HeatHack Test Book!!!!

Last updated on None.

ntents

his is a test of Github Pages rendering, before we ommit to Jupyter Book. It's mostly the default Jupyter ook with a few simple format tests added to the eginning.

his is public, but of no use to anyone but the evelopers yet (sorry).

ee the Jupyter Book documentation for ocumentation.

dmonitions formatting

Note

Notes have a nice format with a relevant icon and a specific colour.

Warning

See, this one is a bit different because it's a warning. We want a bunch of possibilities like this that aren't types they have natively: Key Concept (which is a some science), Fun Fact!, and Further Readings.

dmonitions are the general case where you can say ets in the "header", but how do you set the icon and olour - where does the "class" go and what are the nes that are already defined?

Key Concepts

Here's some text about some science.

Further Readings

How do we make this one different?

Fun Fact!

And this one - can it be fun?

est of panels

t some point we want to know, how hard is it to put a vascript simulation in (for now, any simple one will o)?

est of sidebars

his is a place to look at the side bar rendering. Maybe can have classes like admonitions do. We might ant to have some of our admonition types be side ars instead.

orem ipsum dolor sit amet, consectetur adipiscing it, sed do eiusmod tempor incididunt ut labore et olore magna aliqua. Ut enim ad minim veniam, quis ostrud exercitation ullamco laboris nisi ut aliquip ex a commodo consequat. Duis aute irure dolor in eprehenderit in voluptate velit esse cillum dolore eu giat nulla pariatur. Excepteur sint occaecat cupidatat on proident, sunt in culpa qui officia deserunt mollit nim id est laborum.

orem ipsum dolor sit amet, onsectetur adipiscing elit, sed o eiusmod tempor incididunt labore et dolore magna iqua. Ut enim ad minim eniam, quis nostrud xercitation ullamco laboris nisi aliquip ex ea commodo onsequat. Duis aute irure olor in reprehenderit in oluptate velit esse cillum olore eu fugiat nulla pariatur. xcepteur sint occaecat upidatat non proident, sunt in ulpa qui officia deserunt mollit nim id est laborum.

orem ipsum dolor sit amet, onsectetur adipiscing elit, sed o eiusmod tempor incididunt labore et dolore magna iqua. Ut enim ad minim eniam, quis nostrud xercitation ullamco laboris nisi aliquip ex ea commodo onsequat. Duis aute irure olor in reprehenderit in oluptate velit esse cillum olore eu fugiat nulla pariatur. xcepteur sint occaecat upidatat non proident, sunt in ulpa qui officia deserunt mollit nim id est laborum.

Here's a sidebar!

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

olore magna aliqua. Ut enim ad minim veniam, quis ostrud exercitation ullamco laboris nisi ut aliquip ex a commodo consequat. Duis aute irure dolor in prehenderit in voluptate velit esse cillum dolore eu giat nulla pariatur. Excepteur sint occaecat cupidatat on proident, sunt in culpa qui officia deserunt mollit nim id est laborum.

orem ipsum dolor sit amet, consectetur adipiscing it, sed do eiusmod tempor incididunt ut labore et olore magna aliqua. Ut enim ad minim veniam, quis ostrud exercitation ullamco laboris nisi ut aliquip ex a commodo consequat. Duis aute irure dolor in prehenderit in voluptate velit esse cillum dolore eu giat nulla pariatur. Excepteur sint occaecat cupidatat on proident, sunt in culpa qui officia deserunt mollit nim id est laborum.

Markdown Files

Whether you write your book's content in Jupyter notebooks (`.ipynb`) or in regular markdown files (`.md`), you'll write in the same flavor of markdown called **MyST Markdown**. This is a simple file to help you get started and show off some syntax.

What is MyST?

MyST stands for "Markedly Structured Text". It is a slight variation on a flavor of markdown called "CommonMark" markdown, with small syntax extensions to allow you to write **roles** and **directives** in the Sphinx ecosystem.

For more about MyST, see the MyST Markdown overview.

Sample Roles and Directivs

Roles and directives are two of the most powerful tools in Jupyter Book. They are kind of like functions, but written in a markup language. They both serve a similar purpose, but **roles are written in one line**, whereas **directives span many lines**. They both accept different kinds of inputs, and what they do with those inputs depends on the specific role or directive that is being called.

Here is a "note" directive:

> **Note**
>
> Here is a note

It will be rendered in a special box when you build your book.

Here is an inline directive to refer to a document: notebooks with MyST Markdown.

Citations

`cite}`holdgraf_evidence_2014`` will render like is: [HdHPK14].

oreover, you can insert a bibliography into your page ith this syntax: The `{bibliography}` directive must e used for all the `{cite}` roles to render properly. or example, if the references for your book are stored `references.bib`, then the bibliography is inserted ith:

IdHPK14] Christopher Ramsay Holdgraf, Wendy de Heer, Brian N. Pasley, and Robert T. Knight. Evidence for Predictive Coding in Human Auditory Cortex. In *International Conference on Cognitive Neuroscience*. Brisbane, Australia, Australia, 2014. Frontiers in Neuroscience.

earn more

his is just a simple starter to get you started. You can arn a lot more at jupyterbook.org.

lotebooks with MyST Markdown

upyter Book also lets you write text-based notebooks sing MyST Markdown. See the Notebooks with MyST arkdown documentation for more detailed structions. This page shows off a notebook written in yST Markdown.

n example cell

ith MyST Markdown, you can define code cells with directive like so:

```
print(2 + 2)
```

```
4
```

hen your book is built, the contents of any `code-cell}` blocks will be executed with your efault Jupyter kernel, and their outputs will be splayed in-line with the rest of your content.

See also

Jupyter Book uses Jupytext to convert text-based files to notebooks, and can support many other text-based notebook files.

reate a notebook with MyST Markdown

yST Markdown notebooks are defined by two things:

1. YAML metadata that is needed to understand if / how it should convert text files to notebooks (including information about the kernel needed).

2. The presence of `{code-cell}` directives, which will be executed with your book.

hat's all that is needed to get started!

Quickly add YAML metadata for MyST Notebooks

you have a markdown file and you'd like to quickly dd YAML metadata to it, so that Jupyter Book will eat it as a MyST Markdown Notebook, run the llowing command:

```
jupyter-book myst init
path/to/markdownfile.md
```

Demonstration that we can produce the graphs we want easily in the Jupyter Book

his is a test file for some of the kinds of plots we will ant in HeatHack-Sessions, so we can see what uilds look like and how much we can automate. lthough this uses csv files uploaded to Github anually, it may be possible to automate temp/RH ed download and the production of books specific to e venues we serve.

asic plot showing a thingspeak temperature feed arked up for 16C, the child care commission inimum (not relevant for this particular data, just a st). This plot is also useful for assessing temperature ontrol, especially on a short test for overshoot that es holding a building at a temperature - cheapest in utumn. We'll want similar plots showing suggested H bounds for the comfort of people and for organs/oil aintings and so on.

ote pan, zoom, etc - not beautiful, but even this basic vel of plot would work. I wonder whether they'll be orried by the rogue readings. We could probably move based on improbably fast temperature anges.

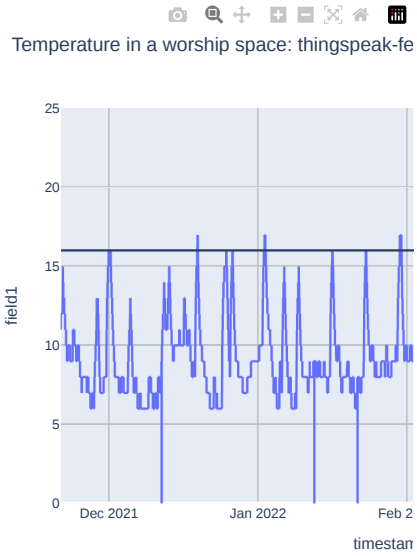
n our current thingspeak feeds, temperature is field1 nd RH is field2 - we may be able to assign better ames in future. I think I had to change the time format o that's either some scripting or a configuration ange on the platform.

lotly express is syntactic sugar over graph_objects; op down into the graph_objects themselves allows ore possibilities.

formation about non-plotly approaches: tps://jupyterbook.org/interactive/interactive.html One onsideration is whether they're going to need internet ccess to look at graphs - they might not have that hen they're together if they meet on the premises. ltair sounded like it might be useful in that situation.

ODO: find out if we can hide the code.

```
import plotly.express as px
import plotly.graph_objects as go

import pandas as pd

filename = "thingspeak-feed"
dfthingspeak = pd.read_csv(filename +
".csv")
dfthingspeak["timestamp"] =
pd.to_datetime(dfthingspeak['created_at'])

fig = px.line(dfthingspeak, x='timestamp',
 y='field1', range_x=['2021-11-21','2022-
04-07'],range_y=[0,25], title="Temperature
in a worship space: " + filename)

fig.add_hline(y=16)

fig.show()
```

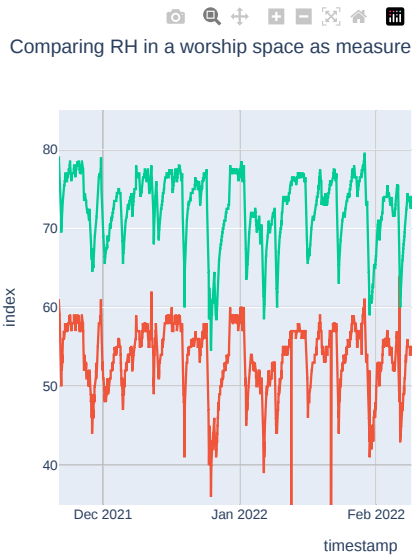


imple demonstration of data from two data frames on e same plot - with the wrinkle that one frame is from lascar logger. We will be roughly exploring the alibration of the RH sensors by running batches of 10 HT22s alongside a few Lascars over an RH range nd showing groups the results, so they can judge ow much to trust the data.

ascars aren't configurable for what they export. I've moved a Unicode character this couldn't deal with legree symbol) and used Excel to change the data rmat. These things should be fixable in code, but we on't use Lascars enough for that to be a priority task. ny processing we need to do on Thingspeak feeds is priority, though.

e need every line labelled. Doing that requires us to ll out each line into a separate command to add it, I ink - this might be an odd way of combining express nd graph_object?

```
import plotly.graph_objects as go

import pandas as pd
dfthingspeak = pd.read_csv("thingspeak-
feed.csv")
dfthingspeak["timestamp"] =
pd.to_datetime(dfthingspeak['created_at'])

dflascar = pd.read_csv("lascar-data.csv")
dflascar['timestamp'] =
pd.to_datetime(dflascar['Time'])

# fig = px.line(dfthingspeak,
x='timestamp', y='field2', range_x=['2021-
11-21','2022-04-07'],range_y=[35,85],
title="Comparing RH as measured by
different devices")
fig = px.line(dfthingspeak, x='timestamp',
 range_x=['2021-11-21','2022-04-
07'],range_y=[35,85], title="Comparing RH
in a worship space as measured by
different devices side by side")
fig.add_scatter(x =
dfthingspeak['timestamp'], y =
dfthingspeak['field2'], name =
'thingspeak')
fig.add_scatter(x = dflascar['timestamp'],
y = dflascar['RH'], name = 'lascar')


fig.show()
```



ertical lines are useful for the start and end time of vents. It would be better rendered as a separate ackground shading when the space is occupied.

erhaps we can set up a worksheet where they put in eir usual weekly schedule with a descriptive short ring to render these. We could use diary export, but if eir diary doesn't have a busy/free option, there's too uch risk of personal data being in there, and there uld be too many diary systems to deal with.

'ODO: It would be helpful if there were a dropdown ntrol for choosing to view a day or a week, and then hich specific day or week. That sort of control could e used to choose the group and venue, as well, so e're only producing one master book for everyone.

```
import plotly.express as px

import pandas as pd
df = pd.read_csv("thingspeak-feed.csv")
df["timestamp"] =
pd.to_datetime(df['created_at'])


#animation_frame and animation_group
should make it possible to add a range
slider??

fig = px.line(df, x='timestamp',
y='field1',  range_x=['2021-12-24','2021-
12-26'],range_y=[0,20], title="Midnight
mass and Christmas morning services in a
worship space.")
fig.add_vline(x='2021-12-24 23:00')
fig.add_vline(x='2021-12-25 00:00')
fig.add_vline(x='2021-12-25 10:00')
fig.add_vline(x='2021-12-25 11:30')

fig.show()
```