



R. BRICEÑO

Assistant Professor - ODU
Research Staff - Jefferson Lab
DOE Early Career Award recipient (2018)
<http://bit.ly/rbricenoPhD>



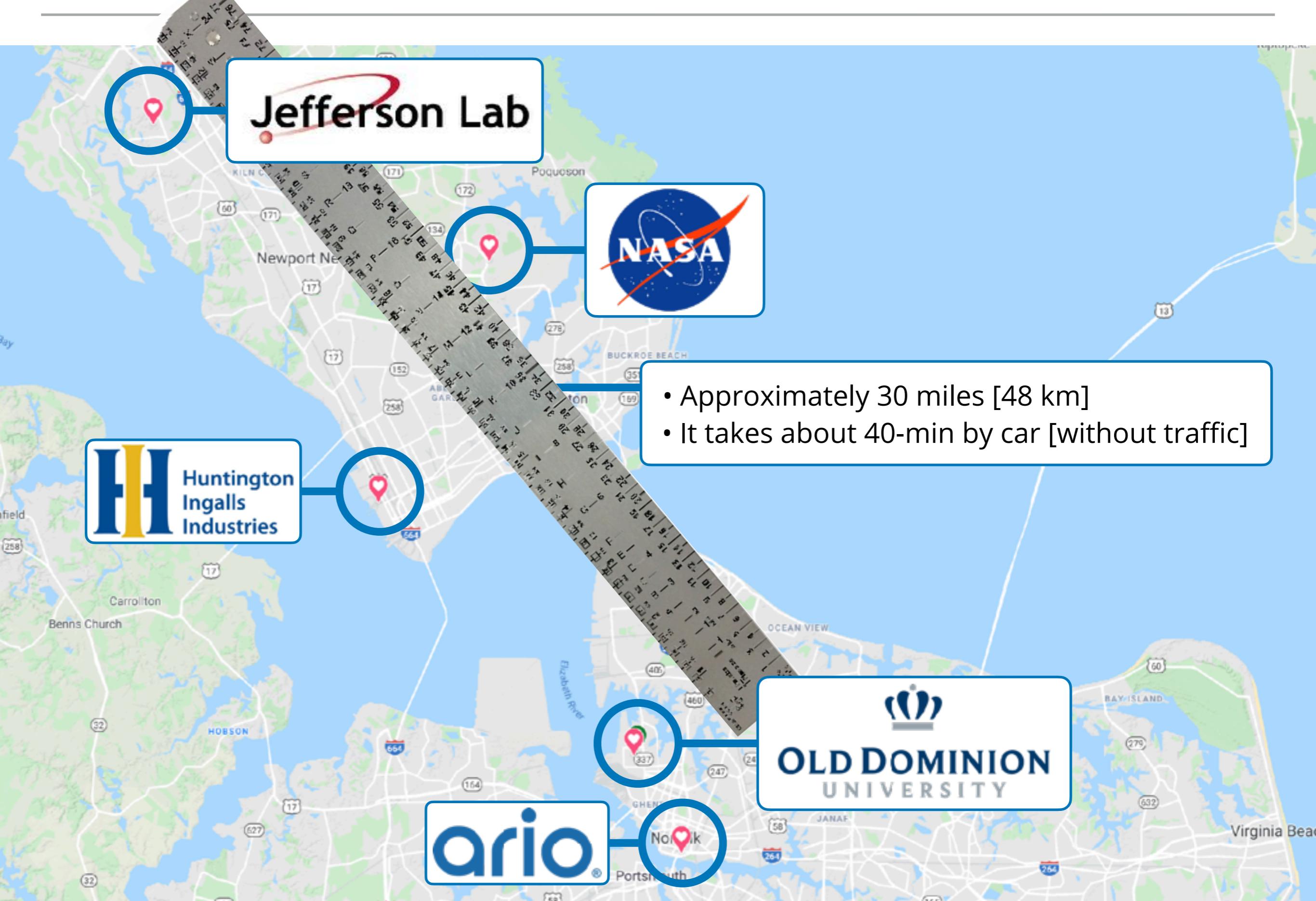
T. ROGERS

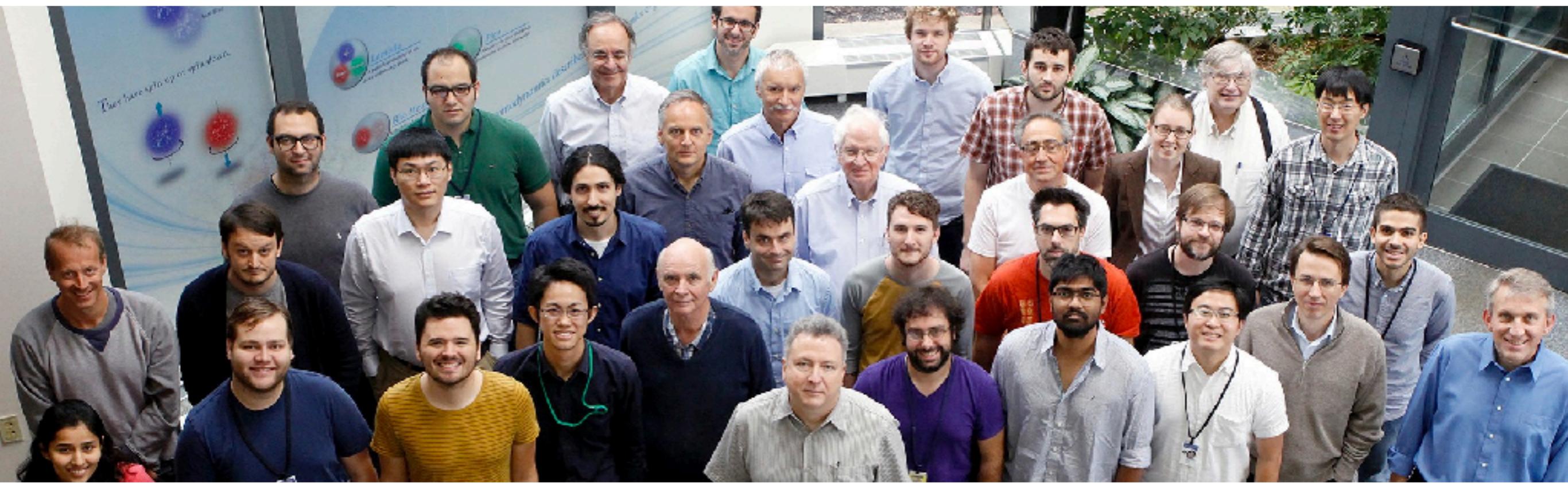
Assistant Professor - ODU
Research Staff - Jefferson Lab
DOE Early Career Award recipient (2017)
<https://sites.google.com/a/odu.edu/tedcrogers/home>

PYTHON 4 PHYSICS INTRO

HAMPTON ROADS TECH

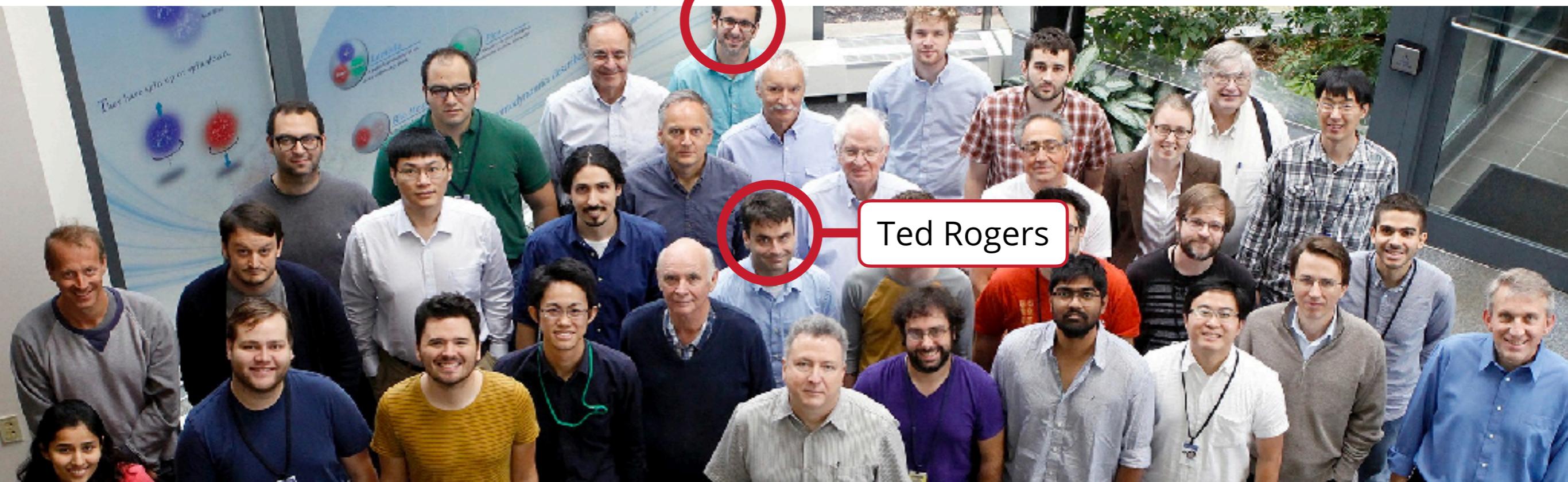






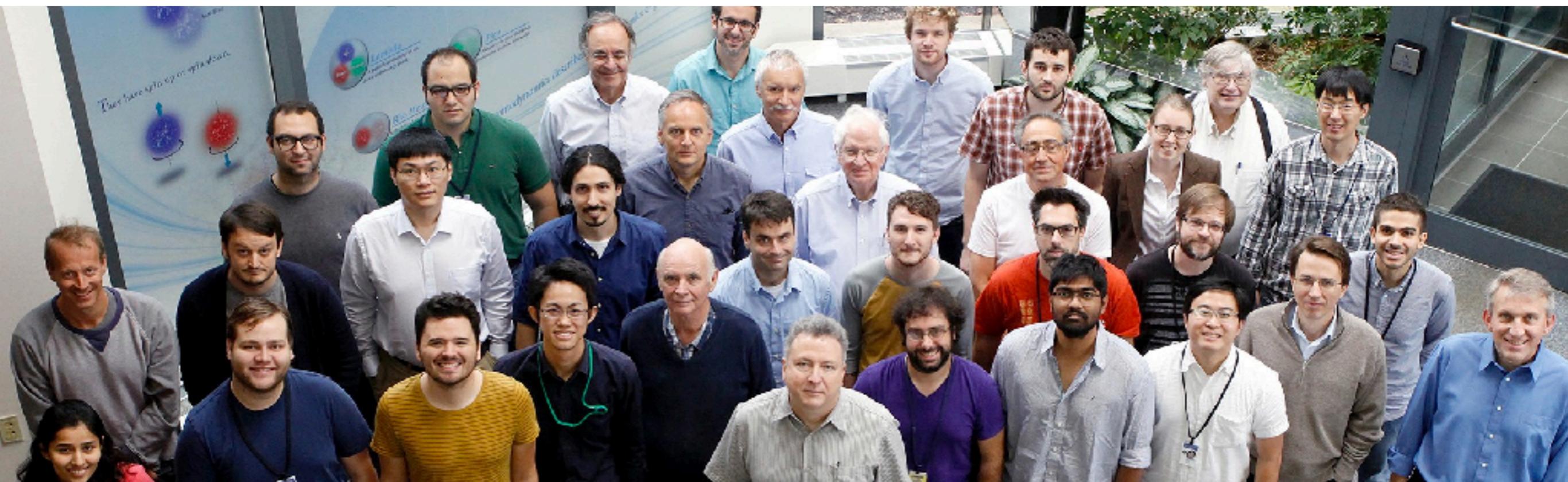
Jefferson Lab

<https://www.jlab.org/theory>



Jefferson Lab

<https://www.jlab.org/theory>



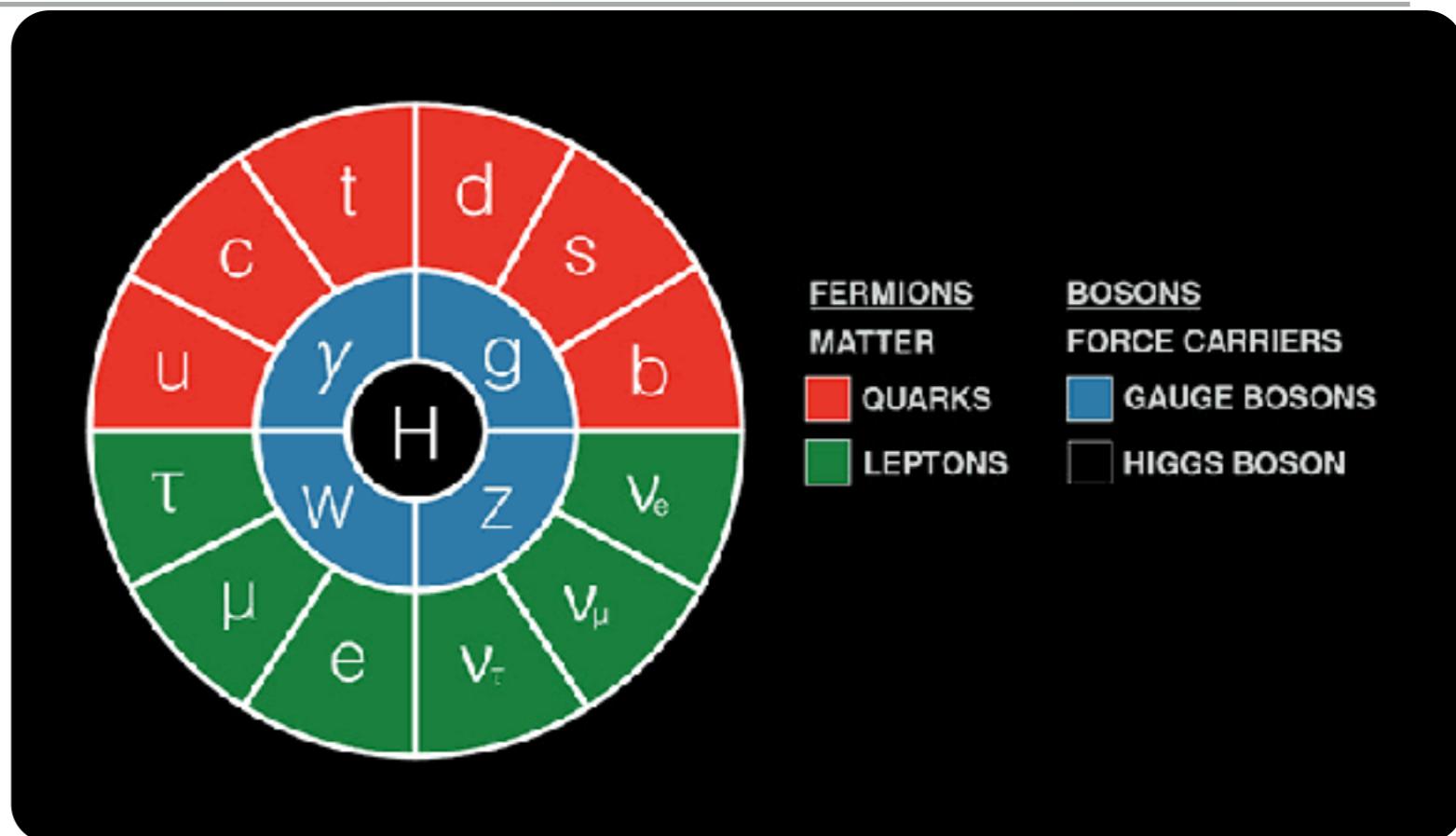
**Virginia's biggest and
fastest supercomputer**



Jefferson Lab

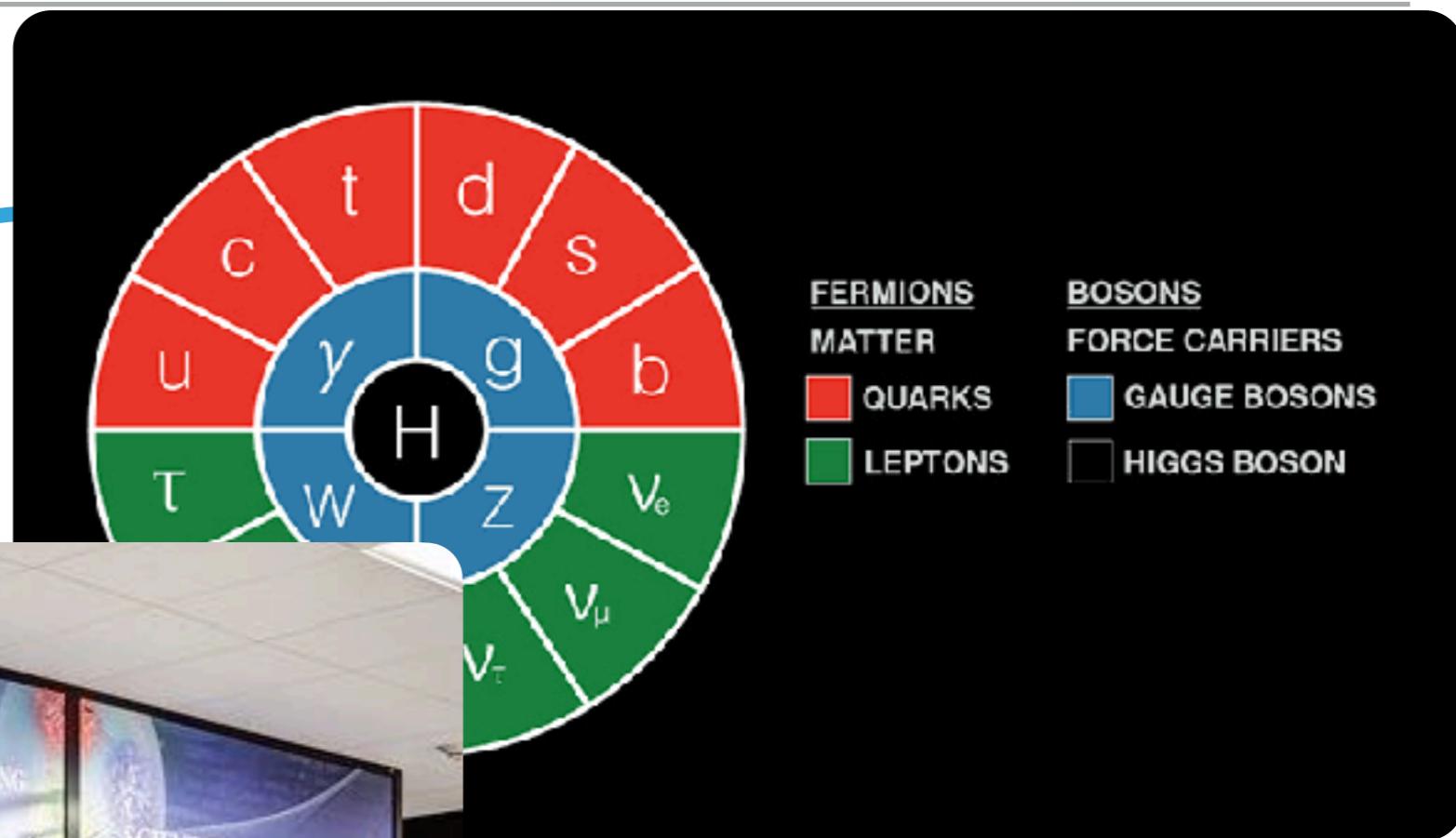
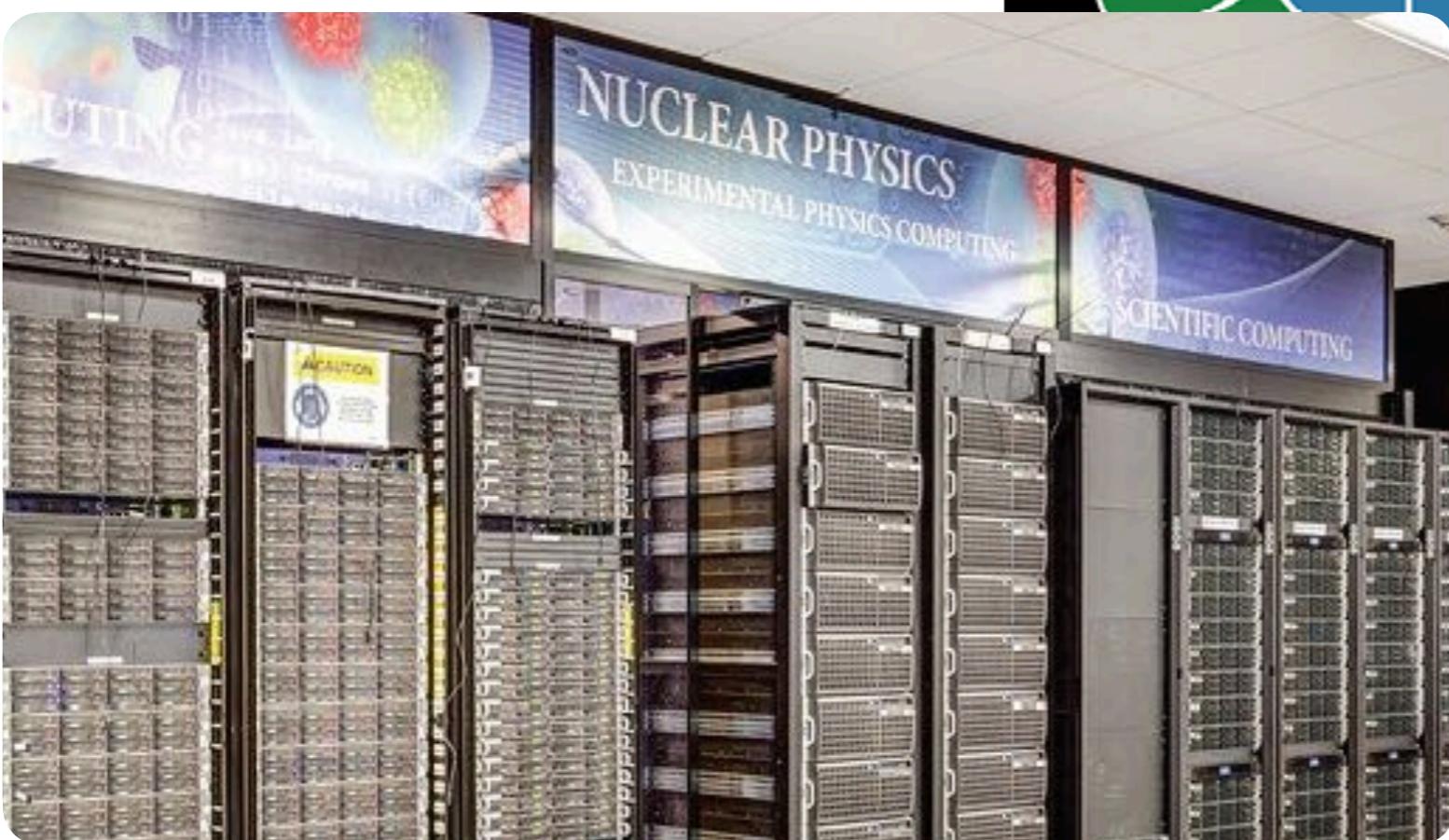
the Standard Model of Particle Physics

"3/4 of the forces we know of in nature"



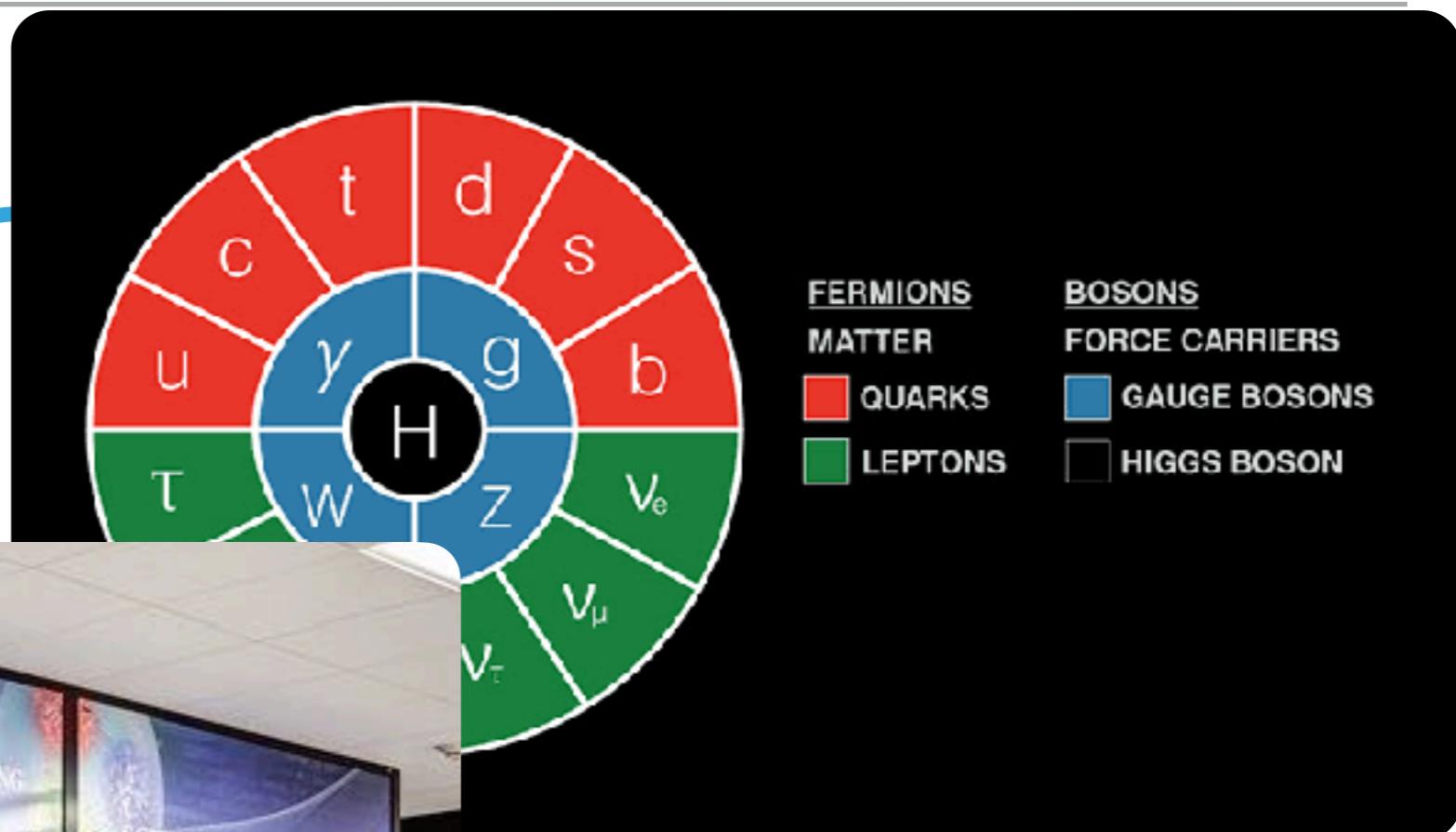
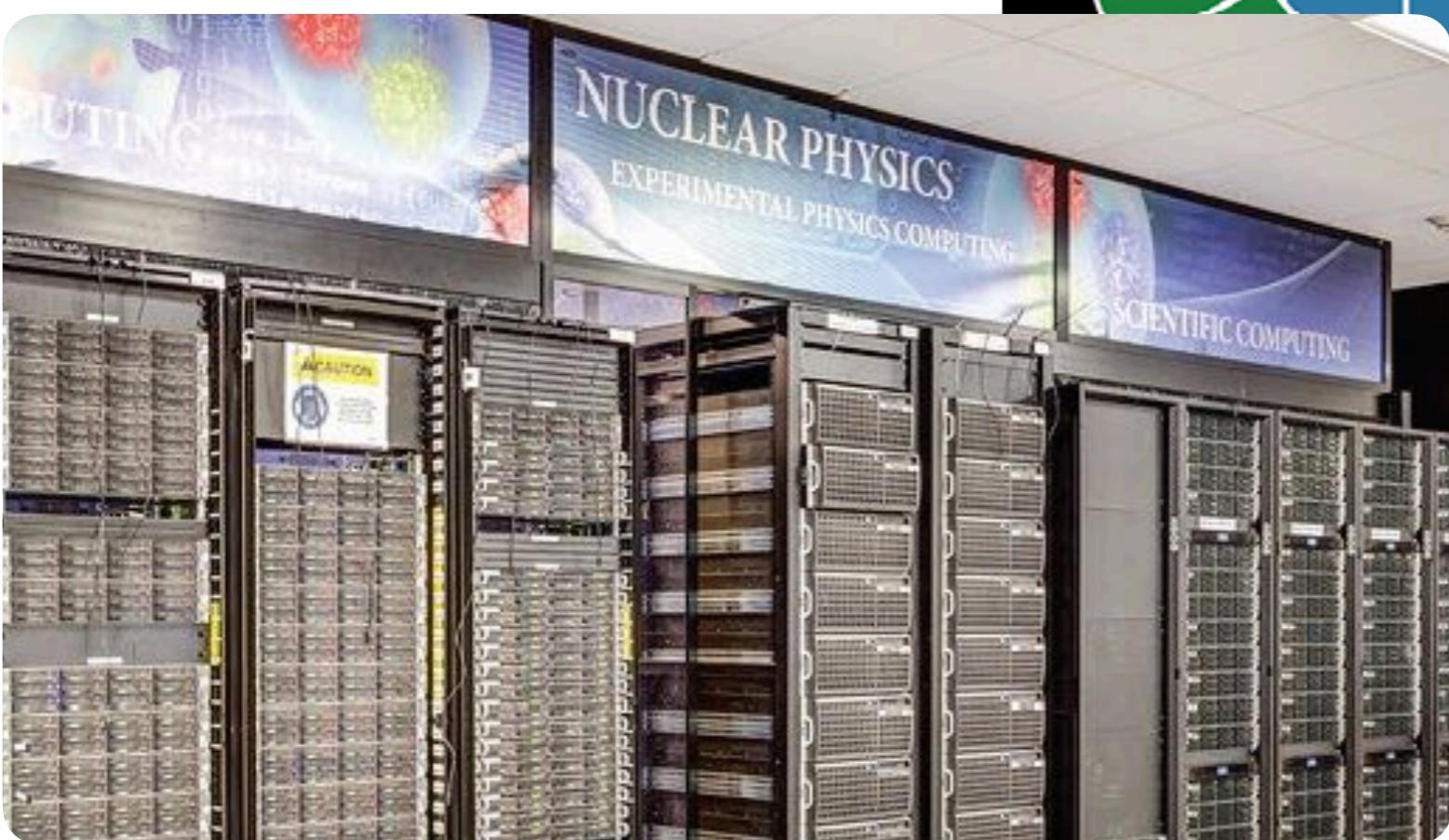
SIMULATING THE UNIVERSE

put the Universe's building blocks and rules into a big enough computer...

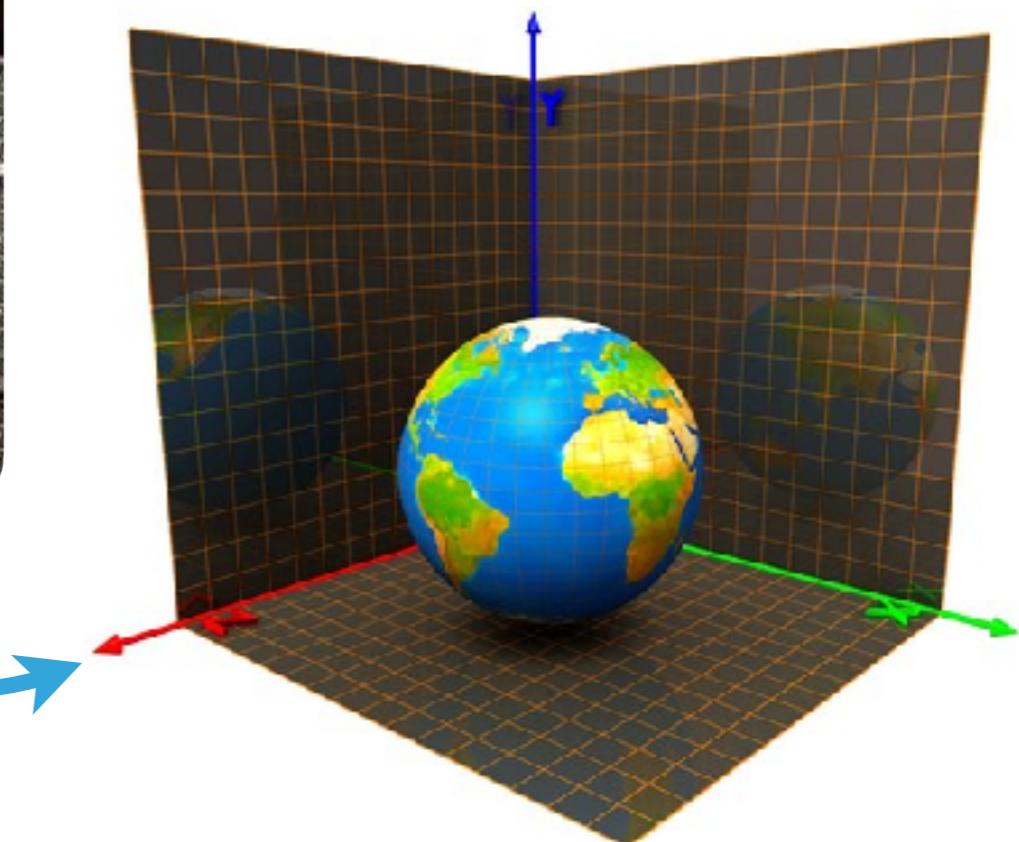


SIMULATING THE UNIVERSE

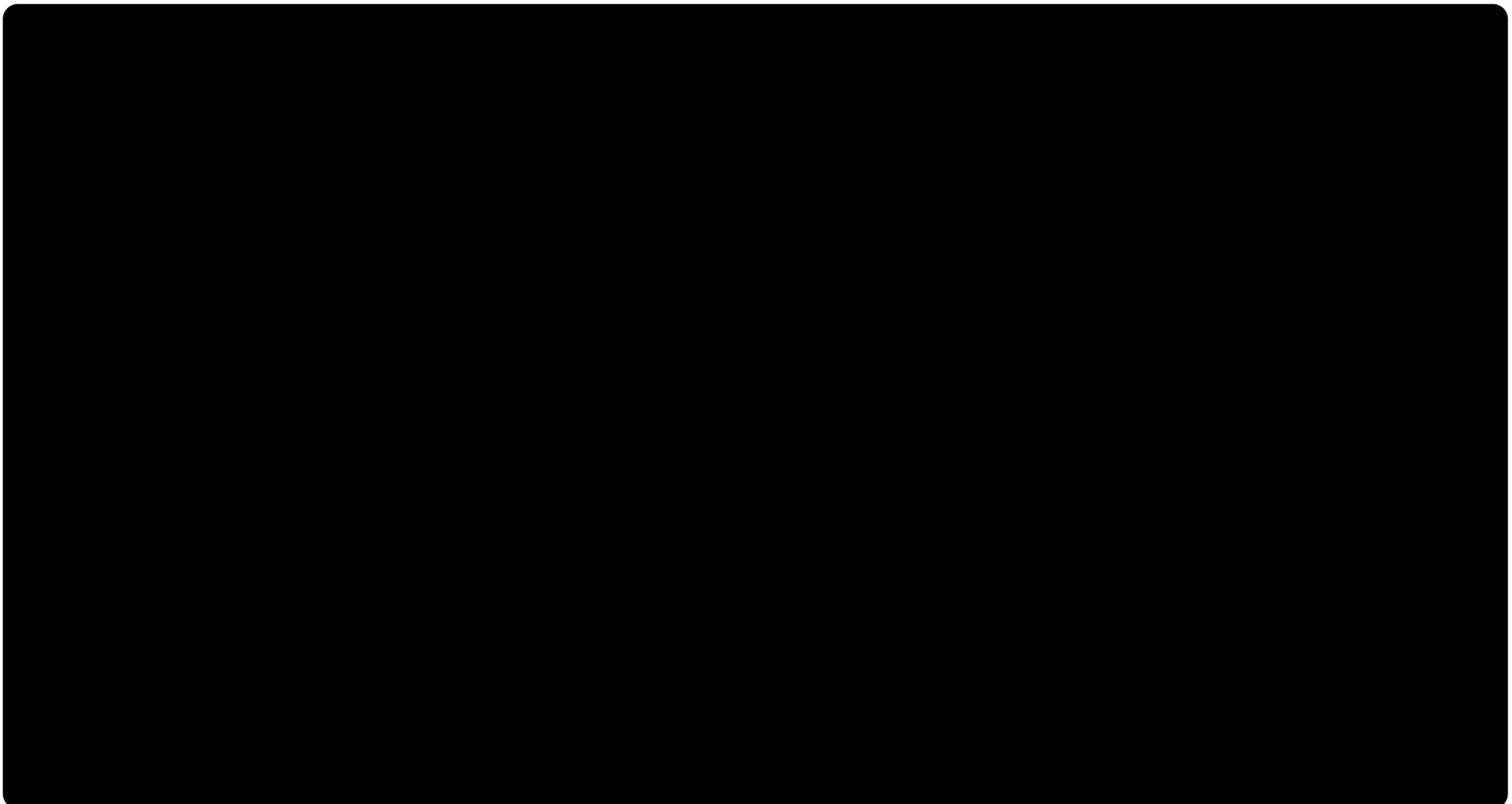
put the Universe's building blocks and rules into a big enough computer...



...and nature should emerge



MODERN DAY NUCLEAR PHYSICS



MODERN DAY NUCLEAR PHYSICS



M. T. Hansen, Ph.D.

Scientific Staff - CERN

*CERN: The past and future
of the world's best microscope*

07/20



C. Andrés, Ph.D.

Postdoctoral fellow - JLab

Looking inside the proton at JLab

08/03



L. Weinstein, Ph.D.

Physics Faculty - ODU

GUESSTIMATION

07/06



OLD DOMINION
UNIVERSITY



F. Ortega-Gama

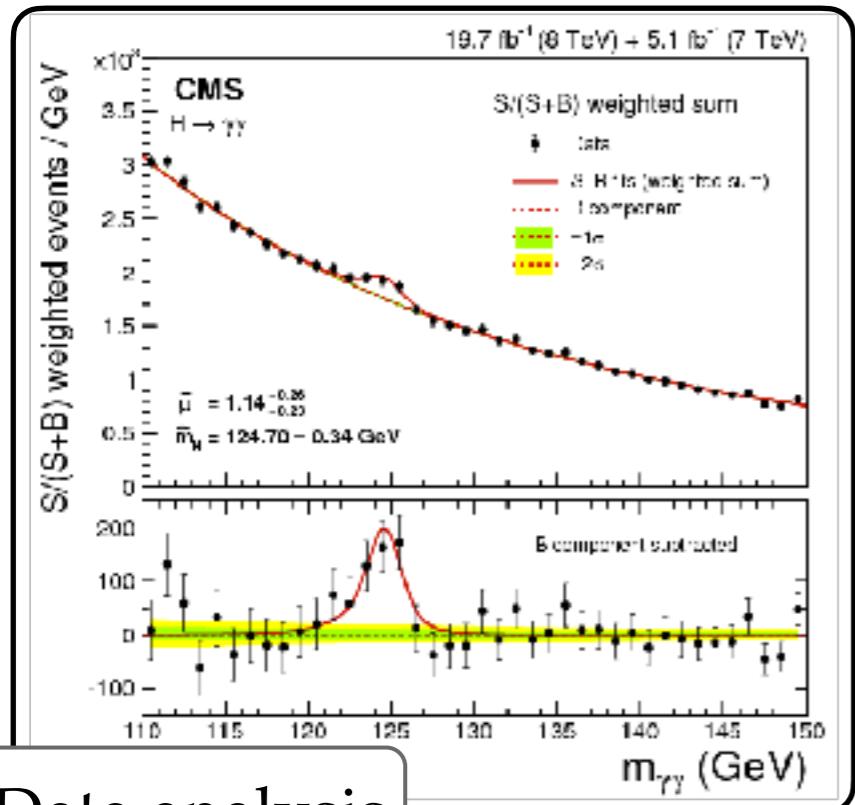
Student - William & Mary

*Symmetries and how to use them
in physics* 07/24

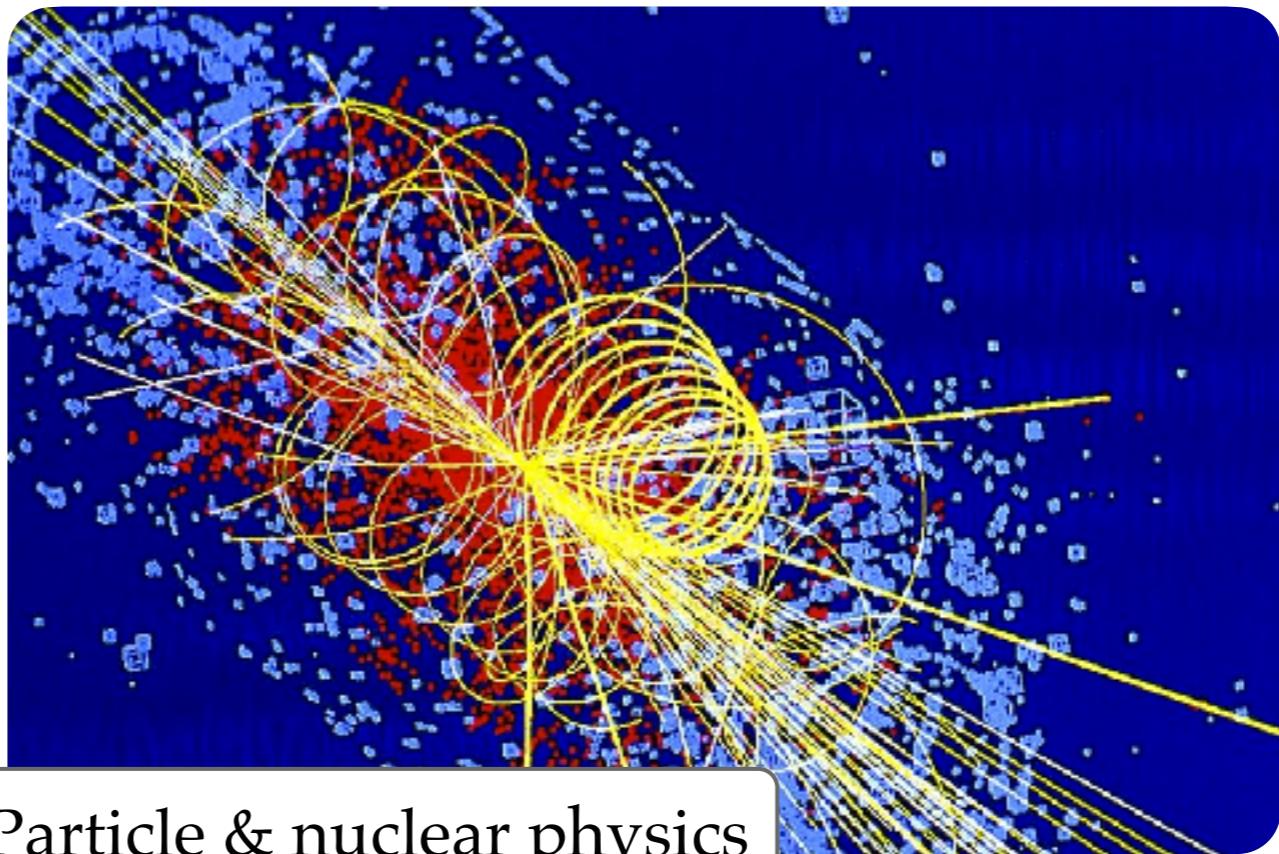


WILLIAM & MARY
CHARTERED 1693

YOU WILL LEARN ABOUT...

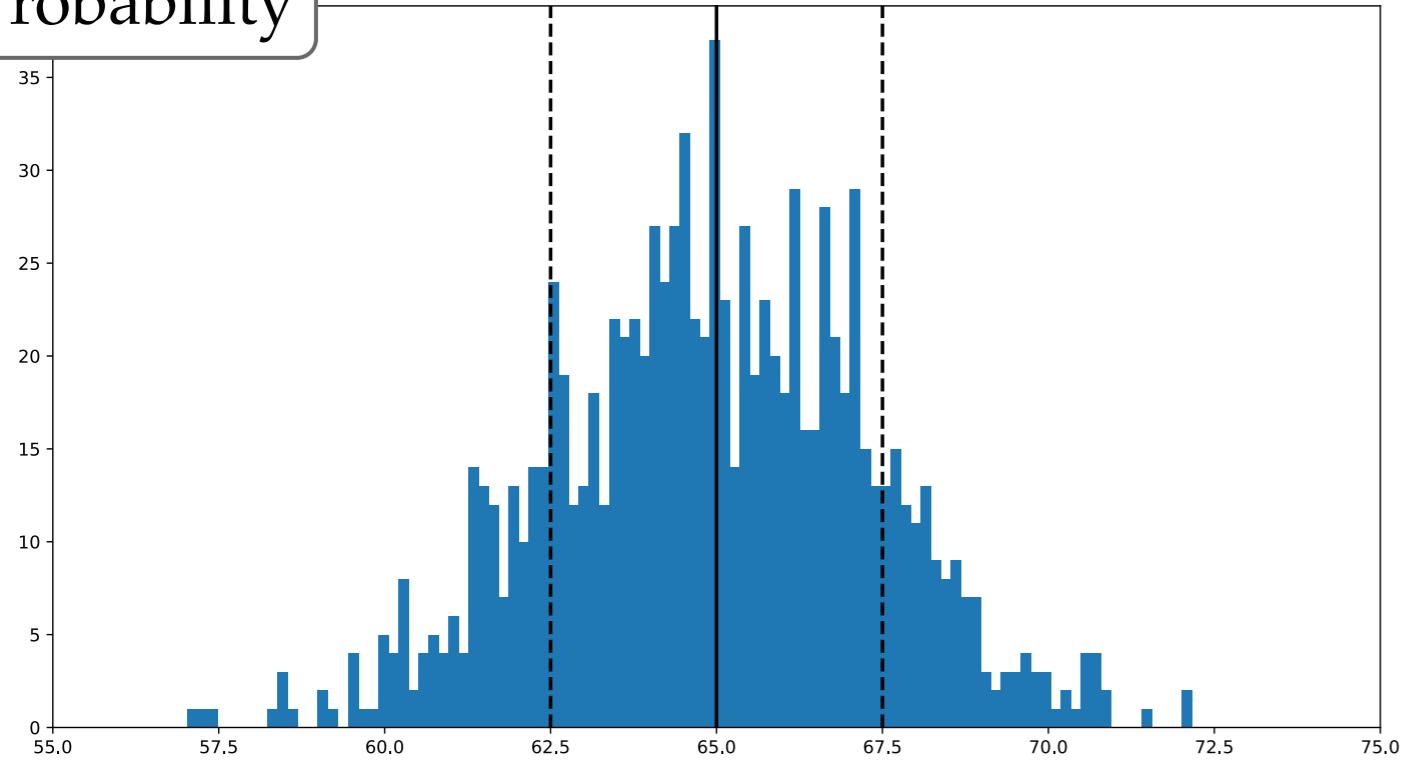


Data analysis

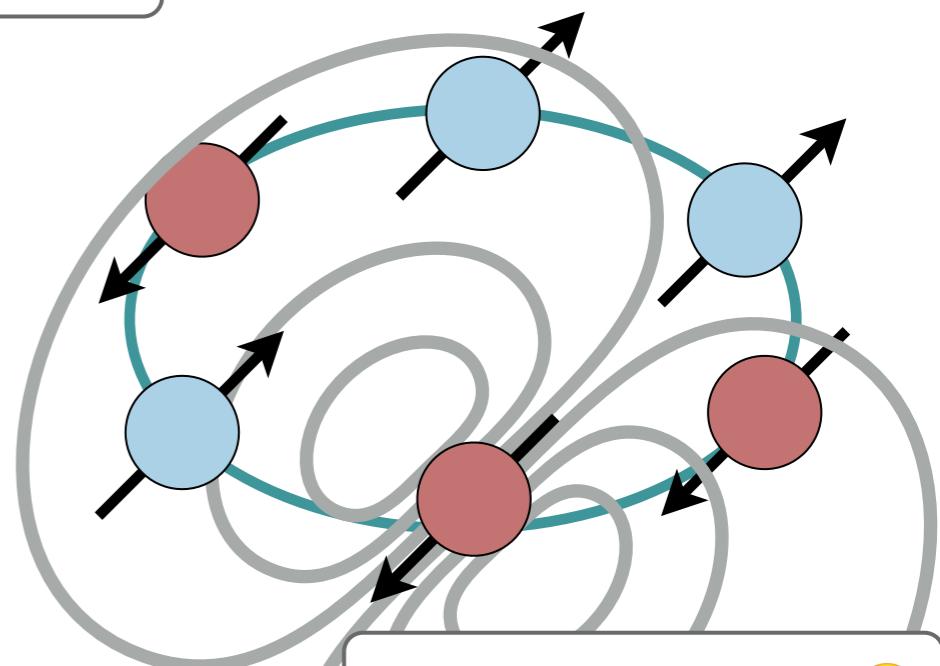


Particle & nuclear physics

Probability



Magnets



...and lots more 😎

MEET JAZZI ...she will be your spiritual animal in P4P 🐶

“less talking, more coding”



DON'T WANT TO DOWNLOAD PYTHON?

Feeling lazy and don't want to download Python or anything?

There are ways to use your browser to run Python code without having to download anything.

Here's an example:

<https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=2fhs6GZ4qFMx>

The screenshot shows the Google Colaboratory interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help', and a message 'Cannot save changes'. To the right are 'Share', 'Settings', and a user profile icon. Below the bar, there's a toolbar with '+ Code', '+ Text', 'Copy to Drive', and controls for 'RAM' and 'Disk'. On the left, a 'Table of contents' sidebar lists sections like 'Getting started', 'Data science', 'Machine learning', 'More Resources', 'Machine Learning Examples', and 'Section'. The main content area features a large title 'What is Colaboratory?' with a sub-section 'Getting started'. It explains that Colaboratory allows writing and executing Python in a browser with zero configuration, free access to GPUs, and easy sharing. It encourages users to learn more via an 'Introduction to Colab' link. At the bottom, there's a code cell with a Python script:

```
def f(x): print("Hello world")
```

 and a toolbar with up/down arrows, a refresh button, settings, and a trash bin.

DON'T WANT TO DOWNLOAD PYTHON?

Feeling lazy and don't want to download Python or anything?

There are ways to use your browser to run Python code without having to download anything.

Here's an example:

<https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=2fhs6GZ4qFMx>

The screenshot shows the Google Colab interface. On the left, there's a sidebar with a 'Table of contents' section containing links like 'Getting started', 'Data science', 'Machine learning', 'More Resources', 'Machine Learning Examples', and 'Section'. The main area displays a notebook page. A large, semi-transparent thought bubble originates from a dog's head in the foreground. Inside the bubble, the text reads: 'But you don't want to admit to being lazy, do you? 🐶'. The background of the thought bubble is white, and it has a soft shadow. The rest of the Colab interface is visible, including the menu bar ('File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help') and various notebook controls.

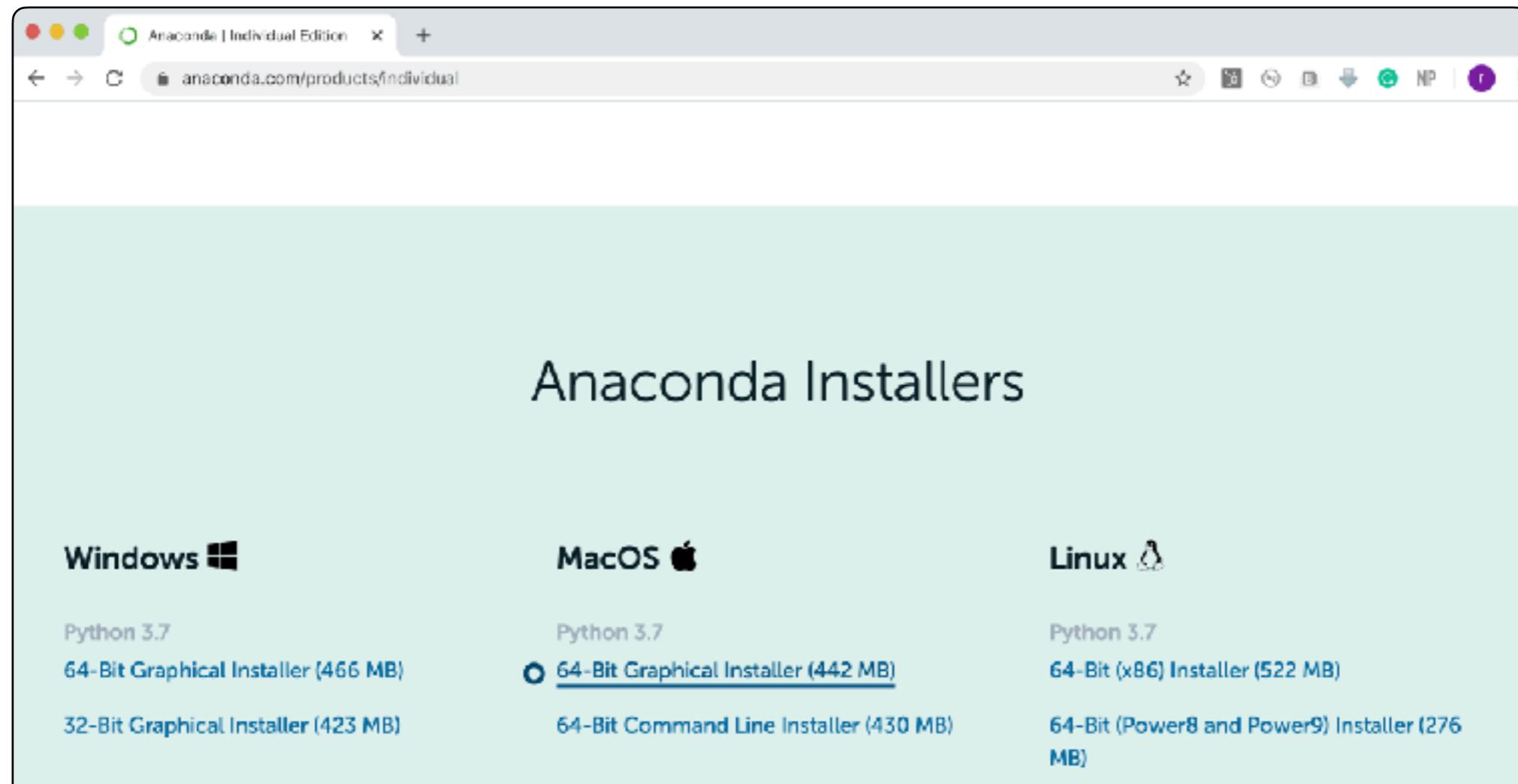
DOWNLOAD PYTHON 3.7 USING ANACONDA

The very first thing you will need to do is download and install Python. We will use the latest version of python, which Python 3.7

The easiest way to do this is to use **Anaconda**, which downloads all the essential libraries.

Here is the page to download Anaconda: <https://www.anaconda.com/products/individual>

Also, there are many tutorials, including through YouTube: <https://www.youtube.com/watch?v=YJC6ldI3hWk>



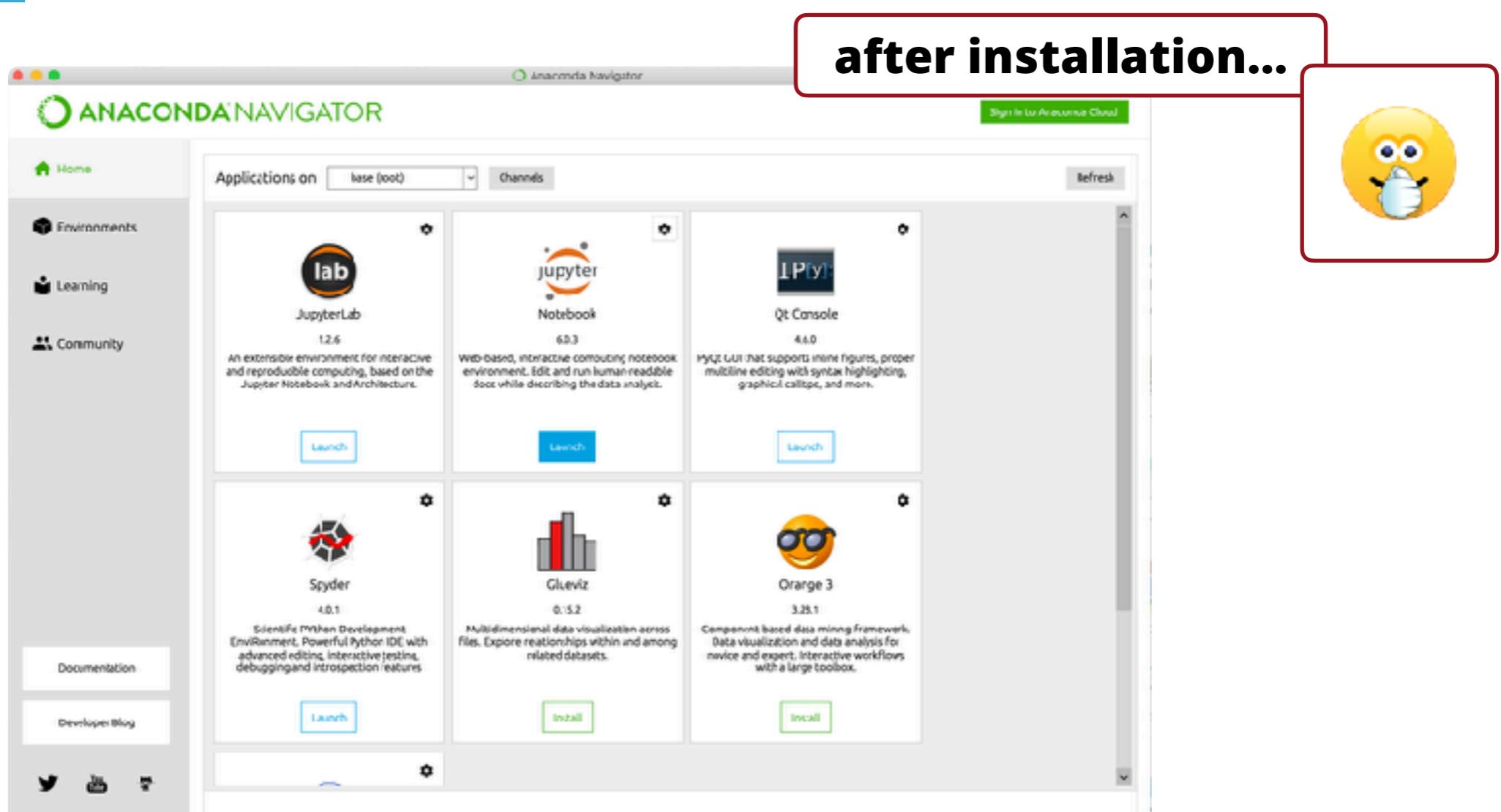
DOWNLOAD PYTHON 3.7 USING ANACONDA

The very first thing you will need to do is download and install Python. We will use the latest version of python, which Python 3.7

The easiest way to do this is to use Anaconda, which downloads all the essential libraries.

Here is the page to download Anaconda: <https://www.anaconda.com/products/individual>

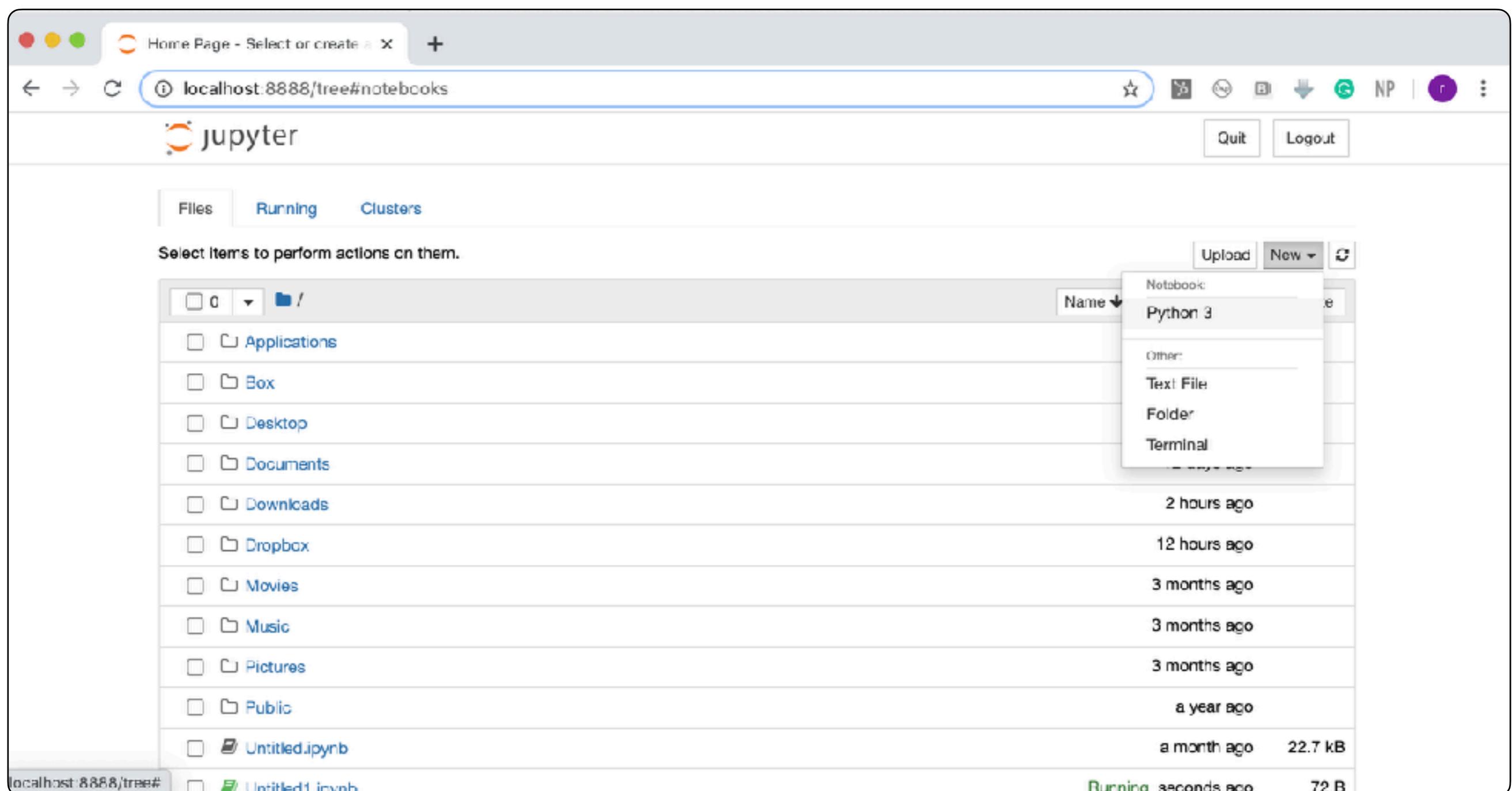
Also, there are many tutorials, including through YouTube: <https://www.youtube.com/watch?v=YJC6ldI3hWk>



JUPYTHON [COMES WITH ANACONDA]

an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

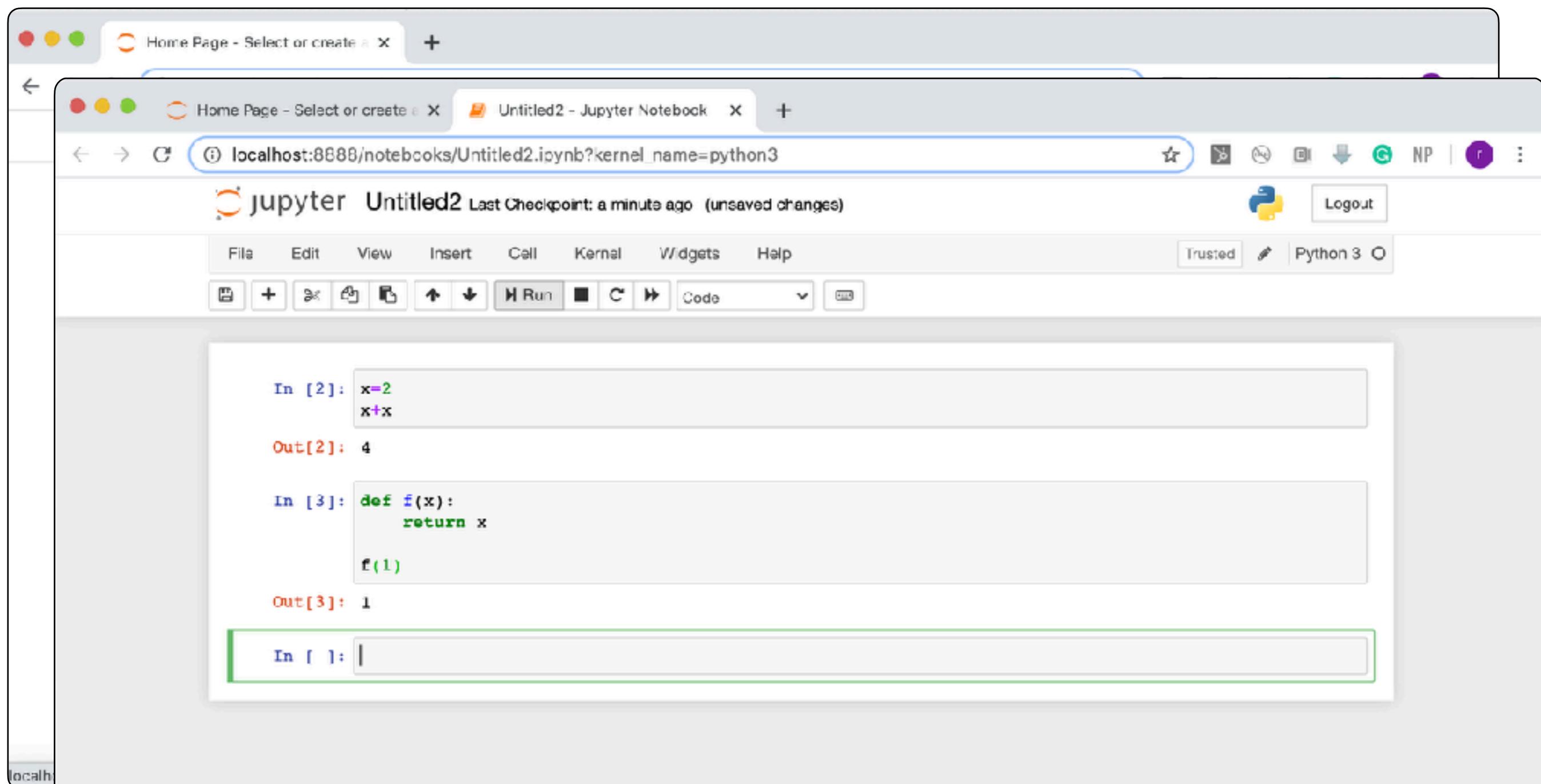
Webpage with detailed description: <https://jupyter.org/>



JUPYTER [COMES WITH ANACONDA]

an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

Webpage with detailed description: <https://jupyter.org/>



THE TERMINAL

One common tool for interfacing with an operating systems and running code is the terminal. The terminal is your friend, do not fear it :)

Below is an image illustrating what the terminal can look like. Also, it shows that I have installed python version 3.7.1

There are many terminal commands that, with a little memorization, can make your coding experience very efficient. There are many tutorials and cheat sheets in the literature with reference to the commands.

The image consists of two side-by-side screenshots. On the left is a screenshot of a Mac OS X terminal window titled 'raulbriceno — bash — 80x24'. The window shows a user's session:

```
Last login: Sun Jun  9 21:15:53 on ttys003
[Rauls-MacBook-Pro:~ raulbriceno$ pwd
/Users/raulbriceno
[Rauls-MacBook-Pro:~ raulbriceno$ which python
/anaconda3/bin/python
[Rauls-MacBook-Pro:~ raulbriceno$ python --version
Python 3.7.1
Rauls-MacBook-Pro:~ raulbriceno$ ]
```

On the right is a screenshot of a Google search results page for 'terminal cheat sheet'. The search bar contains 'terminal cheat sheet'. Below the search bar are navigation links for All, Images, News, Shopping, Maps, More, Settings, and Tools. The main search result is a link to a GitHub gist titled 'Mac Terminal Cheat Sheet · GitHub' with the URL <https://gist.github.com/poopsplat/7195274>. A snippet of the description reads: 'Mac Terminal Cheat Sheet. GitHub Gist: instantly share code, notes, and snippets.' Another result listed is a link to 'A Quick Unix Commands Cheat Sheet to the Unix/Mac Terminal' with the URL <https://learntocodewith.me/command-line/unix-command-cheat-sheet/>. A snippet of the description reads: 'Aug 10, 2018 - That's why I put together a Unix commands cheat sheet of regularly used commands in the Unix command line (see below). ... Instead, my Unix commands cheat sheet has been created to act as a quick reference guide. ... Take a look below at the Unix commands cheat sheet, perfect for your Mac.'

INTERACTIVE PYTHON - IPYTHON [COMES WITH ANACONDA]

IPython: A terminal-based interactive shell for python. For documentation, go to: <https://ipython.readthedocs.io/en/stable/>

Here is an example using IPython via the terminal. Tip: often times, I call ipython with [pylab](#). This imports numpy and matplotlib, two useful libraries for making plots.

```
Rauls-MacBook-Pro:~ raulbriceno$ ipython -pylab
/anaconda3/lib/python3.7/site-packages/IPython/terminal/ipapp.py:299: UserWarning:
`-pylab` flag has been deprecated.
  Use `--matplotlib <backend>` and import pylab manually.
  warnings.warn(`-pylab` flag has been deprecated.\n")
Python 3.7.1 (default, Dec 14 2018, 13:28:58)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.2.0 -- An enhanced Interactive Python. Type '?' for help.
Using matplotlib backend: Qt5Agg

[In [1]: x=arange(10)

[In [2]: x*2
Out[2]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])

[In [3]: x+x
Out[3]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])

In [4]: ]
```

DEMONSTRATIONS

Today we will have three different presentations detailing the presentation of these in three different operating systems:



J. Guerrero, Ph.D.
Postdoctoral fellow - ODU



A. Sturzu
New College of Florida Student



G. Blume
ODU Student

MacOS

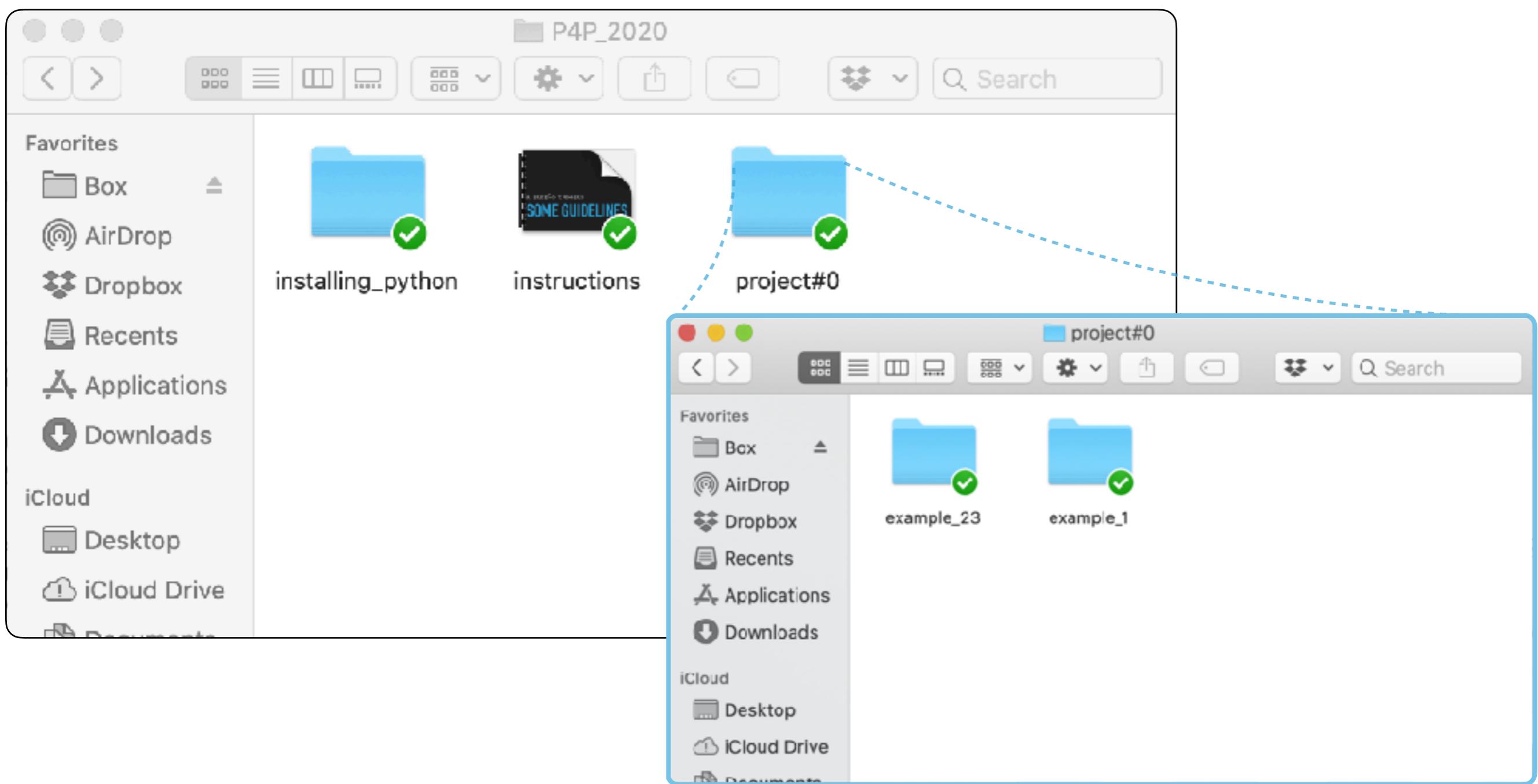
Windows

Linux

DROPBOX

We will make all the material, including these slides available through dropbox. If you do now have a Dropbox account, set one up for free: <https://www.dropbox.com>

Here is the link to the website material: <https://www.dropbox.com/sh/ur6mk8gzl22mq4l/AACRe9R4UlB-4bYAvJG2UI3aa?dl=0>



ASSIGNMENTS

We promise you...WE WILL NOT READ OVER 2k pieces of CODES



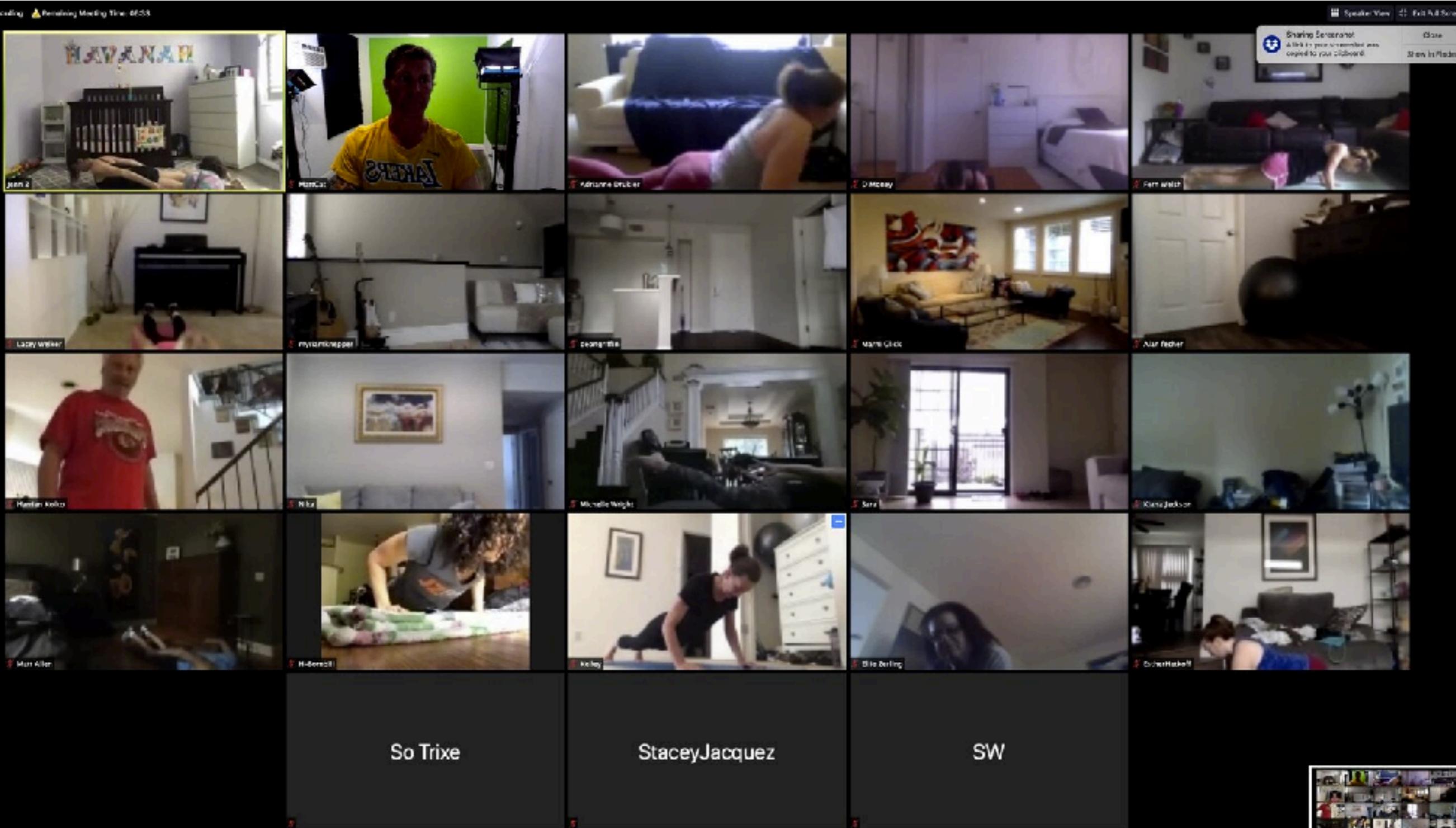
ASSIGNMENTS

We promise you...WE WILL NOT READ OVER 2k pieces of CODES
Instead, think of us as your “zoom workout coach” for programming!



Recording Pending Meeting Time: 6:55

Sharing Document: A link to your submitted assignment has been copied to your clipboard.



Speaker View | Exit Full Screen

Sharing Document: A link to your submitted assignment has been copied to your clipboard.

Names visible in the video feeds:

- Jeanie
- MUSICA
- Adrienne Brubaker
- D'Mossey
- Patti White
- Lacey Walker
- myronknapp
- ceonamita
- Mary Glick
- Alan Fischer
- Hieran Kelso
- Nika
- Michelle Wright
- Sara
- Kiana Jackson
- Matt Allen
- H-Bornell
- Kelley
- Brie Berlin
- Schaeffer

So Trixe

StaceyJacquez

SW

Speaker View | Exit Full Screen

Sharing Document: A link to your submitted assignment has been copied to your clipboard.

EXAMPLE 1

The first example is in the following folder [Dropbox: P4P_2020/project#0/example_1](#)

Note, “/” refers to a subfolder of a file. For example, here **P4P_2020** is main folder. Within this, there is a folder called **project#0**, which itself includes another folder **example_1**.

The example is [example_plotting.py](#)

Look at this code in your preferred choice of the text editor. There, you will see a detailed description of the different pieces of the code and what they do. This makes of two libraries we will use quite a bit, numpy (<https://www.numpy.org/>) and matplotlib (<https://matplotlib.org/>). If there is anything you don’t understand, google it! 🐕

Here is an illustration of this code being run in the terminal and the figure that is produced, which is saved as [example_plot.pdf](#)

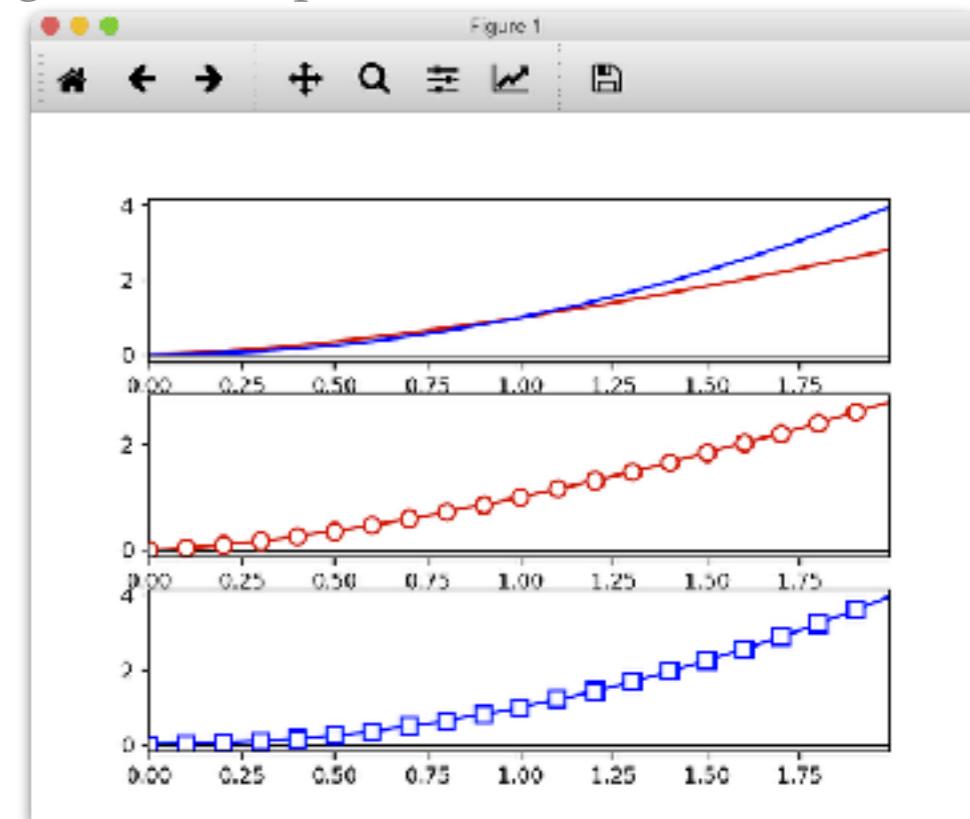
```
Last login: Sun Jun  5 21:56:29 on ttys003
Rauls-MacBook-Pro:~ raulbriceno$ cd ~/Dropbox/python4physics/project#0/example_1/
Rauls-MacBook-Pro:example_1 raulbriceno$ ipython -pylab
~/anaconda3/lib/python3.7/site-packages/IPython/terminal/ipapp.py:239: UserWarning: 
deprecationwarning.
  Use ``--matplotlib checkdeps`` and import pylab manually.
  warnings.warn("``-pylab`` flag has been deprecated.\n")
Python 3.7.1 (default, Dec 14 2018, 13:20:56)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.2.0 -- An enhanced Interactive Python. Type '?' for help.

Using matplotlib backend: qt5agg

In [1]: import example_plotting
*****
x = [0.  0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1  0.11 0.12 0.13
     0.14 0.15 0.16 0.17 0.18 0.19 0.2  0.21 0.22 0.23 0.24 0.25 0.26 0.27
     0.28 0.29 0.3  0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4  0.41
     0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5  0.51 0.52 0.53 0.54 0.55
     0.56 0.57 0.58 0.59 0.6  0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69
     0.7  0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8  0.81 0.82 0.83
     0.84 0.85 0.86 0.87 0.88 0.89 0.9  0.91 0.92 0.93 0.94 0.95 0.96 0.97
     0.98 0.99 1.  1.01 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09 1.1  1.11
     1.12 1.13 1.14 1.15 1.16 1.17 1.18 1.19 1.2  1.21 1.22 1.23 1.24 1.25
     1.26 1.27 1.28 1.29 1.3  1.31 1.32 1.33 1.34 1.35 1.36 1.37 1.38 1.39
     1.4  1.41 1.42 1.43 1.44 1.45 1.46 1.47 1.48 1.49 1.5  1.51 1.52 1.53
     1.54 1.55 1.56 1.57 1.58 1.59 1.6  1.61 1.62 1.63 1.64 1.65 1.66 1.67
     1.68 1.69 1.7  1.71 1.72 1.73 1.74 1.75 1.76 1.77 1.78 1.79 1.8  1.81
     1.82 1.83 1.84 1.85 1.86 1.87 1.88 1.89 1.9  1.91 1.92 1.93 1.94 1.95
     1.96 1.97 1.98 1.99]

*****
x1 = [0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7
     1.8 1.99]

In [2]:
```



EXAMPLE 2 & 3

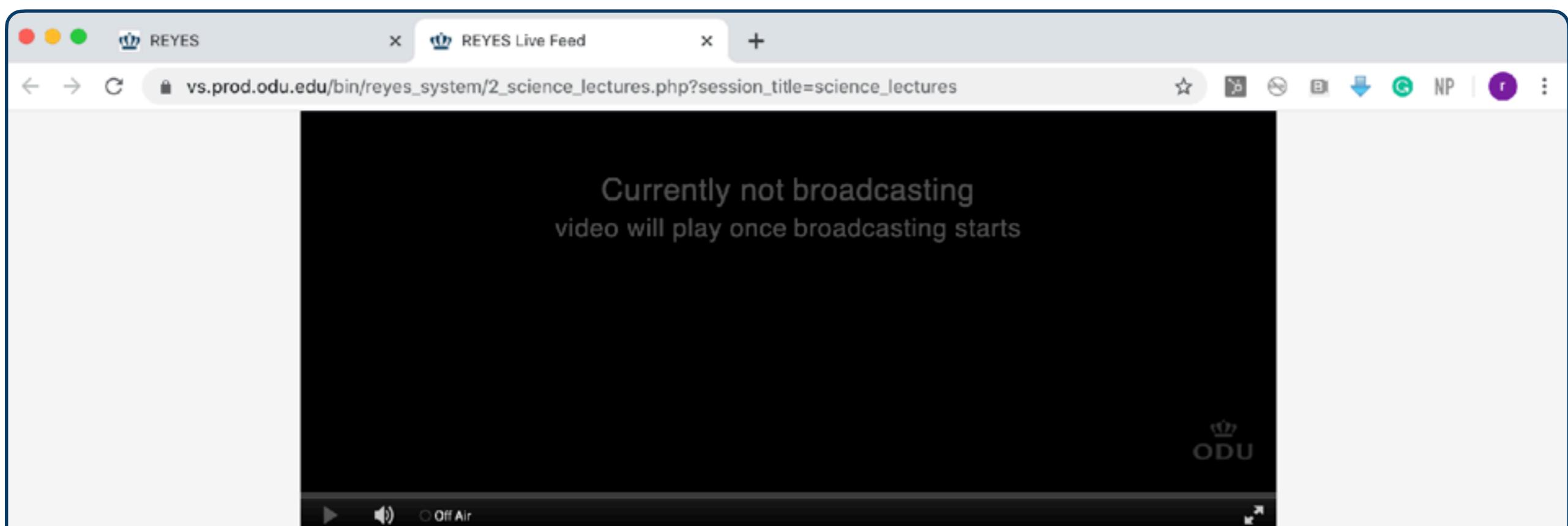
The second and third examples are in the following folder [Dropbox: P4P_2020/project#0/example_23](#)

Again, look at [example#2_Gauss.py](#). Understand it line by line, make sure to understand it. Run it like you did before. This will generate a figure, which is then saved as [example#2_Gauss.pdf](#) and a text file, named [data.txt](#). The text file is a string of number, which are visualized in the figure.

Run the final example, [example#3_loading_data.py](#). This will load the text file and generate a new figure, [example_Gauss_check.pdf](#).

Also, you can copy and paste the code into Jupyter!

VIDEO STREAMS // VS.PROD.ODU.EDU/BIN/REYES_SYSTEM/



Currently not broadcasting
video will play once broadcasting starts

ODU

▶ 🔊 Off Air

Live Questions or Comments? | **Event Information** | **Need Help?** | **Your Feedback?**

Feedback

Thank you for attending this event! We would appreciate 5-minutes of your time to complete a short survey and let us know your thoughts!

Please [click here](#) to provide your feedback.

We want to hear your thoughts!



VIDEO STREAMS // VS.PROD.ODU.EDU/BIN/REYES_SYSTEM/

REYES REYES Live Feed vs.prod.odu.edu/bin/reyes_system/2_science_lectures.php?session_title=science_lectures NP r :)

Currently not broadcasting
video will play once broadcasting starts

OLD DOMINION UNIVERSITY IDEA FUSION

Remote Experience for Young Engineers & Scientists (REYES)

Thank you for attending this event! We would appreciate 5-minutes of your time to complete a short survey and let us know your thoughts!

What is your gender?

Female
 Male

Fee My ethnicity is:

Asian or Asian American, including Chinese, Japanese, and others
 Black or African American
 Black Haitian and Caribbean Islander

Thank you for attending this event! We would appreciate 5-minutes of your time to complete a short survey and let us know your thoughts!

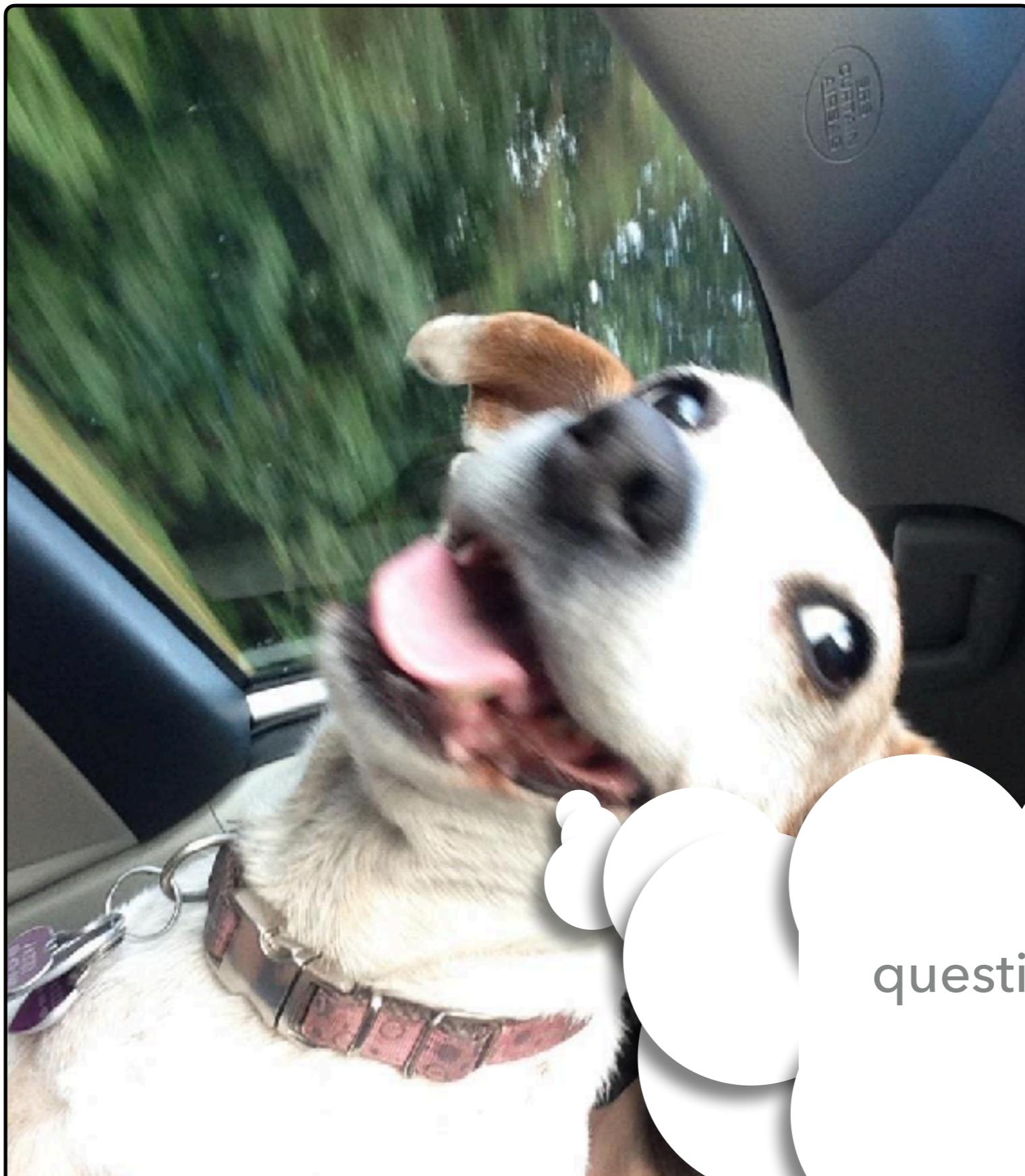
Please [click here](#) to provide your feedback.

We want to hear your thoughts!

Your Feedback?



QUESTIONS?



questions? 🐶

ONTO THE DEMONSTRATIONS

Today we will have three different presentations detailing the presentation of these in three different operating systems:



J. Guerrero, Ph.D.
Postdoctoral fellow - ODU



A. Sturzu
New College of Florida Student



G. Blume
ODU Student

MacOS 

Windows 

Linux 

NEXT EVENT: A PANEL ON “DISCOVER SCIENCE”

REYES

vs.prod.edu/bin/reyes_system/sessions.php

Live Video Stream

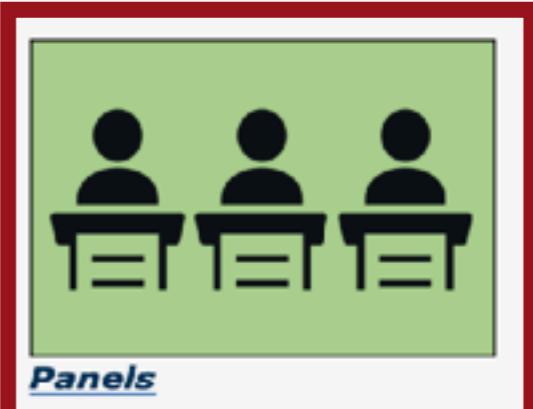
 **REYES: Remote Experience for Young Engineers and Scientists**

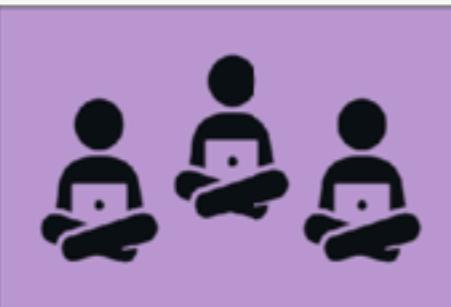
Please select a live session:


[Activity Link](#)


[Science Sessions](#)


[Engineering Sessions](#)


[Panels](#)


[Social Events](#)


[Python4Physics](#)


[Engineering Fundamentals](#)

NEXT EVENT: A PANEL ON “DISCOVER SCIENCE”

Week 1					
Time	Monday, June 22, 2020	Tuesday, June 23, 2020	Wednesday, June 24, 2020	Thursday, June 25, 2020	Friday, June 26, 2020
1:00 PM	1:00pm - 2:00pm Welcoming from the Deans		1:00pm - 2:00pm What is Computational, Modeling and Simulation?		1:00pm - 1:30pm Guest Lecture -What are the new opportunities at NASA?
1:30 PM		1:00pm - 3:00pm Python 4 Physics Course		1:00pm - 3:00pm Python 4 Physics Course	
2:00 PM	2:00pm - 3:00pm Exploring Artificial Intelligence				
2:30 PM					
3:00 PM	2:00pm - 3:00pm Implantable and wearable biosensors				
3:30 PM					
4:00 PM					
4:30 PM					
5:00 PM					
5:30 PM					
6:00 PM					

3:00pm - 4:00pm
Discover Science

Faculty in the College of Science
“why study science?”