



CIT-DATA-101

Welcome to the CIT's Data Cell !

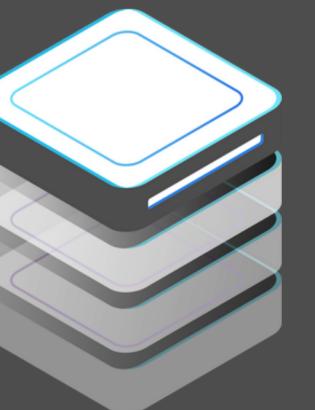


How the Data Cell works ?

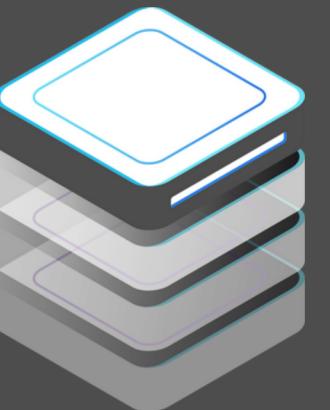
We learn new
skills together.

We collaborate on
projects together.

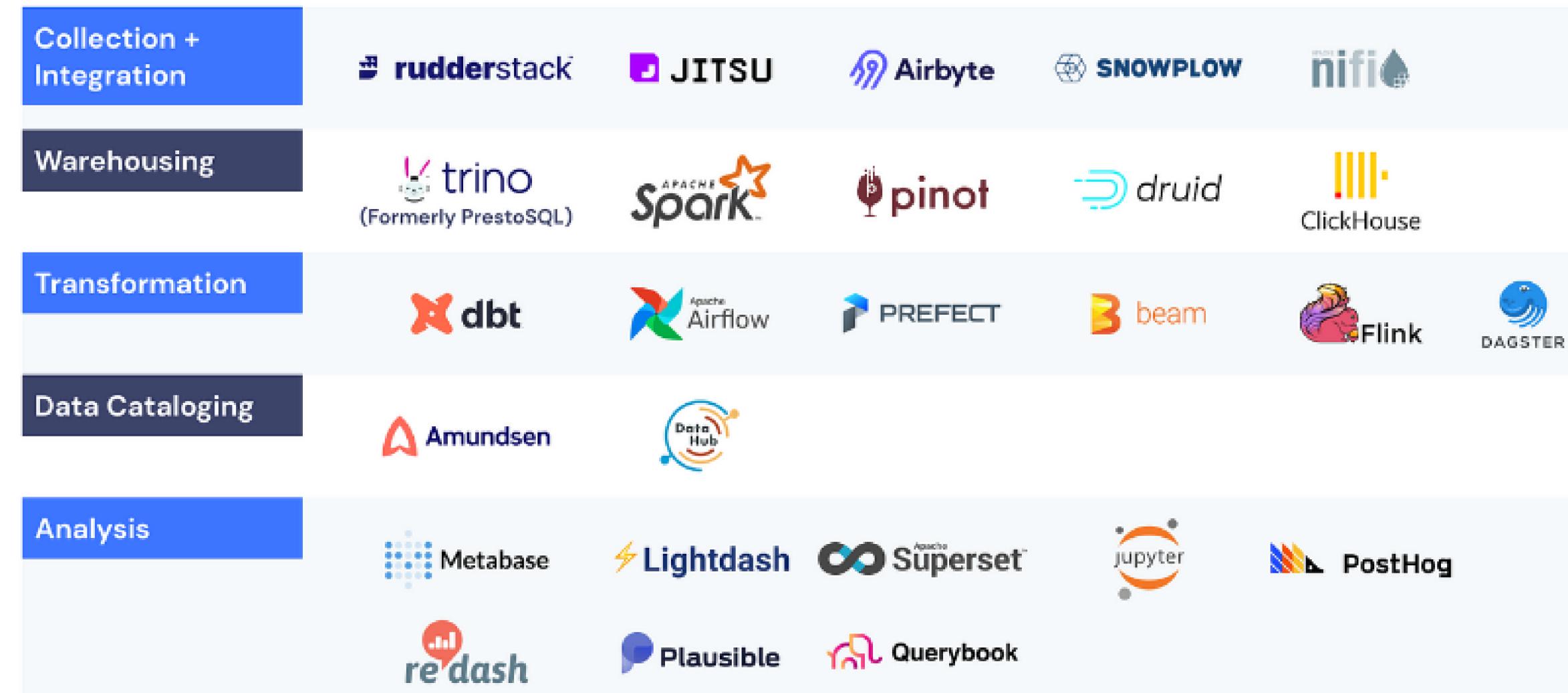
Our community
grows stronger.



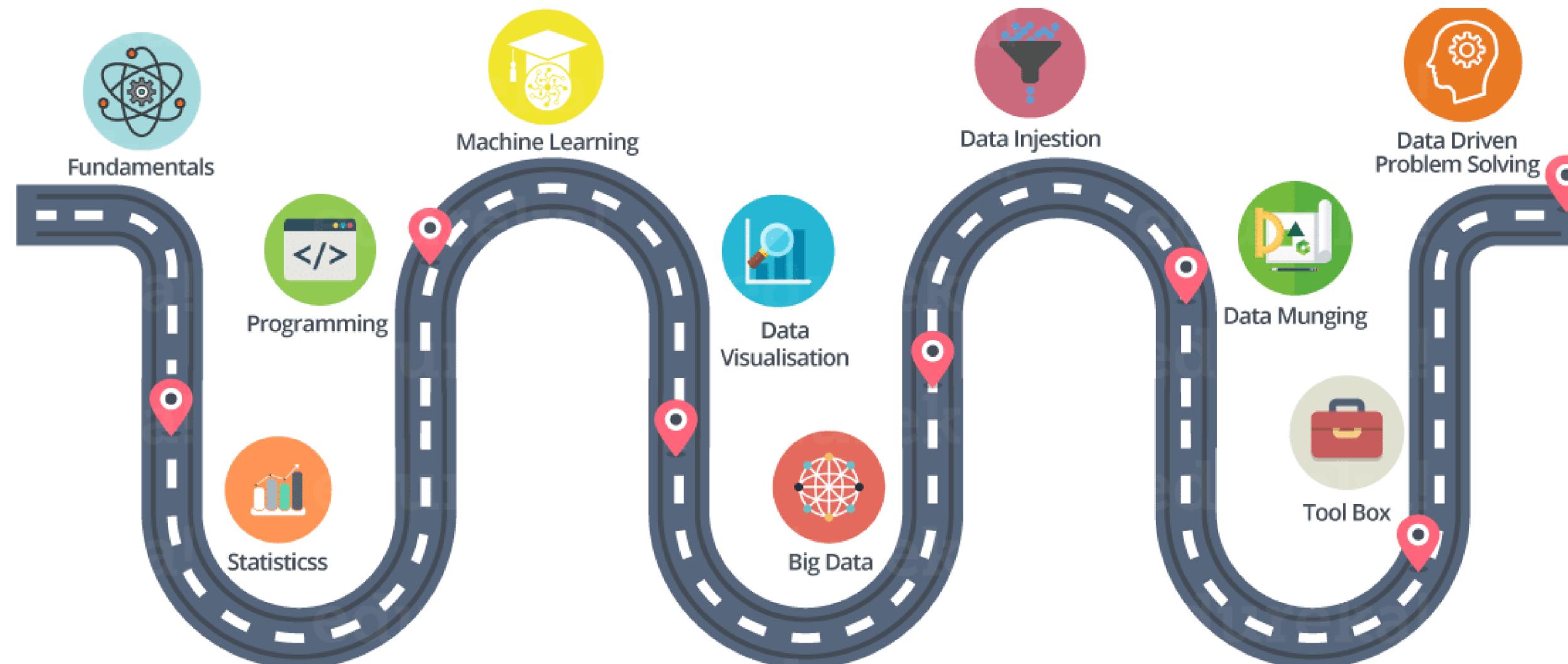
Exploring Data Jobs Together



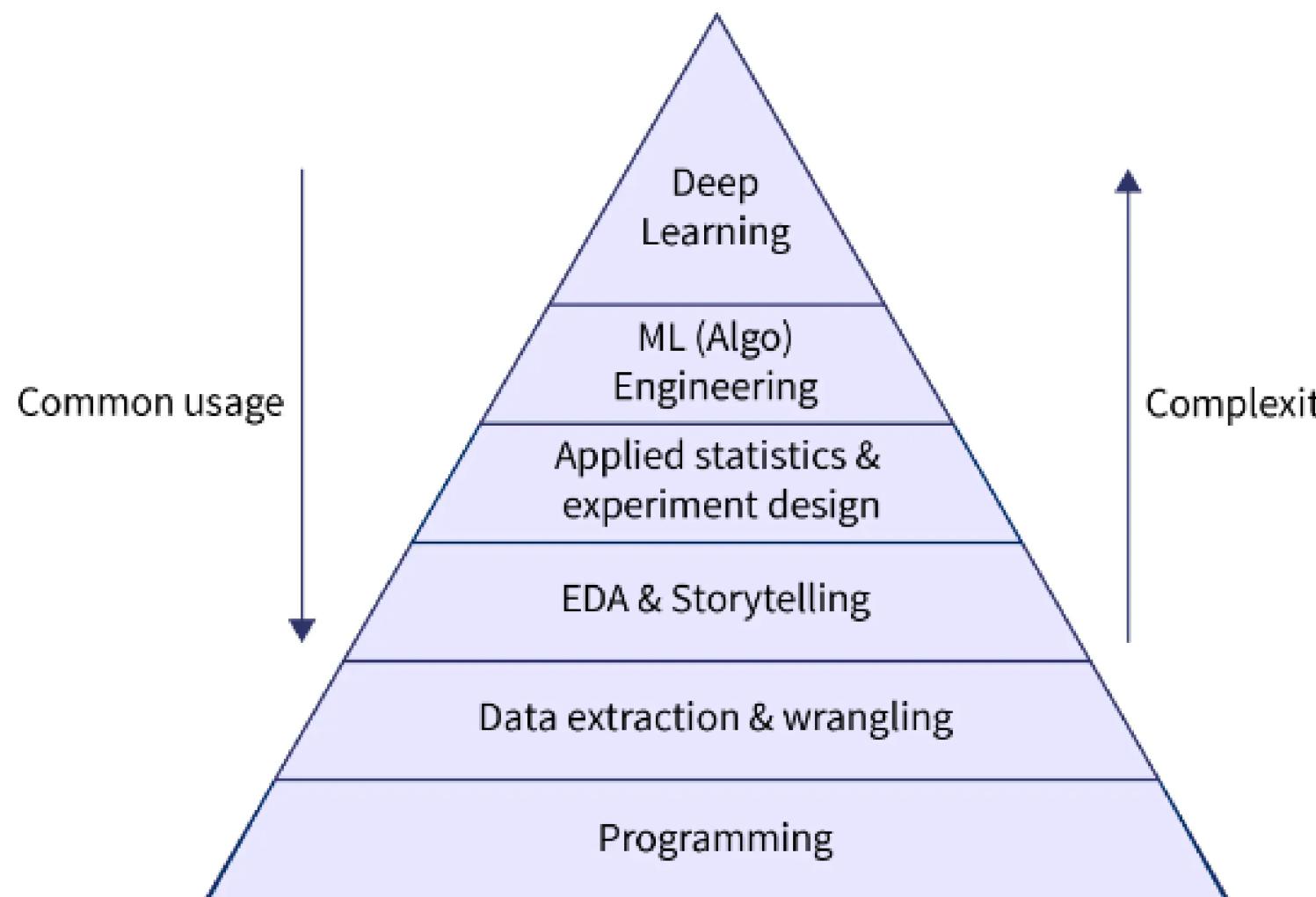
Data Engineer



Data Scientist & Analyst



Machine Learning Engineer

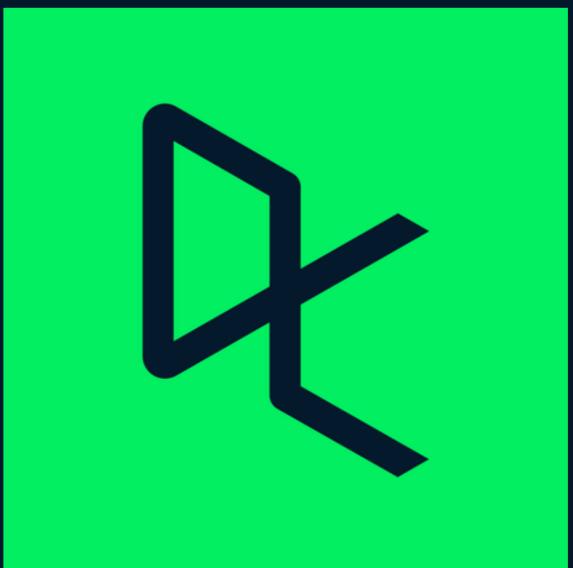


First term

I. Data Exploration

- Data Manipulation
- Data Visualization
- Tips for data storytelling

Presenting our partner



Introduction to OOP in Python

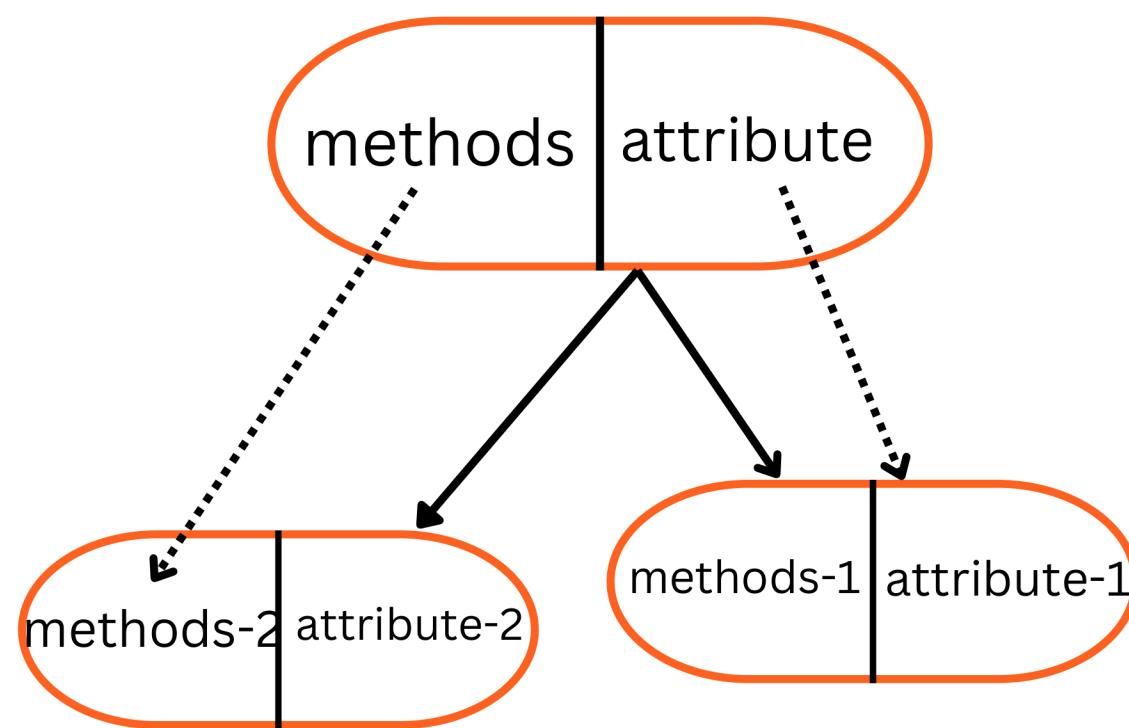
Key concepts

- OOP: A paradigm based on objects.
- Objects: Central to OOP, representing data and behavior.

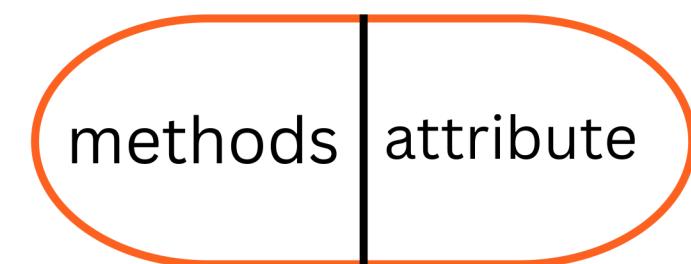


Principles of OOP

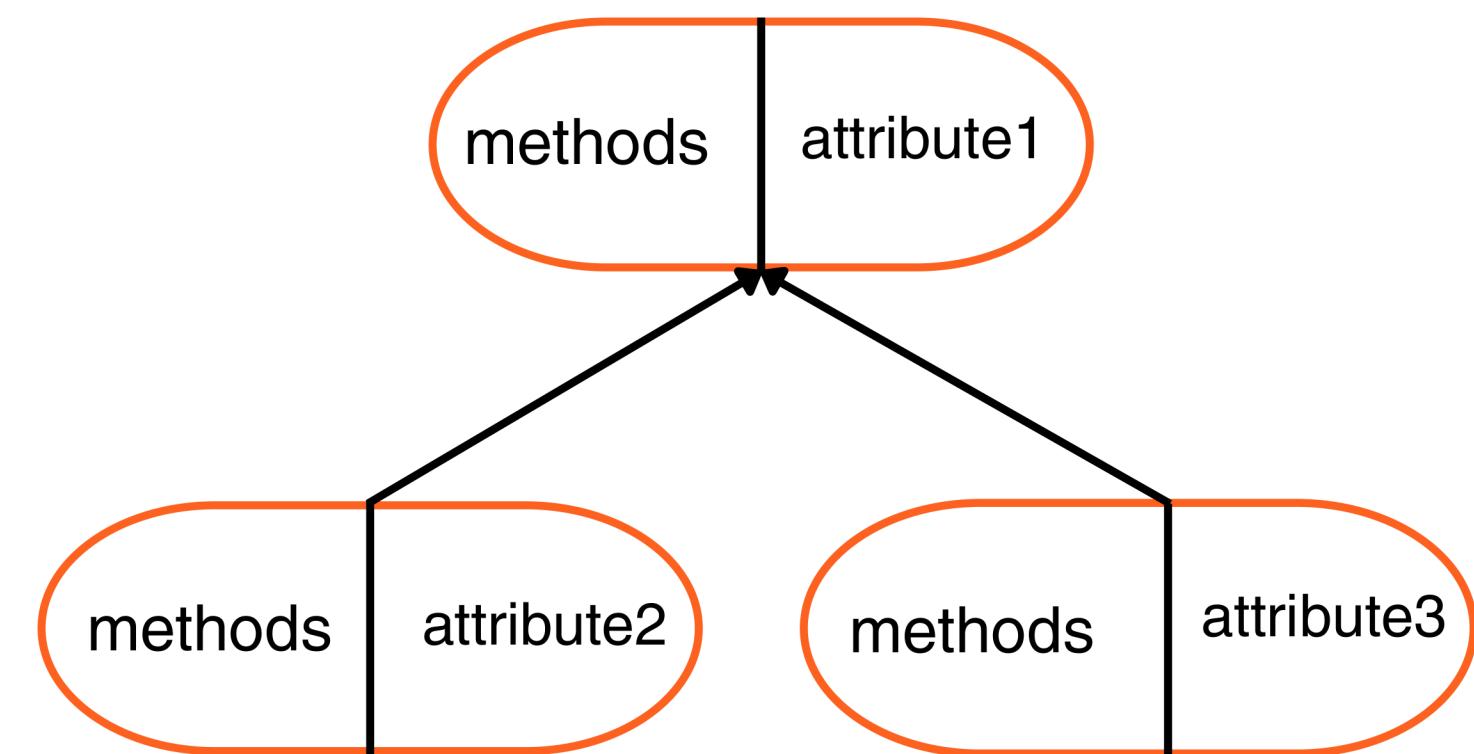
Inheritance



Encapsulation



Polymorphism



Example to sum up

Let's start by defining a class :

class name



constructor



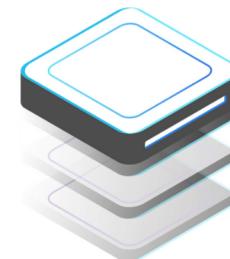
attribute



method



```
class Dog:  
    def __init__(self, name, breed):  
        self.name = name  
        self.breed = breed  
  
    def bark(self):  
        print(f"{self.name} barks loudly!")
```



Inheritance

Child class

```
class GoldenRetriever(Dog):  
    def fetch(self):  
        print(f"{self.name} loves to fetch balls!")
```

Child class

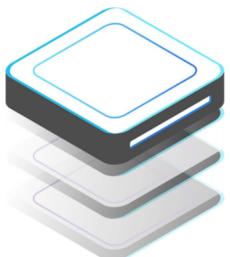
```
class GoldenRetriever(Dog):  
    def fetch(self):  
        print(f"{self.name} loves to fetch  
balls!")  
    def __init__(self, name):  
        super().__init__(name)
```



Polymorphism

```
class Dog:  
    def __init__(self, name, breed):  
        self.name = name  
        self.breed = breed  
  
    def bark(self):  
        pass
```

```
class GoldenRetriever(Dog):  
    def bark(self):  
        return "Woo0000f!"  
  
class GermanShepherd(Dog):  
    def bark(self):  
        return "Woof!"
```



Abstraction

abstraction class



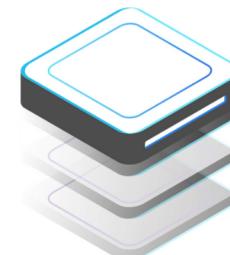
```
from abc import ABC, abstractmethod
```

```
class Animal(ABC):  
    def __init__(self, name):  
        self.name = name
```

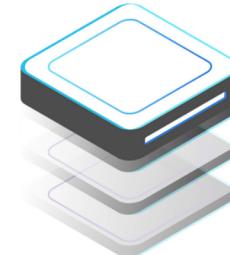
abstract method



```
@abstractmethod  
def speak(self):  
    pass
```



**Check Google Colab
Now let's have fun ...**



Thank you
See you soon ...

