



Welcome!

Artificial Intelligence/Machine Learning Bootcamp

August 5-16, 2024



Agenda

- Welcome (logistics and expectations)
- Overview of DOE and BNL
- Introduce OEP
- Introduce CSI
- Motivation for Bootcamp
- Intro to Python

This Week's Schedule at a Glance

Intro to Python Modules (Monday through Wednesday)

- Introducing Scientific Computing
- Python Fundamentals
- Graphing
- Functions and Logic
- Probability and Statistics
- Random Numbers and Algorithms
- Capstone Project – Make a Game!

This Week's Schedule at a Glance (cont.)

Thursday

- **Internship Day!** *Meet current interns and learn about their experiences in the SULI and CCI programs at BNL*

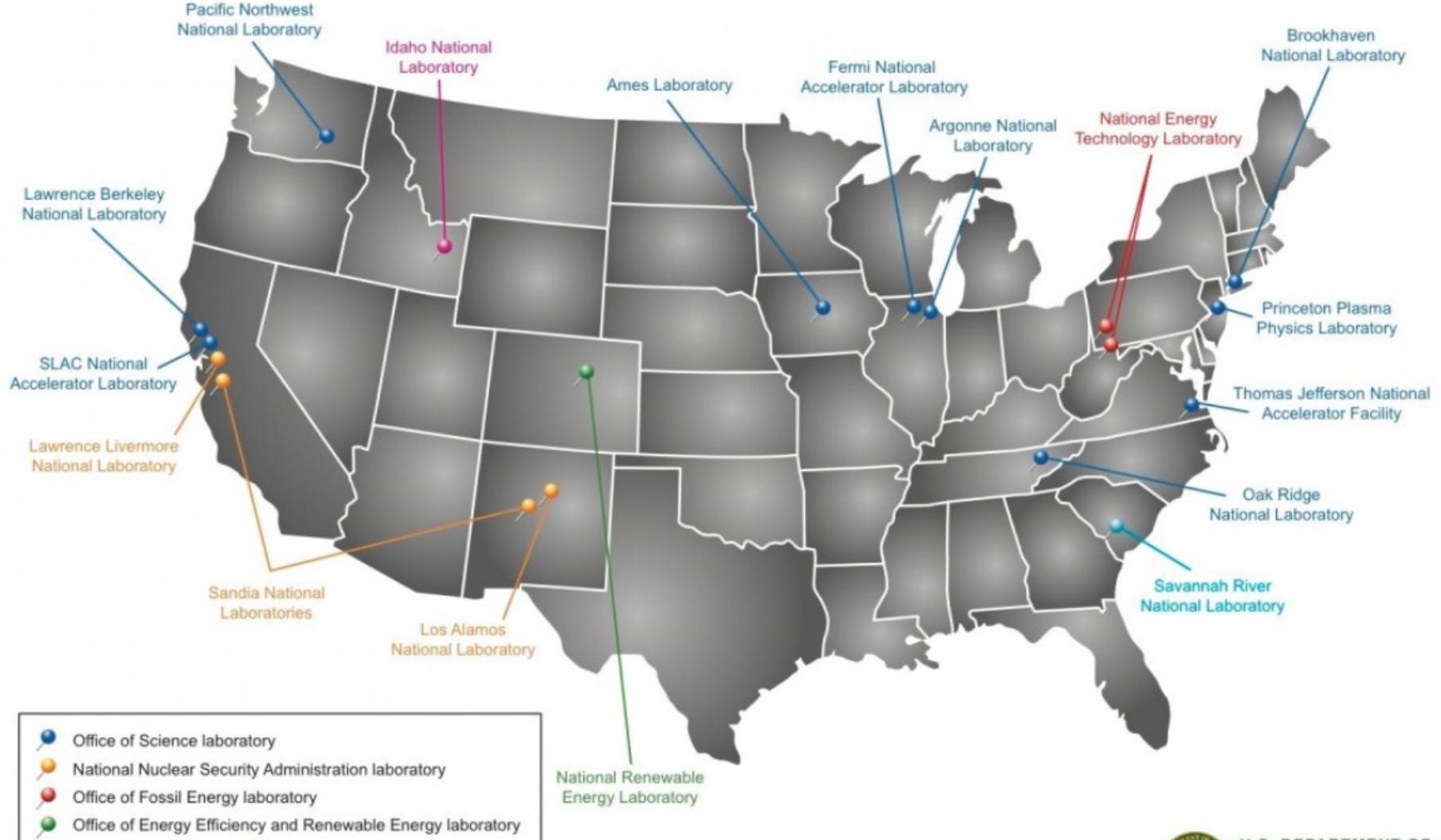
Friday

- The Machine Learning Begins...

Logistics

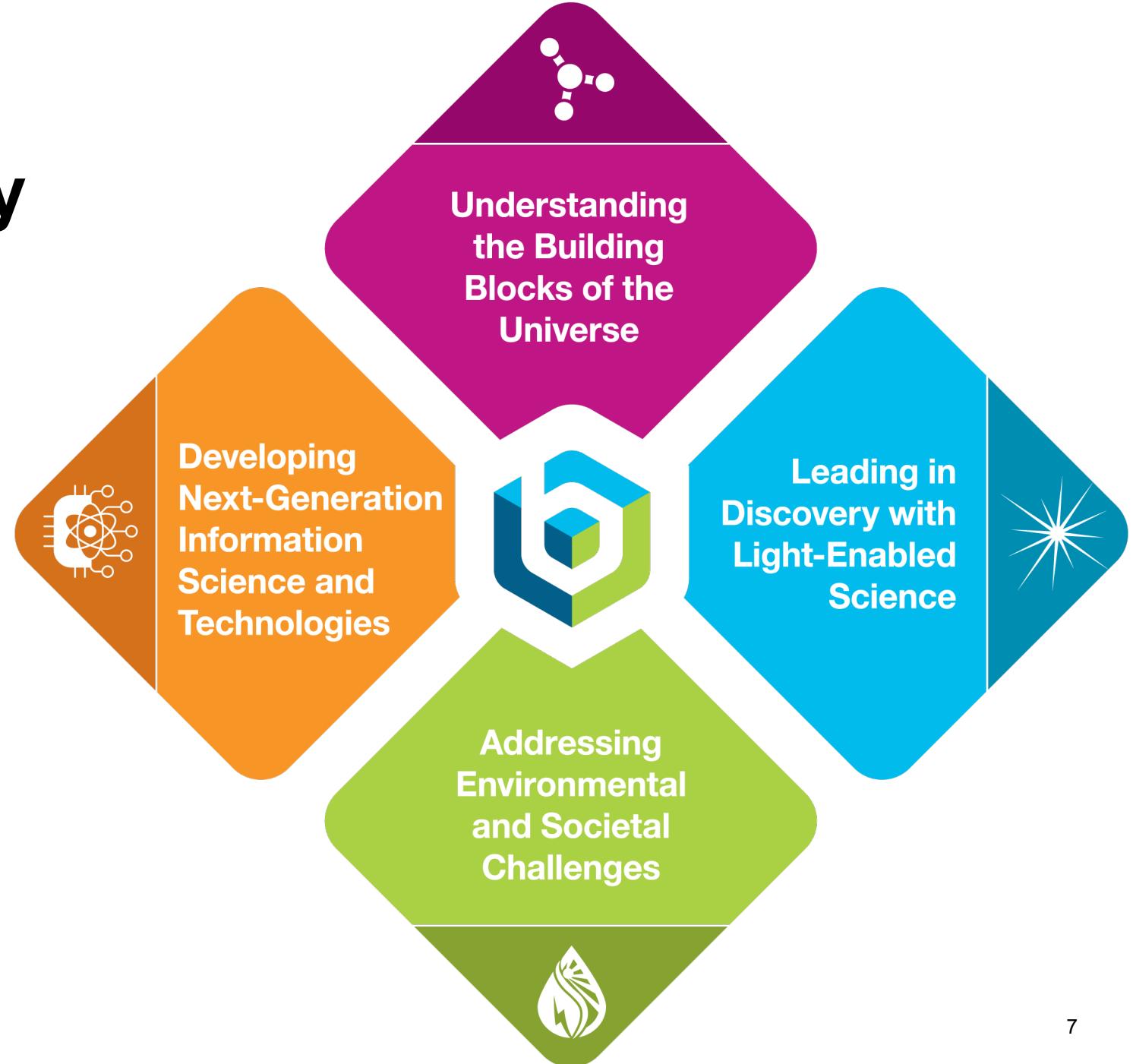
- Drivers, upon check-in will receive a placard to be displayed on the dashboard for the week.
- Adhere to speed limit postings, 30mph
- Building 438 Classroom – home base
- Attire: **must** wear long pants, flat closed-toe shoes and suggest you bring additional layers
- Personal devices (cell phones and laptops) are allowed
- BYO lunch or a few options available for purchase
 - Lunch break is typically from about 12:30pm to about 1:30pm
- Water refill stations located in building 438 and cafeteria

Department of Energy National Laboratories



Brookhaven National Laboratory

Our Vision: To accelerate pathways to scientific discovery and technological innovation that transform the world

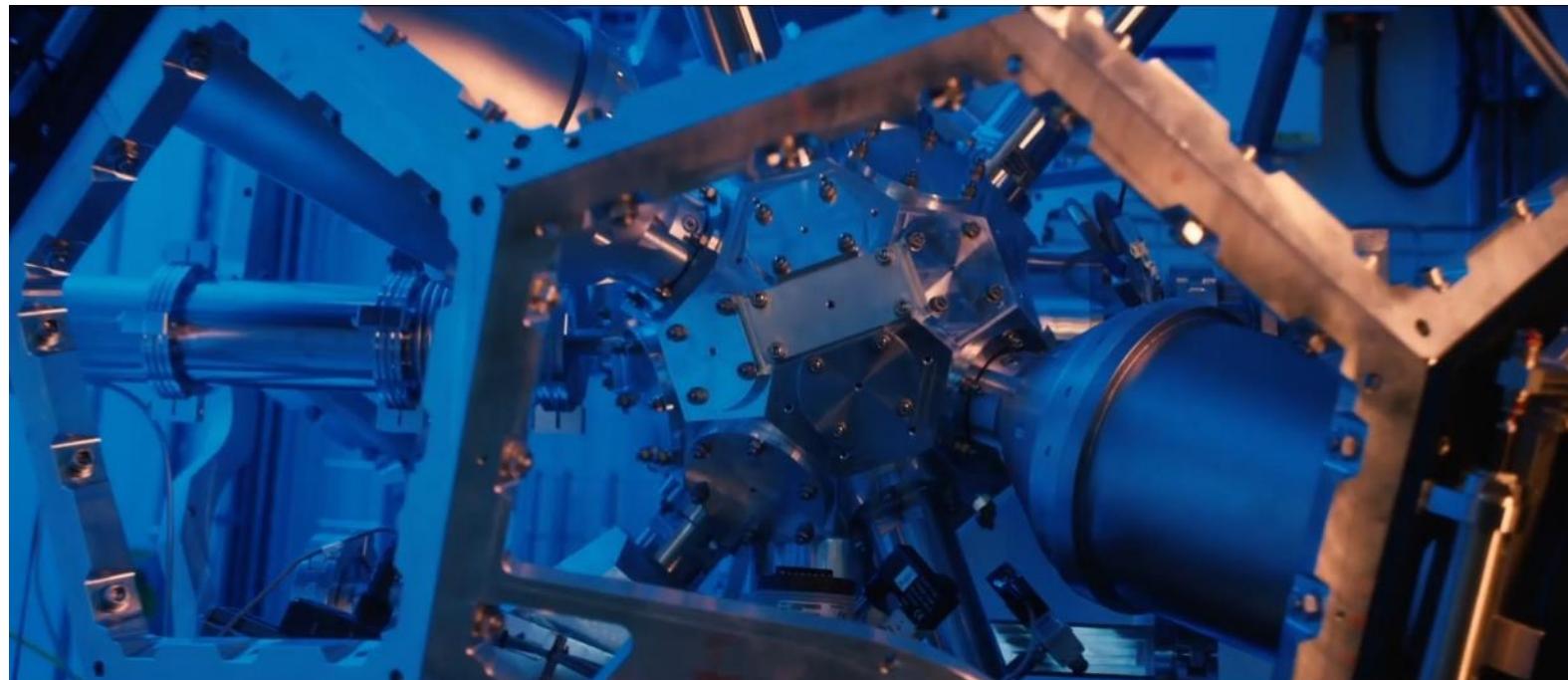


Brookhaven National Laboratory

Our combination of facilities, people, and equipment gives us a unique and vital role in addressing national needs.



About Brookhaven National Laboratory



[YouTube Video - This is Brookhaven Lab](#)

Preparing New Generations of STEM Professionals

Office of Educational Programs

Host ~30,000 students, interns, teachers
and college faculty

Pre-College Programs

- Field trips
- Summer Programs
- Contests
- Teacher PD



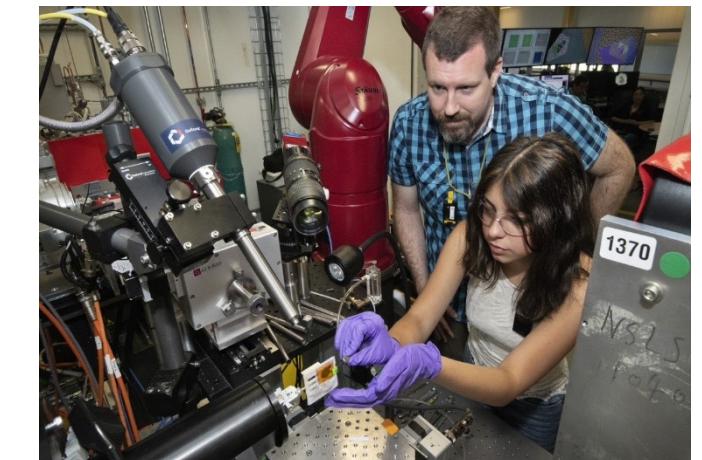
Preparing New Generations of STEM Professionals

Office of Educational Programs

Host ~30,000 students, interns, teachers
and college faculty

Pre-College Programs

- Open Space Stewardship
- High School Research Program
- SPARK



Preparing New Generations of STEM Professionals

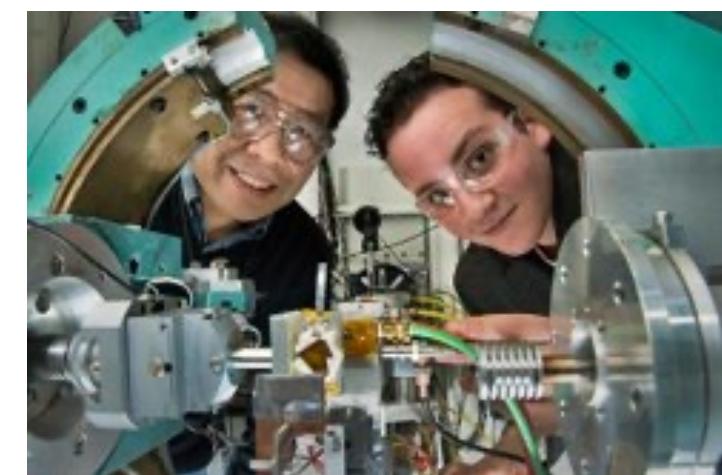
Office of Educational Programs

Host ~30,000 students, interns, teachers
and college faculty

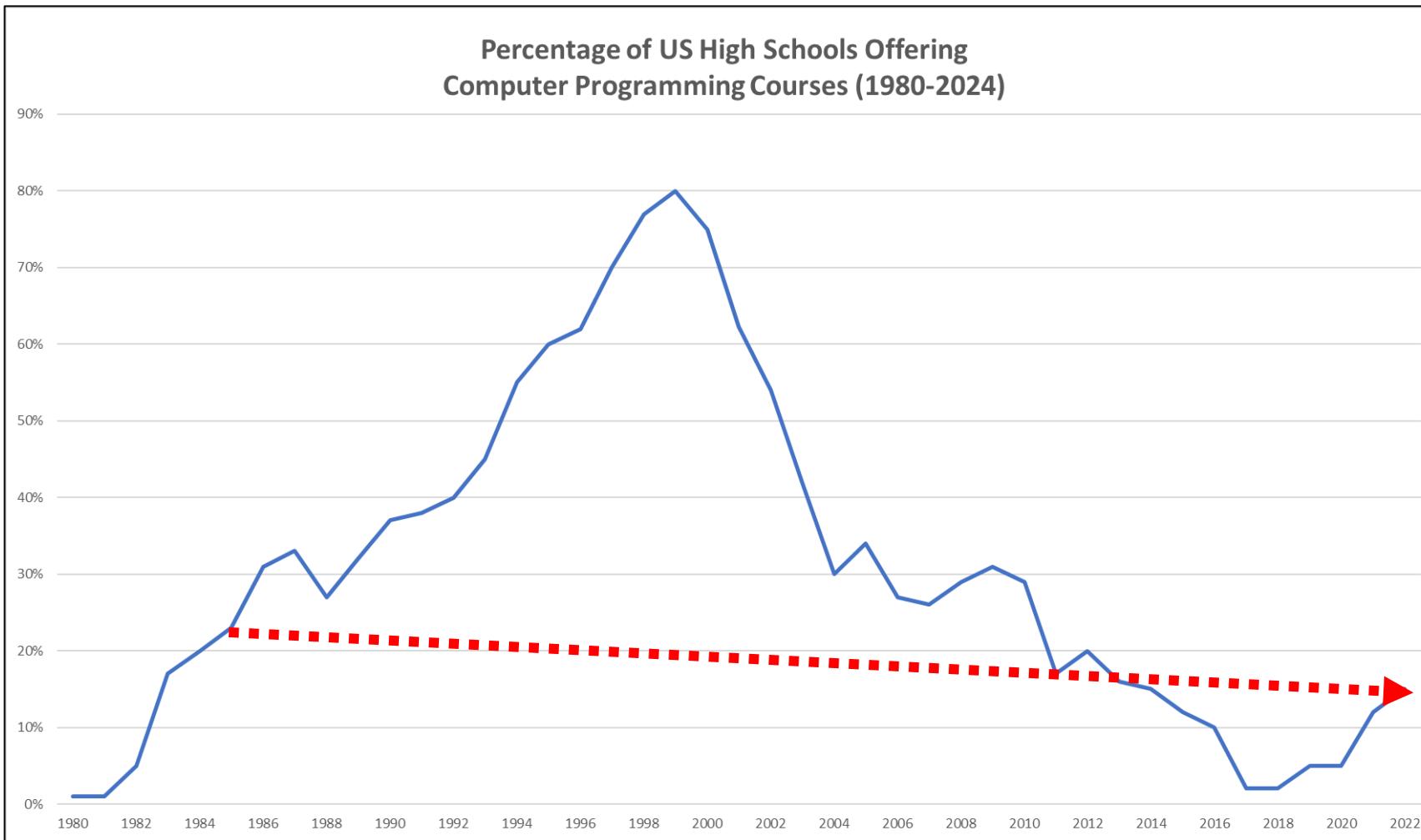
University and Faculty Programs

- College Internship Programs
- Visiting Faculty Program

**There are research opportunities at all
the DOE National Labs!**



A National Challenge



The Challenge to DOE

Scientists across DOE have noted a dramatic rise in the **startup latency** of interns who do not possess foundational programming skills

This latency means that for the ***initial 6-8 weeks of their assignment***, the interns are essentially unproductive as they must first acquire basic knowledge of how to write software

Instead of learning critical scientific principles from their mentor, the interns are spending their precious lab time **often working alone**, figuring out how to instruct the computer to perform rudimentary data processing tasks

From the perspective of DOE scientists, **the need to adeptly wield software tools has never been as urgent as it is today**

The Hype

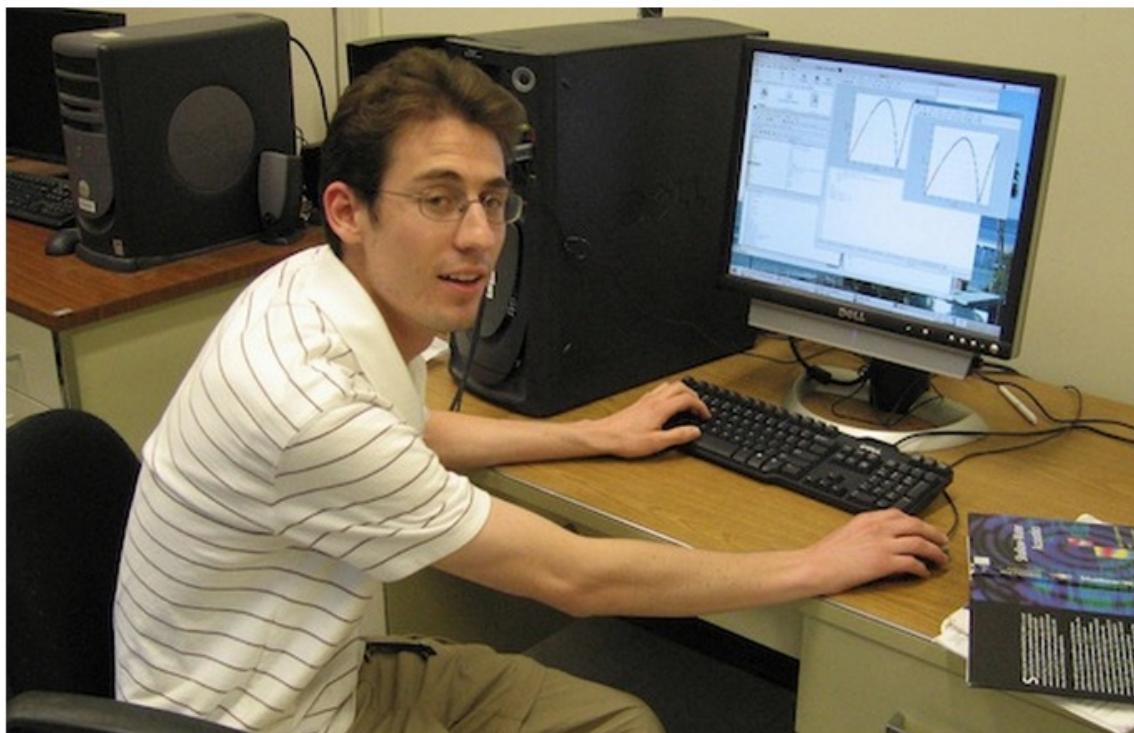
Kevin's Story

Modern-day scientists and engineers are spending more and more of their work days in front of the computer. As an example, consider my friend Kevin, who works in oceanography and mechanical engineering. Whoa, sounds like he's probably spending all day out on high-tech boats rigging together mechanical devices like MacGyver and collecting data from underwater sensors, right? This must be his typical work day -- hard hats and heavy-duty work gloves.



The Reality

Actually, Kevin spends less than 5% of his time out on the ocean; the other 95% of the time, he's sitting in front of the computer writing programs to clean up, transform, process, and extract insights from data collected out in the field. This is what Kevin looks like at work on most days:



The same story plays out for scientists and engineers in all sorts of fields: astronomers, biologists, physicists, aerospace engineers, economists, geneticists, ecologists, environmental engineers, neuroscientists ... the list goes on and on. Modern-day science and engineering is all about processing, analyzing, and extracting insights from data.

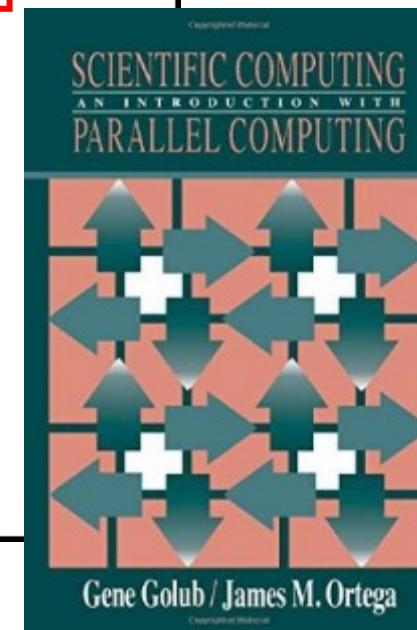
What is Scientific Computing?

Golub and Ortega: “*Scientific computing is the collection of tools, techniques and theories required to solve on a computer mathematical models of problems in science and engineering.*”

Or a more narrow definition: “*Development and use of numerical methods and mathematical models to solve real-world problems efficiently on computers.*”

Interdisciplinary field requiring:

- knowledge about the underlying (physical) problem,
- ability to formulate a mathematical model,
- stable & accurate numerical schemes,
- efficient implementation on high performance computers.



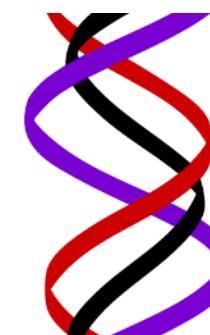
What is Scientific Computing?

Scientific computing problems **cannot be solved** using just a graphing calculator or a spreadsheet program

- A computer should not be viewed as just another closed-form benchtop instrument with fixed functionality
- SciComp does not require writing thousands of lines of code to answer problems – complete code usually fits on **one** slide!

SciComp is **applied** computer science

- The first name of CompSci is *computer*
- The first name of SciComp is science
- A **triple helix** of math, science, and computation



SciComp vs CompSci

Scientific Computing

- Probability and Statistics
- Simulation and Modelling
- Data Visualization
- Storing and Analyzing Very Large Datasets
- Parallel & Distributed Algorithms
- Speed and Accuracy Paramount
- Functional Languages
- Open-Ended Problems with Unknown Solutions

Computer Science

- General Data Structures
- Design Methodologies
- Procedural Languages
- Stand-Alone Programs
- Emphasis on Object-Orientation
- Simple Data Models
- Sequential Algorithms
- Less Graphics Intensive
- Directed Closed-Form Problems with Known Solutions

SciComp = Just Enough CompSci

Data types: int, float, bool, string

Data structures: lists, arrays, dictionaries, classes, modules

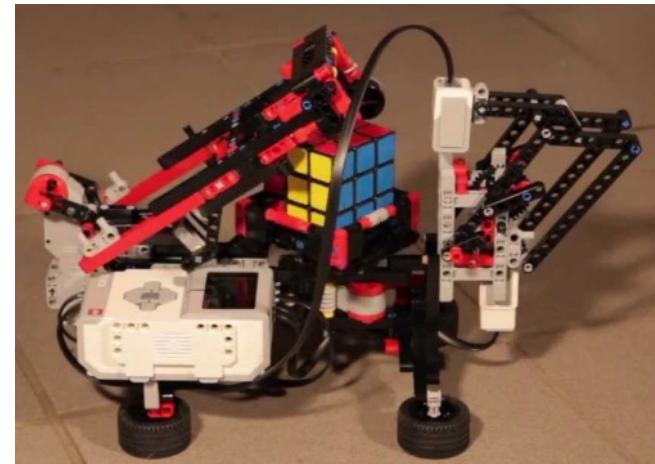
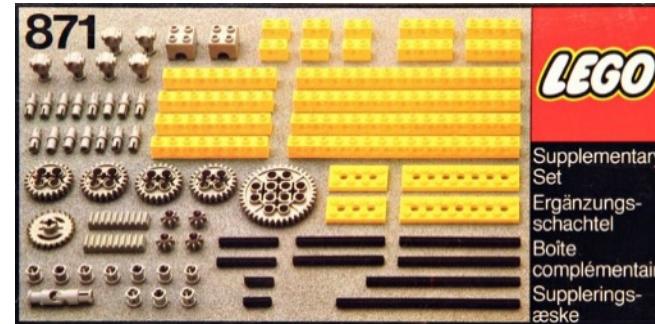
Functions: def, return

Statements: if, for, while, break

Patterns: vectorization, divide & conquer, list comprehensions

Algorithms: GCD, mean/variance, sorting, searching

Modules: numpy, matplotlib, numba, scipy, sympy, scikit-learn



SciComp As Translational Science

"The algorithm that almost won the cold war"

$$u(\mathbf{x}) = \begin{cases} \frac{\sum_{i=1}^N w_i(\mathbf{x}) u_i}{\sum_{i=1}^N w_i(\mathbf{x})}, & \text{if } d(\mathbf{x}, \mathbf{x}_i) \neq 0 \text{ for all } i \\ u_i, & \text{if } d(\mathbf{x}, \mathbf{x}_i) = 0 \text{ for some } i \end{cases}$$

where

$$w_i(\mathbf{x}) = \frac{1}{d(\mathbf{x}, \mathbf{x}_i)^p}$$

```
def calc_idw_height(xi, yi, p):
    sum_weight = 0.0
    sum_height_weight = 0.0
    for si in range(num_samples):
        distance = np.hypot(grid_x[xi], xi - samples_x[si],
                             grid_y[yi], yi - samples_y[si])
        if distance == 0:
            return samples_z[si]
        weight = 1 / np.power(distance, p)
        sum_weight += weight
        sum_height_weight += samples_z[si] * weight
    return sum_height_weight / sum_weight
```

11 lines of code can change the world!

SciComp is the ability to translate mathematical expressions of scientific concepts into correct and efficient software code

US Department of Labor - 2019

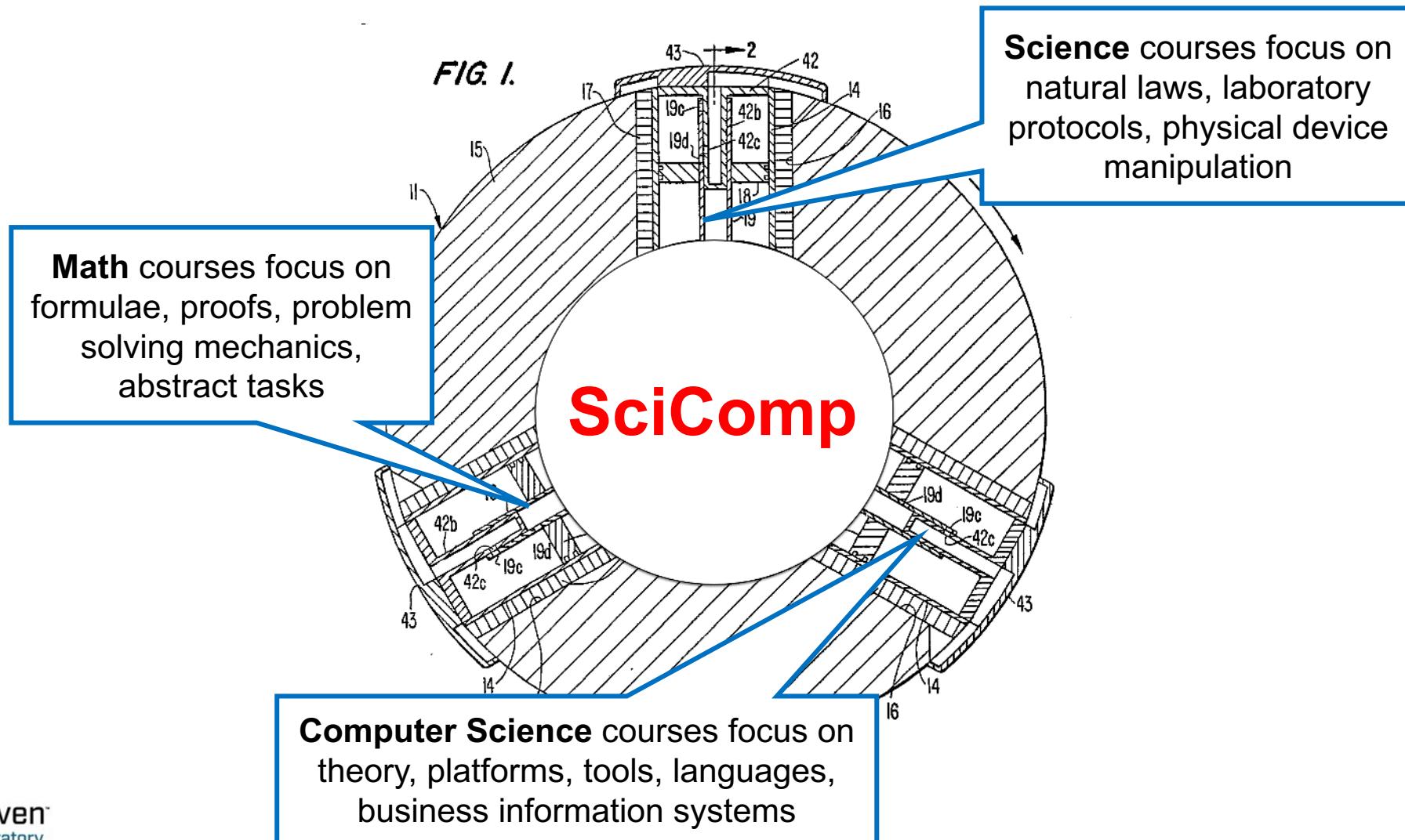
Robotics Engineering Careers

Quick Facts: Electro-mechanical Technicians	
2018 Median Pay ?	\$57,790 per year \$27.78 per hour
Typical Entry-Level Education ?	Associate's degree
Work Experience in a Related Occupation ?	None
On-the-job Training ?	None
Number of Jobs, 2016 ?	13,800
Job Outlook, 2016-26 ?	4% (Slower than average)
Employment Change, 2016-26 ?	500

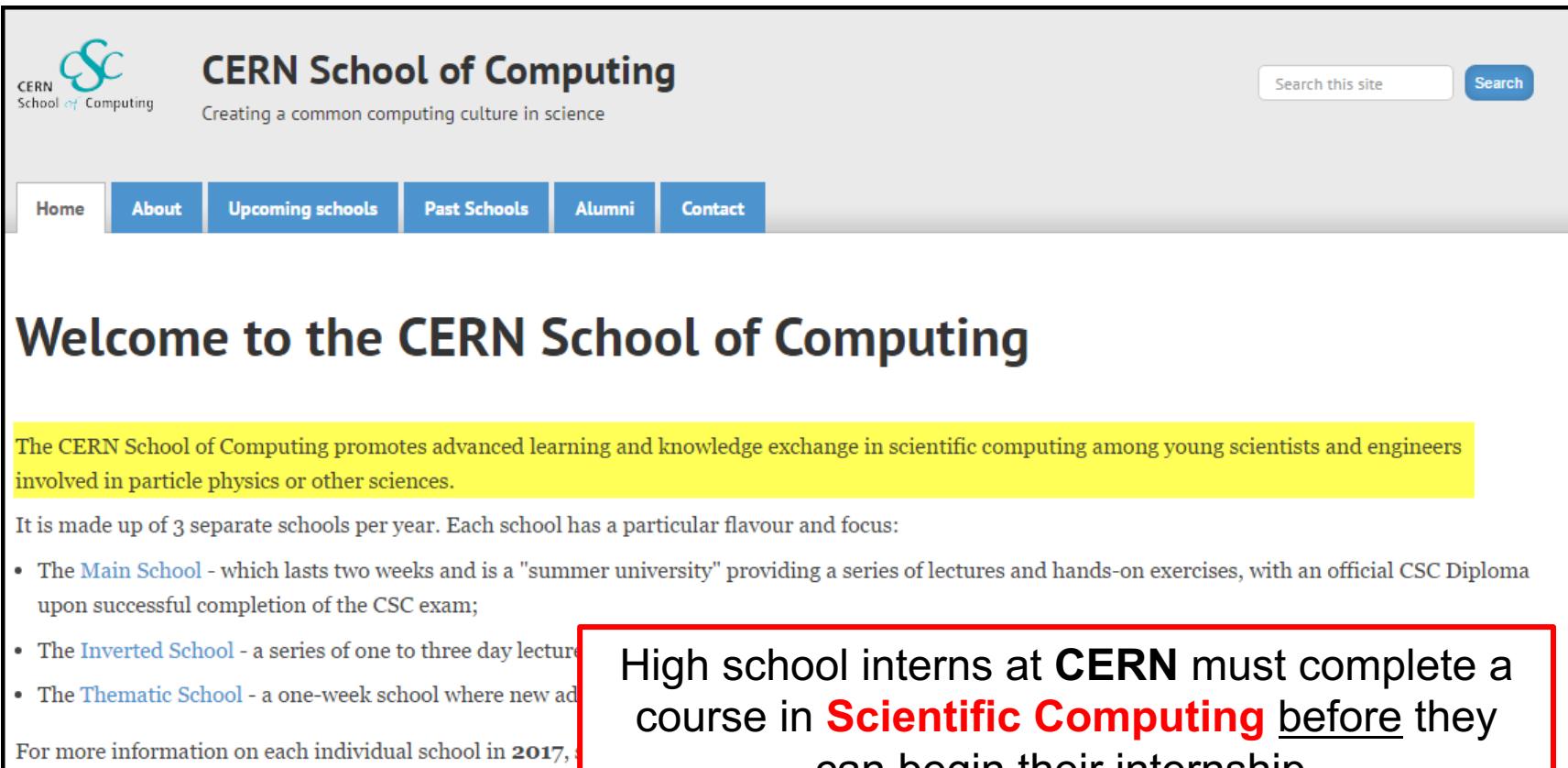
Scientific Computing Careers

Quick Facts: Computer and Information Research Scientists	
2018 Median Pay ?	\$118,370 per year \$56.91 per hour
Typical Entry-Level Education ?	Master's degree
Work Experience in a Related Occupation ?	None
On-the-job Training ?	None
Number of Jobs, 2016 ?	27,900
Job Outlook, 2016-26 ?	19% (Much faster than average)
Employment Change, 2016-26 ?	5,400

SciComp Is Getting Lost in the Middle



SciComp = The Pathway to Internships



The screenshot shows the CERN School of Computing website. At the top, there is a logo for 'CERN School of Computing' with the tagline 'Creating a common computing culture in science'. Below the logo is a navigation bar with links for Home, About, Upcoming schools, Past Schools, Alumni, and Contact. A search bar is located in the top right corner. The main heading 'Welcome to the CERN School of Computing' is displayed prominently. Below the heading, a yellow box contains the text: 'The CERN School of Computing promotes advanced learning and knowledge exchange in scientific computing among young scientists and engineers involved in particle physics or other sciences.' To the right of this text, a bulleted list describes the three types of schools: Main School, Inverted School, and Thematic School. At the bottom of the page, there is a link for more information on individual schools in 2017.

High school interns at CERN must complete a course in Scientific Computing before they can begin their internship.

That is how important CERN views these skills!

Scientific Computing with Python

Python is quickly becoming one of the **most heavily used languages** in science projects

Python runs on all major modern **operating systems** and is completely free and open-source (not vendor controlled)

Python makes it easy for your code to directly integrate with a large spectrum of available 3rd party software

Python code runs **consistently** on different platforms and scales well from small IoT devices to large server clusters

Python benefits from a very active and growing user community that continues to enhance the language

Useful Python Reference Sites

<https://colab.google>

<https://docs.python.org/3/tutorial>

<https://www.learnpython.org>

<https://realpython.com>

<https://www.w3schools.com/python>

<https://www.fullstackpython.com>

Bookmark
It!

Access the Courseware – Step 1

Go to this link: <https://github.com/matthewcarbone/Bootcamp>

Your web browser should then display this page:

The screenshot shows a GitHub repository page for 'matthewcarbone / Bootcamp'. The repository is public and contains 42 commits. The README file is the current view. The repository has 0 stars, 2 forks, and 2 watchers. It includes files like LICENSE, .gitignore, and Modules.

Code | Issues 7 | Pull requests | Actions | Projects | Wiki | Security | Insights

matthewcarbone / Bootcamp

Type ⌘ to search

Watch 2 | Fork 0 | Star 0

Bootcamp Public

master 1 Branch 0 Tags

Go to file Add file Code About

matthewcarbone Update README.md · 30f4385 · 16 hours ago 42 Commits

Modules · Update README.md · 16 hours ago

.gitignore · Create background module · 3 days ago

LICENSE · Update LICENSE · 3 weeks ago

README.md · Update README.md · 17 hours ago

README · BSD-3-Clause license

No description, website, or topics provided.

Readme · BSD-3-Clause license · Activity · 0 stars · 2 watching · 0 forks · Report repository

Releases · No releases published · Create a new release

Packages

Get the Courseware – Step 2

The screenshot shows a GitHub repository page for the 'Bootcamp' repository, which is public and owned by 'matthewcarbone'. The repository has 1 branch and 0 tags. The master branch has 42 commits. A commit by 'matthewcarbone' titled 'Update README.md' is highlighted, showing changes to 'Modules', 'README.md', '.gitignore', and 'LICENSE'. The 'Code' tab is selected. The repository has 0 stars, 2 forks, and 0 issues. It also has 0 releases and 0 packages.

matthewcarbone / Bootcamp

Type ⌘ to search

Code Issues 7 Pull requests Actions Projects Wiki Security Insights

Bootcamp Public

Watch 2 Fork 0 Star 0

master 1 Branch 0 Tags

Go to file Add file Code About

matthewcarbone Update README.md 30f4385 · 16 hours ago 42 Commits

Modules

.gitignore

LICENSE

README.md

Update README.md 16 hours ago

Create background module 3 days ago

Update LICENSE 3 weeks ago

Update README.md 17 hours ago

Readme

BSD-3-Clause license

Activity

0 stars

2 watching

0 forks

Report repository

Releases

No releases published Create a new release

Packages

Bootcamp

Get the Courseware – Step 3

The screenshot shows a GitHub repository named 'matthewcarbone / Bootcamp'. The 'Code' tab is selected. On the left, the file tree shows a 'Modules' folder containing '00_Background' and '01_Introduction_to_Python'. The '01_Introduction_to_Python' folder is circled in blue. On the right, a list of files and their commits is shown. A large blue callout bubble points to the '01_Introduction_to_Python' folder in the file tree.

This contains information about BNL and the Department of Energy National Lab Complex

Name	Message	Last commit date
...		
00_Background	Update README.md	3 days ago
01_Introduction_to_Python	Update README.md	17 hours ago
02_Tabular_Data	Update README.md	19 hours ago
03_Introduction_to_Machine_Learning	Add first draft linear and logistic regression notebook	2 weeks ago
04_Intermediate_Machine_Learning	Added KNN and SVM Notebooks	2 weeks ago
05_Introduction_to_Unsupervised_Learning	Add 05 and 09 scaffolds	yesterday
09_Transformer-based_Architectures	Update README.md	16 hours ago

Get the Courseware – Step 4

Introduction to Python



Welcome!

This is the start of the Bootcamp proper. The point of this module is to give you an introduction to the basics of the Python programming language. It assumes you know nothing, but it also assumes that you will "immerse" yourself to some degree. I.e., you need to "google around", use Stack Overflow and just generally have the attitude that you're going to "debug anything and everything" in order to get the most out of this.

Lecture-guided content

1. [Hello world!](#) - A classic. Contains one line of code. Feel free to use this as scrap too!
2. [Python fundamentals](#) - A brief introduction to some of the Python, and NumPy, fundamentals.
3. [Plotting and more NumPy](#) - Exactly what it sounds like: plotting and more NumPy!
4. [Functions and logic](#) - An introduction to functions and basic Python logic.
5. [Probability and statistics](#) - A basic introduction on how to perform statistics using simple codes in Python, NumPy, etc.
6. [Random number generators and algorithms](#) - The basics of random number generation and algorithms, demonstrated with simple examples.

Scroll down to get to this content

Open hello_world.ipynb

Introduction to Python

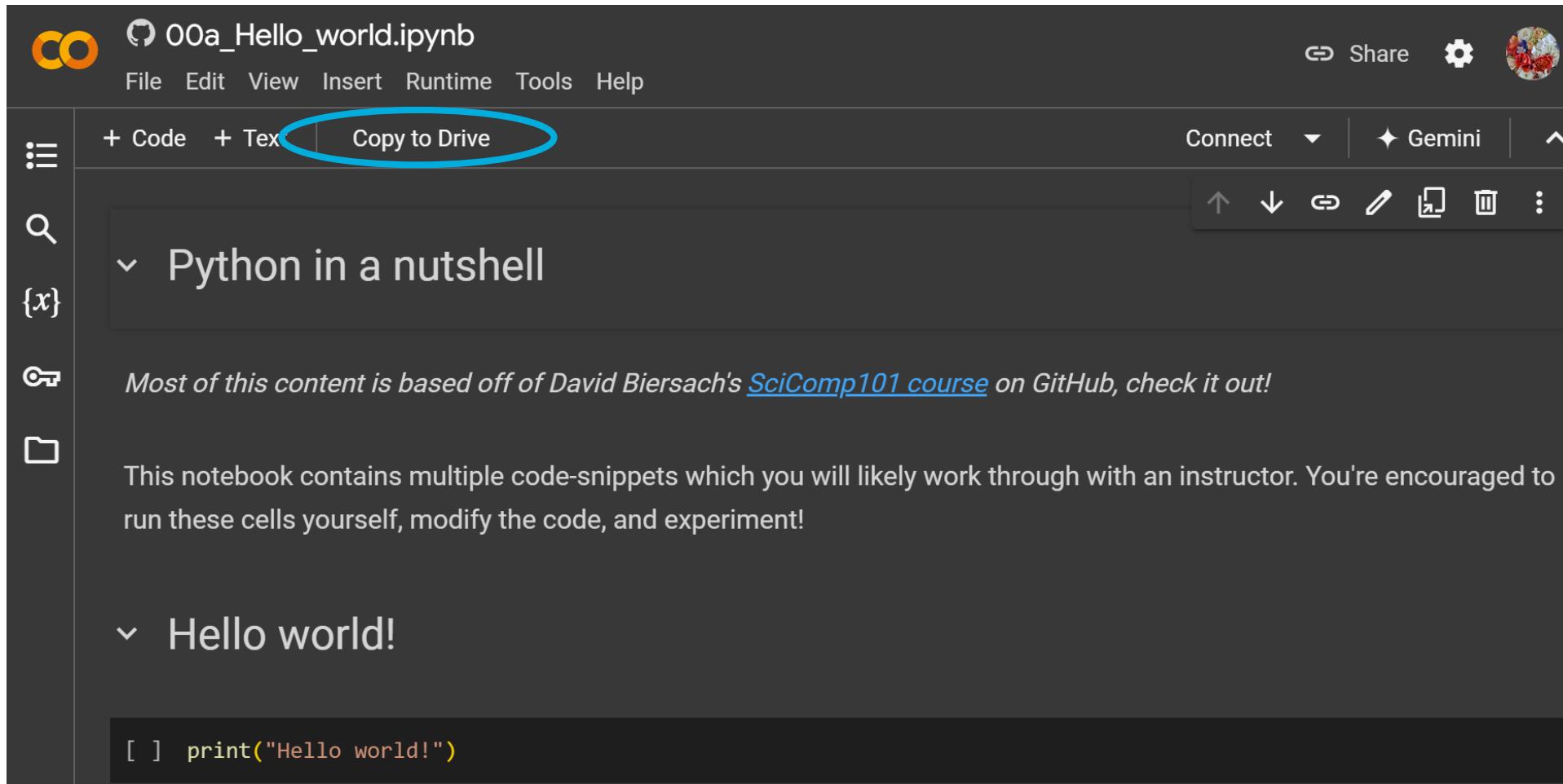
👋 Welcome!

This is the start of the Bootcamp proper. The point of this module is to give you an introduction to the basics of the Python programming language. It assumes you know nothing, but it also assumes that you will "immerse" yourself to some degree. I.e., you need to "google around", use Stack Overflow and just generally have the attitude that you're going to "debug anything and everything" in order to get the most out of this.

Lecture-guided content

1. [Hello world!](#) - A classic. Contains one line of code. Feel free to use this as scrap too!
2. [Python fundamentals](#) - A brief introduction to some of the Python, and NumPy, fundamentals.
3. [Plotting and more NumPy](#) - Exactly what it sounds like: plotting and more NumPy!
4. [Functions and logic](#) - An introduction to functions and basic Python logic.
5. [Probability and statistics](#) - A basic introduction on how to perform statistics using simple codes in Python, NumPy, etc.
6. [Random number generators and algorithms](#) - The basics of random number generation and algorithms, demonstrated with simple examples.

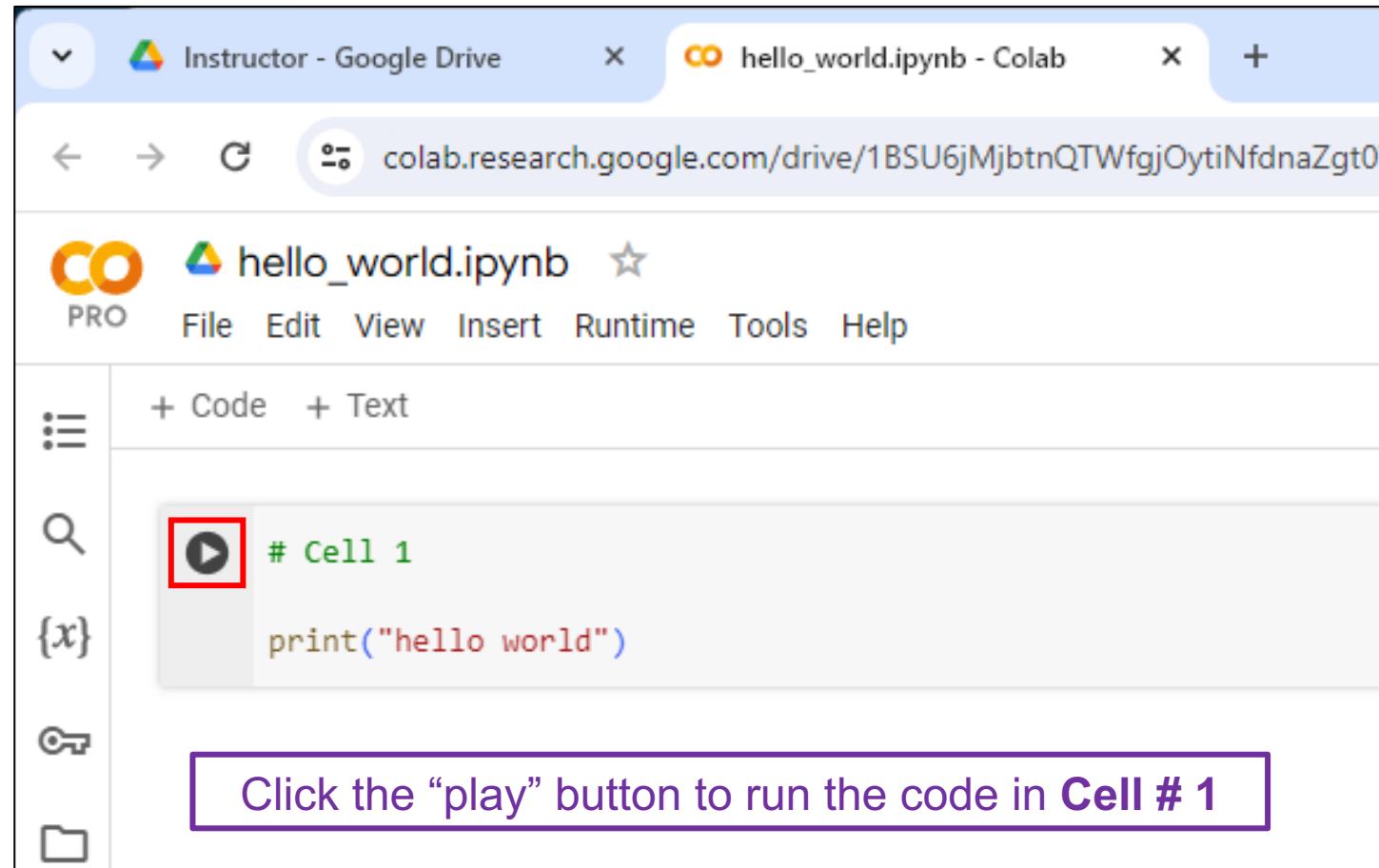
Make Your Own Copy



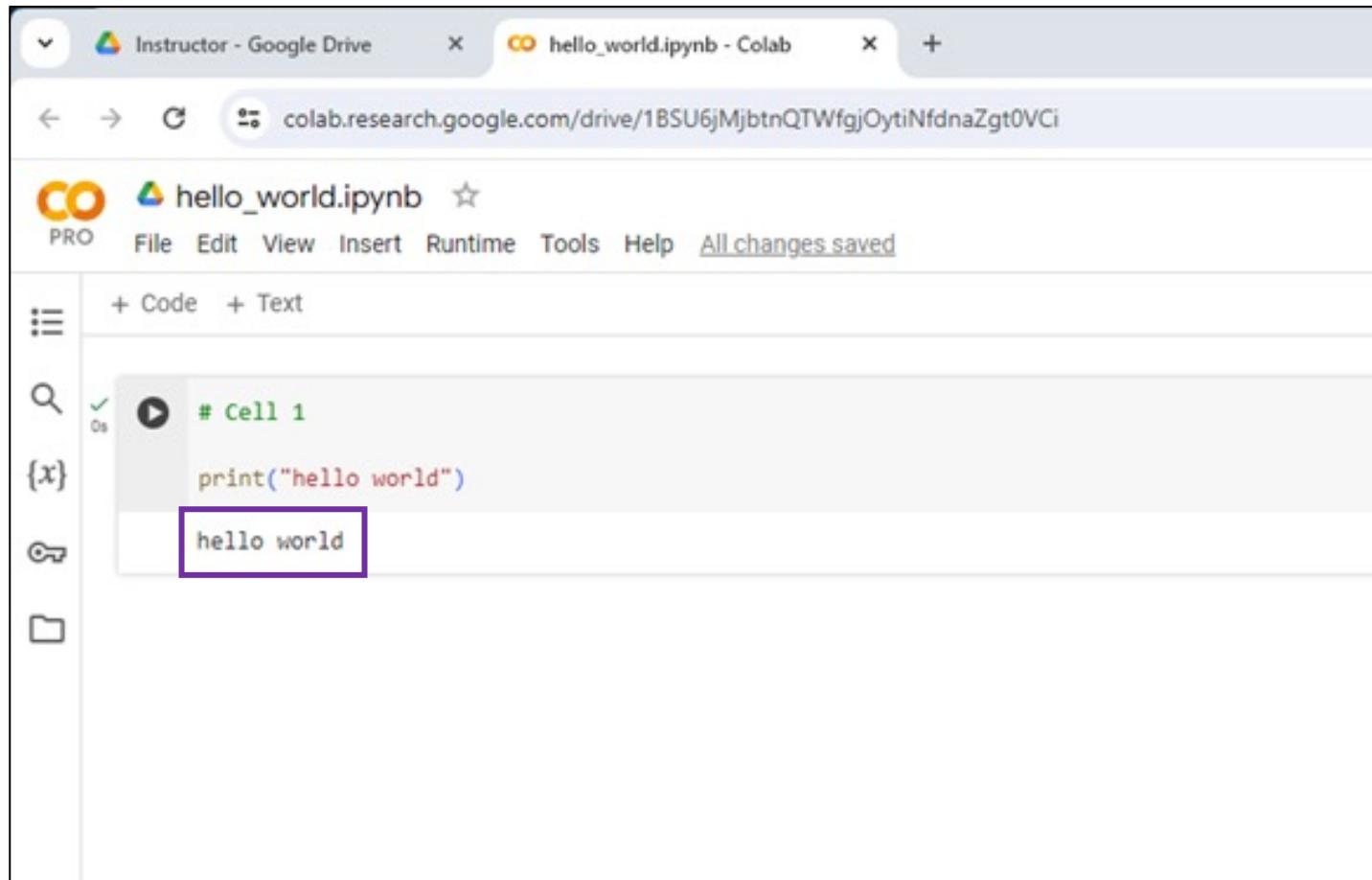
The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** The title is "00a_Hello_world.ipynb".
- Toolbar:** Includes "File", "Edit", "View", "Insert", "Runtime", "Tools", "Help", "Share" (with a gear icon), and a user profile picture.
- Header Buttons:** "+ Code", "+ Tex", and "Copy to Drive" (which is circled in blue).
- Connect and Gemini Buttons:** "Connect" with a dropdown arrow, "Gemini" with a star icon, and other navigation icons.
- Left Sidebar:** Icons for "Code", "Text", "Search", and "Notebook".
- Content Area:**
 - A section titled "Python in a nutshell" with a dropdown arrow.
 - A note: "Most of this content is based off of David Biersach's [SciComp101 course](#) on GitHub, check it out!"
 - A note: "This notebook contains multiple code-snippets which you will likely work through with an instructor. You're encouraged to run these cells yourself, modify the code, and experiment!"
 - A section titled "Hello world!" with a dropdown arrow.
 - A code cell: [] print("Hello world!")

Run hello_world.ipynb



Check hello_world.ipynb



The screenshot shows a Google Colab interface with the title "hello_world.ipynb - Colab". The notebook contains one cell with the following code:

```
# Cell 1
print("hello world")
```

The output of the cell, "hello world", is highlighted with a purple box.

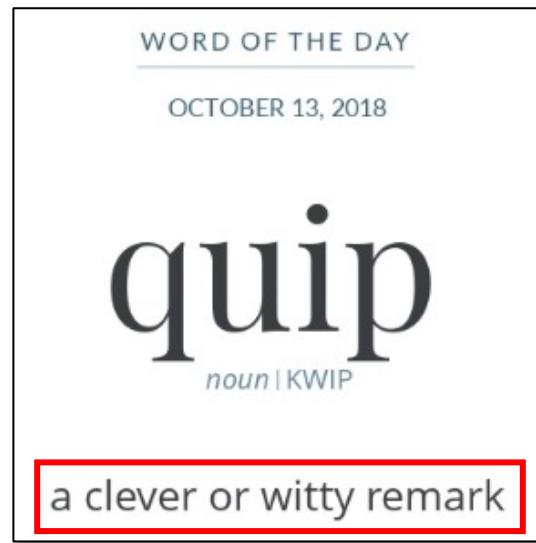


Session 01 – Know You Know...

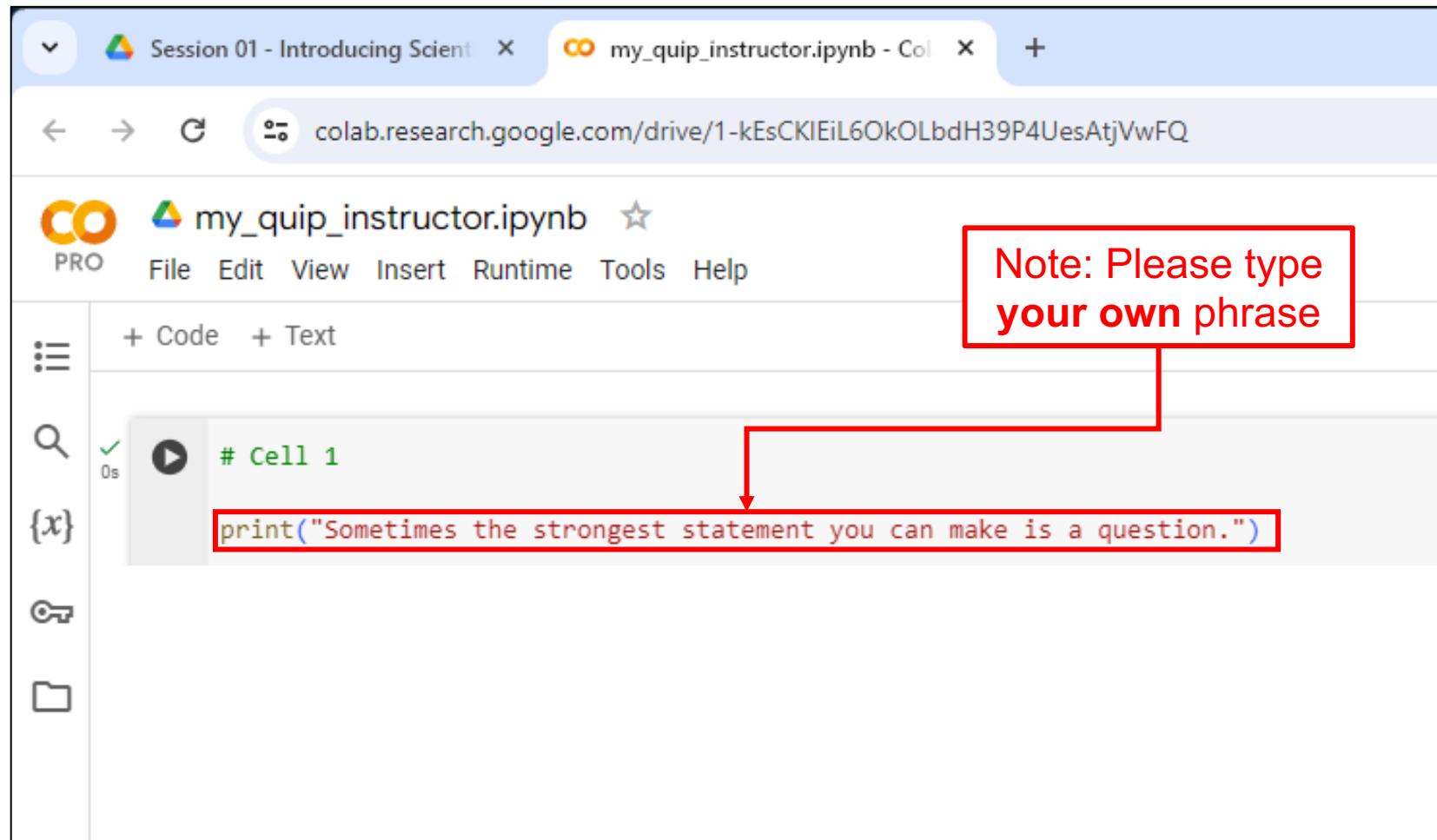
- Scientific Computing **is not the same** as Computer Science
- You cannot do modern science without knowing how to **program** a computer (not just *use* a computer)
- The US Department of Energy offers high school and college internships at 17 National Laboratories across the USA
- One of the best ways to **differentiate yourself** from other students competing for internships is to develop foundational skills in scientific computing
- AP College Board is now offering AP exams in **Python**
- Google **Colab** is a cost-free cloud-hosted service built on Python Notebooks that works with Google **Drive**

TASK 01

Add to your “Hello World” code to make it also display a quip of your choosing or creation



How Do I Do That?

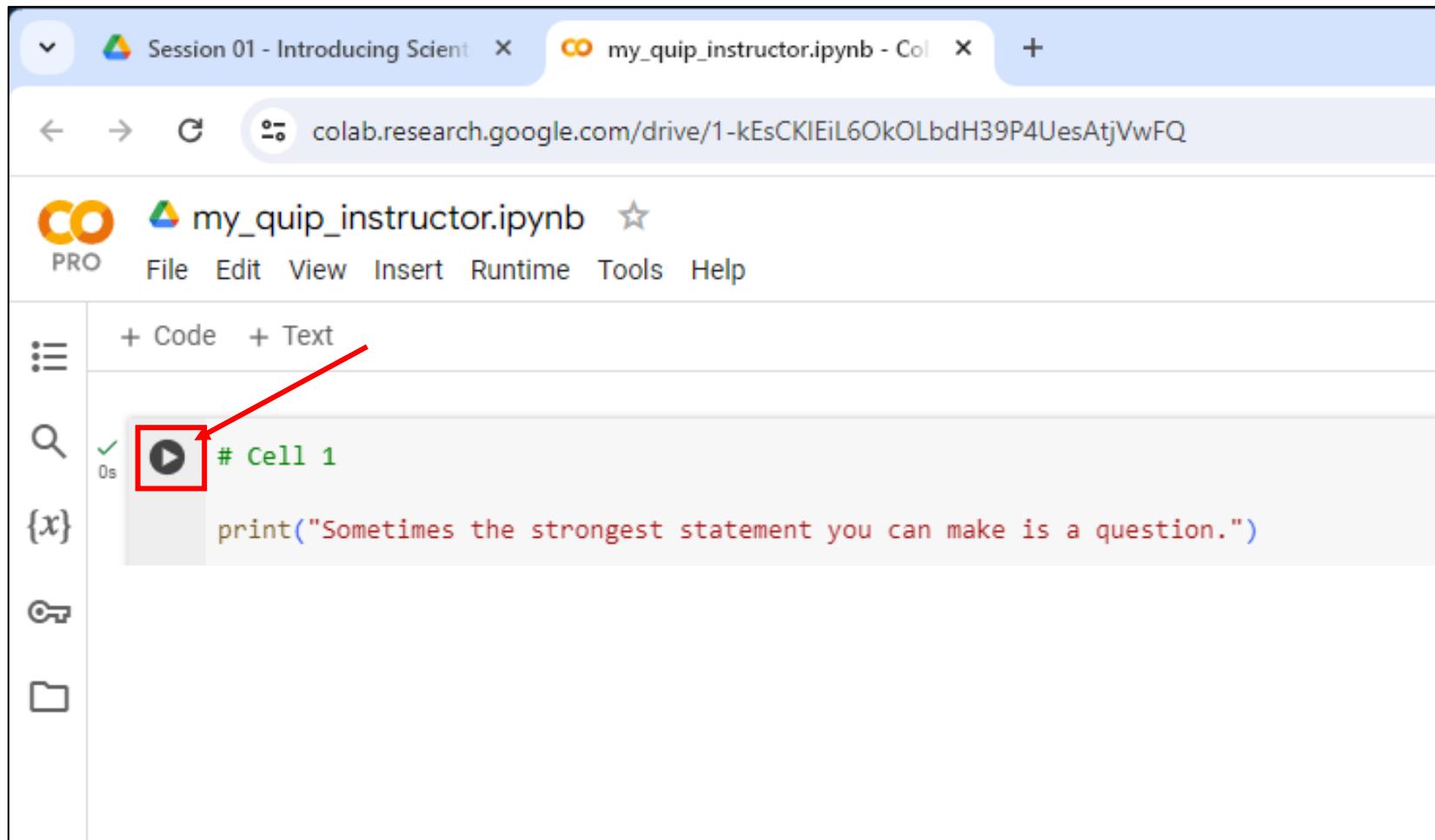


The screenshot shows a Google Colab notebook titled "my_quip_instructor.ipynb". The notebook interface includes a toolbar with file operations like File, Edit, View, Insert, Runtime, Tools, and Help. On the left, there's a sidebar with icons for Code, Text, Search, and a key. The main workspace contains a code cell with the following content:

```
# Cell 1
print("Sometimes the strongest statement you can make is a question.")
```

A red callout box with the text "Note: Please type your own phrase" is positioned above the code cell, with a red arrow pointing from the note to the placeholder text in the code cell.

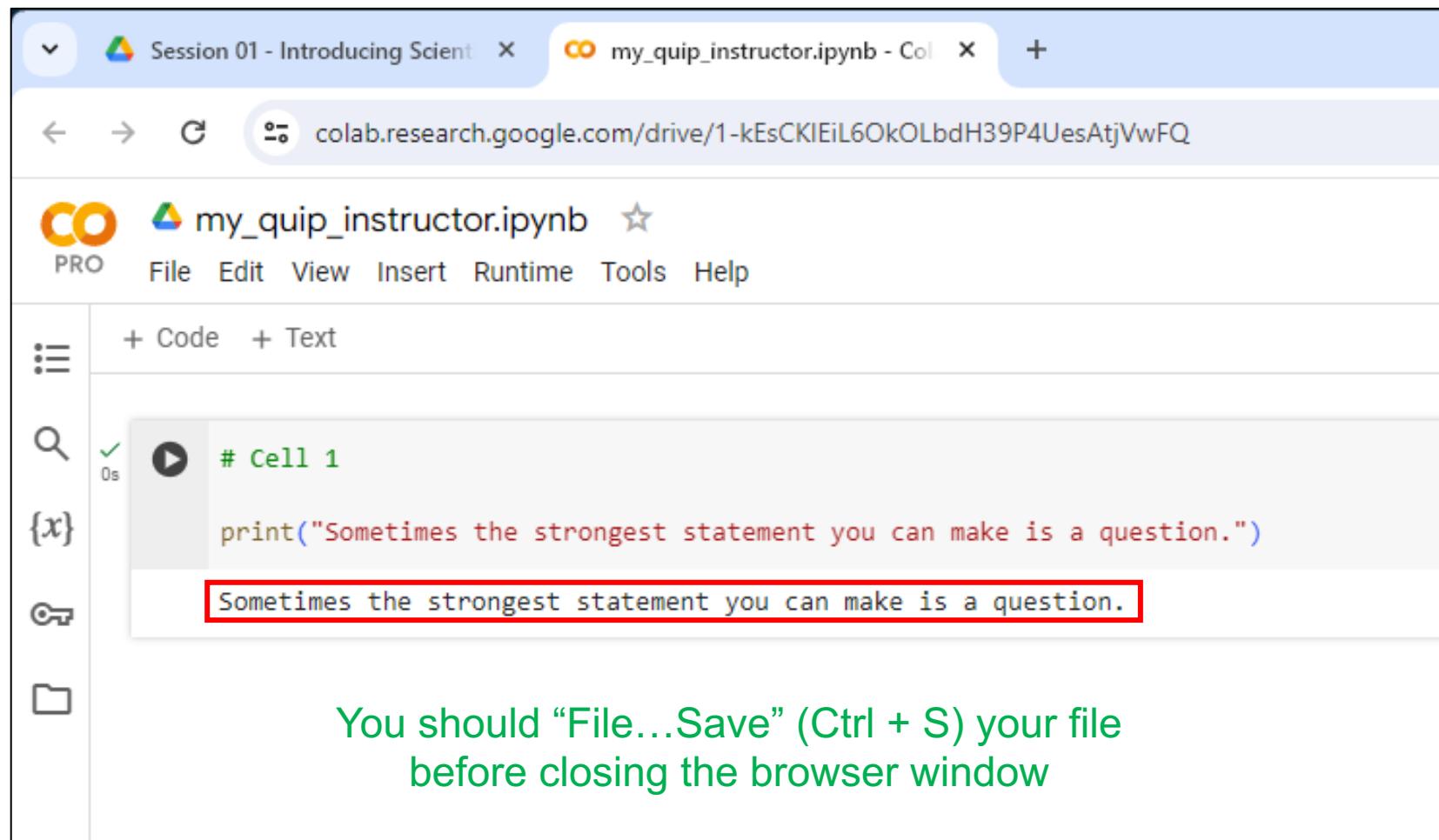
Run my_quip.py



The screenshot shows a Google Colab notebook titled "my_quip_instructor.ipynb". The notebook interface includes a top bar with tabs for "Session 01 - Introducing Scient" and "my_quip_instructor.ipynb - Col". Below the tabs is a URL bar showing the colab URL. The main workspace displays the notebook title and a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the left, there's a sidebar with icons for "Code" and "Text", and a search bar. The main area contains a code cell with a play button icon. A red box highlights the play button, and a red arrow points to it from the top-left. The code cell contains the following Python code:

```
# Cell 1
print("Sometimes the strongest statement you can make is a question.")
```

Check my_quip.py



The screenshot shows a Google Colab notebook titled "my_quip_instructor.ipynb". The code cell contains the following Python code:

```
# Cell 1
print("Sometimes the strongest statement you can make is a question.")
```

The output of the cell is displayed in a box and reads: "Sometimes the strongest statement you can make is a question.", which is highlighted with a red border.

Below the code cell, a green message states: "You should ‘File...Save’ (Ctrl + S) your file before closing the browser window".