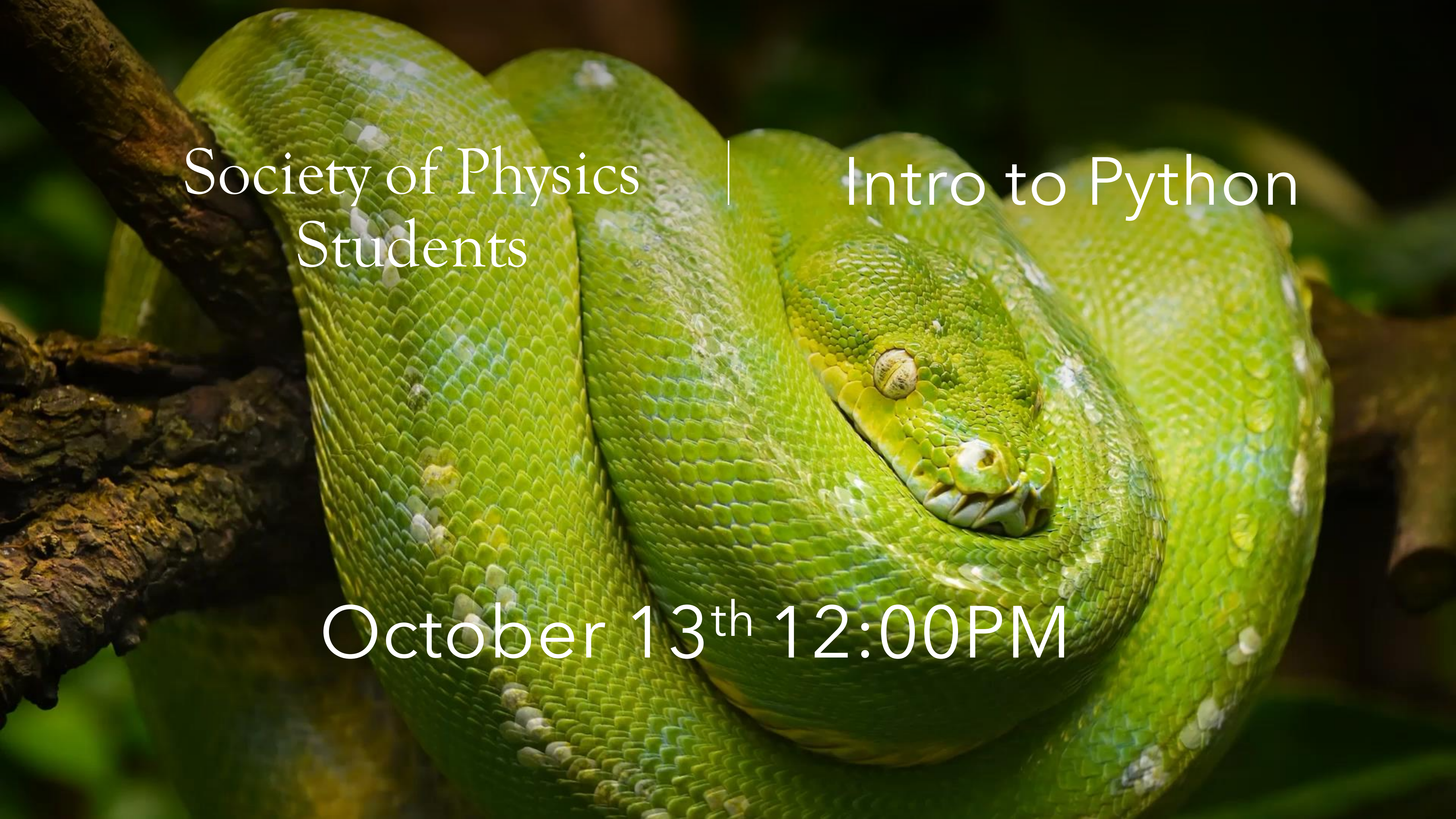
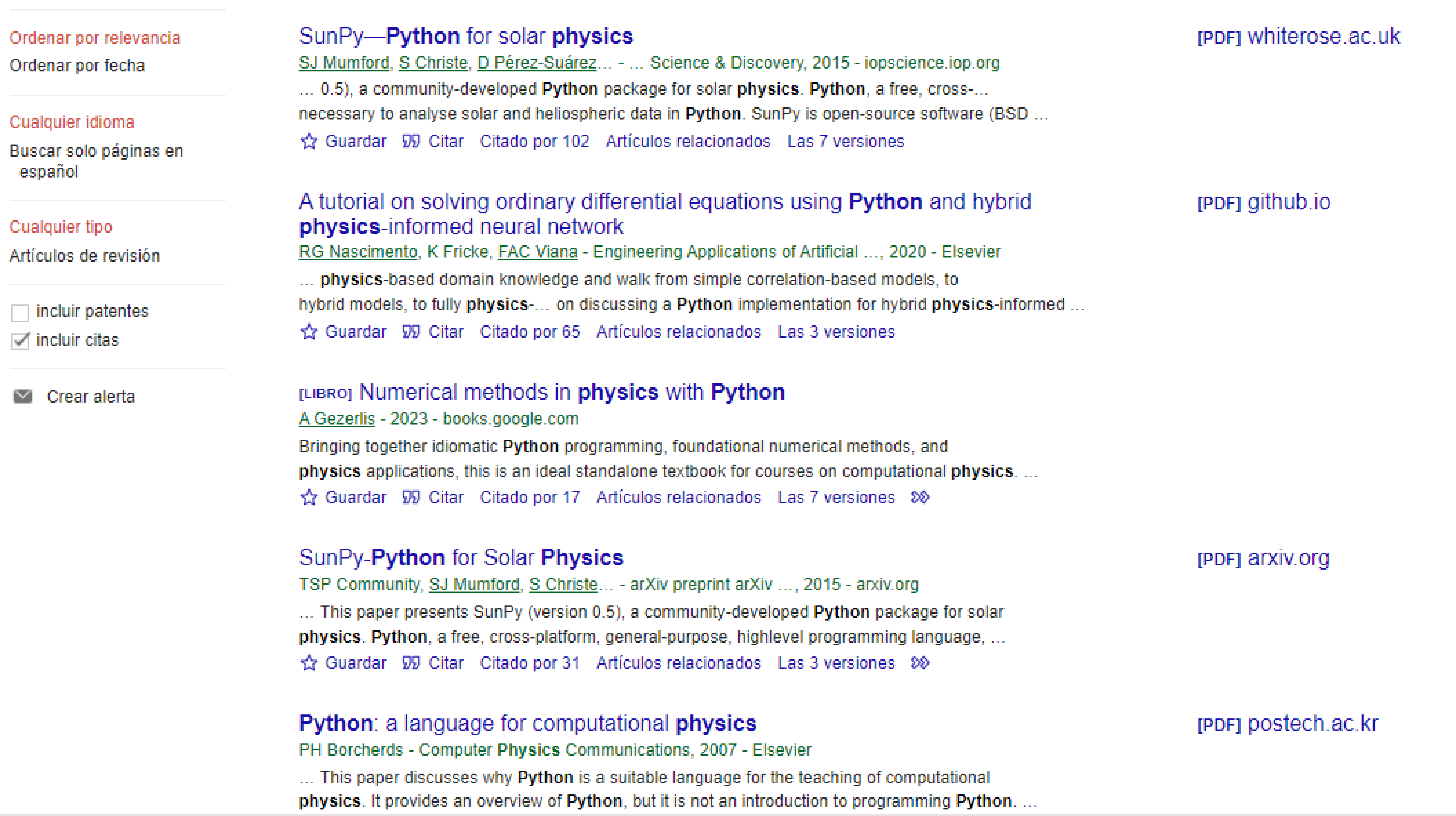


Society of Physics Students

Intro to Python

October 13th 12:00PM

Why Coding as a Physicist is Important?



Ordenar por relevancia
Ordenar por fecha

Cualquier idioma
Buscar solo páginas en español

Cualquier tipo
Artículos de revisión

☐ incluir patentes
☑ incluir citas

Crear alerta

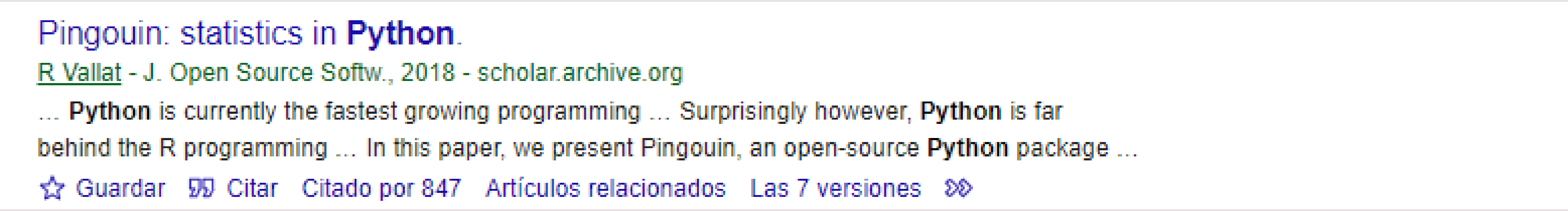
SunPy—Python for solar physics — [PDF] whiterose.ac.uk
SJ Mumford, S Christe, D Pérez-Suárez… - … Science & Discovery, 2015 - iopscience.iop.org
… 0.5), a community-developed **Python** package for solar **physics**. **Python**, a free, cross-… necessary to analyse solar and heliospheric data in **Python**. SunPy is open-source software (BSD …
☆ Guardar Citar Citado por 102 Artículos relacionados Las 7 versiones

A tutorial on solving ordinary differential equations using Python and hybrid physics-informed neural network — [PDF] github.io
RG Nascimento, K Fricke, FAC Viana - Engineering Applications of Artificial …, 2020 - Elsevier
… **physics**-based domain knowledge and walk from simple correlation-based models, to hybrid models, to fully **physics**-… on discussing a **Python** implementation for hybrid **physics**-informed …
☆ Guardar Citar Citado por 65 Artículos relacionados Las 3 versiones

[LIBRO] **Numerical methods in physics with Python**
A Gezerlis - 2023 - books.google.com
Bringing together idiomatic **Python** programming, foundational numerical methods, and **physics** applications, this is an ideal standalone textbook for courses on computational **physics**. …
☆ Guardar Citar Citado por 17 Artículos relacionados Las 7 versiones

SunPy-Python for Solar Physics — [PDF] arxiv.org
TSP Community, SJ Mumford, S Christe… - arXiv preprint arXiv …, 2015 - arxiv.org
… This paper presents SunPy (version 0.5), a community-developed **Python** package for solar **physics**. **Python**, a free, cross-platform, general-purpose, highlevel programming language, …
☆ Guardar Citar Citado por 31 Artículos relacionados Las 3 versiones

Python: a language for computational physics — [PDF] postech.ac.kr
PH Borcherds - Computer Physics Communications, 2007 - Elsevier
… This paper discusses why **Python** is a suitable language for the teaching of computational **physics**. It provides an overview of **Python**, but it is not an introduction to programming **Python**. …

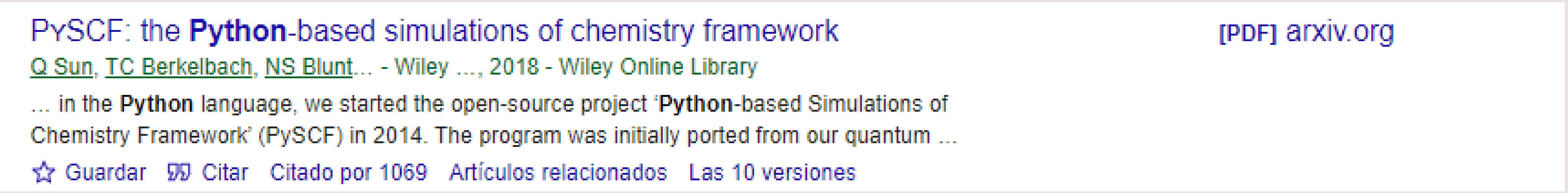


Pingouin: statistics in **Python**.

R Vallat - J. Open Source Softw., 2018 - scholar.archive.org

... **Python** is currently the fastest growing programming ... Surprisingly however, **Python** is far behind the R programming ... In this paper, we present Pingouin, an open-source **Python** package ...

Guardar Citar Citado por 847 Artículos relacionados Las 7 versiones



PySCF: the **Python**-based simulations of chemistry framework [PDF] arxiv.org

Q Sun, TC Berkelbach, NS Blunt... - Wiley ..., 2018 - Wiley Online Library

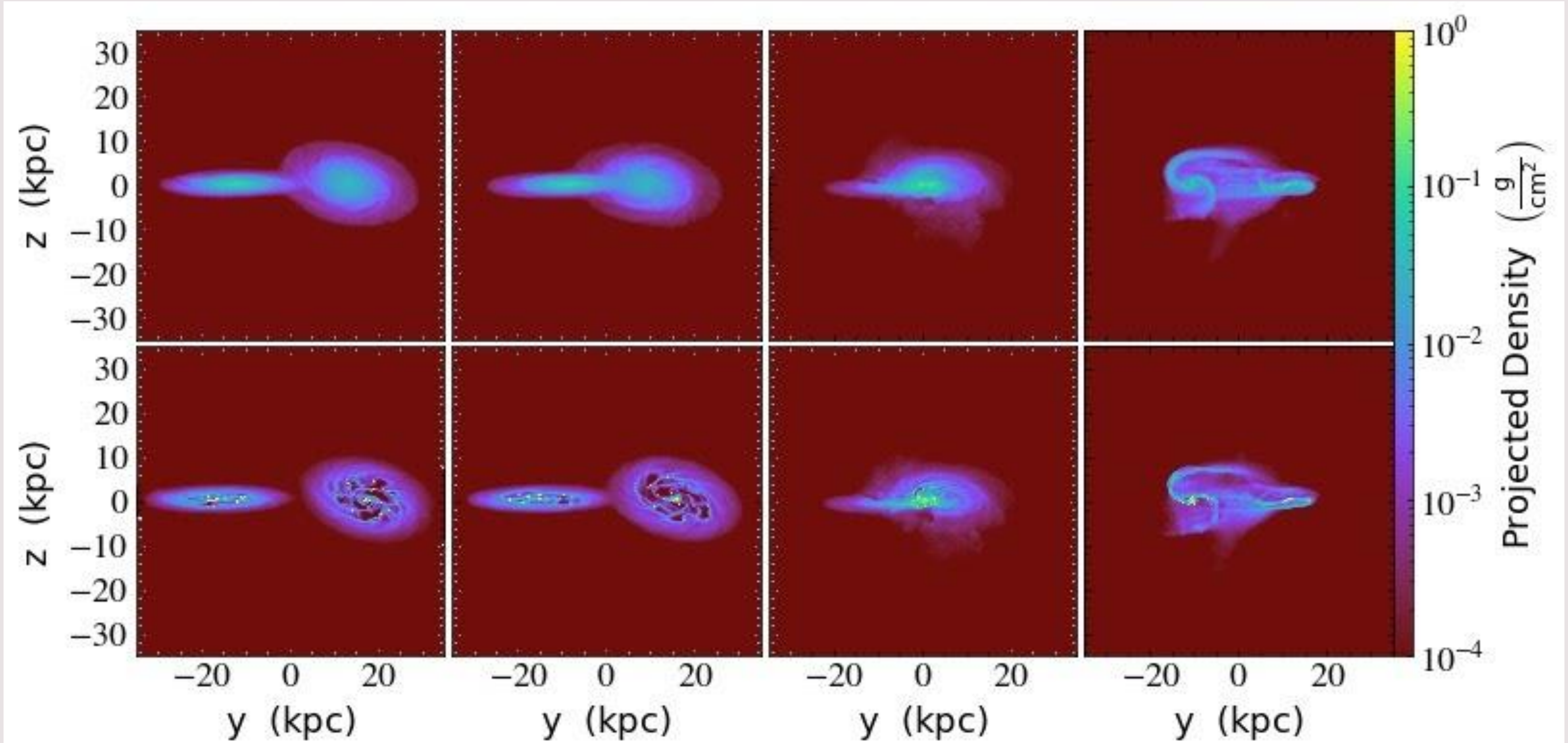
... in the **Python** language, we started the open-source project '**Python**-based Simulations of Chemistry Framework' (PySCF) in 2014. The program was initially ported from our quantum ...

Guardar Citar Citado por 1069 Artículos relacionados Las 10 versiones

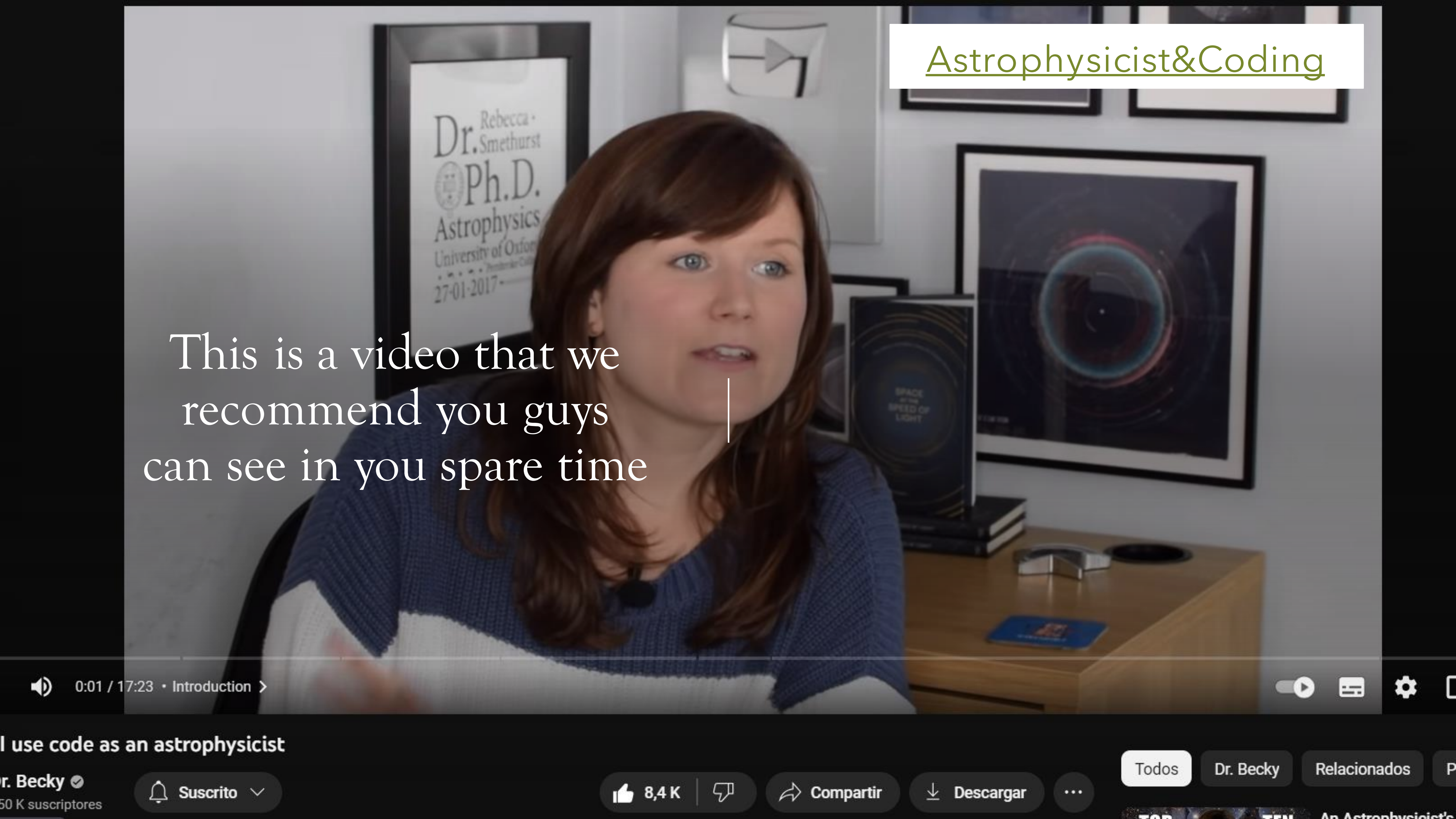
To which we can list of a series of famous physics simulation and montecarlo softwares

- https://en.wikipedia.org/wiki/Geant4
- https://en.wikipedia.org/wiki/DelPhi
- https://en.wikipedia.org/wiki/ROOT
- https://en.wikipedia.org/wiki/WIEN2k

Gadget4 Simulation (Implemented in C++) And Data Analysis with (Python and Project yt library)



Astrophysicist&Coding



This is a video that we recommend you guys can see in you spare time

Who are we?

Aldo Sanchez
Computer Science Senior

- Google ML Intern – Python
- Microsoft Intern
- MCA Foundation Intern

Tadeo De La Rocha
Computer Science Junior

- Software Engineer Intern Google (x2)

Byron Encinas Velazquez
Physics Graduate

- Scholarship with Conacyt for the development of materials fracture simulations
- National Physics Congress, Star Formation Data Analysis with Gadget-4

With no further ado...

- Let's get started with Python!

Outline

- Introduction
- Setting up Python
- Variables and Data types
- Arithmetic operations
- Conditionals
- Loops
- Functions

Intro to Python Workshop

- In this workshop, we'll introduce you to Python, a powerful programming language that is widely used in physics and various other fields.
- Our goal is to equip you with the foundational Python skills you need for your physics studies.

What is Python?

- Python is a high-level programming language.
- It's known for its versatility and is used in various scientific and engineering disciplines, including physics.
- Python is an excellent tool for data analysis, simulations, and automating repetitive tasks.
- Lots of libraries including Numpy, Matplotlib, Simpy, etc.

Getting started with Python

1. **Install Python**: First, make sure you have Python installed on your computer. You can check if Python is already installed by opening a command prompt (on Windows) or a terminal (on macOS or Linux) and running the following command:

   ```
   python --version
   ```

   If not, you can download the latest Python installer for your operating system from the official Python website (https://www.python.org/downloads/).

2. **Install a Code Editor**: **PyCharm**: A Python-specific IDE developed by JetBrains.

3. **Write Your First Python Program**: Open PyCharm and create a new Python file with a .py extension.

4. In your Python file, write a simple "Hello, World!" program to get started:

   ```
   print("Hello, World!")
   ```

5. **Run Your Python Program**: Save the Python file and then run it. To do this, open a terminal or command prompt, navigate to the directory where your Python file is located, and enter the following command:

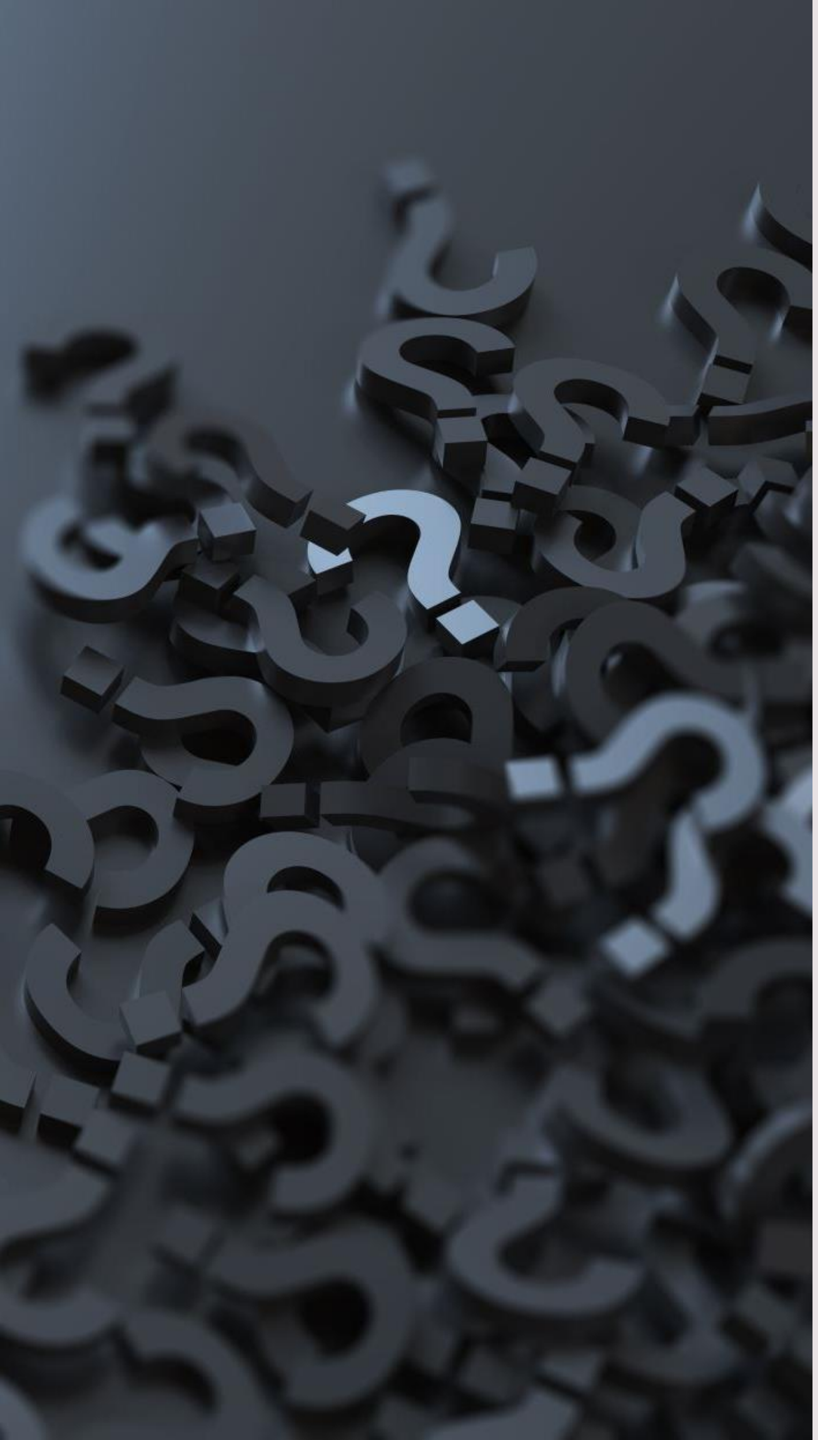
   ```
   python hello.py
   ```

6. You should see the output "Hello, World!" displayed in the terminal.

What are Variables?

Variables like containers that can store different types of data or values.

Think of variables as labeled boxes where you can put different things (data).



Python Variable Naming Rules:

Example (GoogleColabNotebook)

Rules for naming variables in Python:

1. Variable names can only contain letters, numbers, and underscores.
2. Variable names must start with a letter or underscore but **not a number**.
3. Variable names are case-sensitive

Are myVar and myvar different variables?

Which of these examples are allowed as variable names:

Counter_1 index&3 Vector_12 first_force_x 2nd_force_y

Data Types in Python

Python has various data types to represent different kinds of data.

Most common data types:

1. **Integers (int):** Whole numbers like 5, -3, 0.
2. **Floats (float):** Decimal numbers like 3.14, -0.5, 2.0.
3. **Strings (str):** Text enclosed in single or double quotes, e.g., "Hello, World!" or 'Python'.
4. **Booleans (bool):** Represents True or False.

Data Type Examples

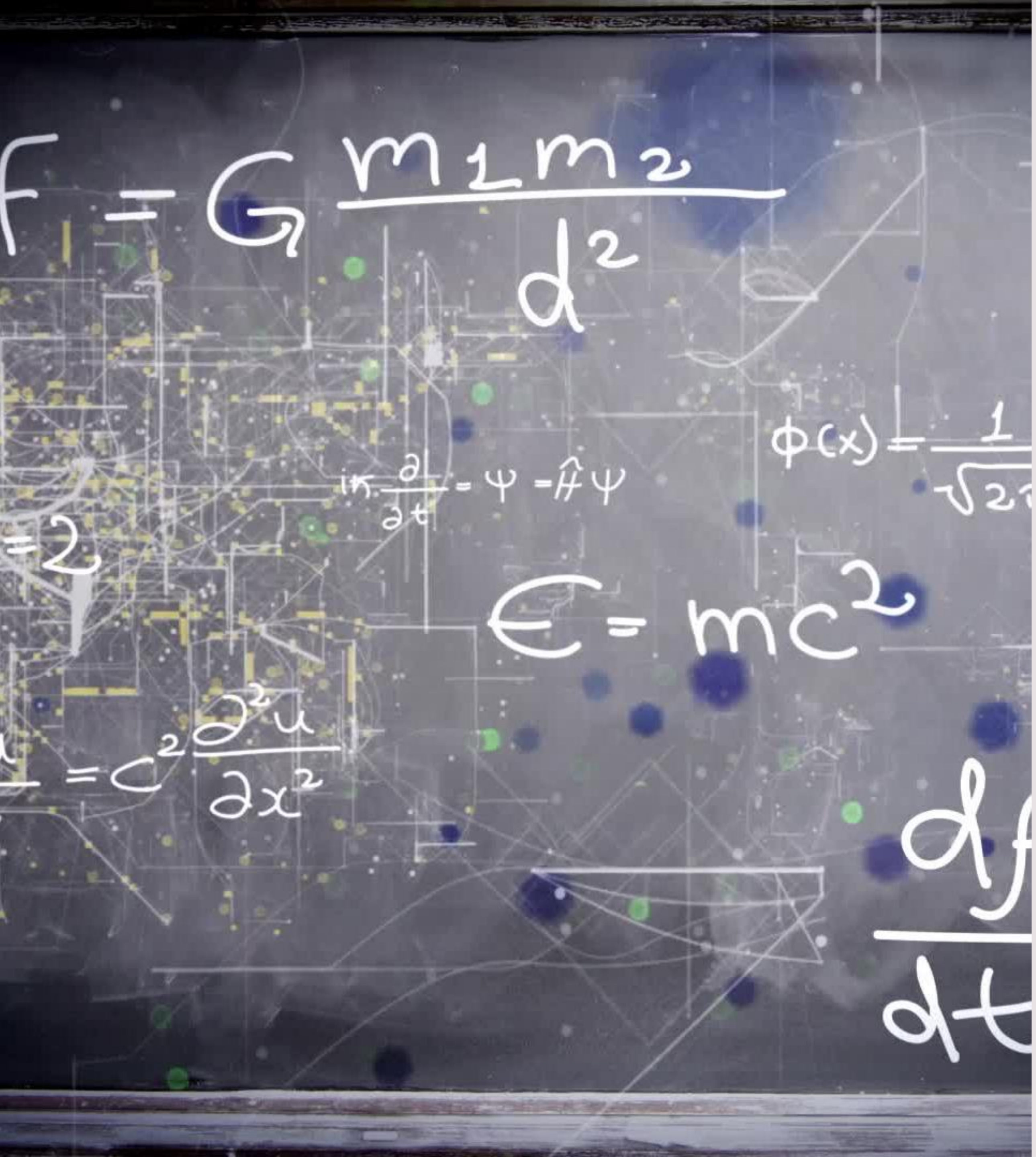
- What data type should each of these be:
  - Cost of a flight
  - Name of a pet
  - Capacity of a classroom
  - Attendance of a person
  - Shoe size
  - Company stock value
  - Area of circle
  - Pi

Lists and indexing

```
myList = ["Hello", "this", "is", "a list"]
```

"Hello"	"this"	"is"	"a list"
myList[0]	myList[1]	myList[2]	myList[3]

- Lists are a collection of variables ordered numerically.
- We can access the values inside a list by using brackets after the name of our list



Variable Assignment:

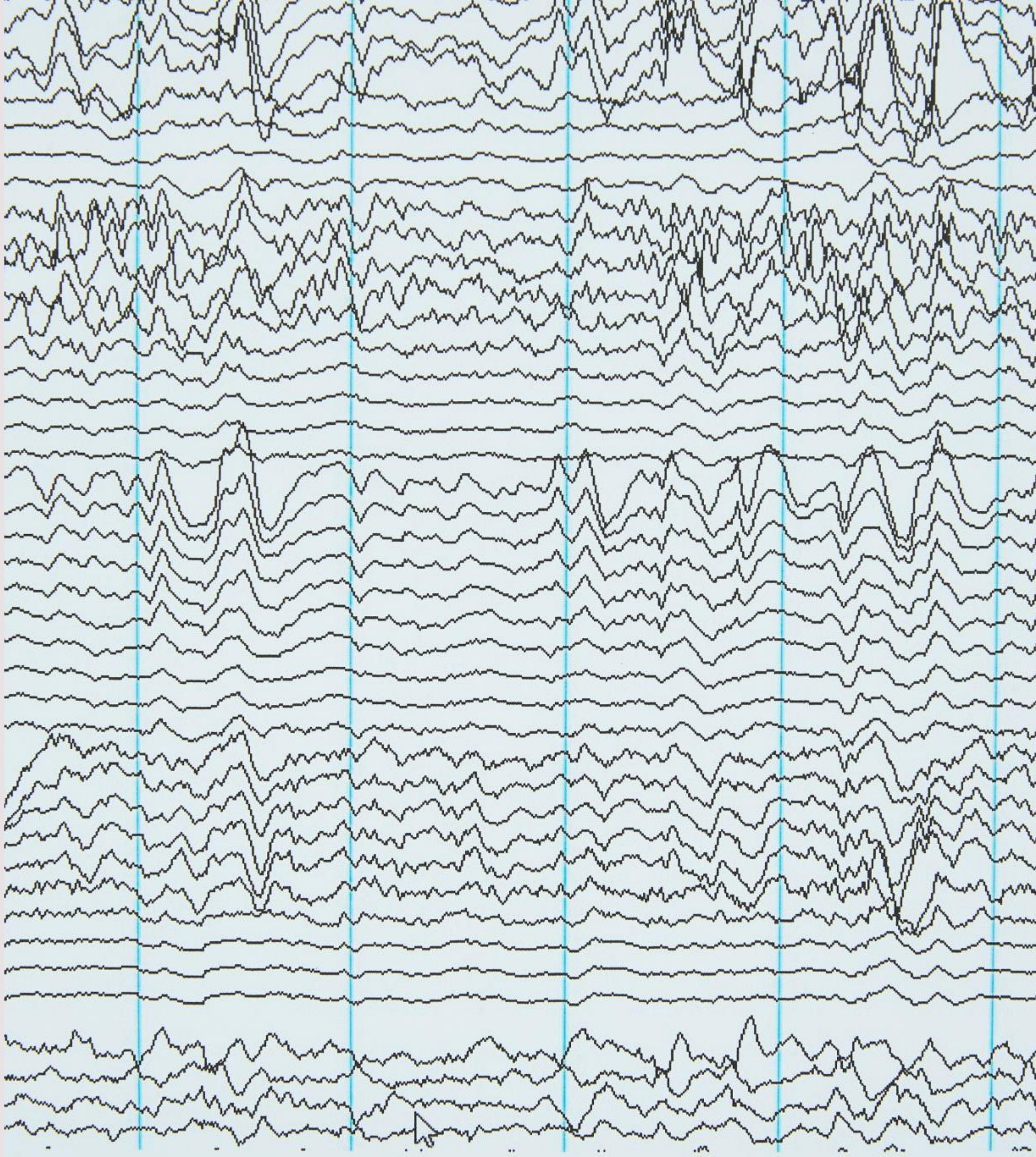
1. Assign values to variables using the = operator.
   - x = 2
   - x = 5
   - print(x)
2. Variable assignments with different data types
   - x = 5   # Integer
   - y = 3.14   # Float
   - name = "John"   # String
   - is_ready = True   # Boolean

Using Variables:

How to use variables in simple expressions or print statements:

- age = 25
- increase = 3
- updated_age = age + increase
- print("I am", updated_age , "years old.")

- age = 25
- age = age + 3
- print("I am", updated_age , "years old.")



Data Type Conversion:

Python allows you to convert between data types if needed.

- num = 10
- num_str = **str**(num)   # Convert to string
- num_float = **float**(num)# Convert to float

Arithmetic in Python

Addition (+): Use the + operator to add two numbers together.

Subtraction (-): Use the - operator to subtract one number from another.

Multiplication (*): Use the * operator to multiply two numbers.

Division (/): Use the / operator to divide one number by another.

(In Python 3, division returns a float by default.)

Modulo (%): Use the % operator to find the remainder when one number is divided by another.

```
num1 = 15
num2 = 7
remainder = num1 % num2
print("Modulo:", remainder) # Output: Modulo: 1
```

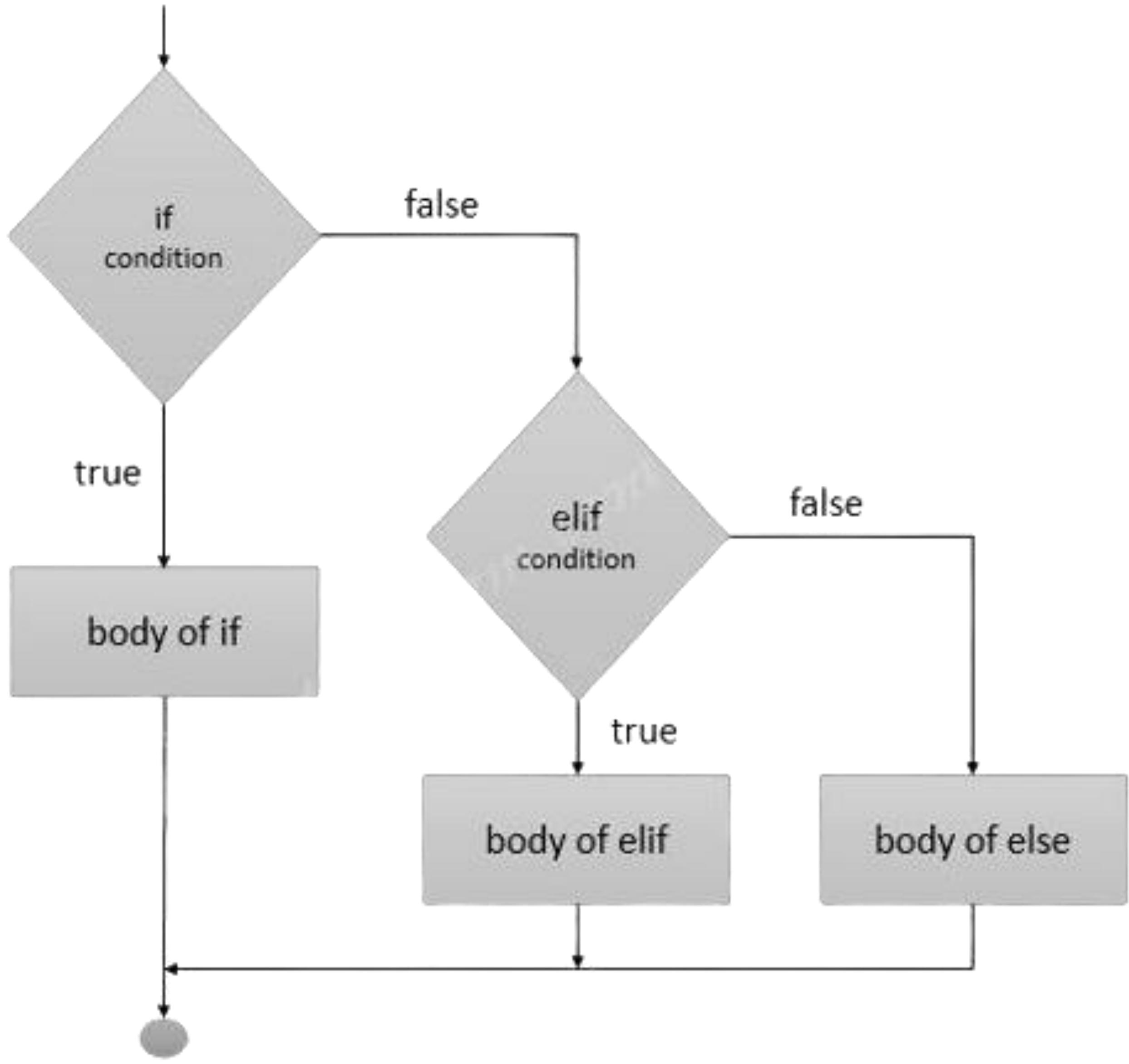
Using Parentheses for Order of Operations:

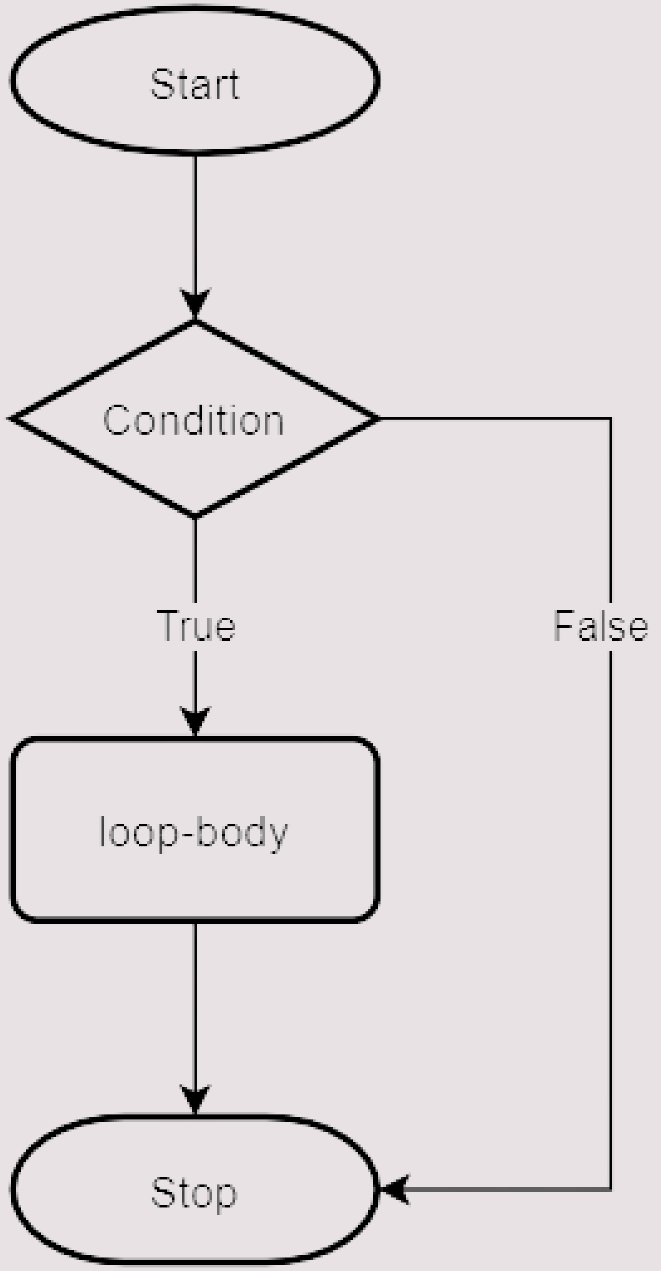
```
expression = (4 + 5) * 2
print("Expression with Parentheses:", expression) # Output: Expression with Parentheses: 18
```

Comments (lines that start with #) can help explain the purpose of your code and calculations code after # is not run

Conditionals

- **If Statements:** Whenever we want to run a piece of code only if a certain condition is true we can use an if statement.
- **Else Statements:** We can place an else statement after an if statement that will execute whenever our if condition fails.
- **Elif (else if) Statements:** They are similar to else statements but they only execute if an additional condition is met.





Loops

- **While loops:** They will execute a specific block of code as long as the condition given is true.
- **For loops:** They are used to repeat a specific block of code a known number of times.

Defining a function

- Functions are blocks of code, this is useful as it allows us to run them as many times as we want without having to repeat the code over and over again.

```
Def addNumbers (int a, int b) {
    c = a + b
    Return c
}
```

Calling a Function

- We can call a function by simply typing its name and passing it the parameters required, we can even assign the result to a variable if we know it will return something.

```
result = addNumbers (5, 6)
```

Taking input

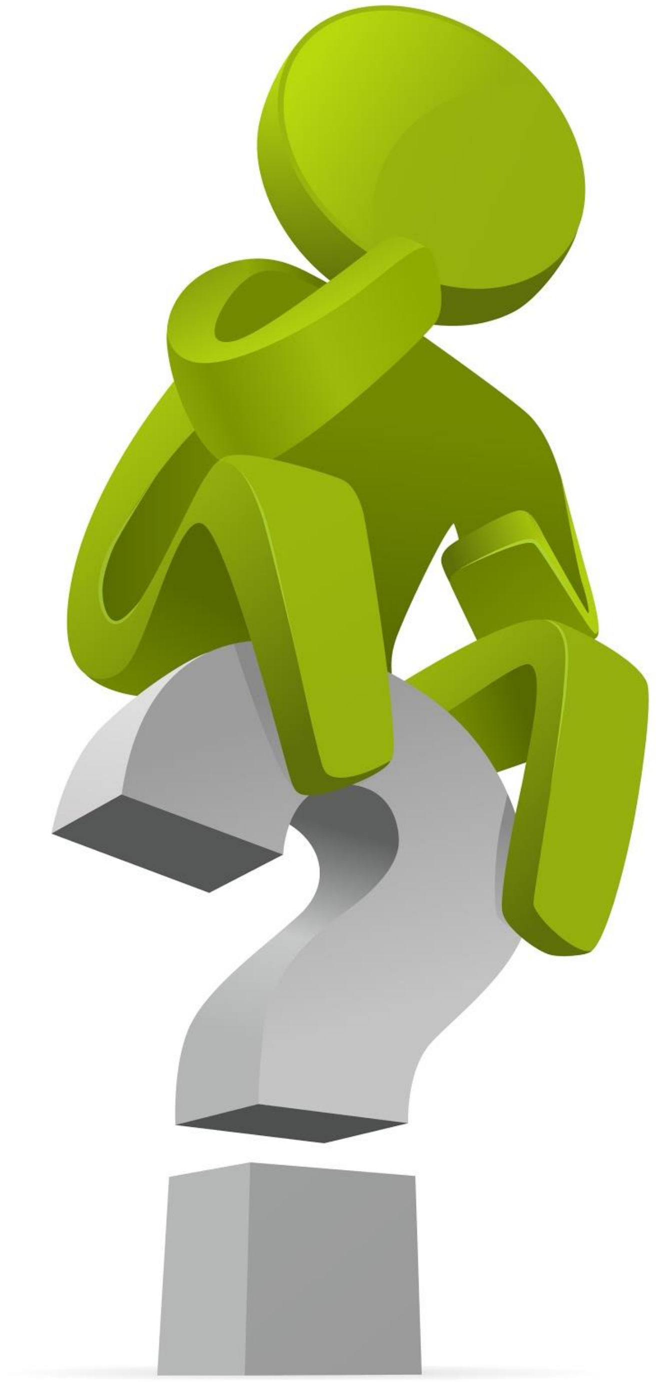
We can take in user input by using the input function.

```
favoriteFruit = input("Type in your favorite fruit:")
```

The string we pass in will be printed before the user input is taken.

Working with Libraries

What libraries are most useful for a scientist?



Working with Libraries

What libraries are most useful for a scientist?

- Numpy (numerical and mathematical manipulation of data)
- Matplotlib (Data Visualization and Ploting)
- Simpy (Symbolic Algebra)
- Pandas (Database treatment)