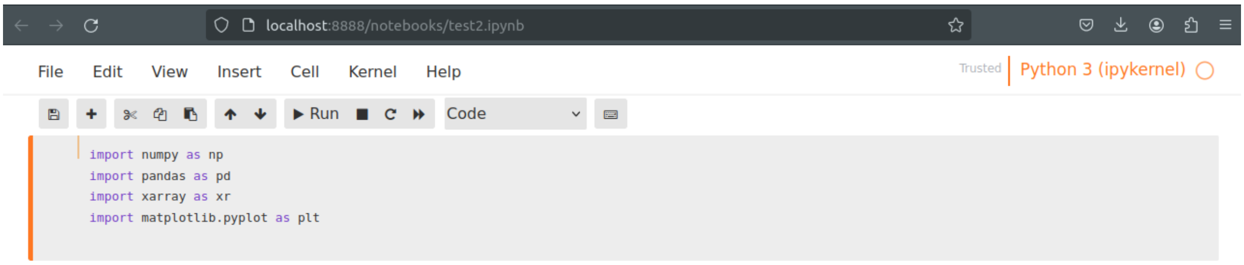
My favourite light mode setting is below

> jt -t grade3 -fs 96 -altp -tfs 12 -nfs 120 -cursc o -cursw 5 -cellw 96% -T -altmd -kl -T -N

It will look like this;



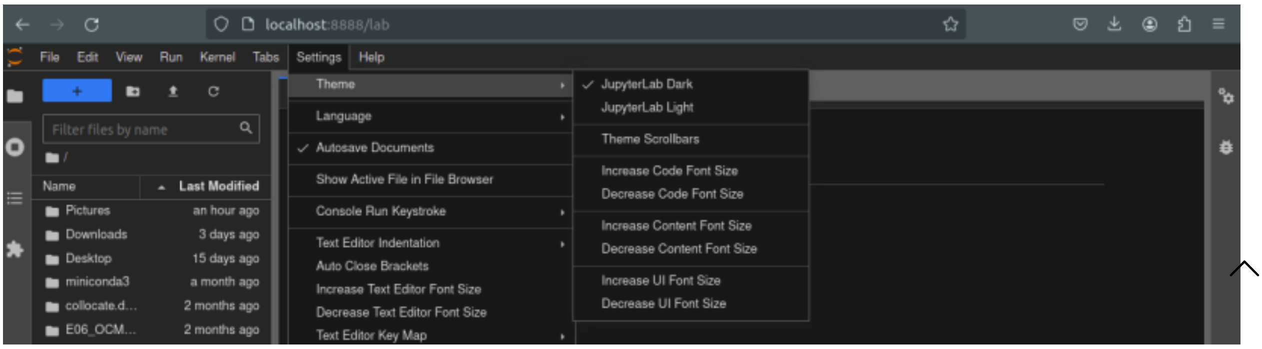
	Name	Last Modified	File size
☑	test.ipynb	seconds ago	602 B
□	test2.ipynb	a minute ago	874 B



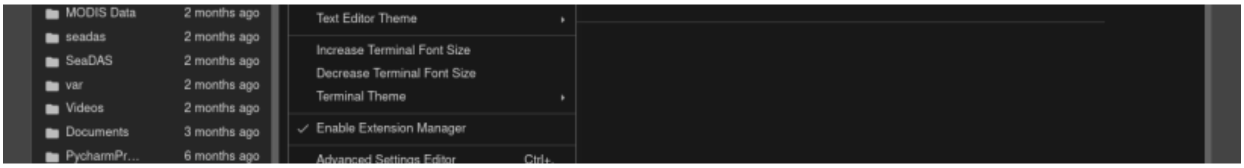
```
import numpy as np
import pandas as pd
import xarray as xr
import matplotlib.pyplot as plt
```

In Jupyter Lab

The above method is for Jupyter Notebook. In Jupyter Lab, you have the direct option to change themes as follows:



Name	Last Modified
Pictures	an hour ago
Downloads	3 days ago
Desktop	15 days ago
miniconda3	a month ago
collocate.d...	2 months ago
E06_OCM...	2 months ago



MODIS Data	2 months ago
seadas	2 months ago
SeaDAS	2 months ago
var	2 months ago
Videos	2 months ago
Documents	3 months ago
PycharmPr...	6 months ago

Text Editor Theme
Increase Terminal Font Size
Decrease Terminal Font Size
Terminal Theme
✓ Enable Extension Manager
Advanced Settings Editor Ctrl+

Every detail matters when it comes to enhancing your productivity and reducing eye strain. Even a simple customization like adjusting the font size can make a significant difference. By fine-tuning your Jupyter environment to suit your preferences, you're not just improving aesthetics but also optimizing your workflow.

Imagine being able to read code and documentation comfortably without squinting or straining your eyes. It's these little adjustments that contribute to a more pleasant and efficient coding experience. Plus, when you're working in a setup that feels right for you, it can help clear your mind and boost your focus.

So, don't overlook the power of customization. Take the time to explore the various options available and tailor your Jupyter environment to your liking. Your eyes and your productivity will thank you for it!

I'm thrilled to share this latest blog post with you, and I owe it all to the encouragement and inspiration from dear friends Ullas and Anand. After receiving their insightful feedback on my last blog, I couldn't resist diving back in to explore a topic that piqued their interest.

Feel free to reach out to the author if you have any questions or need any scientific assistance: Akash (akashspunnayil@gmail.com)

Happy coding!

Cheers!

Subscribe, share and comment.

Stay subscribed for more classes.

Tags: python-classes

Facebook Twitter

You might like

View all →

Post a Comment (0)

‹ Previous Post Next Post ›

Follow Us

Popular Articles

Kick start your R journey Getting started with R

Midhunshah - August 13, 2023

One Ocean Expedition - A Voyage of...

August 06, 2023

Why you became an amatuer...

January 29, 2017

The Fascinating World of Jellyfish, A joy ...

July 21, 2023

Stay Informed

Subscribe to our mailing list to get the new updates.

Email Address

Subscribe

Main Tags

chronicles python-classes

⚠ Report Abuse

Search This Blog

Search

May 2024	(1)
December 2023	(1)
November 2023	(3)
October 2023	(3)
August 2023	(2)
July 2023	(1)
January 2023	(1)
October 2017	(1)
January 2017	(1)

Break

Listen to those who research

Home About Us Privacy Policy Contact Us