CIT-DATA-101

Welcome to the CIT’s Data Cell !

How the Data Cell works ?

We collaborate on projects together.

We learn new skills together.

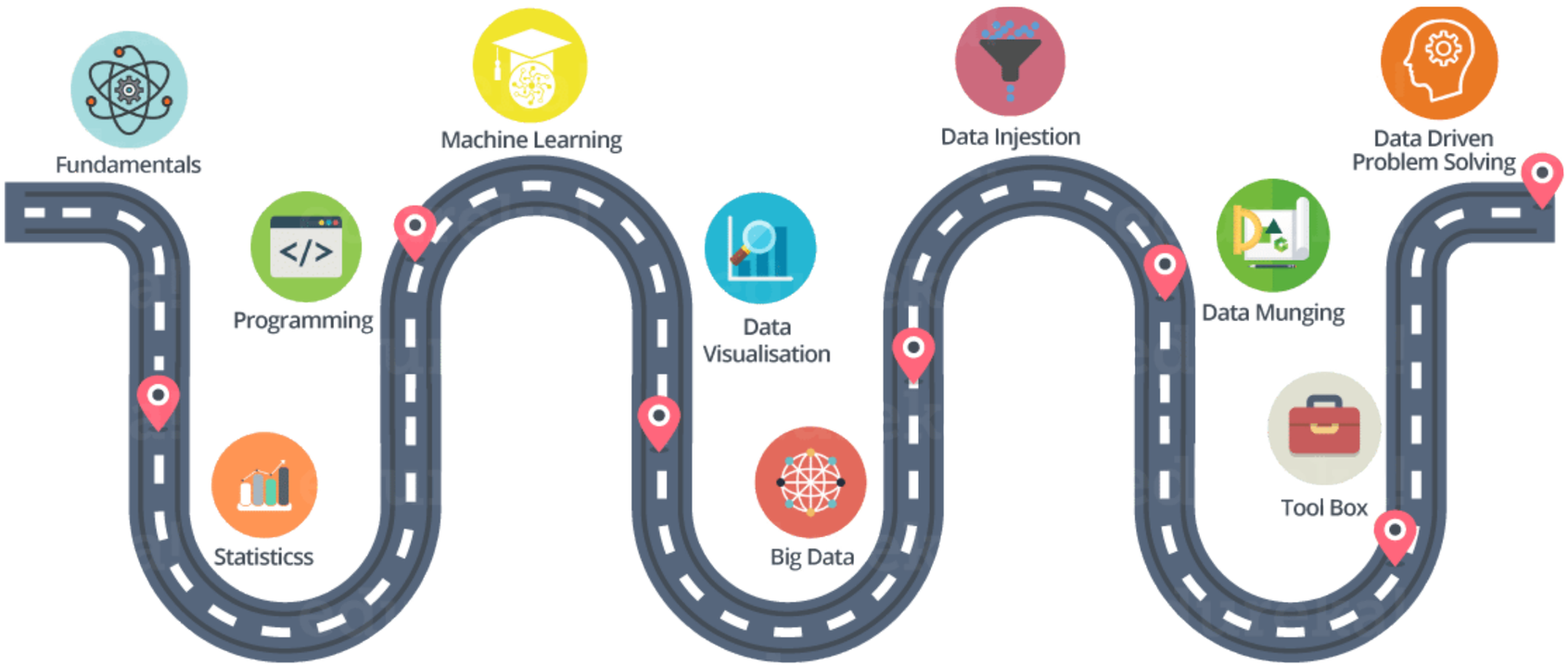
Our community grows stronger.

Exploring Data Jobs Together

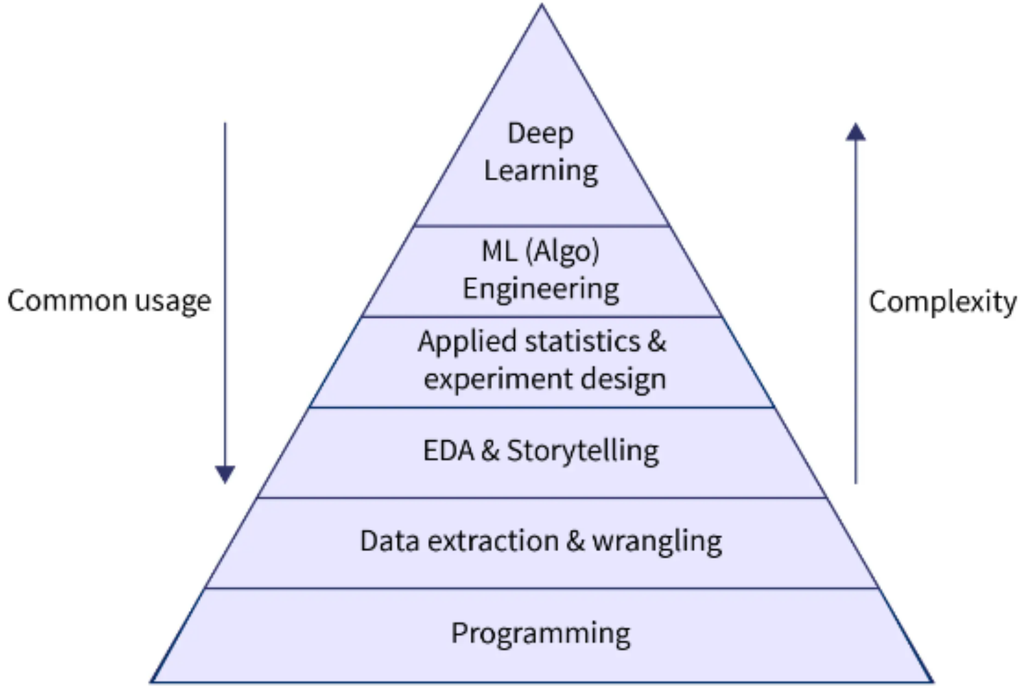
Data Engineer

Collection + Integration	rudderstack, JITSU, Airbyte, SNOWPLOW, nifi
Warehousing	trino (Formerly PrestoSQL), Apache Spark, pinot, druid, ClickHouse
Transformation	dbt, Apache Airflow, PREFECT, beam, Flink, DAGSTER
Data Cataloging	Amundsen, Data Hub
Analysis	Metabase, Lightdash, Apache Superset, jupyter, PostHog, re dash, Plausible, Querybook

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

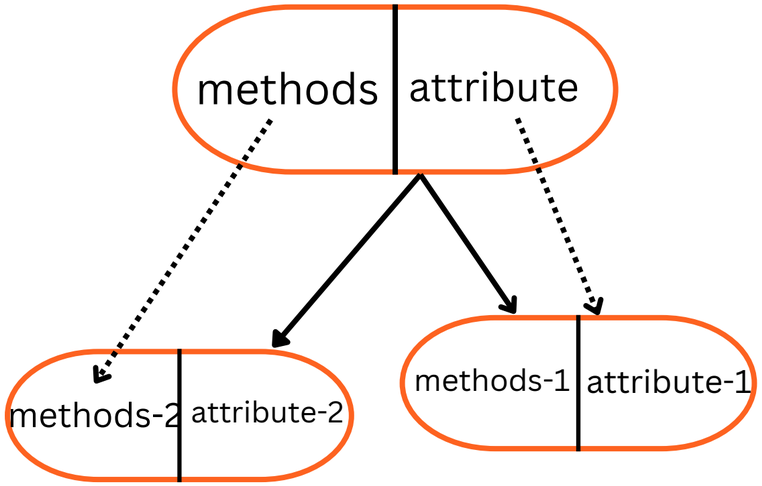
Key concepts

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

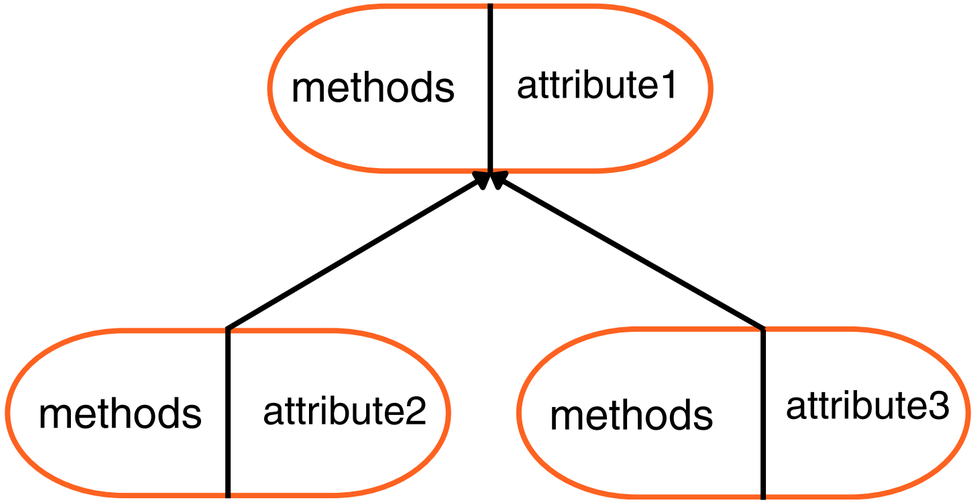
Principles of OOP

Encapsulation

Inheritance



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

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

    def bark(self):
        pass
```

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


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

Abstraction

abstraction class ----------→

abstract method ----------→

```
from abc import ABC, abstractmethod

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

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

Check Google Colab
Now let’s have fun …

Thank you

See you soon ...