

OOPS In python

Friday, August 19, 2022 11:36 AM

A

elliPaat

Agenda

- | | |
|--------------------------------------|----------------------------------|
| 01 Introduction to OOPs | 02 Real-world OOP example |
| 03 OOPs – Classes and Objects | 04 Inheritance in Python |
| 05 Encapsulation in Python | 06 Polymorphism in Python |
| 07 Python Modules | 08 Standard library |
| 09 Installing Packages | 10 Exception Handling |

Basic Principle of OOPS



What is object && class

What are Objects and Classes?

- 01** Object is the basic unit of object-oriented programming
- 02** An object represents a particular instance of a class
- 03** There can be more than one instance of an object
- 04** Each instance of an object can hold its own relevant data
- 05** Objects with similar properties and methods are grouped together to form a Class

In it method

__init__() method in Python

Example

```
class Student(object):  
    def __init__(self, name, branch, year):  
        self.name = name  
        self.branch = branch  
        self.year = year  
        print("A student object is created.")  
  
    def print_details(self):  
        print("Name:", self.name)  
        print("Branch:", self.branch)  
        print("Year:", self.year)  
  
ob1= Student( "Paul", "CSE", 2019)  
ob1.print_details()
```

```
A student object is created.  
Name: Paul  
Branch: CSE  
Year: 2019
```

- __init__ is a special method in Python classes is a constructor method for a class
- __init__ is called when ever an object of the class is constructed

```
>_
class Dog:
    def __init__(self, name):
        self.name = name

    def talk(self):
        print("Woof!")

    def printName(self):
        print("My name is: {}".format(self.name))

dog = Dog("Charlie")

dog.talk()
dog.printName()
```