

# Computational Physics Club

---

Yachay Tech

# Mision

The Computational Physics Club promotes the learning and application of Python and Mathematica in physics, providing a collaborative space where students can develop skills in simulation, modeling, symbolic computation, and data analysis, fostering scientific and technological innovation.

# Vision

To become a leading academic and research hub in computational physics, specializing in Python and Mathematica, and to equip students with strong skills in programming, numerical methods, and symbolic computation for solving complex physical problems.



The primary consequence of the computational nature of the universe is that the universe naturally generates complex systems, such as life. Although the basic laws of physics are comparatively simple in form, they give rise, because they are computationally universal, to systems of enormous complexity.

— *Seth Lloyd* —

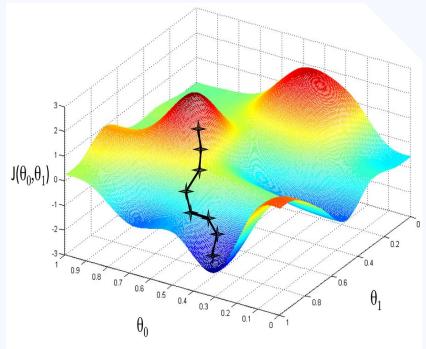
AZ QUOTES

# Objectives



1. **Training:** Offer workshops on Python (NumPy, SciPy, Matplotlib, SymPy) and Mathematica for numerical and symbolic computation.
2. **Research Projects:** Develop simulations and analytical models using Python and Mathematica in various areas of physics.
3. **Multidisciplinary Collaboration:** Encourage teamwork across physics, engineering, and applied mathematics using computational tools.
4. **Competitions and Publications:** Participate in coding challenges, modeling contests, and encourage article publications.
5. **Scientific Outreach:** Organize demonstrations and talks showcasing Python and Mathematica applications in physics.

# Our Directive



**Juanda Vasconez**

Ped áspera ad astra



**Santiago Reascos**

Always learning

# Our Directive



**Yorlan Males-Araujo**

Like doing sweet stuff.



**Alejandro Silva**

I love reading and I want to be an astronaut



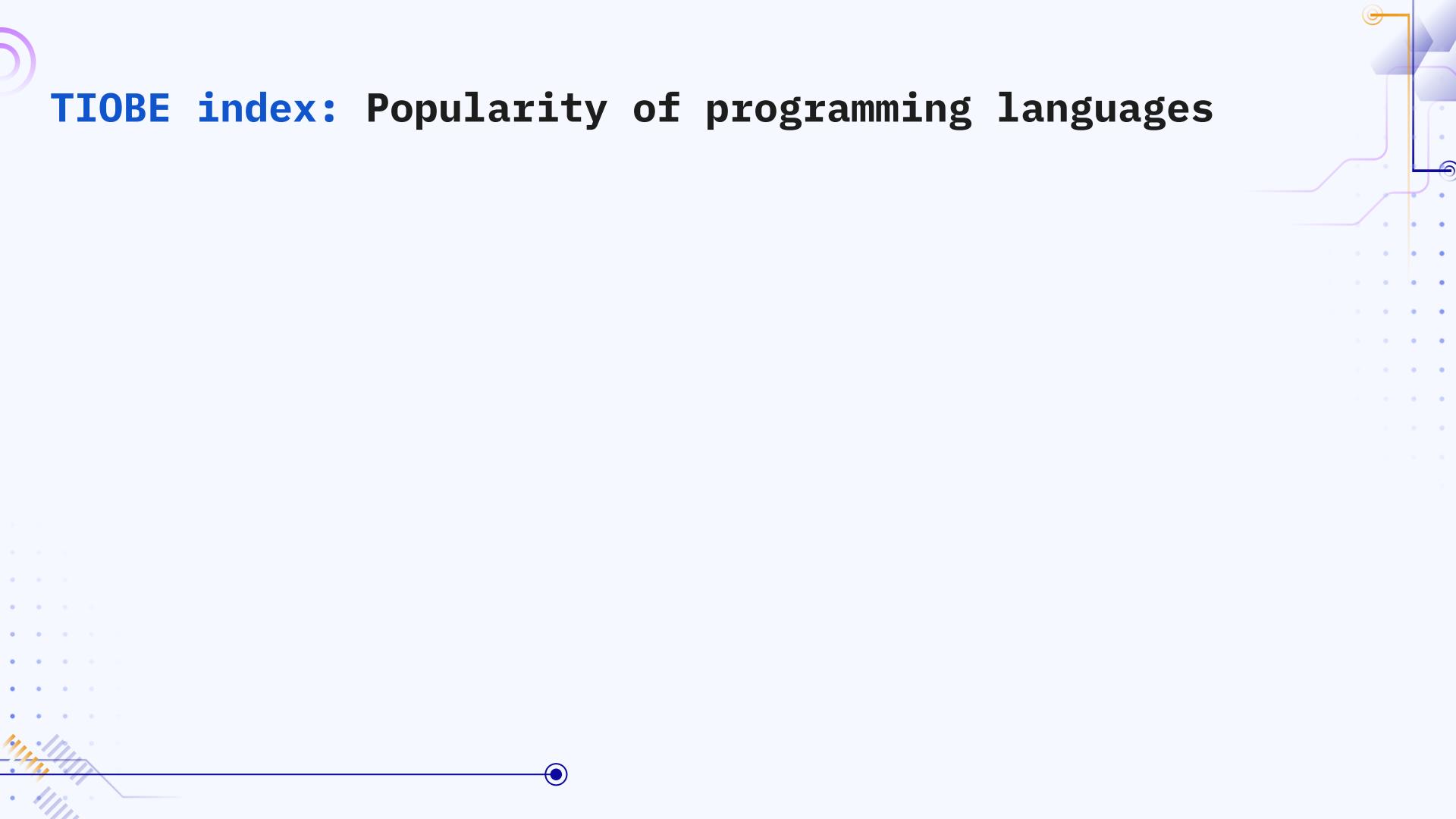
**Alexander Andrade**

I don't fully understand the universe, but I enjoy exploring it.

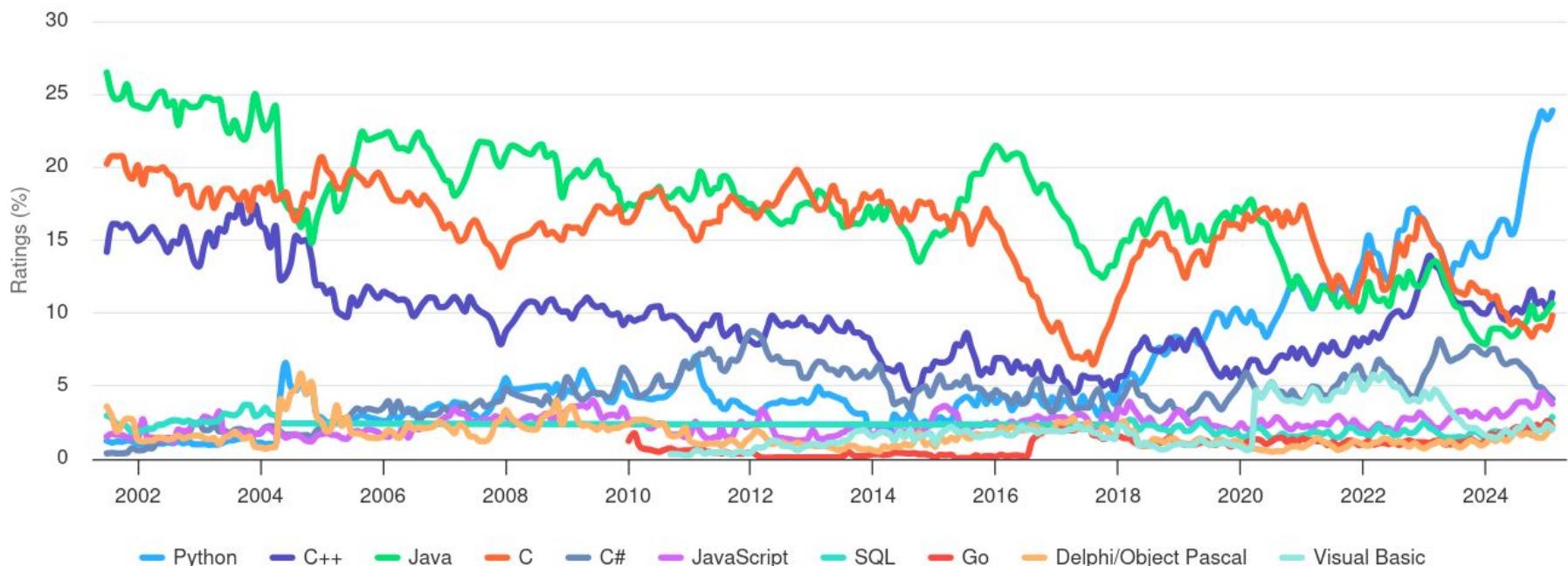
# Why Python?

---

# TIOBE index: Popularity of programming languages

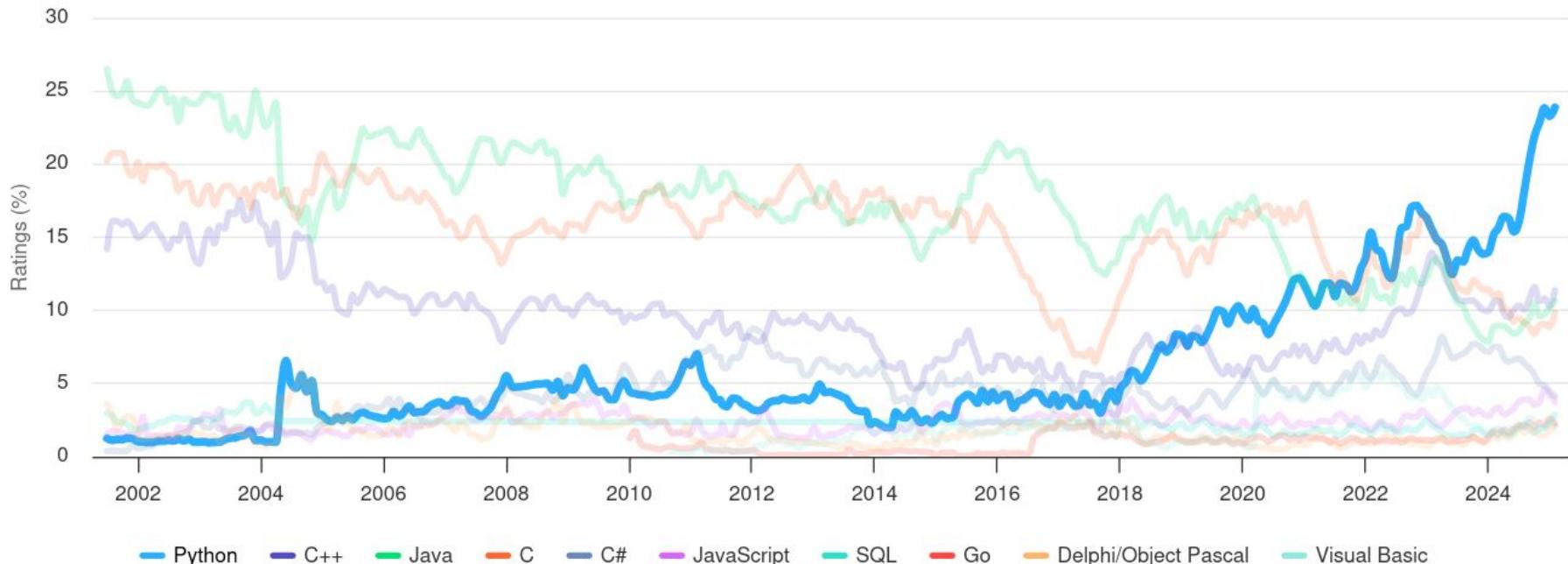


# TIobe index: Popularity of programming languages



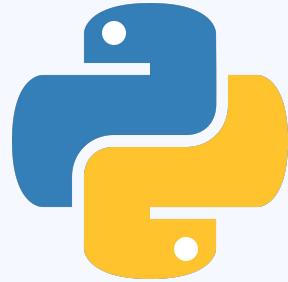
Source: <https://www.tiobe.com/tiobe-index/>

# TIOBE index: Popularity of programming languages



Source: <https://www.tiobe.com/tiobe-index/>

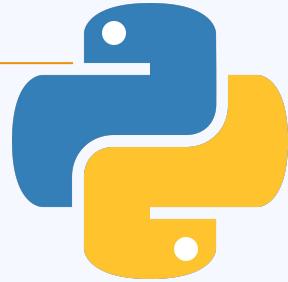
# Python main features



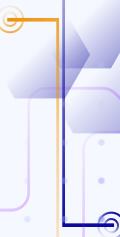
# Python main features

**High-level  
language**

*Close to coding in  
human language.*



## Syntax comparison



# Syntax comparison



```
someList = [x**2 for x in array]           Python
```

# Syntax comparison



```
someList = [x**2 for x in array]           Python
```



C

```
#include <stdlib.h>

int* square_array(int* array, int size)
{
    int* result = (int*) malloc(size * sizeof(int));

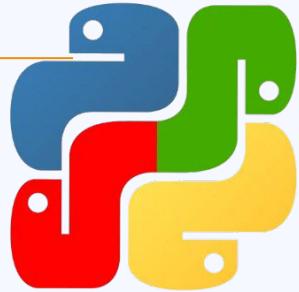
    for (int i = 0; i < size; i++)
    {
        result[i] = array[i] * array[i];
    }

    return result;
}
```

# Python main features

**High-level  
language**

*Close to coding in  
human language.*



# Python main features

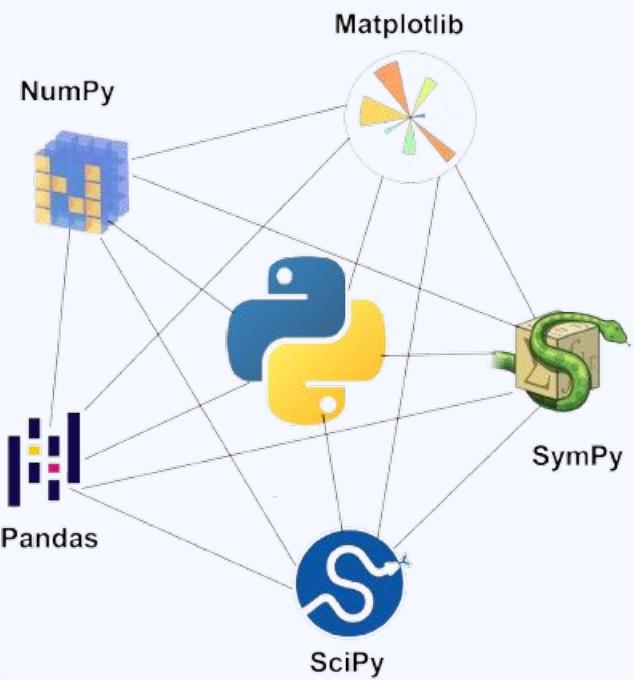
## High-level language

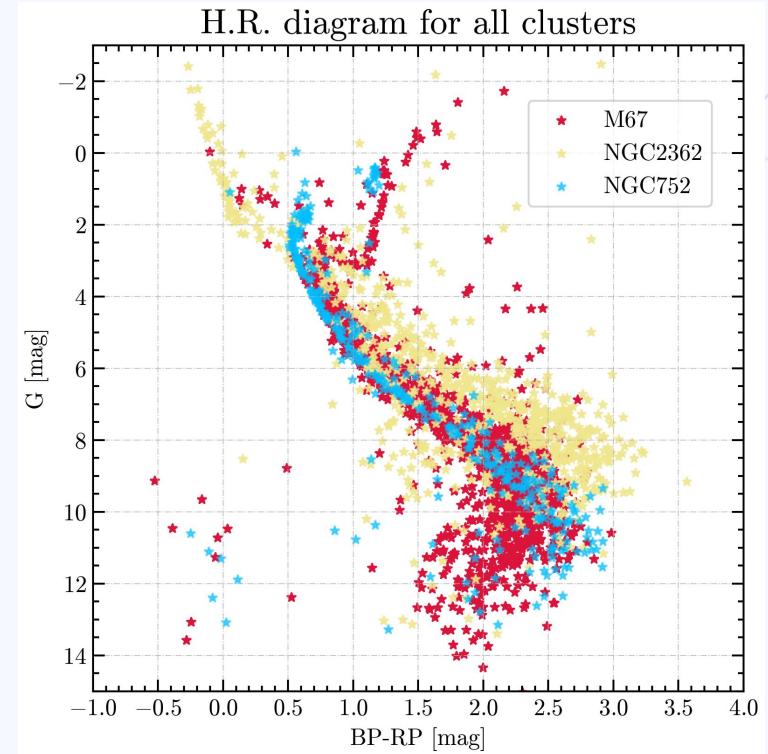
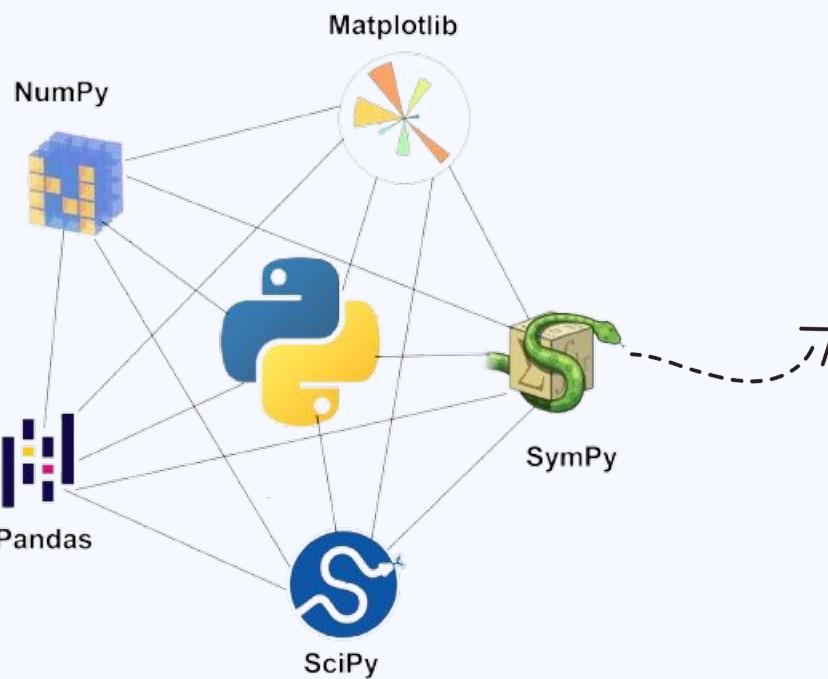
*Close to coding in human language.*



## Tons of good libraries

No need to do it all from scratch!





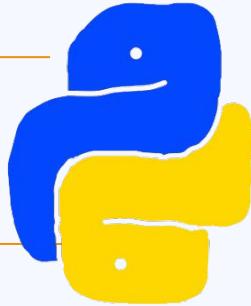
# Python main features

## High-level language

*Close to coding in human language.*

## Tons of good libraries

No need to do it all from scratch!



# Python main features

## High-level language

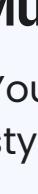
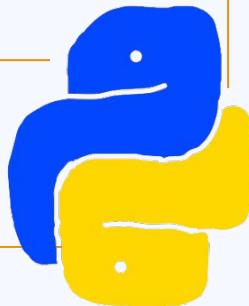
*Close to coding in human language.*

## Tons of good libraries

No need to do it all from scratch!

## Multi-paradigm

You're free to use any style you like!



# Python main features

## High-level language

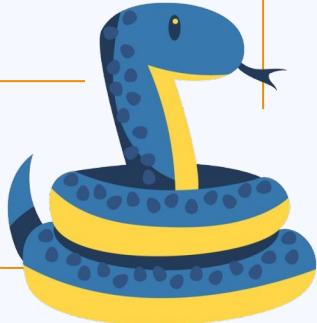
*Close to coding in human language.*

## Tons of good libraries

No need to do it all from scratch!

## Multi-paradigm

You're free to use any style you like!



# Python main features

## High-level language

*Close to coding in human language.*

## Tons of good libraries

No need to do it all from scratch!



## Multi-paradigm

You're free to use any style you like!

## Open-source

Freedom, collaboration, and transparency.



numpy

Public

[Sponsor](#)[Watch 597](#)[Fork 10.5k](#)[Star 28.9k](#)[main](#)[31 Branches](#)[254 Tags](#)[Go to file](#)[t](#)[Add file](#)[Code](#)

 **ngoldbaum** Merge pull request #28383 from star1327p/numba-ctyp...   9962899 · 44 minutes ago  38,113 Commits

 .circleci CI: update circleci to python3.11.10, limit parallel builds. ... 3 months ago

 .devcontainer MAINT: Fix codespaces setup.sh script 2 years ago

 .github cleanup 4 days ago

 .spin MAINT: Update spin lint command 2 months ago

 benchmarks fix benchmarks last month

 branding/logo DOC: correct Logo SVG files rendered in dark by Figma (...) 2 years ago

 doc Merge pull request #28383 from star1327p/numba-ctyp... 44 minutes ago

 meson\_cpu Loongarch: modify lsx optimization(25215PR) for newest... 2 months ago

 numpy BUG: Remove unnecessary copying and casting from out... 10 hours ago

## About

The fundamental package for scientific computing with Python.

 [numpy.org](#)

[python](#)

[numpy](#)

 [Readme](#)

 [View license](#)

 [Code of conduct](#)

 [Security policy](#)

 [Cite this repository](#)

 [Activity](#)

 [Custom properties](#)

 **28.9k stars**

 **597 watching**

 **10.5k forks**



numpy

Public

[Sponsor](#)[Watch 597](#)[Fork 10.5k](#)[Star 28.9k](#)[main](#)[31 Branches](#)[254 Tags](#)[Go to file](#)[t](#)[Add file](#)[Code](#)

 **ngoldbaum** Merge pull request #28383 from star1327p/numba-ctyp...   9962899 · 44 minutes ago  38,113 Commits

 .circleci CI: update circleci to python3.11.10, limit parallel builds. ... 3 months ago

 .devcontainer MAINT: Fix codespaces setup.sh script 2 years ago

 .github 4 days ago

 .spin months ago

 benchmarks last month

 branding/logo DOC: correct Logo SVG files rendered in dark by Figma (...) 2 years ago

 doc Merge pull request #28383 from star1327p/numba-ctyp... 44 minutes ago

 meson\_cpu Loongarch: modify lsx optimization(25215PR) for newest... 2 months ago

 numpy BUG: Remove unnecessary copying and casting from out... 10 hours ago

## Merge pull request #28383

### About

The fundamental package for scientific computing with Python.

 [numpy.org](#)

[python](#)

[numpy](#)

 [Readme](#)

 [View license](#)

 [Code of conduct](#)

 [Security policy](#)

 [Cite this repository](#)

 [Activity](#)

 [Custom properties](#)

 **28.9k stars**

 **597 watching**

 **10.5k forks**



numpy

Public

Sponsor

Watch 597

Fork 10.5k

Star 28.9k

main

31 Branches 254 Tags

Go to file

t

Add file

Code

ngoldbaum Merge pull request #28383 from star1327p/numba-ctyp... 9962899 · 44 minutes ago 38,113 Commits

.circleci

CI: update circleci to python3.11.10, limit parallel builds. ... 3 months ago

.devcontainer

MAINT: Fix codespaces setup.sh script 2 years ago

.github

4 days ago

.spin

months ago

benchmarks

last month

branding/logo

DOC: correct Logo SVG files rendered in dark by Figma (...) 2 years ago

doc

Merge pull request #28383 from star1327p/numba-ctyp... 44 minutes ago

meson\_cpu

Loongarch: modify lsx optimization(25215PR) for newest... 2 months ago

numpy

BUG: Remove unnecessary copying and casting from out... 10 hours ago

Merge pull request #28383

## About

The fundamental package for scientific computing with Python.

[numpy.org](#)

python

numpy

Readme

[View license](#)

Code of conduct

Security policy

Cite this repository

Activity

Custom properties

28.9k stars

597 watching

40.5k forks

**Everyone can contribute to make better software!**



numpy

Public

Sponsor

Watch 597

Fork 10.5k

Star 28.9k

main

31 Branches 254 Tags

Go to file

Add file

Code

ngoldbaum Merge pull request #28383 from star1327p/numba-ctyp... 9962899 · 44 minutes ago 38,113 Commits

.circleci

CI: update circleci to python3.11.10, limit parallel builds. ... 3 months ago

.devcontainer

MAINT: Fix codespaces setup.sh script 2 years ago

.github

4 days ago

.spin

months ago

benchmarks

last month

branding/logo

DOC: correct Logo SVG files rendered in dark by Figma (...) 2 years ago

doc

Merge pull request #28383 from star1327p/numba-ctyp... 44 minutes ago

meson\_cpu

Loongarch: modify lsx optimization(25215PR) for newest... 2 months ago

numpy

BUG: Remove unnecessary copying and casting from out... 10 hours ago

Merge pull request #28383

## About

The fundamental package for scientific computing with Python.

[numpy.org](#)

python numpy

Readme

View license

Code of conduct

Security policy

Cite this repository

Activity

Custom properties

28.9k stars

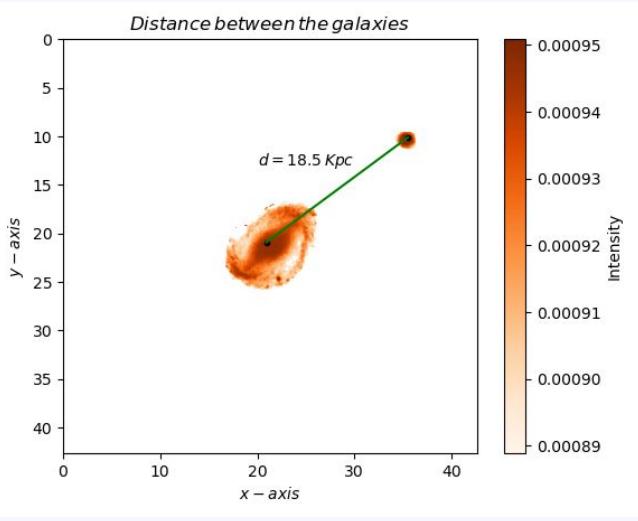
597 watching

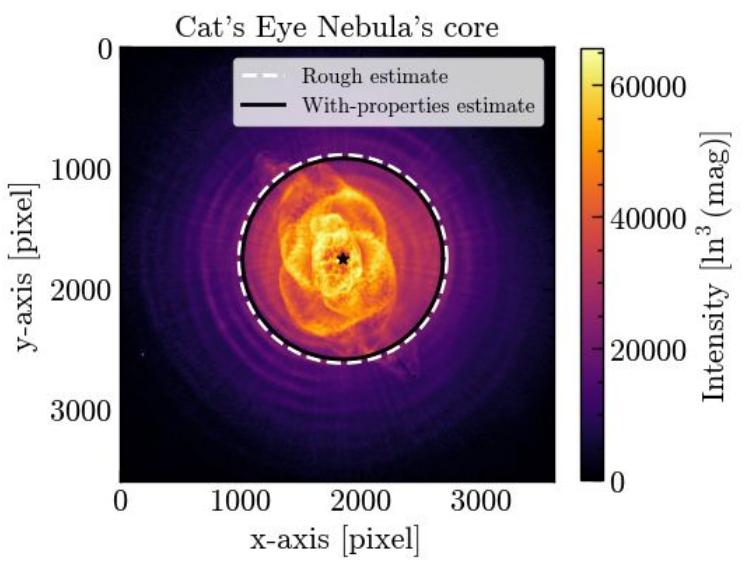
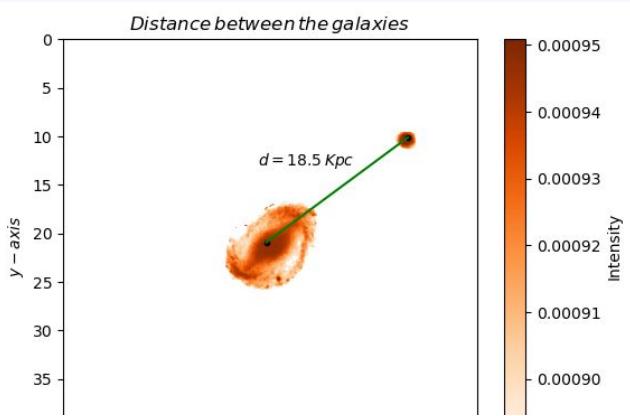
40.5k forks

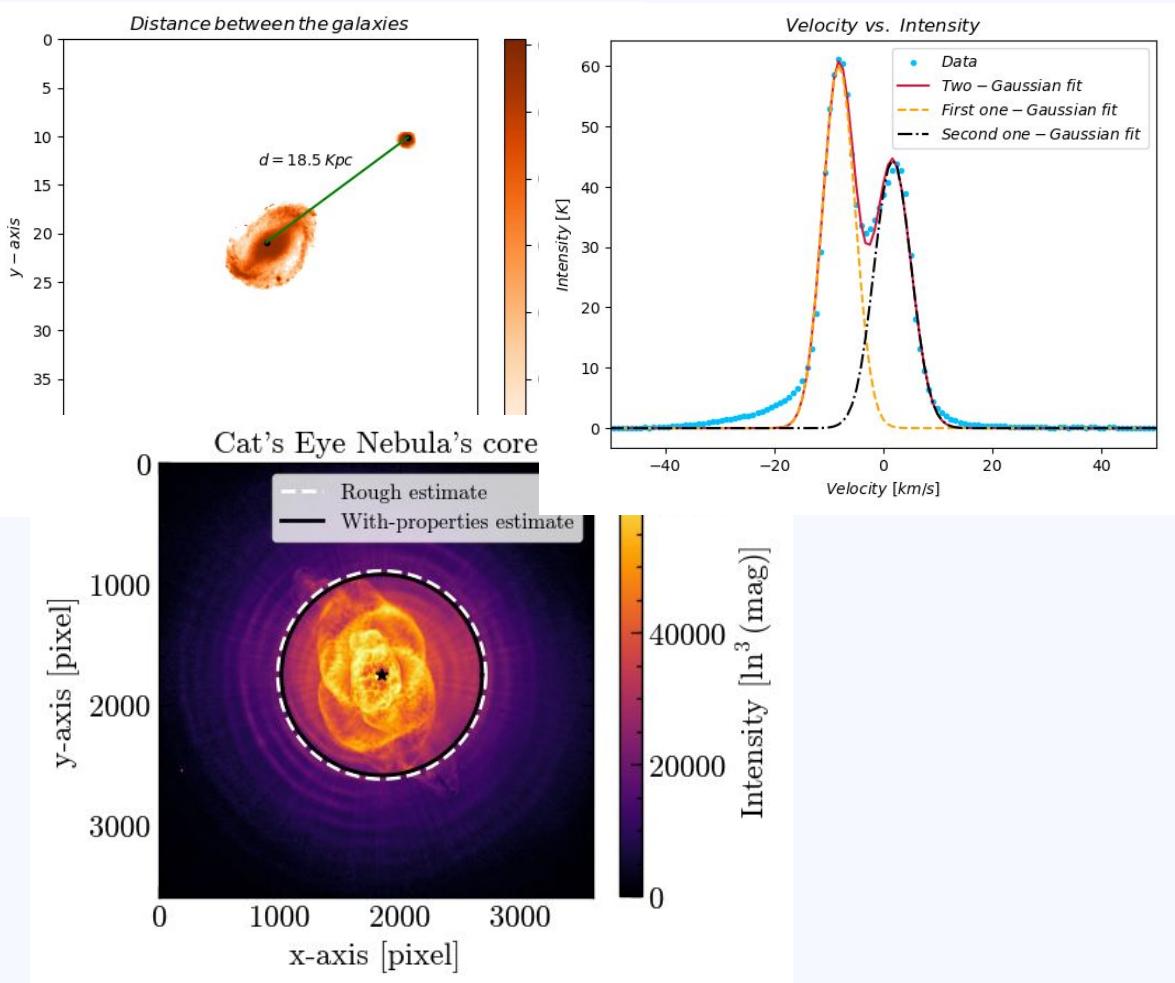
**Everyone can contribute to make better software!**  
And you're no exception.

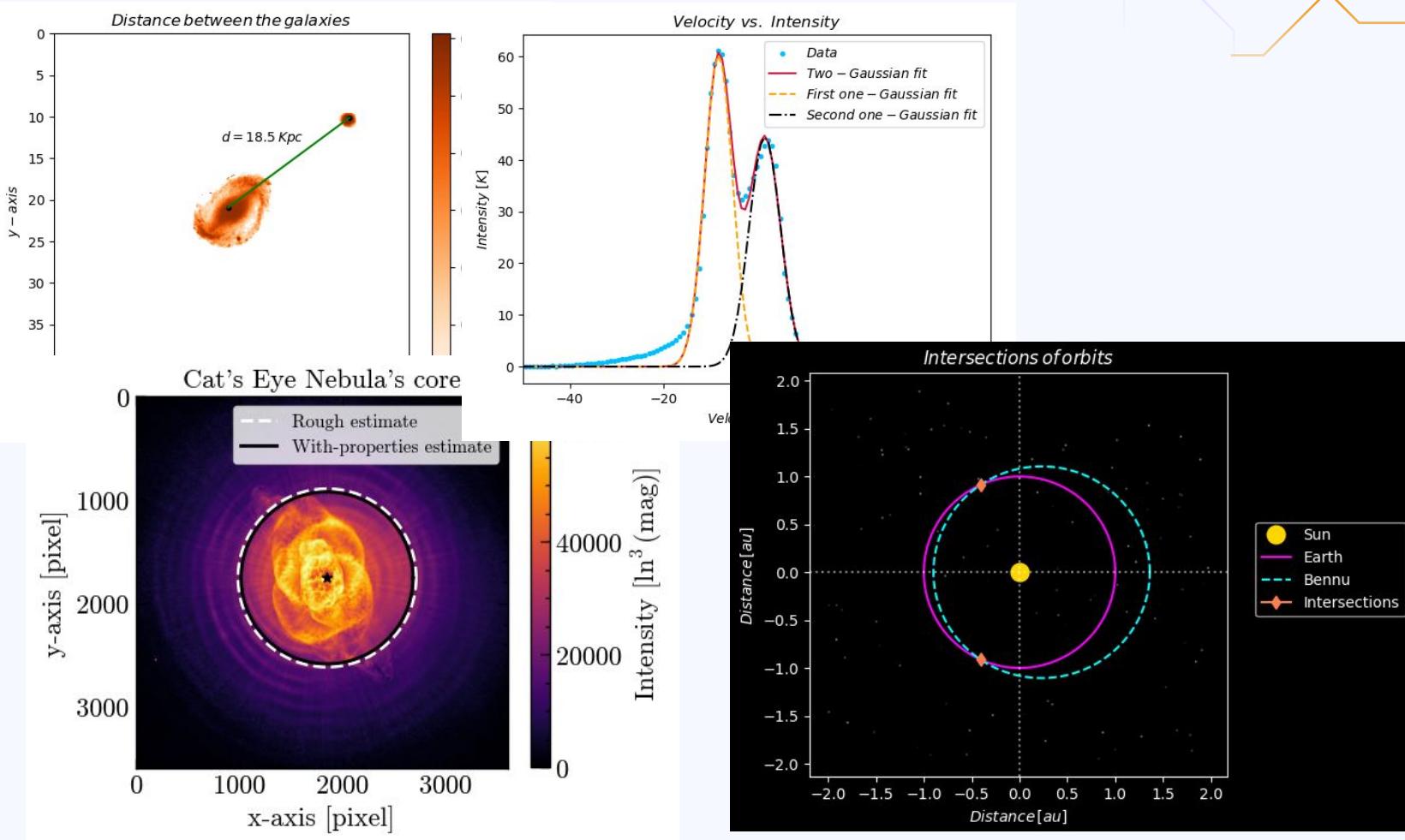
# Python in physics :)

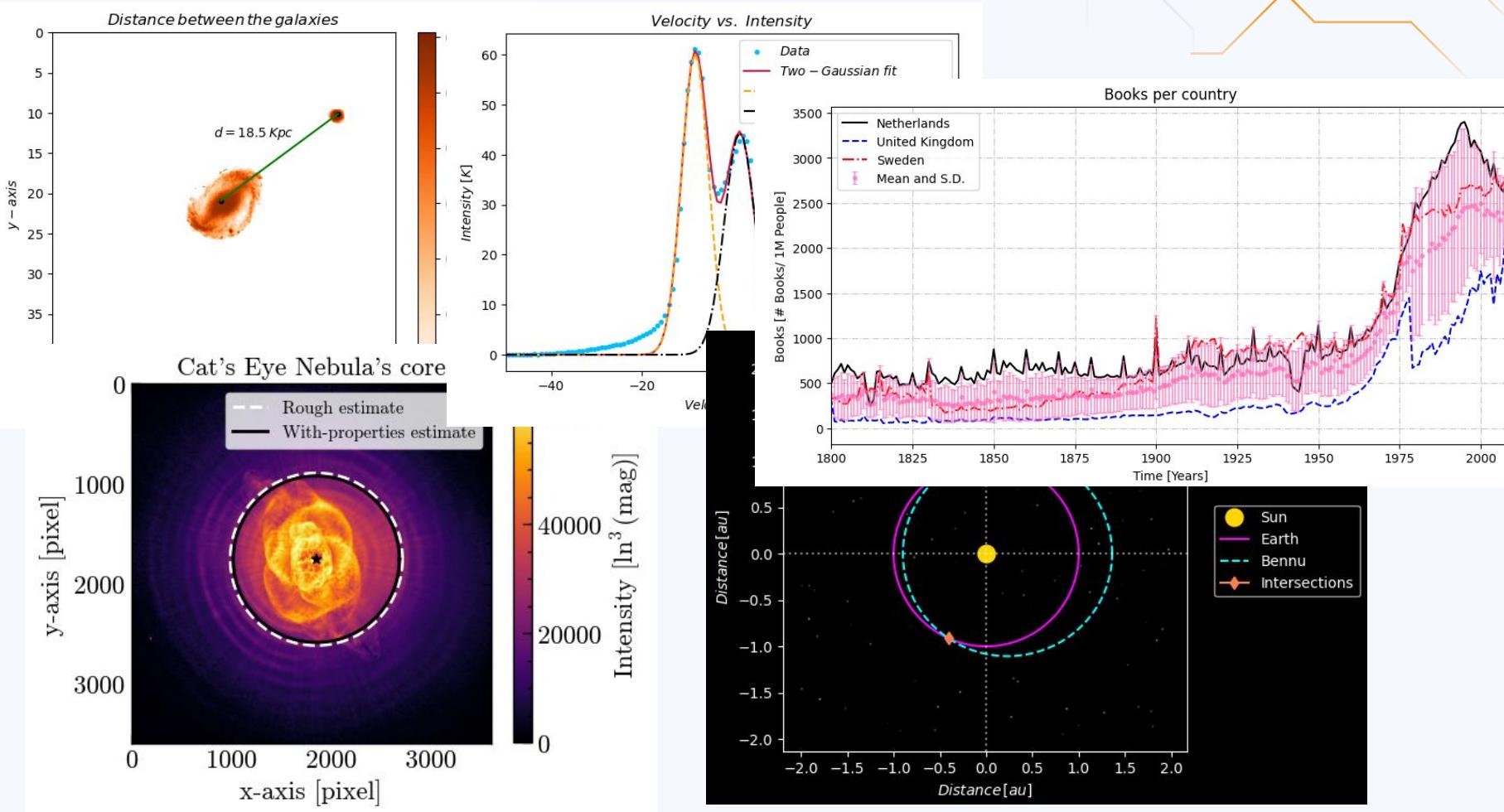
---

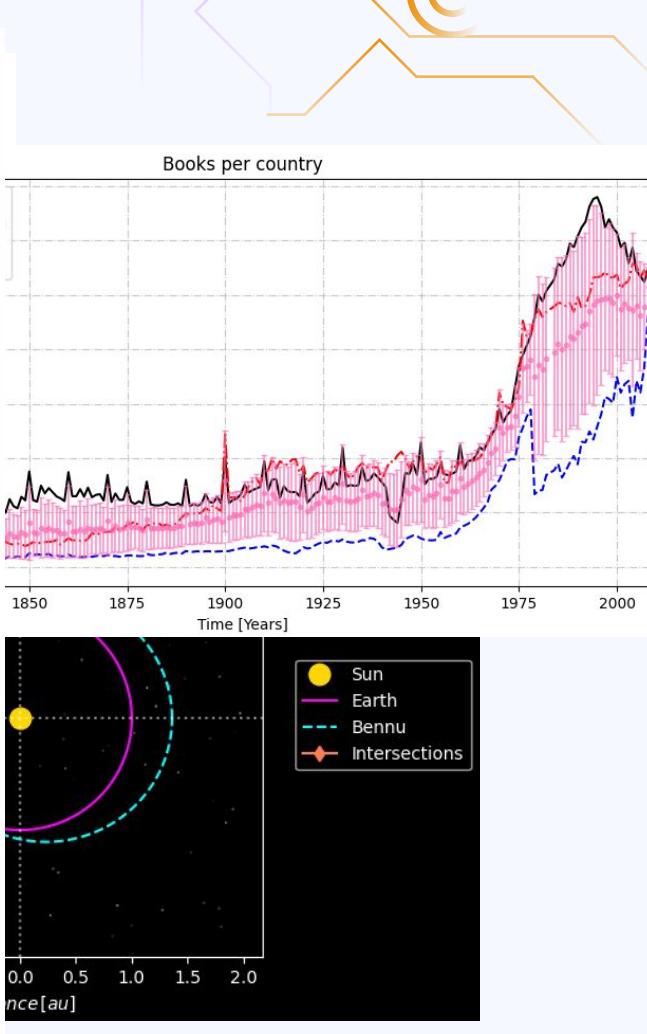
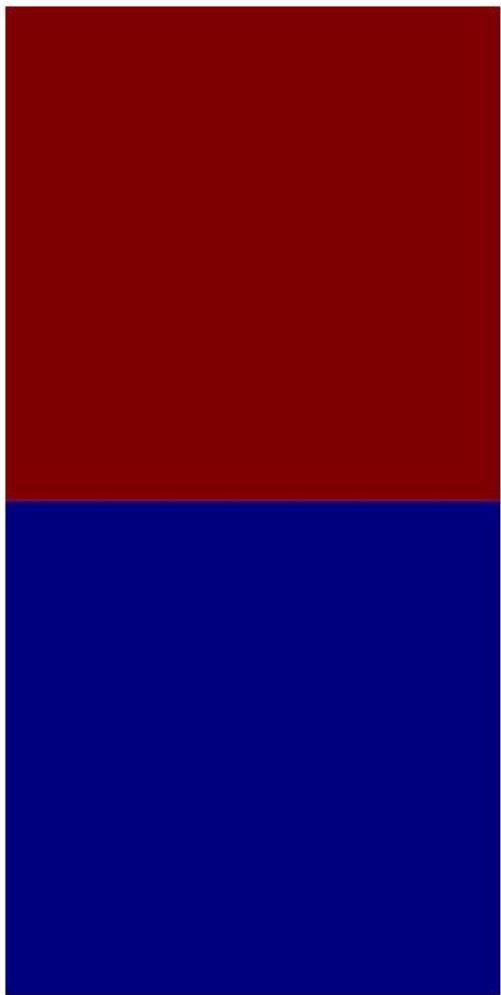
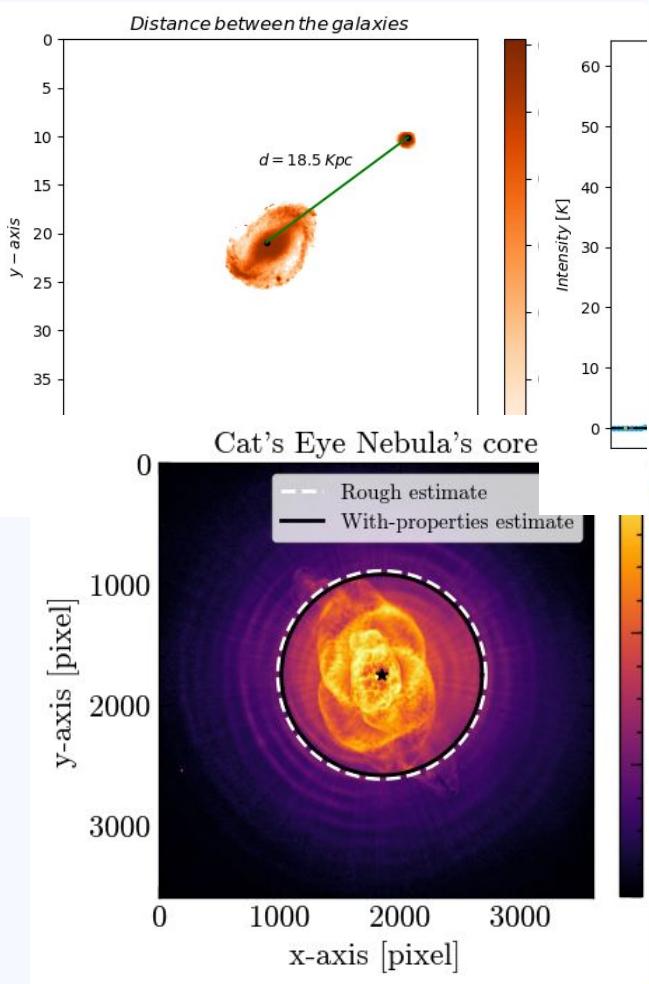














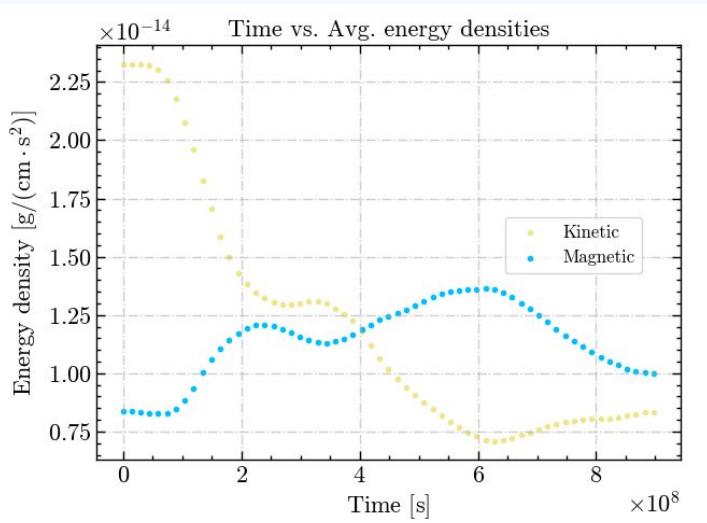
# 1. Data obtention (might include doing some calcs)

Time [s]	Avg. gas temperature [K]	Avg. kinetic energy density [g/(cm s <sup>2</sup> )]	Avg. magnetic energy density [g/(cm s <sup>2</sup> )]	Avg. thermal energy density [g/(cm s <sup>2</sup> )]	Avg. thermal pressure [g/(cm s <sup>2</sup> )]	Avg. density [g/cm <sup>3</sup> ]	
0	0.000000e+000	43.298048	2.323611e-14	8.365000e-15	4.182500e-14	2.788333e-14	4.647222e-24
1	1.485263e+07	43.297840	2.324853e-14	8.351031e-15	4.182655e-14	2.788437e-14	4.647222e-24
2	2.980500e+07	43.293574	2.326259e-14	8.315291e-15	4.184823e-14	2.789882e-14	4.647222e-24
3	4.457468e+07	43.276272	2.321478e-14	8.276087e-15	4.193524e-14	2.795683e-14	4.647222e-24
4	5.920091e+07	43.235030	2.301887e-14	8.259261e-15	4.214798e-14	2.809865e-14	4.647222e-24
...	...	...	...	...	...	...	
56	8.373374e+08	54.652604	8.103655e-15	1.034459e-14	5.497787e-14	3.665191e-14	4.647222e-24
57	8.522842e+08	54.792106	8.179468e-15	1.019357e-14	5.505307e-14	3.670205e-14	4.647222e-24
58	8.673548e+08	54.934191	8.256370e-15	1.007416e-14	5.509558e-14	3.673039e-14	4.647222e-24
59	8.825692e+08	55.095862	8.315059e-15	1.001807e-14	5.509298e-14	3.672865e-14	4.647222e-24
60	8.976000e+08	55.268832	8.344249e-15	1.000056e-14	5.508130e-14	3.672087e-14	4.647222e-24

# 1. Data obtention (might include doing some calcs)

Time [s]	Avg. gas temperature [K]	Avg. kinetic energy density [g/(cm s <sup>2</sup> )]	Avg. magnetic energy density [g/(cm s <sup>2</sup> )]	Avg. thermal energy density [g/(cm s <sup>2</sup> )]	Avg. thermal pressure [g/(cm s <sup>2</sup> )]	Avg. density [g/cm <sup>3</sup> ]	
0	0.000000e+000	43.298048	2.323611e-14	8.365000e-15	4.182500e-14	2.788333e-14	4.647222e-24
1	1.485263e+007	43.297840	2.324853e-14	8.351031e-15	4.182655e-14	2.788437e-14	4.647222e-24
2	2.980500e+007	43.293574	2.326259e-14	8.315291e-15	4.184823e-14	2.789882e-14	4.647222e-24
3	4.457468e+007	43.276272	2.321478e-14	8.276087e-15	4.193524e-14	2.795683e-14	4.647222e-24
4	5.920091e+007	43.235030	2.301887e-14	8.259261e-15	4.214798e-14	2.809865e-14	4.647222e-24
...	...	...	...	...	...	...	
56	8.373374e+008	54.652604	8.103655e-15	1.034459e-14	5.497787e-14	3.665191e-14	4.647222e-24
57	8.522842e+008	54.792106	8.179468e-15	1.019357e-14	5.505307e-14	3.670205e-14	4.647222e-24
58	8.673548e+008	54.934191	8.256370e-15	1.007416e-14	5.509558e-14	3.673039e-14	4.647222e-24
59	8.825692e+008	55.095862	8.315059e-15	1.001807e-14	5.509298e-14	3.672865e-14	4.647222e-24
60	8.976000e+008	55.268832	8.344249e-15	1.000056e-14	5.508130e-14	3.672087e-14	4.647222e-24

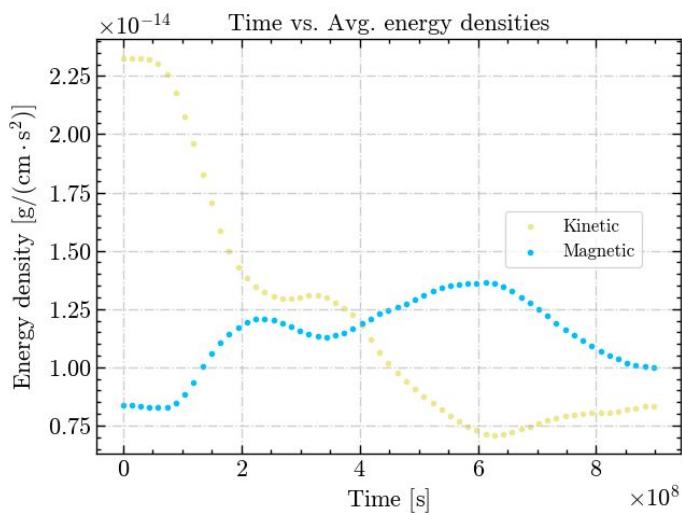
## 2. Analysis & interpretation



# 1. Data obtention (might include doing some calcs)

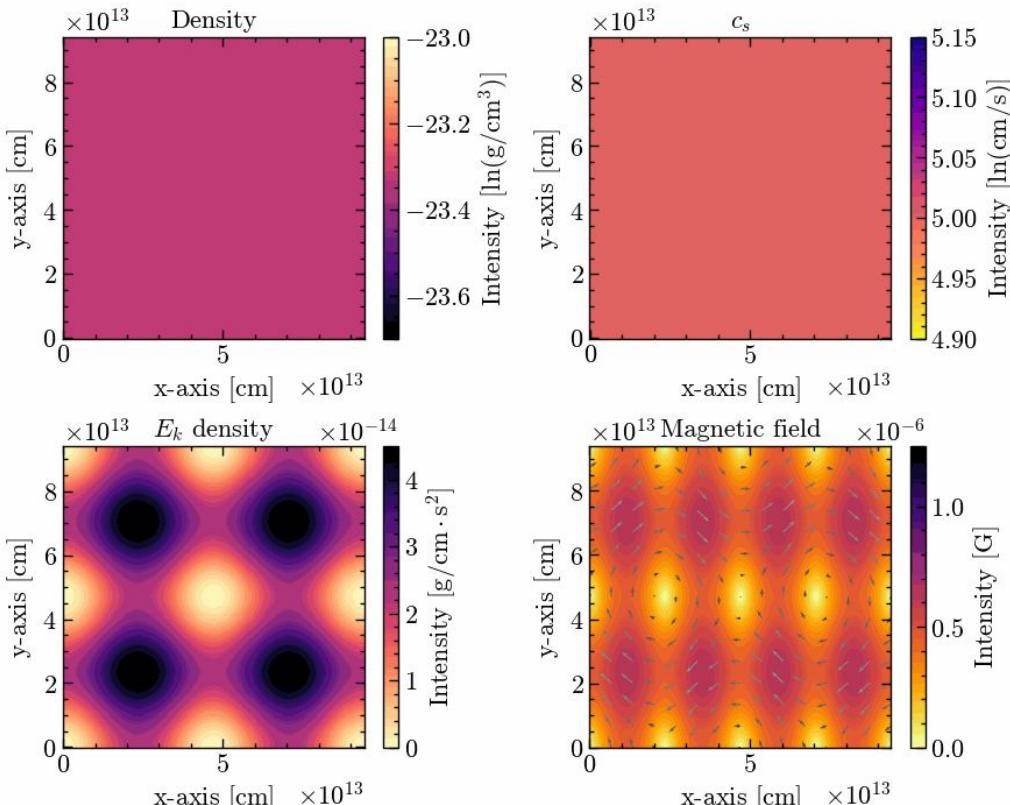
Time [s]	Avg. gas temperature [K]	Avg. kinetic energy density [ $\text{g}/(\text{cm s}^2)$ ]	Avg. magnetic energy density [ $\text{g}/(\text{cm s}^2)$ ]	Avg. thermal energy density [ $\text{g}/(\text{cm s}^2)$ ]	Avg. thermal pressure [ $\text{g}/(\text{cm s}^2)$ ]	Avg. density [ $\text{g}/\text{cm}^3$ ]
0	0.000000e+000	43.298048	2.323611e-14	8.365000e-15	4.182500e-14	4.647222e-24
1	1.485263e+007	43.297840	2.324853e-14	8.351031e-15		
2	2.980500e+007	43.293574	2.326259e-14	8.315291e-15		
3	4.474680e+007	43.276272	2.321478e-14	8.276087e-15		
4	5.920091e+007	43.235030	2.301887e-14	8.259261e-15		
...	...	...	...	...		
56	8.373374e+008	54.652604	8.103655e-15	1.034459e-14		
57	8.522842e+008	54.792106	8.179468e-15	1.019357e-14		
58	8.673548e+008	54.934191	8.256370e-15	1.007416e-14		
59	8.825692e+008	55.095862	8.315059e-15	1.001807e-14		
60	8.976000e+008	55.268832	8.344249e-15	1.000056e-14		

# 2. Analysis & interpretation



# 3. Present results!

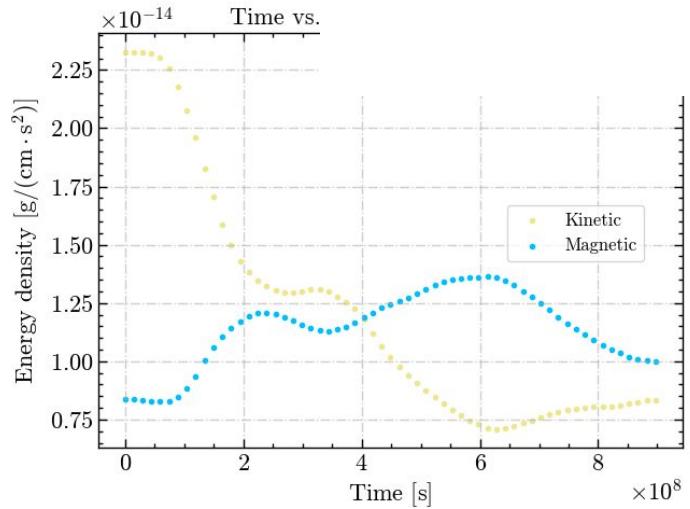
State at  $t = 0.00\text{e}+00$  seconds



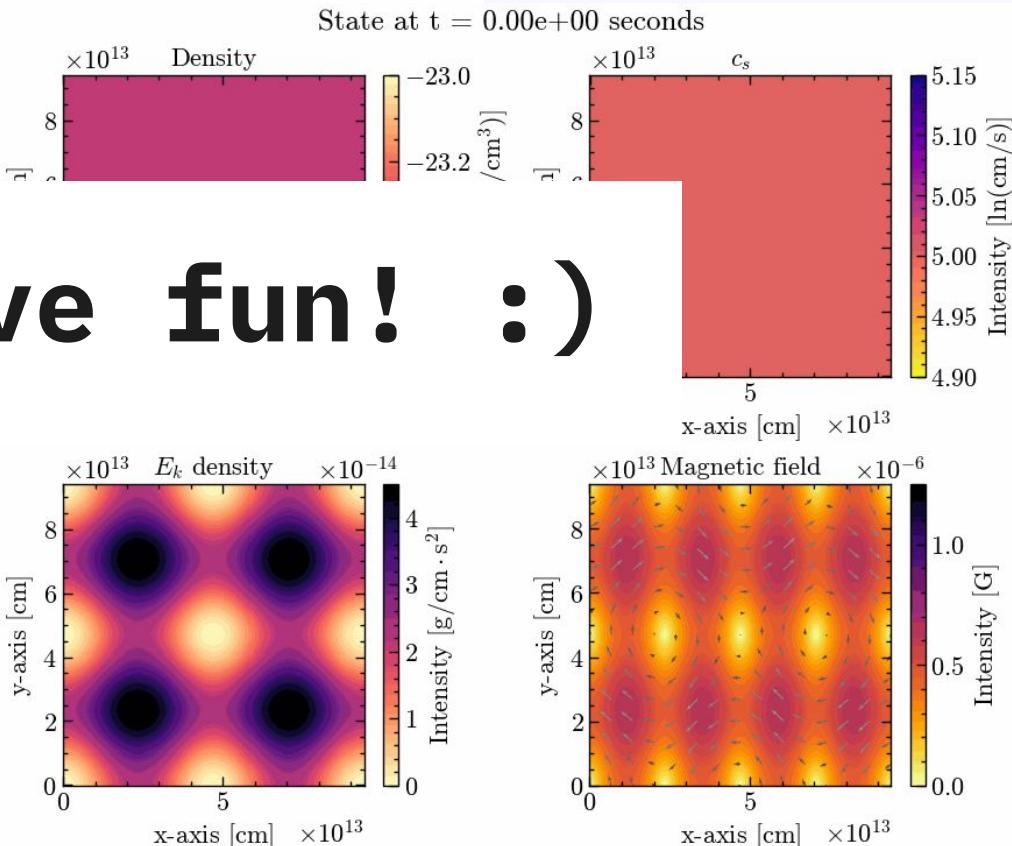
# 1. Data obtention (might include doing some calcs)

Time [s]	Avg. gas temperature [K]	Avg. kinetic energy density [ $\text{g}/(\text{cm s}^2)$ ]	Avg. magnetic energy density [ $\text{g}/(\text{cm s}^2)$ ]	Avg. thermal energy density [ $\text{g}/(\text{cm s}^2)$ ]	Avg. thermal pressure [ $\text{g}/(\text{cm s}^2)$ ]	Avg. density [ $\text{g}/\text{cm}^3$ ]
0	0.000000e+000	43.298048	2.323611e-14	8.365000e-15	4.182500e-14	4.647222e-24
1	1.485263e+007	43.297840	2.324853e-14	8.351031e-15		
2	2.980500e+007	43.293574	2.326259e-14	8.315291e-15		
3	4.474680e+007	43.276272	2.321478e-14	8.276087e-15		
4	5.920091e+007	43.235030	2.301887e-14	8.259261e-15		
...	...	...	...	...		
56	8.373374e+008	54.652604	8.103655e-15	1.034459e-14		
57	8.522842e+008	54.792106	8.179468e-15	1.019357e-14		
58	8.673548e+008	54.934191				
59	8.825692e+008	55.095862				
60	8.976000e+008	55.268832				

# 2. Analysis &



# 4. Have fun! :)



# 3. Present results!

# Problem: Get the average distance in a unit square.

In principle, we would calculate it with this integral

$$\iiint \int_{[0,1]^4} \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} dx_0 dx_1 dy_0 dy_1,$$

but the operation is computationally expensive to do (try it!). We need to look at the problem in a different way.

# Problem: Get the average distance in a unit square.

In principle, we would calculate it with this integral

$$\iiint \int_{[0,1]^4} \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} dx_0 dx_1 dy_0 dy_1,$$

but the operation is computationally expensive to do (try it!). We need to look at the problem in a different way.

## Solution

Via probability theory, the integral above can be found to be equivalent to

$$\langle d \rangle = 4 \int \int_{[0,1]^2} \underbrace{(1-x)(1-y)}_{\text{PDF}} \underbrace{\sqrt{x^2 + y^2}}_{\text{distance}} dx dy.$$

Remember that an expected value is obtained by multiplying the value by its probability.

# Problem: Get the average distance in a unit square.

```
# Define the symbols  
x, y = sp.symbols("x y", real = True)  
  
# and the integrand  
integrand = 4*(1-x)*(1-y)*sp.sqrt(x**2 + y**2)
```

Python

# Problem: Get the average distance in a unit square.

```
# Define the symbols  
x, y = sp.symbols("x y", real = True)  
  
# and the integrand  
integrand = 4*(1-x)*(1-y)*sp.sqrt(x**2 + y**2)
```

Python

```
# 1st integral  
first = sp.integrate(integrand, (x, 0, 1))  
  
# 2nd integral  
result = sp.integrate(first, (y, 0, 1))
```

Python

# Problem: Get the average distance in a unit square.

```
# Define the symbols
x, y = sp.symbols("x y", real = True)

# and the integrand
integrand = 4*(1-x)*(1-y)*sp.sqrt(x**2 + y**2)
```

Python

```
# 1st integral
first = sp.integrate(integrand, (x, 0, 1))

# 2nd integral
result = sp.integrate(first, (y, 0, 1))
```

Python

```
# See the first result
first
```

Python

$$-(1-y) \left( 2y^2 \begin{cases} 0 & \text{for } y^2 > 0 \\ \log(2|y|) & \text{otherwise} \end{cases} - \frac{4y^2|y|}{3} \right) + (1-y) \left( 2y^2 \begin{cases} \operatorname{asinh}\left(\frac{1}{|y|}\right) & \text{for } y^2 > 0 \\ \log\left(2\sqrt{y^2+1}+2\right) & \text{otherwise} \end{cases} + \left(\frac{2}{3} - \frac{4y^2}{3}\right)\sqrt{y^2+1} \right)$$

# Problem: Get the average distance in a unit square.

```
# Define the symbols  
x, y = sp.symbols("x y", real = True)  
  
# and the integrand  
integrand = 4*(1-x)*(1-y)*sp.sqrt(x**2 + y**2)
```

```
# 1st integral  
first = sp.integrate(integrand, (x, 0, 1))  
  
# 2nd integral  
result = sp.integrate(first, (y, 0, 1))
```

```
# See the first result  
first
```

```
# And the exact result  
result
```

Python

$$\frac{\sqrt{2}}{15} + \frac{2}{15} + \frac{\log(1 + \sqrt{2})}{3}$$

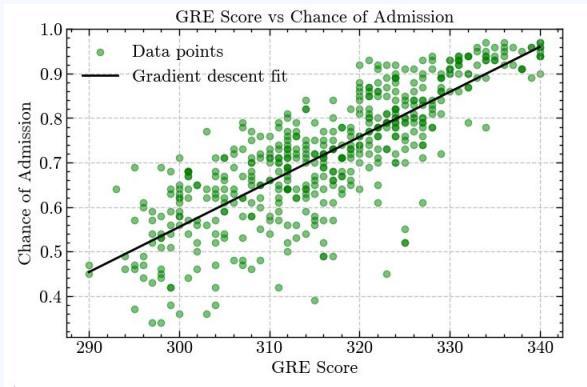
Python

Not so awful.

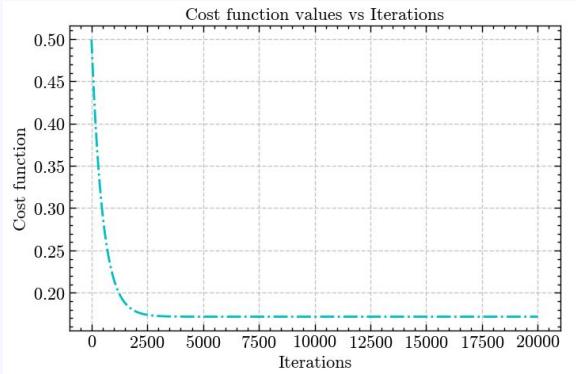
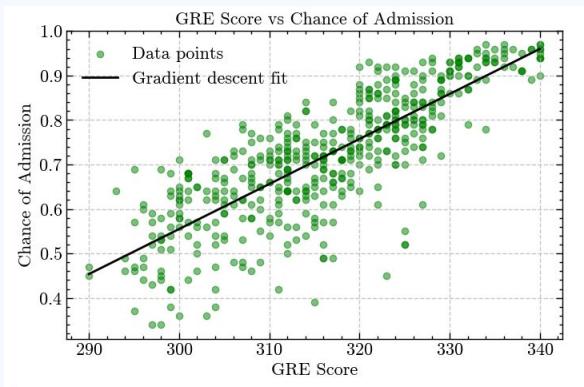
$$-(1 - y) \left( 2y^2 \begin{cases} 0 & \text{for } y^2 > 0 \\ \log(2|y|) & \text{otherwise} \end{cases} - \frac{4y^2|y|}{3} \right) + (1 - y) \left( 2y^2 \begin{cases} \operatorname{asinh}\left(\frac{1}{|y|}\right) & \text{for } y^2 > 0 \\ \log\left(2\sqrt{y^2 + 1} + 2\right) & \text{otherwise} \end{cases} + \left(\frac{2}{3} - \frac{4y^2}{3}\right)\sqrt{y^2 + 1} \right)$$

Python

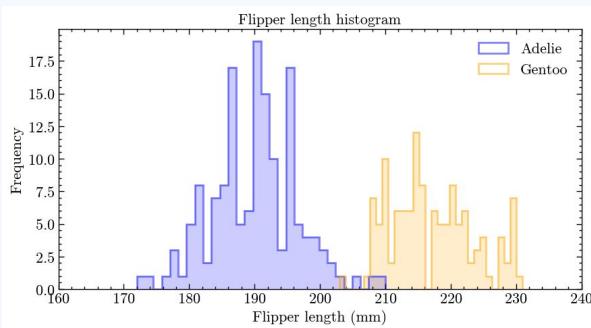
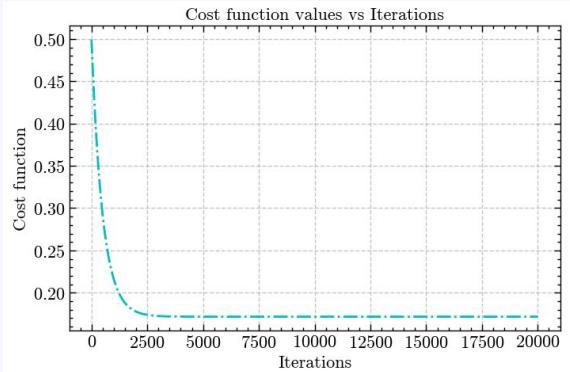
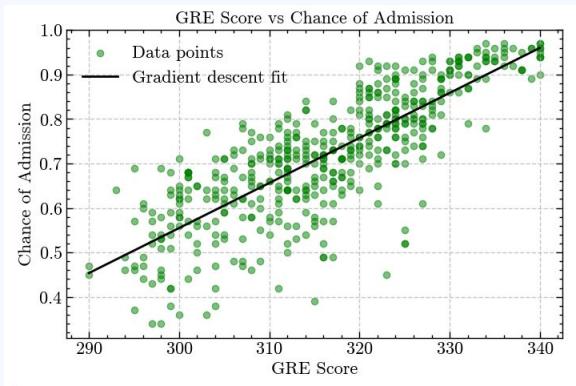
# Machine learning too :)



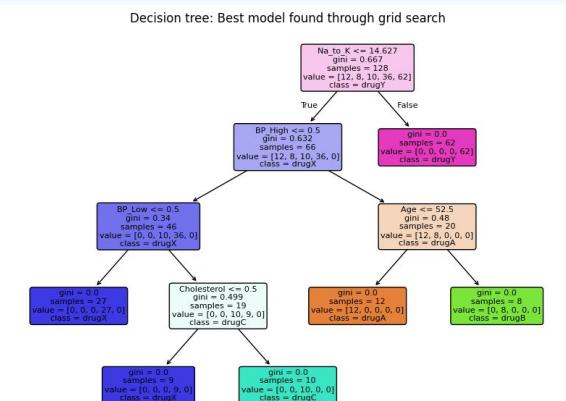
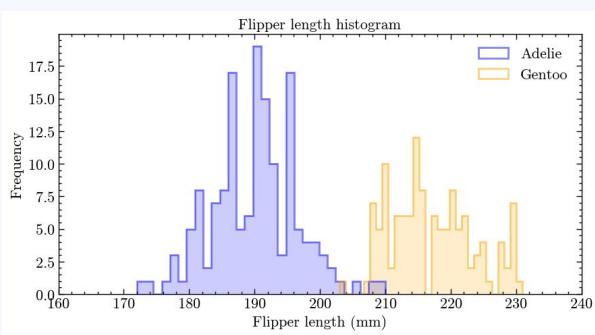
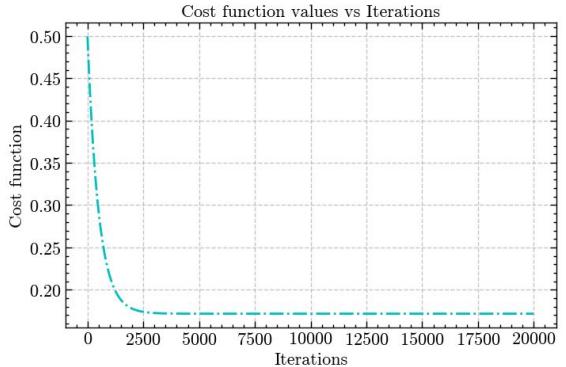
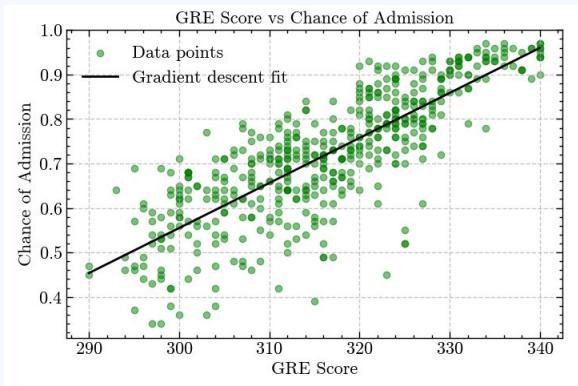
# Machine learning too :)



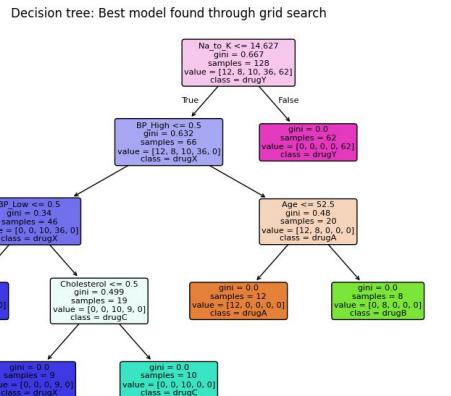
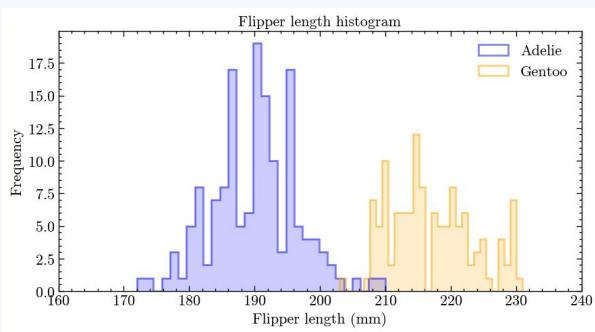
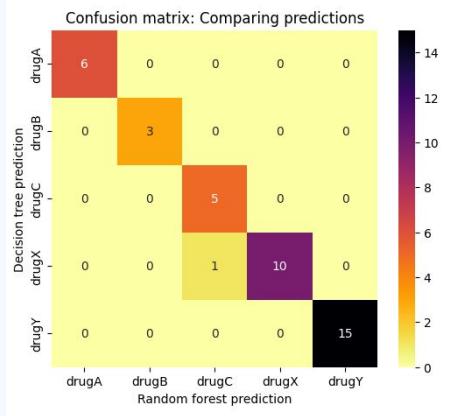
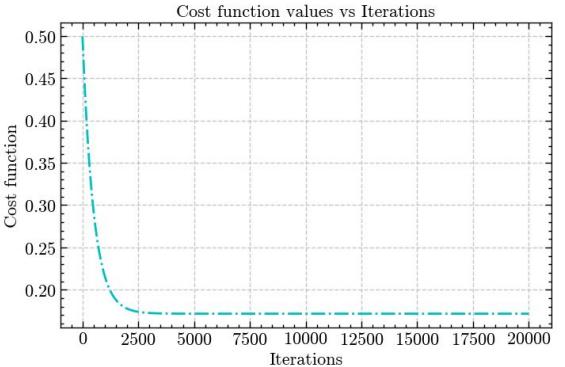
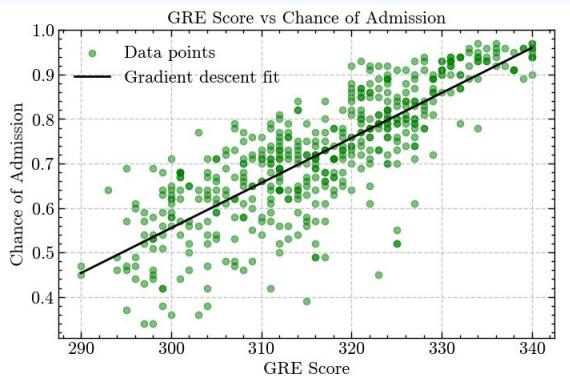
# Machine learning too :)



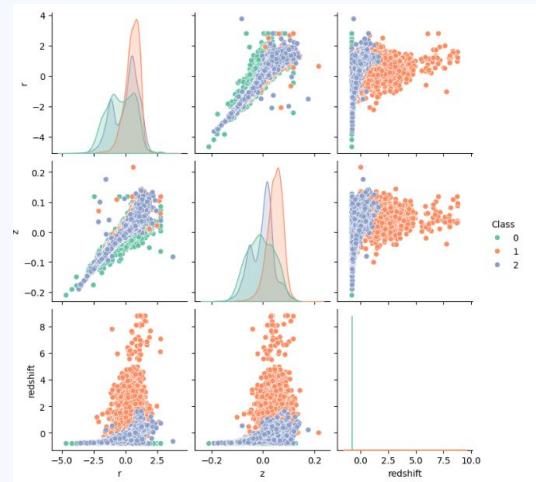
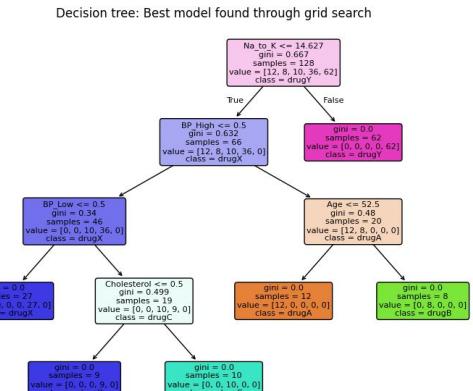
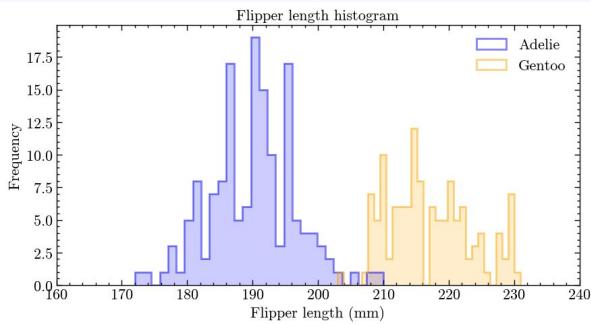
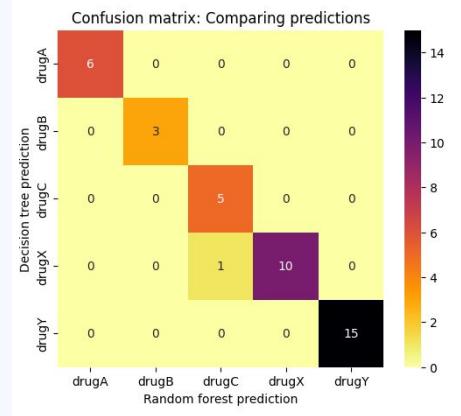
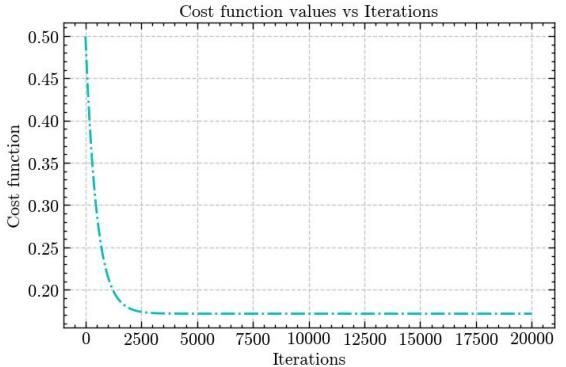
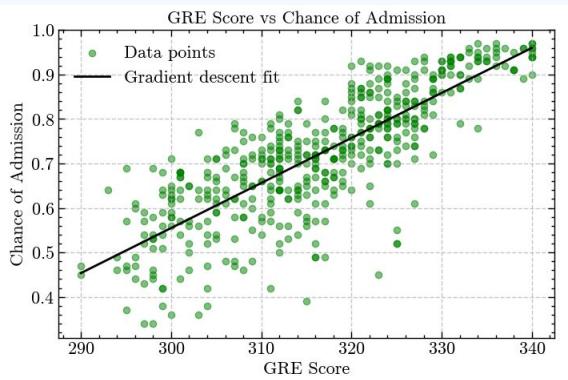
# Machine learning too :)



# Machine learning too :)



# Machine learning too :)



**Or anything you're interested in!**

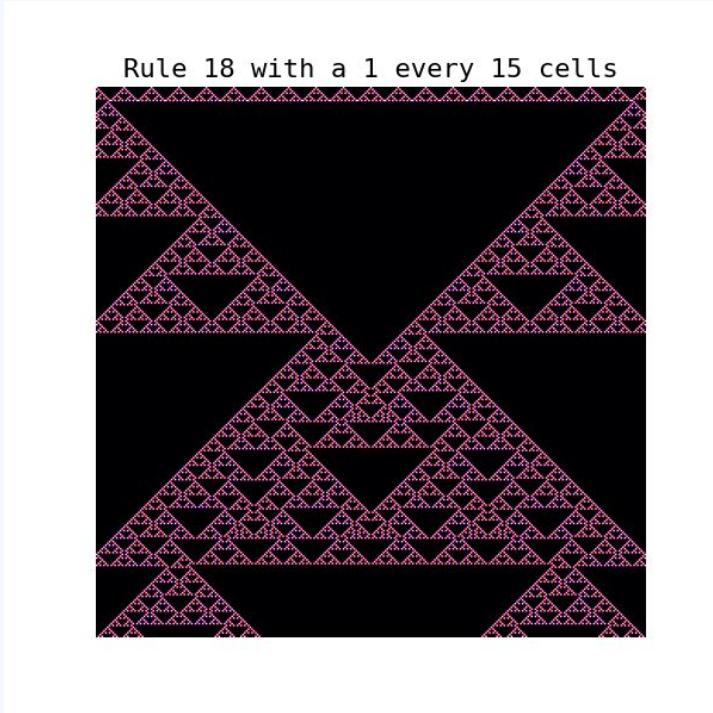


Or anything you're interested in!

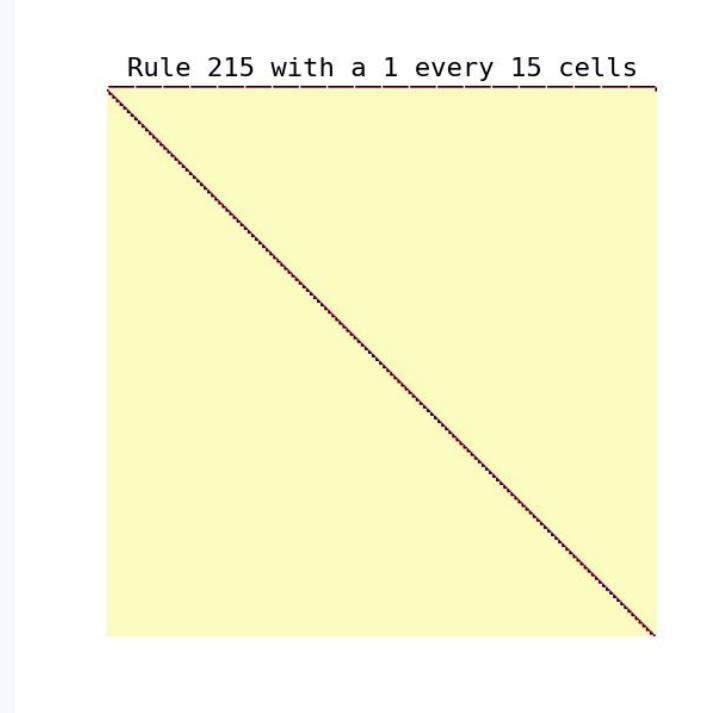
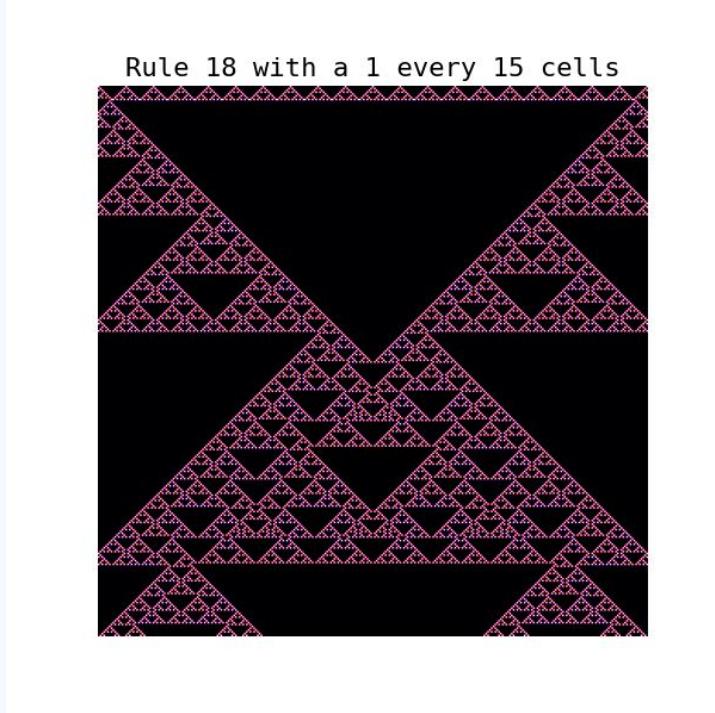
In my case, Cellular Automata.



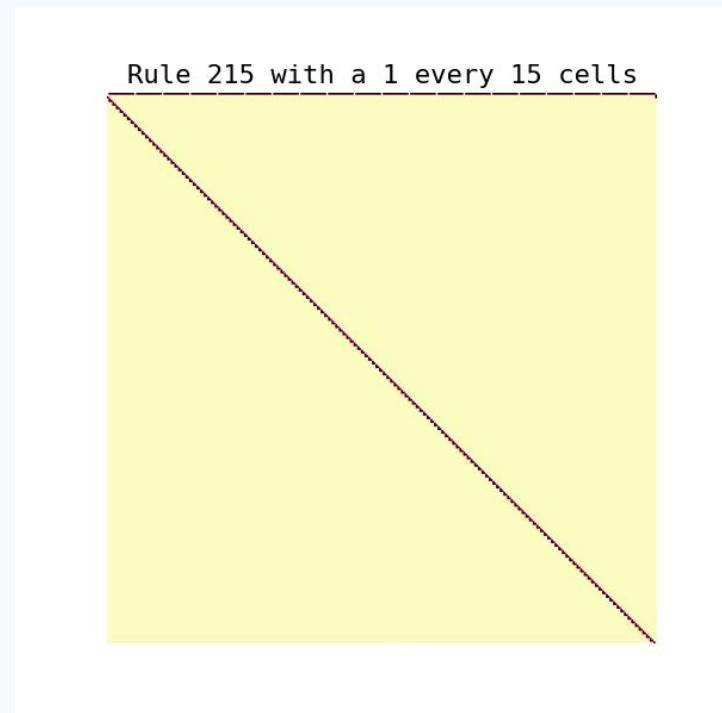
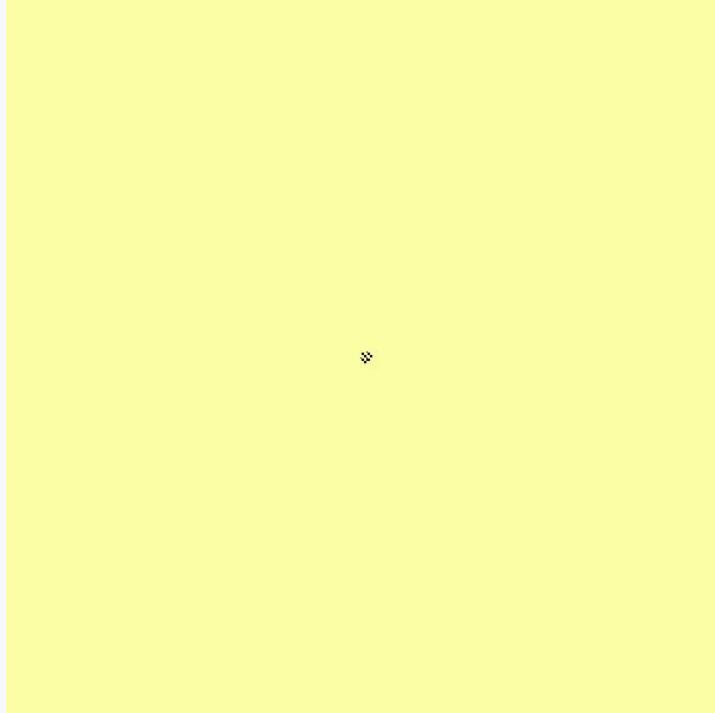
Or anything you're interested in!  
In my case, Cellular Automata.



Or anything you're interested in!  
In my case, Cellular Automata.

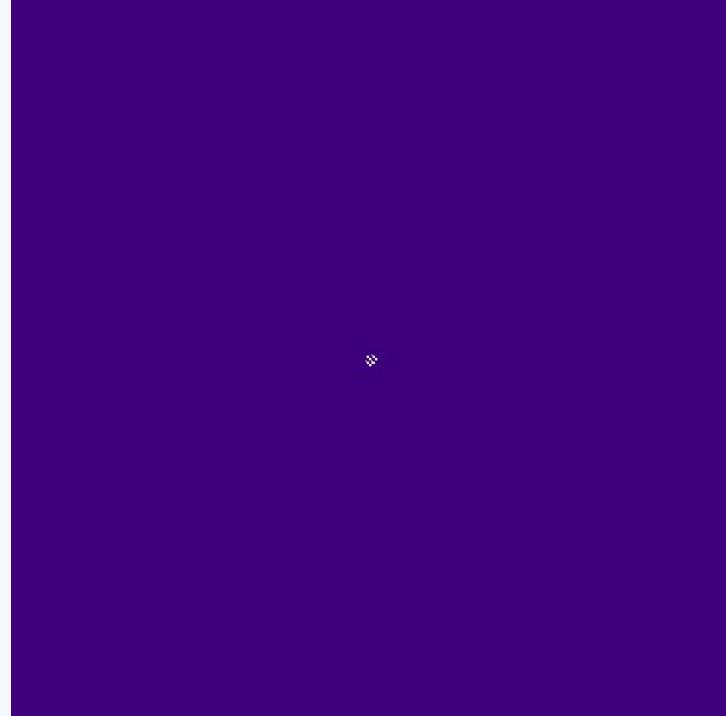
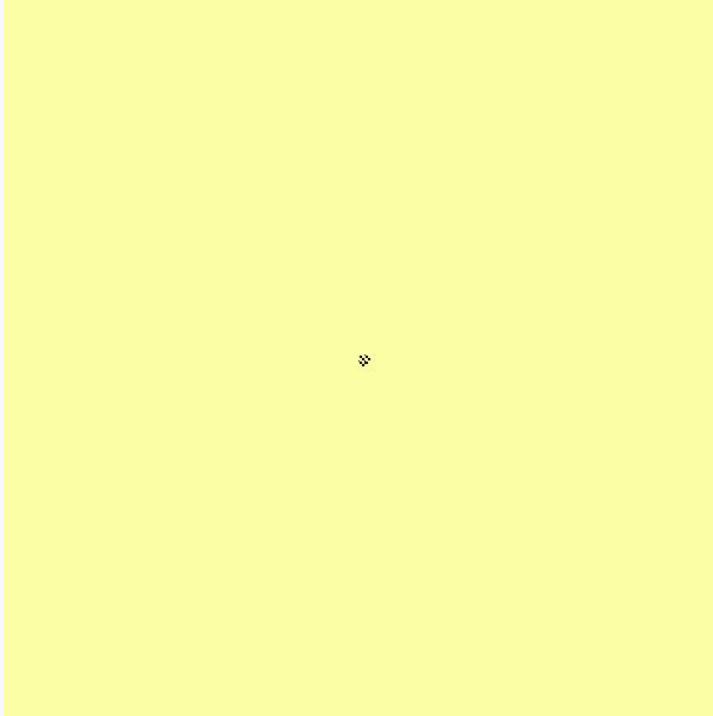


Or anything you're interested in!  
In my case, Cellular Automata.



**Or anything you're interested in!**

**In my case, Cellular Automata.**



Or anything you're interested in!

In my case, Cellular Automata.



# How can you get python?

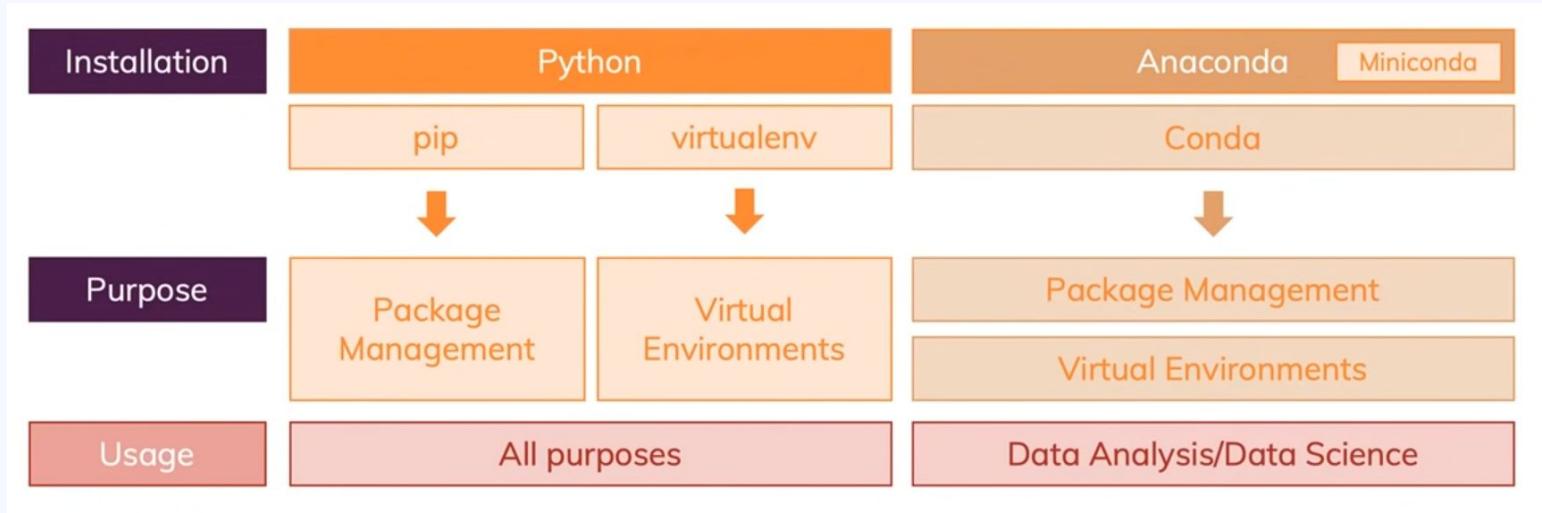
There are two ways you can install it into your system:

Directly in your computer

Via package management system



# How can you get python?

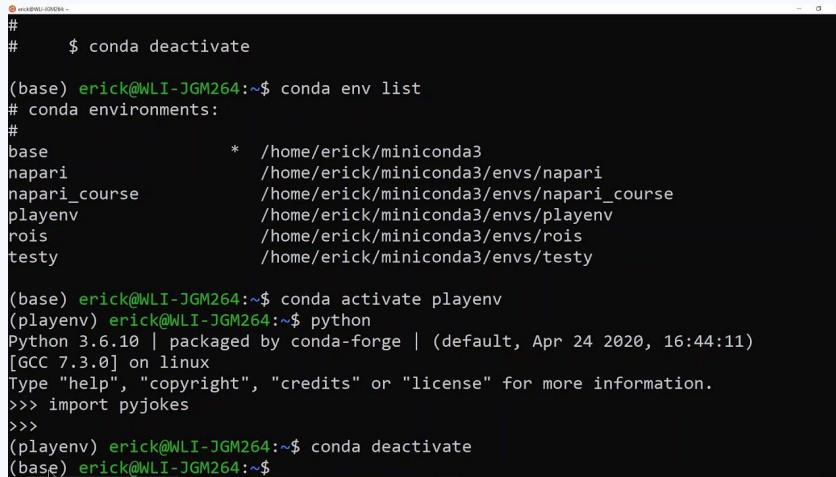


# Why use a package manager?

It can avoid a lot of problems within packages.

Let's you install different versions of Python within your system

Organize your libraries and packages associated to an environment.



The screenshot shows a terminal window with the following session:

```
#      $ conda deactivate
(base) erick@WLI-JGM264:~$ conda env list
# conda environments:
#
base            * /home/erick/miniconda3
napari          /home/erick/miniconda3/envs/napari
napari_course   /home/erick/miniconda3/envs/napari_course
playenv         /home/erick/miniconda3/envs/playenv
rois             /home/erick/miniconda3/envs/rois
testy           /home/erick/miniconda3/envs/testy

(base) erick@WLI-JGM264:~$ conda activate playenv
(playenv) erick@WLI-JGM264:~$ python
Python 3.6.10 | packaged by conda-forge | (default, Apr 24 2020, 16:44:11)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyjokes
>>>
(playenv) erick@WLI-JGM264:~$ conda deactivate
(base) erick@WLI-JGM264:~$
```

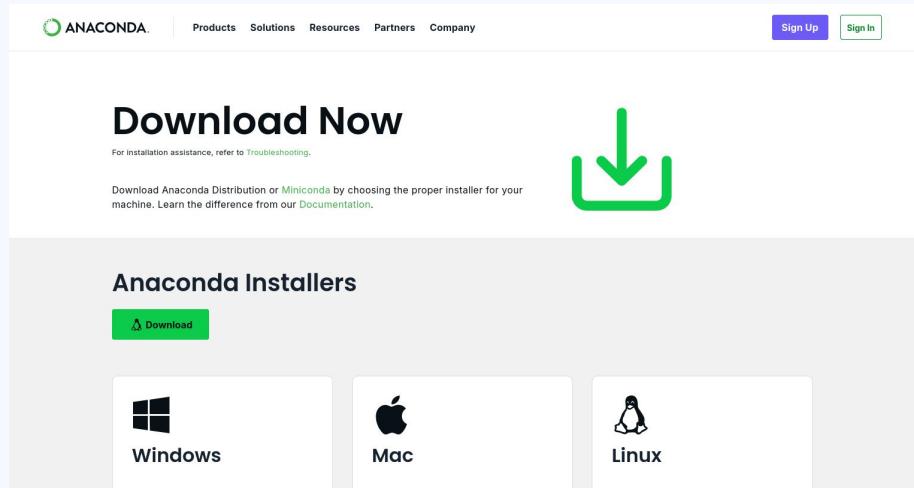


# Let's install Python Together!

---

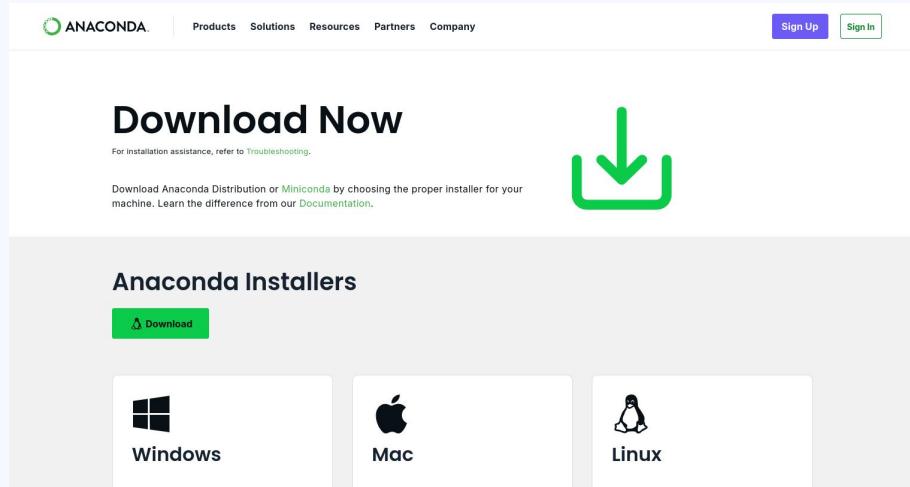
# Windows

1. Visit <https://www.anaconda.com/download>
2. Download the installer
3. Run the installer and enjoy!!



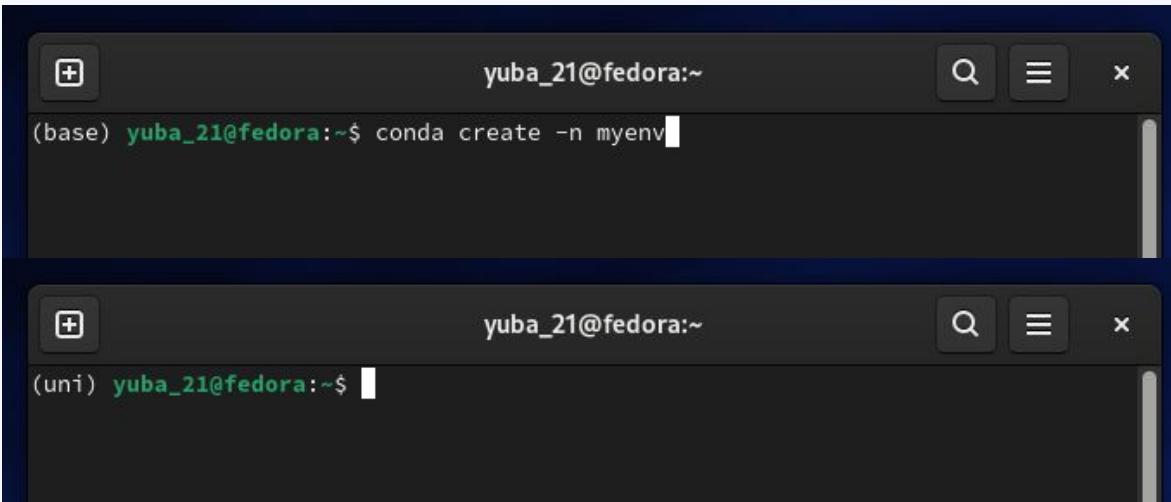
# Linux

1. Visit <https://www.anaconda.com/download>
2. Download the installer.
3. Run `bash <file-you-just-downloaded>.sh` on your terminal.
4. You have conda installed now!



# Some recommendations

1. Do not install anything in the base environment
2. Start with creating a new environment for your work
  - a. Run `conda create -n myenv python=3.9`
  - b. Check with `conda env list`
  - c. Change your working environment `conda activate myenv` Here you should see in your terminal if you changed from the base environment.
  - d. Install some packages! Ex. `conda install numpy`



The image displays two terminal windows side-by-side, both titled "yuba\_21@fedora:~". The top terminal window shows the command `(base) yuba_21@fedora:~$ conda create -n myenv` being typed into the terminal. The bottom terminal window shows the command `(uni) yuba_21@fedora:~$` being typed into the terminal. Both terminals have a dark background and light-colored text.



# Let's get started!

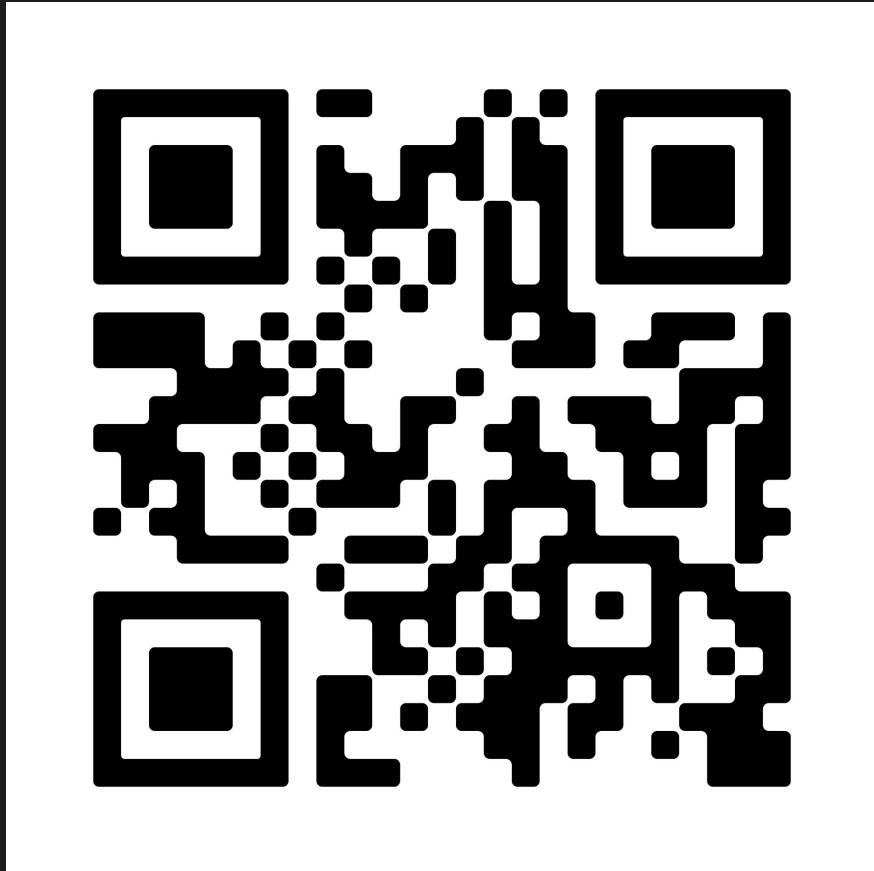
---

# Git and GitHub



# CLUB REPOSITORY

[https://github.com/DaVas1410/Computational\\_Physics\\_Club](https://github.com/DaVas1410/Computational_Physics_Club)

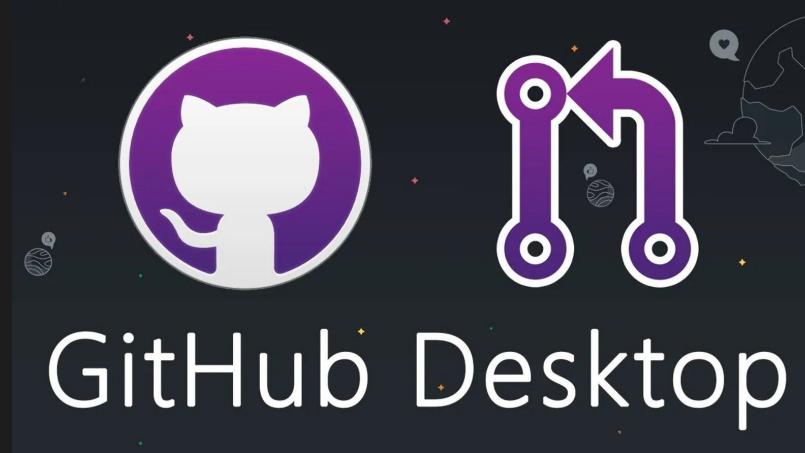


# What is Git?

- Version Control System
  - Keep careful track of changes in your files
  - Collaborate with others on your projects more easily
  - Test changes without losing the original versions
  - Revert back to older versions when/if needed
- GitHub: web-based hosting service for git
  - Provides a "remote" location for storing your git workspaces
  - Useful if you lose/break your computer, etc.

# GitHub Desktop

<https://desktop.github.com/download/>



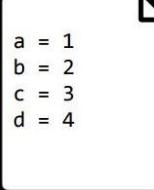
# Using Git

- Installation
  - <https://github.com/join>
  - <https://help.github.com/articles/set-up-git/>
- How it works
  - Create a "repository" (workspace) for your project
  - Add/remove/save/edit files
  - Push local files online to GitHub / pull remote files from GitHub to your local workspace
  - And more!

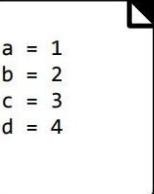
# git clone <url>

- Downloads an existing repository from GitHub
- Creates a synced, local copy

```
git clone <url>
```



```
a = 1  
b = 2  
c = 3  
d = 4
```



```
a = 1  
b = 2  
c = 3  
d = 4
```

# git add

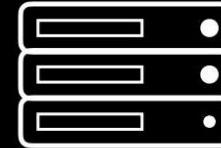
- Signals to git that the specified file should be “tracked” for changes
  - Places modified file(s) in the “staging area”
- Files not added in this way are essentially ignored by git
- git add -A signals to git that it should track all existing files

```
git add <filename>
```

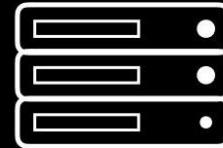
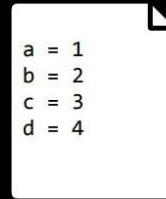


```
a = 1  
b = 2  
c = 3  
d = 4
```

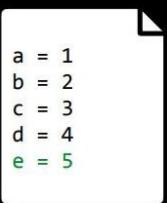
```
a = 1  
b = 2  
c = 3  
d = 4
```



```
git add <filename>
```



```
git add foo.py
```



Changes to be committed:

modified: foo.py

```
git commit -m "message"
```

- Takes a "snapshot" of all files currently on the staging area and commits it to git's memory
- The "snapshot" is captioned with the given message as a brief description for the commit

```
git commit -m "message"
```

A white outline icon of a document with a folded bottom-right corner, containing code. The code is:

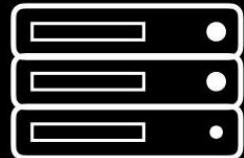
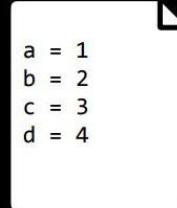
```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

The last line, 'e = 5', is colored green, indicating it is a new addition.A white outline icon of a single document file, containing code. The code is:

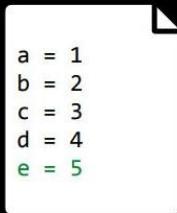
```
a = 1  
b = 2  
c = 3  
d = 4
```



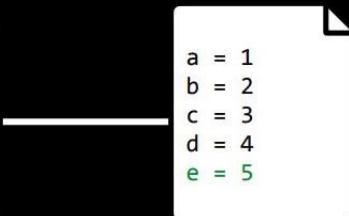
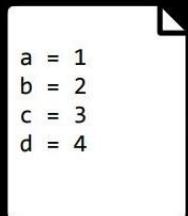
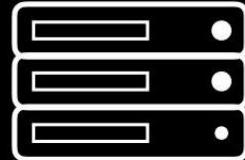
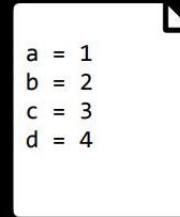
```
git commit -m "message"
```



```
git commit -m  
"Add line"
```



```
git commit -m "message"
```



Add line

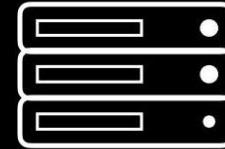
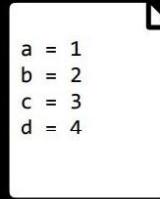
```
git commit -am  
"message"
```

- Nearly identical to previous command, with the added step of applying `git add` to all existing tracked files first
  - Ignores untracked files

# git

- Displays useful ~~information~~ ~~about~~ ~~your~~ ~~repository~~ (e.g., current branch, tracked/untracked files, differences between local and remote versions)

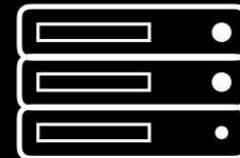
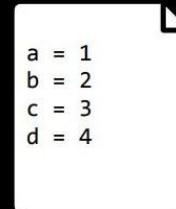
# git status



Add line



# git status



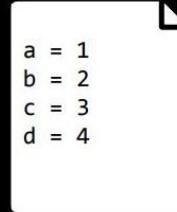
On branch master  
Your branch is ahead of 'origin/master' by 1 commit.  
(use "git push" to publish your local commits)

# git

# push

- Uploads local commits to the remote repository (i.e., from your computer to GitHub)

git push



Add line

# git

## pull

- Downloads remote commits to the local repository (i.e., from GitHub to your computer)

# Merge Conflicts

- When two collaborators make conflicting changes to the same file, a merge conflict may arise
- Git will complain when you attempt to `git pull` and you will need to manually resolve the conflict

# Merge Conflicts



git pull

```
a = 1
<<<< HEAD
b = 2
=====
b = 0
>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```

# git

## log

- Displays history of commits made in the repository from newest to oldest

# Branching

- Each repository by default has a "master" branch where all your work lives
- Sometimes useful to create separate branches in your repository (to test new features, separate work among collaborators, etc.)



first  
commit

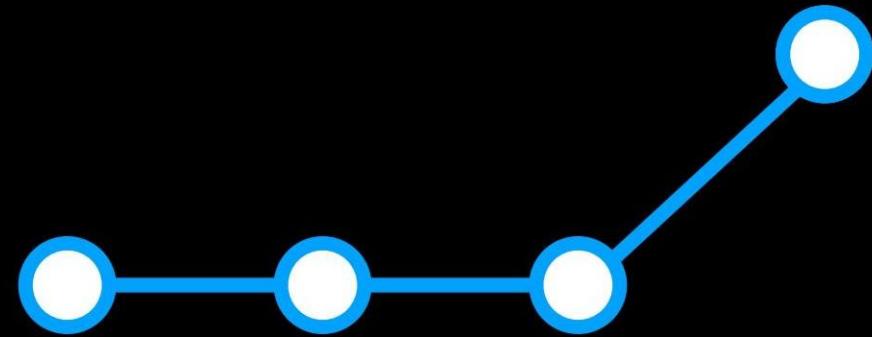


first  
commit

changes

more  
changes

start new  
feature

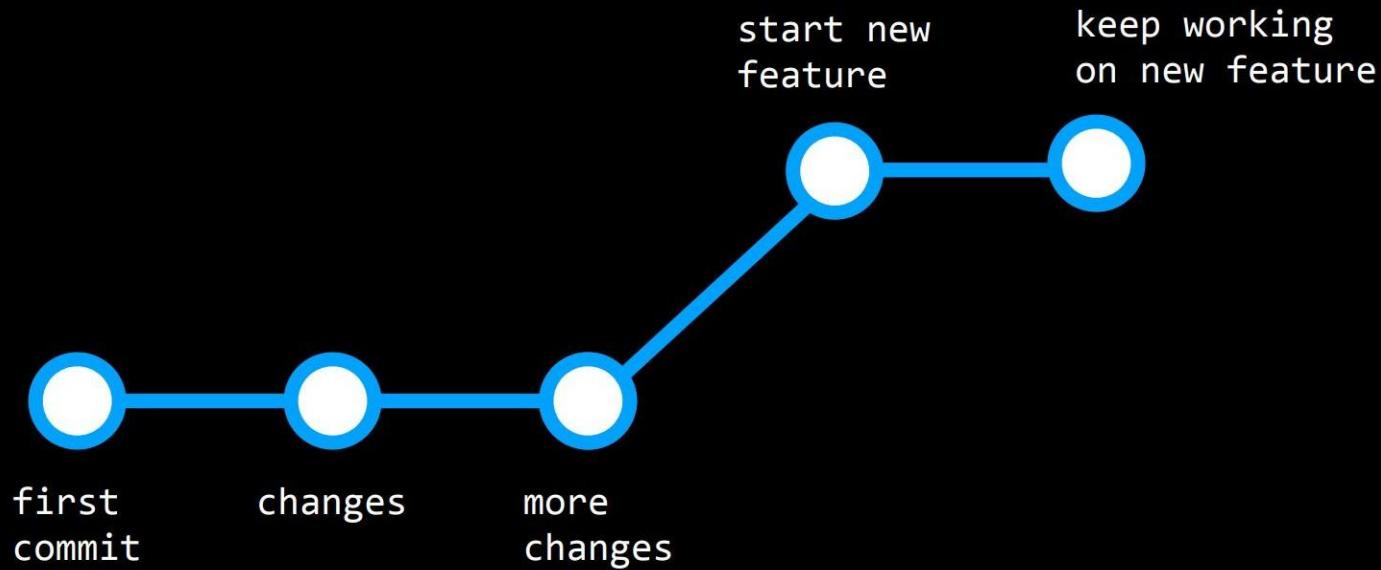


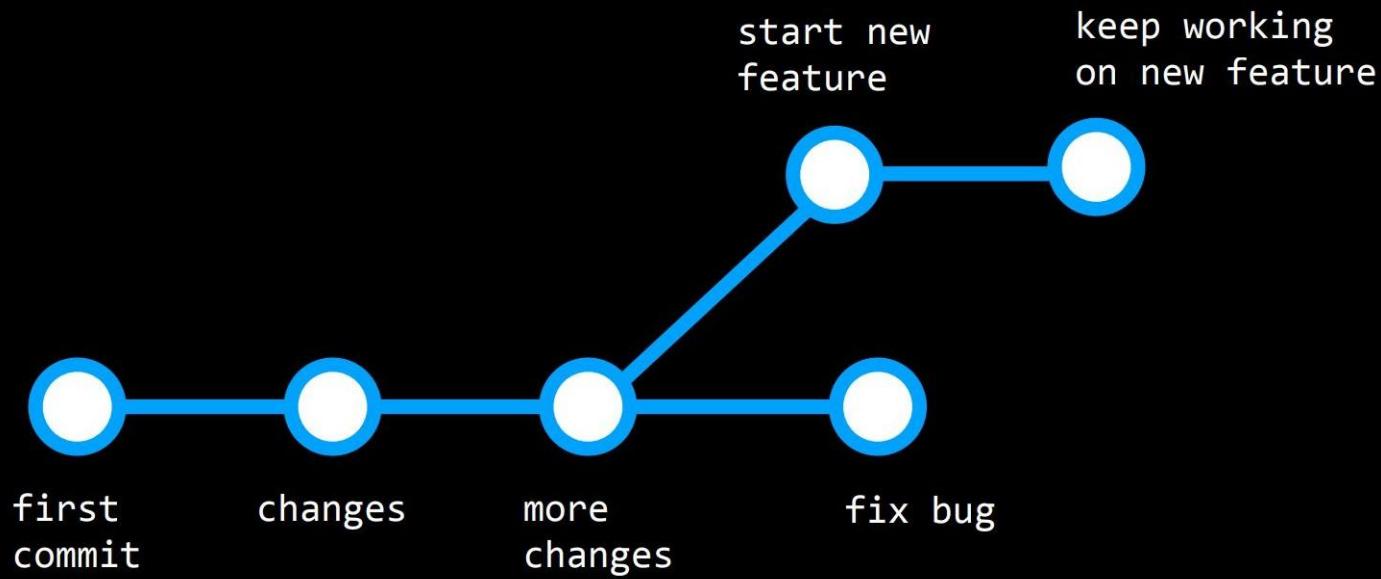
first  
commit

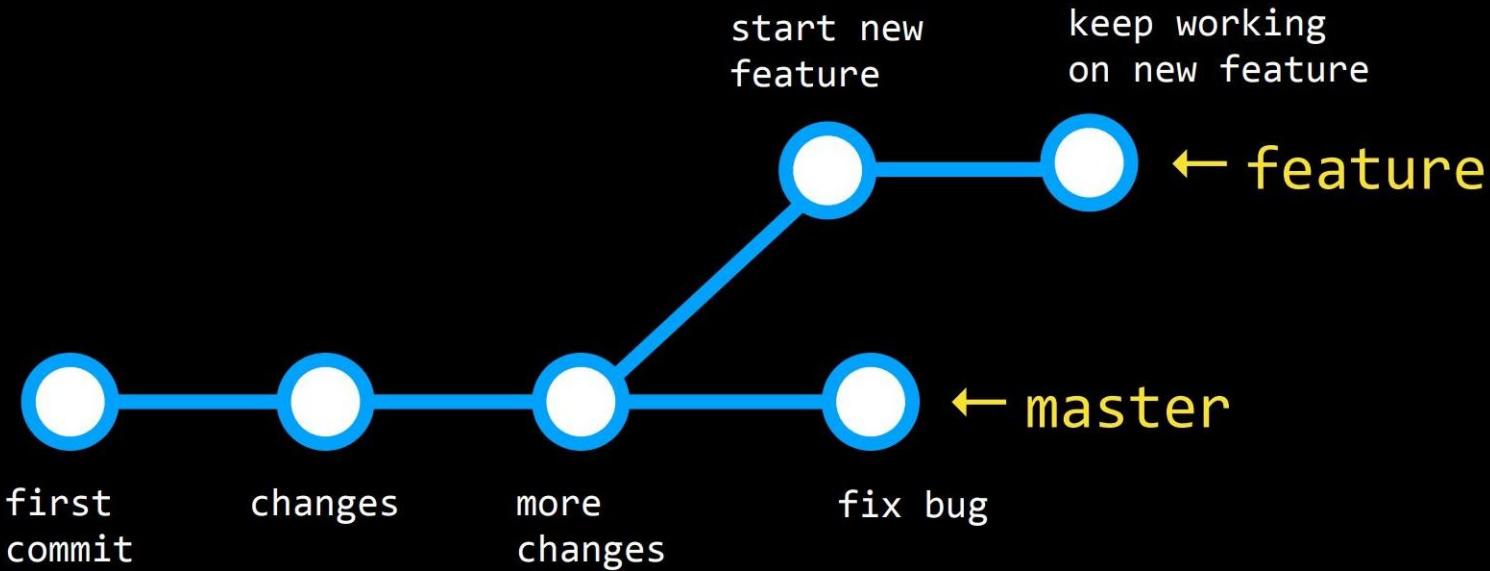
changes

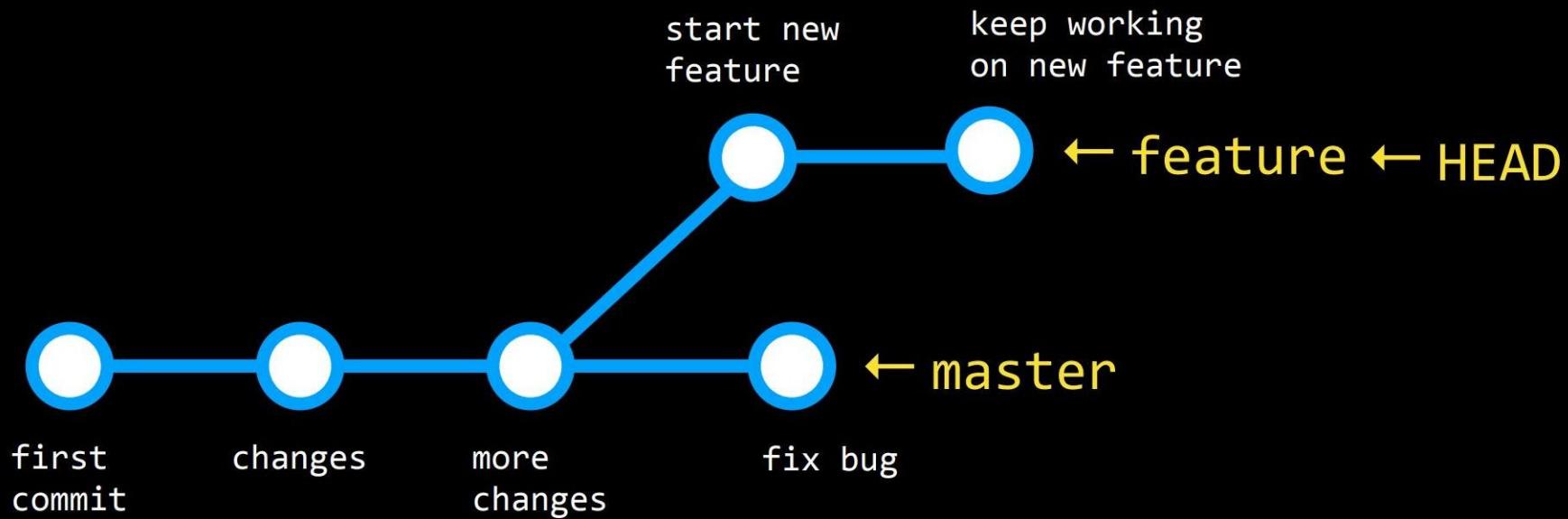
more  
changes

start new  
feature









# git

## branch

- By default, lists all of the branches in your repository, but has a few other variations:
- git branch <branch>
  - Creates a new branch with the given name
- git branch -d <branch>
  - Deletes the specified branch

# git checkout <branch>

- Switches from the current branch to the specified branch  
(must already exist)

# git checkout -b <branch>

- First creates a new branch with the given name, then switches to it

# git merge <branch>

- Merges the specified branch to the current branch

# Forking

- Create a copy of someone else's repository on your profile so that you can contribute to their project

# Open Source

- GitHub is a popular home for "open source" projects (i.e., projects whose source code is freely available online and may be redistributed and modified).