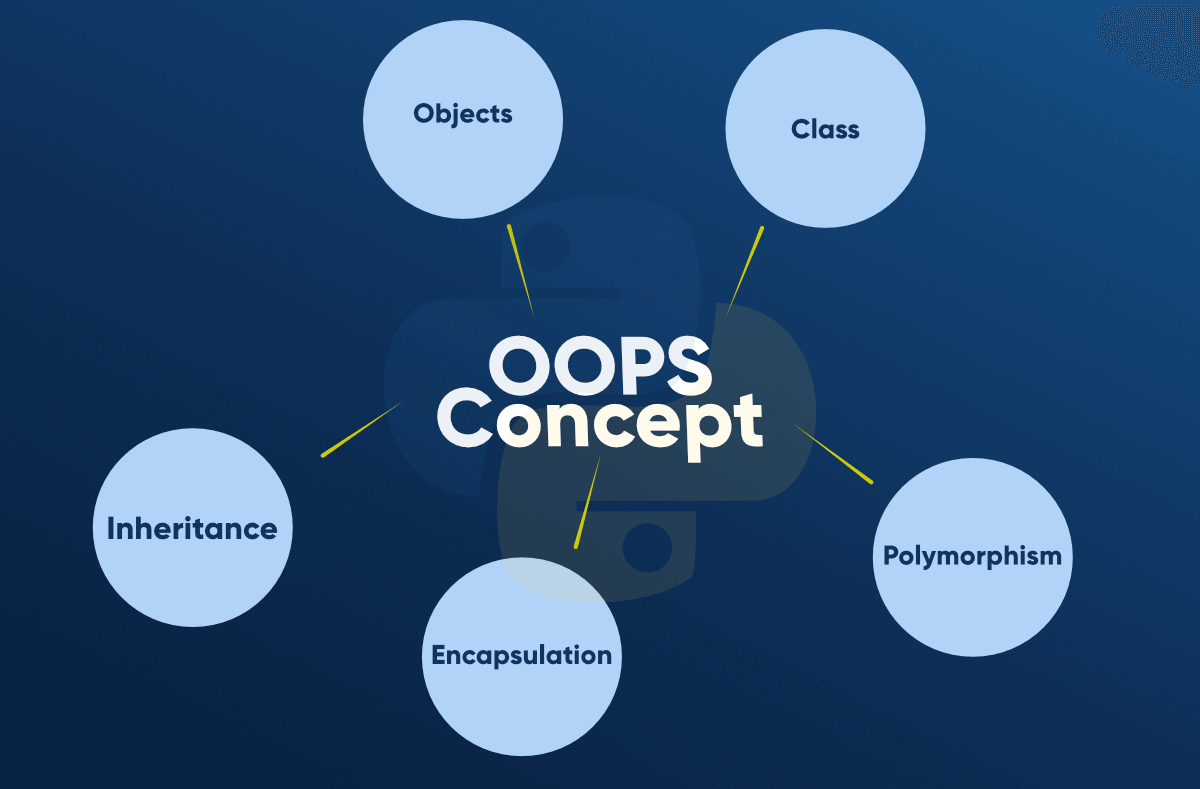
**Python OOPs Concepts:**

In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming. The main concept of OOPs is to bind the data and the functions that work on that together as a single unit so that no other part of the code can access this data.

**Main Concepts of Object-Oriented Programming (OOPs)**

* Class
* Objects
* Polymorphism
* Encapsulation
* Inheritance



**Class :**

A class is a collection of objects. A class contains the blueprints or the prototype from which the objects are being created. It is a logical entity that contains some attributes and methods.

**Some points on Python class:**

Classes are created by keyword class.

Attributes are the variables that belong to a class.

Attributes are always public and can be accessed using the dot (.) operator. **Eg.:** Myclass.Myattribute

**Objects :**

The object is an entity that has a state and behavior associated with it. It may be any real-world object like a mouse, keyboard, chair, table, pen, etc. Integers, strings, floating-point numbers, even arrays, and dictionaries, are all objects.

**An object consists of :**

**State:** It is represented by the attributes of an object. It also reflects the properties of an object.

**Behavior :** It is represented by the methods of an object. It also reflects the response of an object to other objects.

**Identity:** It gives a unique name to an object and enables one object to interact with other objects.

**Methods in Python**

## **Types of Methods in Python**

There are basically three types of methods in Python:

* Instance Method
* Class Method
* Static Method

# Abstraction in Python

Abstraction is used to hide the internal functionality of the function from the users. The users only interact with the basic implementation of the function, but inner working is hidden. User is familiar with that **"what function does"** but they don't know **"how it does."**

In simple words, we all use the smartphone and very much familiar with its functions such as camera, voice-recorder, call-dialing, etc., but we don't know how these operations are happening in the background.

**Inheritance :**

Inheritance is the capability of one class to derive or inherit the properties from another class. The benefits of inheritance are:

* It represents real-world relationships well.
* It provides reusability of a code. We don’t have to write the same code again and again. Also, it allows us to add more features to a class without modifying it.
* It is transitive in nature, which means that if class B inherits from another class A, then all the subclasses of B would automatically inherit from class A.

**Polymorphism:**

The word polymorphism means having many forms. In programming, polymorphism means the same function name (but different signatures) being used for different types.

**The self**

Class methods must have an extra first parameter in the method definition. We do not give a value for this parameter when we call the method, Python provides it. If we have a method that takes no arguments, then we still have to have one argument.