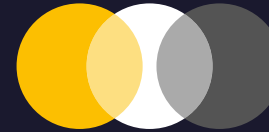




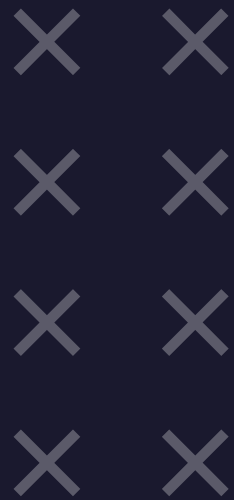
**Analytics Club IITM**  
Data Science and Deep Learning



COMPUTER VISION AND  
INTELLIGENCE GROUP

ANALYTICS + CVI  
PRESENTS

**C $\Phi$  SUMMER SCHOOL**



# Schedule

- **6th July** - Basics of Python and GiT  
Introduction to Numpy and Pandas
- **8th July** - Visualization using Matplotlib  
Intro to Machine learning- Implementation of  
Linear, Logistic and Polynomial Regression

Task 1: Data Visualization Challenge

# Schedule

- **11th July** - Understanding ML Models - SVM, Naive Bayes  
Common performance metrics in ML
- **13th July** - Understanding ML Models - Tree-based Models  
Ensemble Methods & Recommendation Systems
- **15th July** - Beyond Linear models: Why Neural Networks?  
Perceptron Learning Algorithms

Task 2: Data Science Contest on Kaggle

# Schedule

- **18th July** - Back Propagation, Gradient Descent, and other variants
- **20th July** - Strategies for training Neural Networks  
Introduction to Open CV
- **22nd July** - State-of-the-art Computer Vision Models  
Brief on Natural Language Processing and  
Reinforcement Learning

Task 3: Data Learning Hackathon

# What is Python?



Python is a high level programming language created by Guido Van rossum.  
(Now he works for microsoft :))

## Why Python?

It's easier to learn and use in comparison to other languages like C++ or Java, due to its simpler syntax.

It has a huge, rapidly growing supportive community

It has a large number of libraries and frameworks made by third parties. This makes it even easier to use python

# "HELLO WORLD"

```
print("HELLO WORLD")
```

The print() function prints the specified message to the screen.



# CONTENTS

- Data types
- Operators
- Conditional Statements
- Loops
- Functions
- Classes
- Lists
- Tuples
- Dictionary
- Python Library
- Git & GitHub

# Data Types

- Variables are classified using data types
- Each data type gives a variable different properties
- Using data types lets us know what is expected from each variable;





**3.14**

**5**



The most commonly used data types are :

Numeric:

- int (integers)
- float (decimal values)
- strings (characters, words and sentences)

bool (boolean values, i.e. True or False)

Sequential Data Types:

- list
- tuple
- range
- set (Sets)

dict (Dictionaries)



# OPERATORS

Operators perform processes on variables and constants

There are mainly 3 types of operators

- 1) Arithmetic operators (+, -, /, \*, %, \*\*)
- 2) Comparison operators (<, >, ==)
- 3) Logical operators (AND, OR, NOT)

# CONDITIONAL STATEMENTS

**Conditional Statements are used in decision making.**

The 3 keywords used in conditional statements are :

- If
- Elif
- Else



# LOOPS

Looping means repeating something over and over until a particular condition is satisfied.

There are 2 types of loops in Python

- For loop - When no. of iterations are known.
- While loop - When no. of iterations are not known.  
Loop is terminated based on criteria.



# FUNCTIONS

- Functions are modules of code that accomplish a specific task.
- A function executes only when it is called.
- Functions usually take inputs (data), process it, and output (or return) result. Once a function is written, it can be used over and over again.



# OOP

- OOP stand for object oriented programming.
- In OOP we use objects to contain data in the form of attributes and code in the form of procedures(methods).



# WHAT ARE OBJECTS?





# WHAT ARE OBJECTS?





# WHAT ARE CLASSES?

Classes contain blueprints that are used to make objects

For eg.

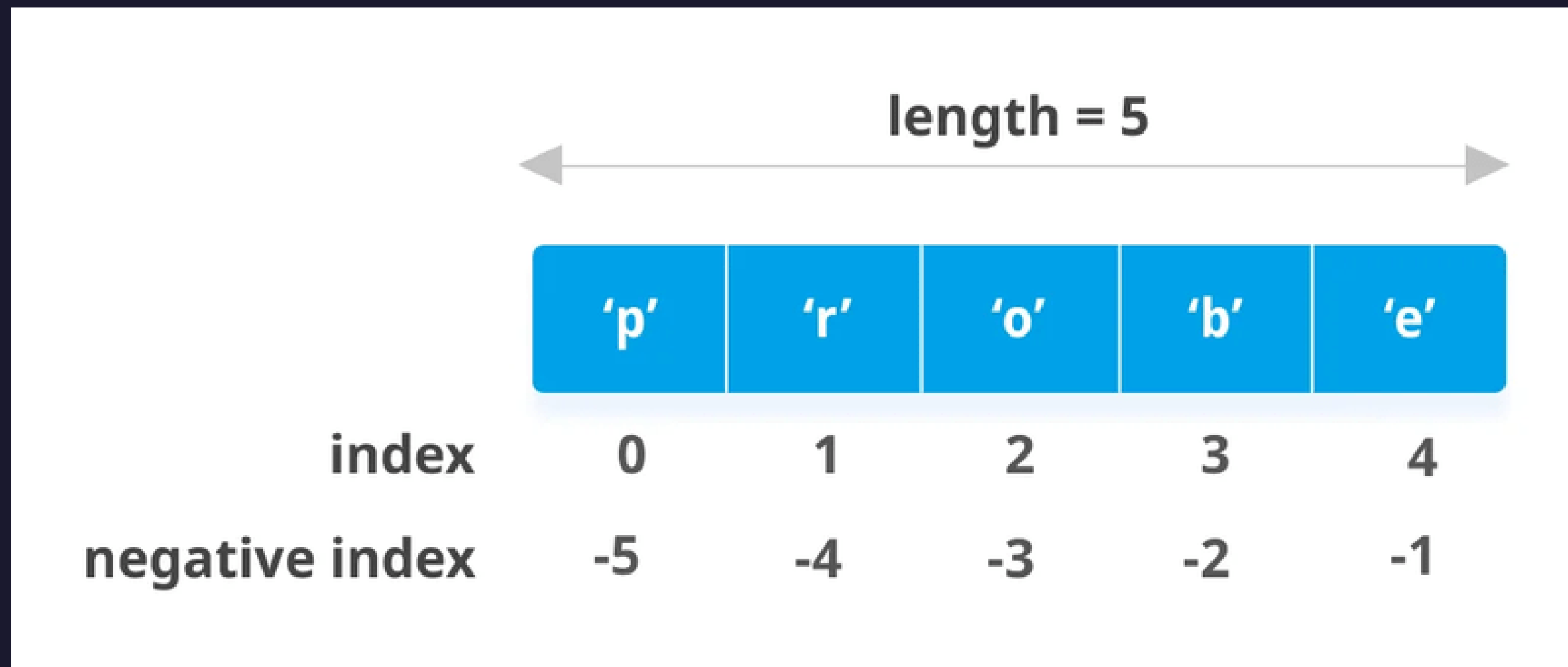
lets make a car (object).

we need some basic info like color, brand etc ie the blue print(classes)

- Classes are made of two things  
fields or attributes.  
eg. Color of the car etc.
- Methods.  
eg. Driving modes in the car.

# Lists

A list is a collection of entries that is ordered and can be changed. They also allow duplicate entries. Entries of a list don't need to be of the same datatype.



# Tuples

- A tuple is a collection of entries which are ordered in the same way as lists but cannot be changed.
- All the regular list operations apply to tuples except for those which attempt to change it's own value

# Dictionaries

A dictionary is a collection of key value pairs which is ordered, changable and does not allow duplicate members.

In dictionaries, keys are immutable and the values are mutable

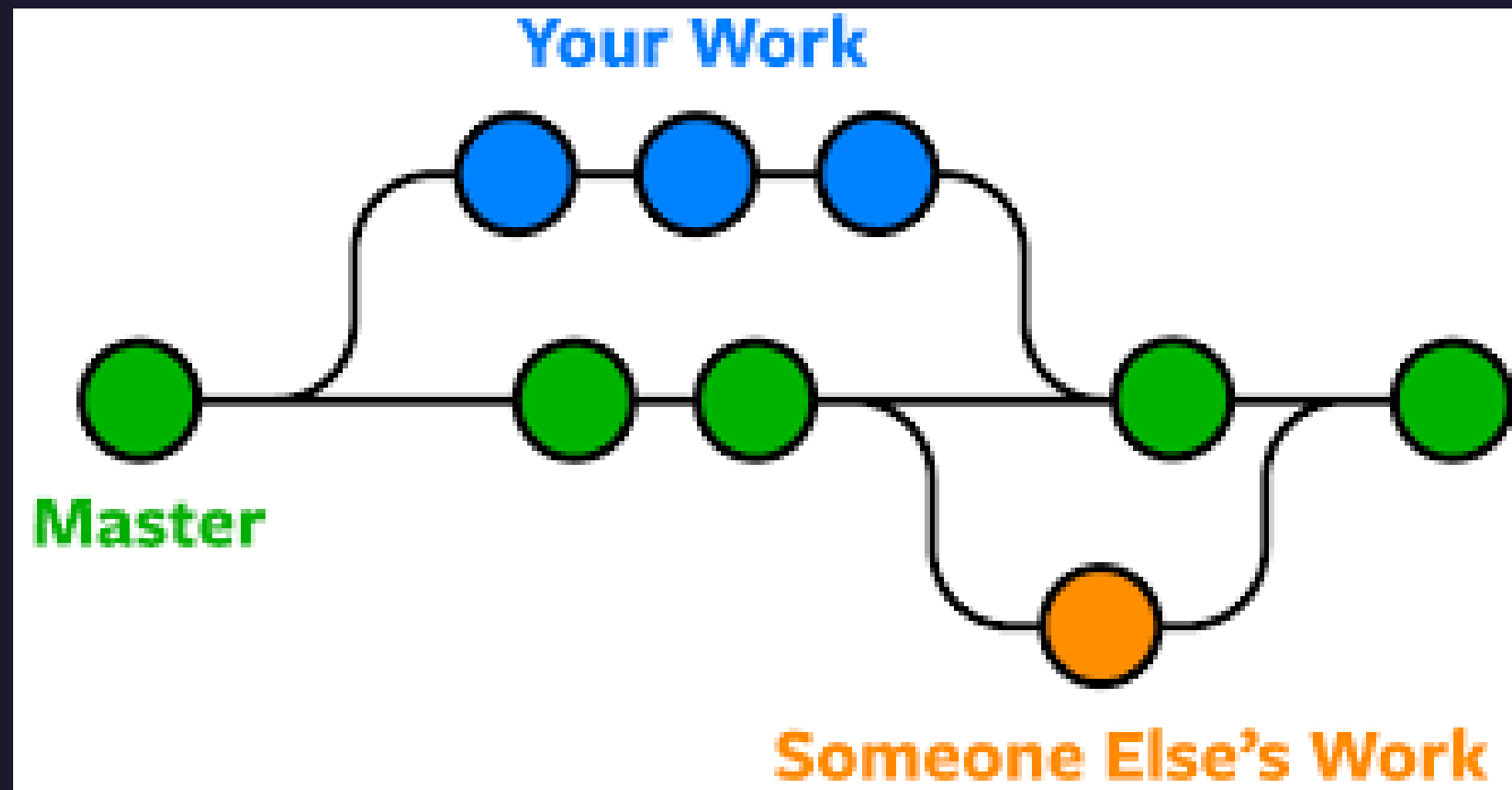
Name	Phone Number
Sam Behn	(916) 176-6345
Cecile Walasik	(227) 795-8943
Dale Hassur	(340) 593-7208
Rutter Maddra	(447) 289-2223
Rosemarie Trett	(647) 592-9991
Auria Swain	(137) 489-5635
Anna-diane Abra	(310) 788-2541
Renard De Rugg	(757) 544-0020
Felecia Guillet	(413) 838-3183
Jolie Juanico	(206) 172-8987
Hedi McMurtyr	(759) 206-9814
Portie Andino	(897) 243-2551

# List Vs Set Vs Dictionary Vs Tuple

Lists	Sets	Dictionaries	Tuples
List = [10, 12, 15]	Set = {1, 23, 34} Print(set) -> {1, 23, 24} Set = {1, 1} print(set) -> {1}	Dict = {"Ram": 26, "mary": 24}	Words = ("spam", "eggs") Or Words = "spam", "eggs"
Access: print(list[0])	Print(set). Set elements can't be indexed.	print(dict["ram"])	Print(words[0])
Can contains duplicate elements	Can't contain duplicate elements. Faster compared to Lists	Can't contain duplicate keys, but can contain duplicate values	Can contains duplicate elements. Faster compared to Lists
List[0] = 100	set.add(7)	Dict["Ram"] = 27	Words[0] = "care" -> TypeError
Mutable	Mutable	Mutable	Immutable - Values can't be changed once assigned
List = []	Set = set()	Dict = {}	Words = ()
Slicing can be done print(list[1:2]) -> [12]	Slicing: Not done.	Slicing: Not done	Slicing can also be done on tuples
<u>Usage:</u> Use lists if you have a collection of data that doesn't need random access. Use lists when you need a simple, iterable collection that is modified frequently.	<u>Usage:</u> - Membership testing and the elimination of duplicate entries. - when you need uniqueness for the elements.	<u>Usage:</u> - When you need a logical association b/w key:value pair. - when you need fast lookup for your data, based on a custom key. - when your data is being constantly modified.	<u>Usage:</u> Use tuples when your data cannot change. A tuple is used in combination with a dictionary, for example, a tuple might represent a key, because its immutable.

# GIT

Git is a type of version control system (VCS). A VCS enables you to record changes to files over time



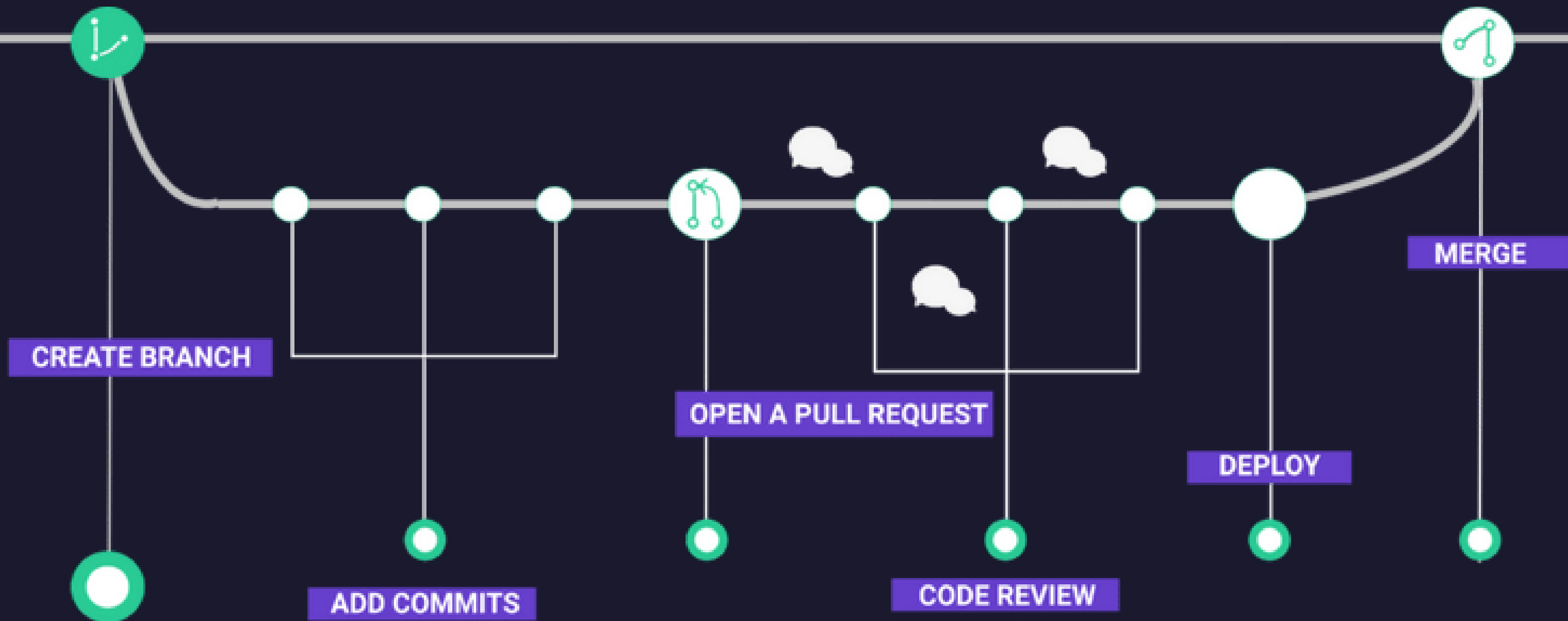


# GitHub

GitHub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.

# Instructions on using github

- create a new repository
- clone the github repo in your local system
- add files and commit the changes

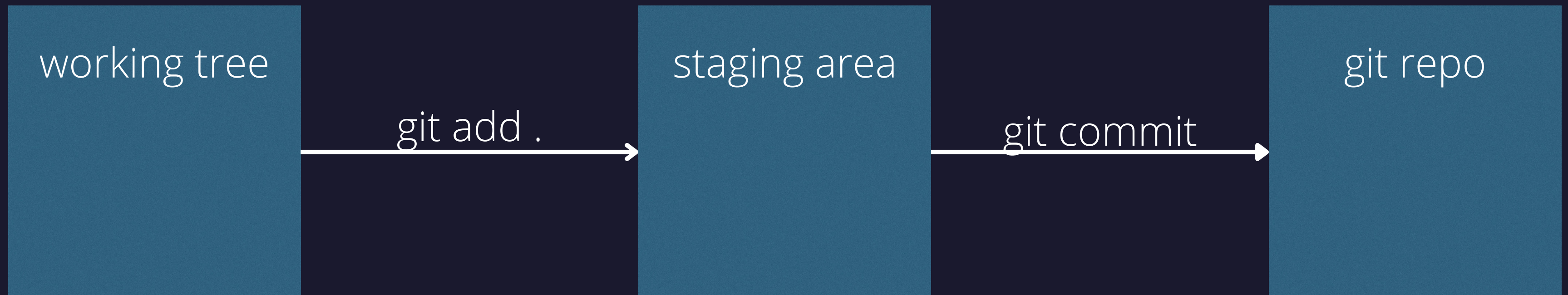


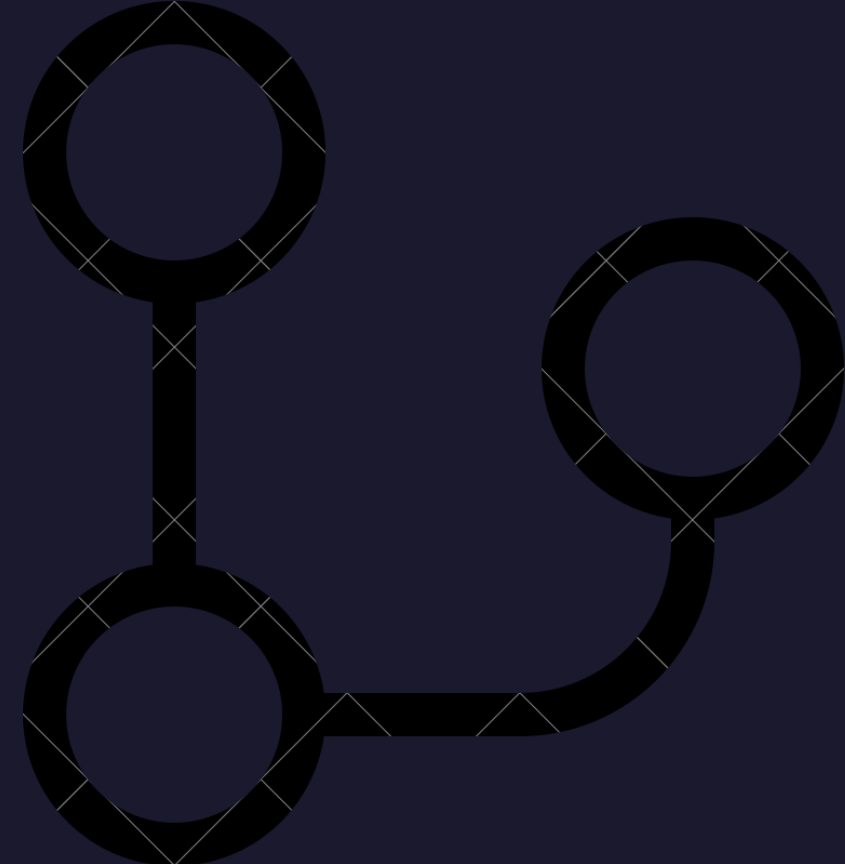
# Setting up git repository

- It can be done in github
- then cloned to local system for changes to be made

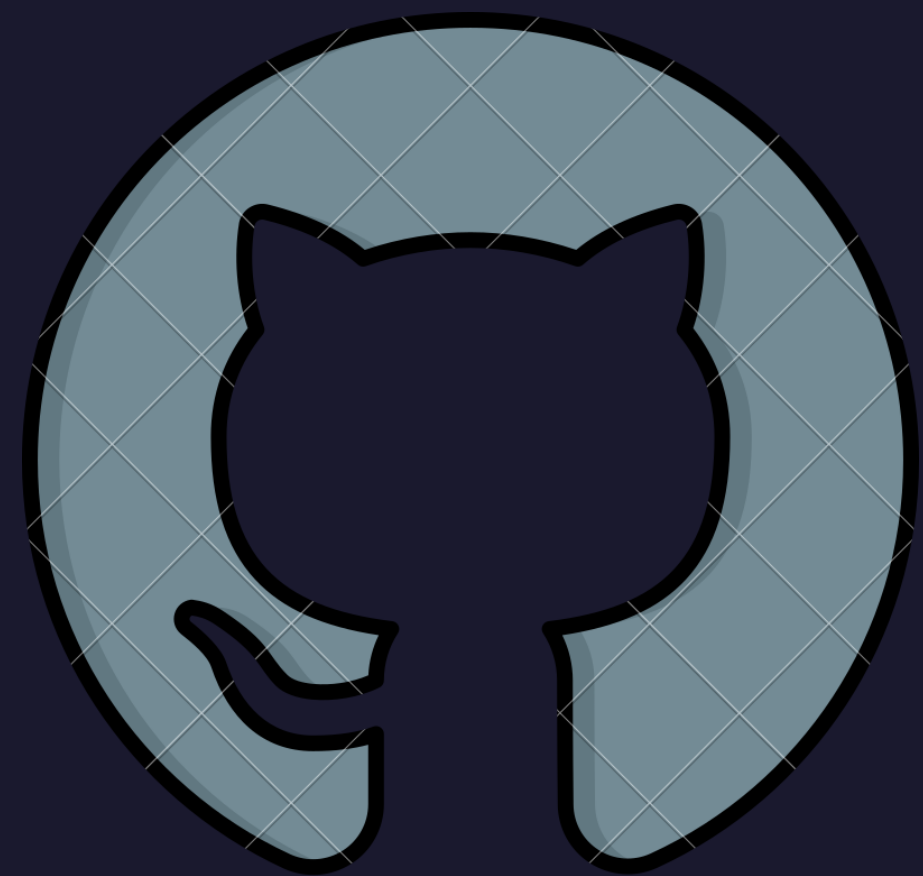
# Adding file using terminal

- `git add .`
- `git commit -m"your message"`
- `git push origin main`





Why branching and forking?



# Python Libraries

- NumPy
  - Pandas
  - OpenCV
  - Matplotlib
  - Scipy
  - Tensorflow
  - PyTorch
- and many more...

**When you finish writing your code and find out there's already a library which does it.**



**Well, today was a fantastic  
waste of time**