

Using Jupyter Notebooks in the Classroom

Spring 2026 LTF Event at the Center for Teaching and Learning

Part One: Lecture/Demonstration

1. Familiarize ourselves with the structure and functionality of a Jupyter Notebook
2. Explore how Jupyter Notebooks can be used in the classroom to enhance student learning
3. Discuss Universal Design for Learning (UDL) principles and how Notebooks can be used for implementation

Part Two: Activity

Brainstorm and create an outline for a homework assignment that you might create in Jupyter notebooks. If you have Jupyter downloaded, feel free to code it up! Work in small groups, and we'll spend the last ten minutes sharing ideas.

Note: We'll be visualizing data from TRACE-21K, a transient climate model simulation spanning the last 21 thousand years, as a real-world example for how some of these principles might be applied. However, the workshop material itself is subject agnostic! If you aren't familiar with Earth Sciences, fear not.

```
title

1. Interleave small blocks of text with blocks of code to:

    (A) Demonstrate concepts that come up in lecture
    (B) Make lecture materials more interactive
    (B) Help introduce coding concepts/packages in an accessible way to students who are not experienced

• Documents can be interacted with during classroom sessions to demonstrate concepts.
• Hide cell output prior to the beginning of the class, and create visualizations as you go!
• Embed activities by importing outside modules
```

```
In [ ]: # Import coding packages -- each package contains certain functions that we will need
import numpy as np
import xarray as xr
import pandas as pd
import matplotlib.pyplot as plt
import cartopy.crs as ccrs

# Import our TRACE transient climate model temperature dataset
temp = xr.open_dataset('r/Users/red/Desktop/TRACE_ALL/trace.01-36.22000BP.cam2.TREFHT.22000BP_decavg_400BCE.nc')
temp
```

Mock Lecture Material

Let's Look at Differences in Simulated Global Temperature from the Last Glacial Maximum to Present

21K years ago ice sheets covered much of the Northern Hemisphere.

1. What does our present day temperature distribution look like? Turn to your partner and describe some general features and *why* those features exist (e.g. why might the tropics be generally warmer than the poles?)..
2. What do you expect the LGM temperature distribution to look like?
3. Where do you anticipate to see the most significant differences?

```
In [ ]: ## QUESTION 1

# create a figure using matplotlib.pyplot and cartopy
figure = plt.figure(figsize = (10,5))
ax = plt.axes(projection=ccrs.PlateCarree(central_longitude = 180))

# show continental coastlines
ax.coastlines()

# take a snapshot of global temperature at 0 ka BP (present) and 21 ka BP (LGM)
# average data over 5 decades
LGM = temp.isel(time = [0,1,2,3,4,5]).TREFHT.mean('time')
present = temp.isel(time = [-1,-2,-3,-4,-5]).TREFHT.mean('time')

# plot present-day data
cf = ax.contourf(present.lon, present.lat, present, cmap = 'coolwarm', extend='both',
                transform=ccrs.PlateCarree(), levels = np.arange(225,315,10))

# add a colorbar
cbar = plt.colorbar(cf, ax=ax, label = 'Temperature (K)')

plt.title('Present day')
plt.show()
```

```
In [ ]: # Question 2

# create a figure using matplotlib.pyplot and cartopy
figure = plt.figure(figsize = (10,5))
ax = plt.axes(projection=ccrs.PlateCarree(central_longitude = 180))
ax.coastlines()

# plot data
cf = ax.contourf(LGM.lon, LGM.lat, LGM, cmap = 'coolwarm', extend='both',
                transform=ccrs.PlateCarree(), levels = np.arange(225,315,10))
cbar = plt.colorbar(cf, ax=ax, label = 'Temperature (K)')

plt.title('LGM (21ka BP)')
plt.show()
```

```
In [ ]: # Question 3

# create a figure using matplotlib.pyplot and cartopy
figure = plt.figure(figsize = (10,5))
ax = plt.axes(projection=ccrs.PlateCarree(central_longitude = 180))
ax.coastlines()

# plot the difference between LGM and present day data
difference = present-LGM

cf = ax.contourf(difference.lon, difference.lat, difference, cmap = 'coolwarm', extend='both',
                transform=ccrs.PlateCarree(), levels = np.arange(-15,15))
cbar = plt.colorbar(cf, ax=ax, label = 'Temperature (K)')

plt.title('LGM (21ka BP)')
plt.show()
```

Is anything surprising?

Let's look more closely at some temperature time series for specific locations...

Discussion Question: What climate events might you be looking for? When do they occur?

```
In [ ]: # define a latitude and longitude of interest
#latitude = 71.2906
#longitude = 360 - 156.7886 # currently Barrow, Alaska
latitude = 0
longitude = 30

# compare with new york city
nyc_lat = 40
nyc_lon = 360 - 74.0060

# subset original temperature dataset to isolate the nearest grid cell to that latitude
# and longitude
nyc_series = temp.sel(lat = nyc_lat, lon = nyc_lon, method = "nearest").TREFHT
temp_series = temp.sel(lat = latitude, lon = longitude, method = "nearest").TREFHT

# plot data and label axes.
# note that in TRACE data, time is in negative kyrs before present

figure = plt.figure(figsize = (10,4))
plt.plot(temp.time*1000, nyc_series-nyc_series.mean(), color = 'blue', label = 'NYC')
plt.plot(temp.time*1000, temp_series-temp_series.mean(), color = 'red')
plt.title("Latitude = " + str(latitude) + ", Longitude = " + str(longitude))
plt.ylabel('ΔTemperature (K)', fontsize = 13)
plt.xlabel('Time (yrs BP)', fontsize = 13)
plt.legend()
```

Applying Jupyter in the Classroom: Finding Resources

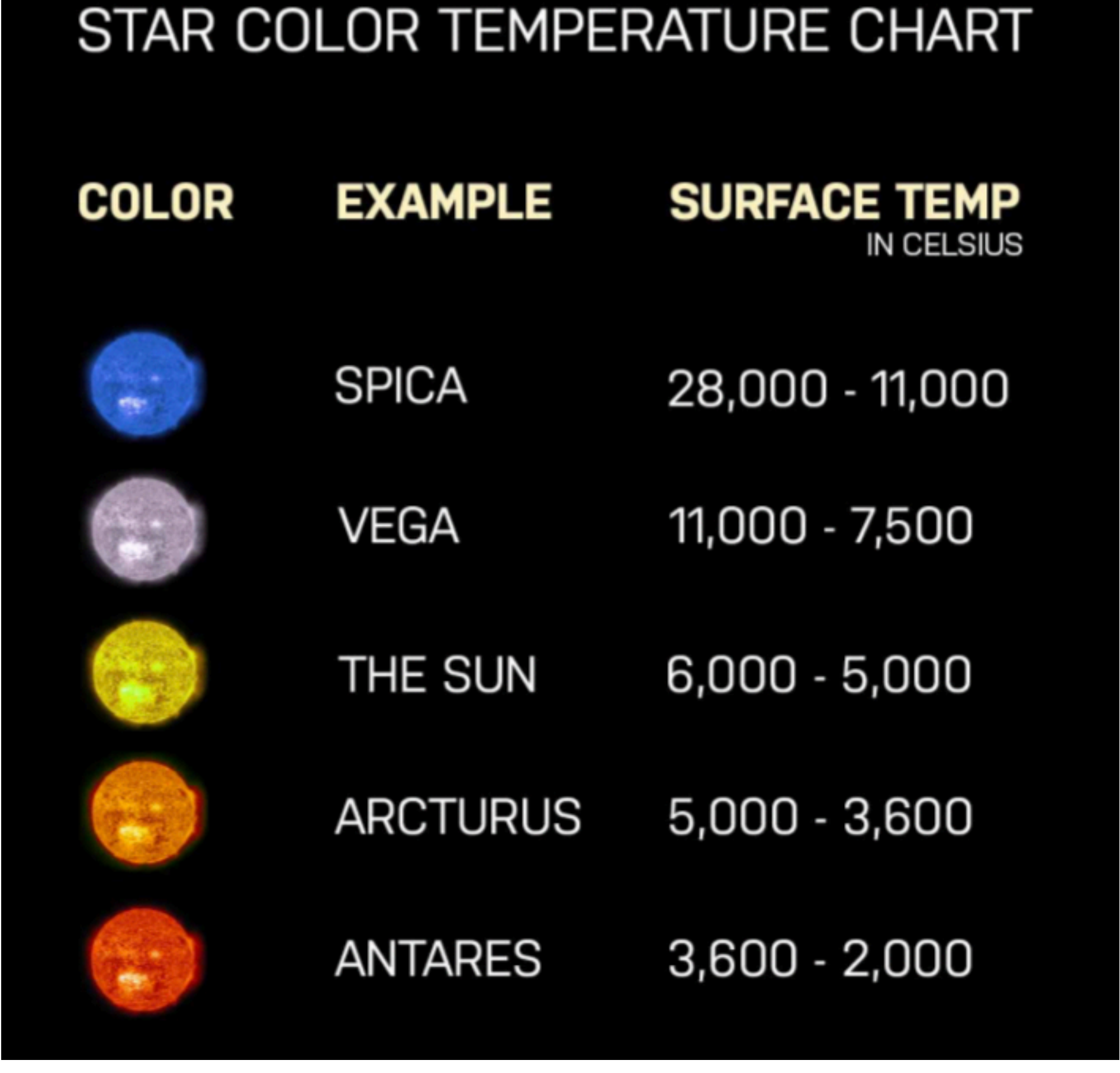
One of the key benefits of using Jupyter with Python in the classroom is that there are thousands of plugins and coding packages already available, and new ones are constantly in development.

1. Install external packages to create live quizzes that can be answered with clickers or students' phones
2. Create GUIs with sliders, toggle buttons, or text boxes that allow you to quickly demonstrate a concept
3. Create gifs/videos for data visualization

Most common plugins/coding packages are exceptionally well documented, but that documentation can be dense. ChatGPT and other AI tools can be very helpful for debugging and finding software, though they do make mistakes. StackOverflow and other community forums are also helpful, and there are a number of websites that specifically describe how to best use Jupyter in pedagogical settings.

Mock Activity: Designing a GUI for Classroom Experiments

The fact that Earth can support life should not be taken for granted! Earth exists in a "goldilocks zone" in our solar system, where water can exist as a liquid, and the surface is not extremely hot or extremely cold. What would happen if we swapped out our sun for a different model?



```
In [ ]:

In [ ]: # Import widgets
from ipywidgets import interact, FloatSlider
import math

In [ ]: # Question: how does the temperature of Earth's surface change if the sun was hotter/cooler?

def earth_temp(Tsun):
    o = 5.67 * 10**-8 # Boltzmann Constant
    dist=1.496*10**11
    r_sun = 6.96*10**8
    S = 0*Tsun**4*(4*math.pi*r_sun**2) # Energy emitted by sun in all directions (W)
    radius = 6.371 * 10**7 # Earth's radius

    S_dist = S/(4*math.pi*dist**2) # solar energy flux in Watts/m^2 at set distance from the sun
    S = S_dist/4 # solar energy normal to Earth's surface, divided by Earth's surface area

    ## calculate the average surface temperature of the Earth
    Tearth = np.power(S/o,.25)

    #####

    # now plot Stefan Boltzmann Law to see how Tsun effects emission spectra.. this is the equation that we saw earlier

    h = 6.626e-34 # plank's constant
    c = 3e8 # speed of light
    k = 1.380649e-23 # Boltzmann constant

    # wavelength in meters: 100 nm to 3000 nm
    lam = np.linspace(100e-9, 3000e-9, 1000)

    Bsun = (2*h*c**2)/((lam**5*(np.exp(h*c/(lam*k*Tsun)) - 1))
    Bsun_real = (2*h*c**2)/((lam**5*(np.exp(h*c/(lam*k*5770)) - 1))

    fig, ax = plt.subplots(figsize = (10,5))

    #plot sun's spectra on one yaxis
    plt.plot(lam, Bsun, color = 'black')
    plt.plot(lam, Bsun_real, color = 'gold', label = 'Sun')
    plt.xlabel('Wavelength (nm)')
    plt.ylabel('Radiance (Sun)')
    plt.ylim(0,4e13)
    plt.xlim(100e-9, 3000e-9)

    #####

    # let's define some common Temperatures for different stars for comparison
    arcturus = np.max((2*h*c**2)/((lam**5*(np.exp(h*c/(lam*k*4000)) - 1)))
    antares = np.max((2*h*c**2)/((lam**5*(np.exp(h*c/(lam*k*3000)) - 1)))

    # plot the maximum radiance for each
    plt.hlines(arcturus, xmin = 1e-9, xmax = 35000e-9,
              color = 'darkorange', label = 'Arcturus, 4000', linestyle = '-')
    plt.hlines(antares, xmin = 1e-9, xmax = 35000e-9,
              color = 'red', label = 'Antares, 3000', linestyle = '-')

    #####

    # Use Wien's law to show how the maximum wavelength changes
    b = 2.8977193e-3 # Wien's constant
    lam_max = b/Tsun

    plt.vlines(lam_max, ymin = 0,ymax = 4e13, color = 'black', linestyle = '-', label = '$\lambda_{max}$')
    plt.text(lam_max, 3e13, ' $\lambda_{max}$', ha = 'left', fontsize = 15)
    plt.legend()

    return "Earth's Average Surface Temperature: " + str(round(Tearth,0)-273) + "°C"

In [ ]: interact(earth_temp,Tsun=FloatSlider(value=5770, min=3000, max=6500, step=5, description='Tsun (K)',
continuous_update=False))
```

Jupyter Notebooks as an Implementation Tool for Universal Design for Learning (UDL) Principles

Design Multiple Means of Engagement →

Design Multiple Means of Representation →

Design Multiple Means of Action & Expression →

Support

Design Options for Sustaining Effort & Persistence (8) →

Design Options for Language & Symbols (2) →

Design Options for Expression & Communication (5) →

Final Workshop Activity

1. Brainstorm and create an outline for a Jupyter Notebook activity or homework assignment that you might create for a class in your department. Feel free to be creative! This doesn't necessarily have to be a "realistic" exercise.
2. If you have Jupyter Notebook downloaded, feel free to begin coding your work. Otherwise, pen and paper is fine.
3. We'll spend the last ten minutes of this workshop sharing ideas within small groups.
4. You can also take this time to download and install Jupyter Notebook.

Quick plug: Check out more resources from the CTL!



Please remember to write down your name and UNI on the sign out sheet when you leave!

Thank you for coming :)

```
In [202]: !jupyter nbconvert --to html CTL_Event.ipynb

[NbConvertApp] Converting notebook CTL_Event.ipynb to html
[NbConvertApp] Writing 2325632 bytes to CTL_Event.html
```

```
In [ ]:
```