## **👉 \_\_init\_\_() – the constructor**

This is perfect as **Part 2** in your OOP series. Let’s break it down clearly 👇

# **🧱 Python OOP – Part 2: Understanding \_\_init\_\_() Constructor**

## **🧠 What is \_\_init\_\_() in Python?**

\_\_init\_\_() is a **special method** in Python classes that runs **automatically** when you create an object.  
 It is used to **initialize the object’s data (attributes)**.

✅ It’s also called the **constructor** in Python.

## **🔧 Why Do We Need \_\_init\_\_()?**

Without it:

* You’d have to manually set data for every object after creating it.
* Risk of forgetting to set attributes → errors

With \_\_init\_\_():

* Object is created **with values** right away
* Clean, safe, and structured code

## **💡 Real-Life Analogy:**

Think of \_\_init\_\_() like an **ID card form** you fill when you join a school —  
 You enter name, age, roll number just once — and it’s stored in your profile automatically.

## **🧪 Simple Python Example with \_\_init\_\_():**

class Student:

def \_\_init\_\_(self, name, age): # constructor

self.name = name # store data in object

self.age = age

def display(self):

print(f"Name: {self.name}, Age: {self.age}")

s1 = Student("Gowtham", 33)

s1.display()

### **✅ Output:**

Name: Gowtham, Age: 33

## **🔍 Breakdown Line-by-Line:**

| **Line** | **Explanation** |
| --- | --- |
| def \_\_init\_\_(...) | Constructor method – runs when object is created |
| self.name = name | Saves the name inside the object |
| s1 = Student(...) | Creates an object and auto-calls \_\_init\_\_() |
| s1.display() | Accesses object data using self |

## **💬 Tamil Explanation Snippet for Video:**

"\_\_init\_\_() na constructor. Object create pannumbodhu, Python automatic-a idha call pannum. Namma inga name, age madhiri values kudukkumbodhu, andha object-la store aagidum. Apparam ellaa method-layum self use pannitu andha data access pannalaam."

## **🎯 Why self is still needed in \_\_init\_\_()?**

self refers to the object being created.  
 It lets you store the values inside that specific object — so every object can hold different data.

## **🧪 Bonus: Create 2 Students**

s1 = Student("Gowtham", 33)

s2 = Student("Priya", 28)

s1.display() # Gowtham, 33

s2.display() # Priya, 28

Each object keeps its **own data** — thanks to \_\_init\_\_() and self.

### **✅ With \_\_init\_\_() and self (Clean OOP style):**

class Employee:

def \_\_init\_\_(self, name, aadhaar):

self.name = name

self.aadhaar = aadhaar

def enter\_office(self):

print(f"{self.name} enters using Aadhaar {self.aadhaar}.")

def open\_bank\_account(self):

print(f"Bank account opened for {self.name} with Aadhaar {self.aadhaar}.")

emp1 = Employee("Gowtham", "1234")

emp1.enter\_office()

emp1.open\_bank\_account() # ✅ No need to pass again

## **❌ When NOT to use \_\_init\_\_() in Python?**

You don't need \_\_init\_\_() when your class:

* Doesn’t store any data
* Only contains utility methods (like calculations or tools)
* Is used like a helper or toolbox

### **✅ Example 1: Class without \_\_init\_\_() (utility class)**

class MathTools:

def square(self, n):

return n \* n

def cube(self, n):

return n \* n \* n

tool = MathTools()

print(tool.square(4)) # Output: 16

* No object-specific data is stored
* So \_\_init\_\_() is **not needed**

### **About the Author**

**Gowtham SB** is a **Data Engineering expert, educator,** **and content creator** with a passion for **big data technologies, as well as cloud and Gen AI** . With years of experience in the field, he has worked extensively with **cloud platforms, distributed systems, and data pipelines**, helping professionals and aspiring engineers master the art of data engineering.

Beyond his technical expertise, Gowtham is a **renowned mentor and speaker**, sharing his insights through engaging content on **YouTube and LinkedIn**. He has built one of the **largest Tamil Data Engineering communities**, guiding thousands of learners to excel in their careers.

Through his deep industry knowledge and hands-on approach, Gowtham continues to **bridge the gap between learning and real-world implementation**, empowering individuals to build **scalable, high-performance data solutions**.

𝐒𝐨𝐜𝐢𝐚𝐥𝐬

🎥𝐘𝐨𝐮𝐓𝐮𝐛𝐞 - https://www.youtube.com/@dataengineeringvideos

📸𝐈𝐧𝐬𝐭𝐚𝐠𝐫𝐚𝐦 - <https://instagram.com/dataengineeringtamil>

📸𝐈𝐧𝐬𝐭𝐚𝐠𝐫𝐚𝐦 - [https://instagram.com/](https://instagram.com/dataengineeringtamil)thedatatech.in

🤝𝐂𝐨𝐧𝐧𝐞𝐜𝐭 𝐟𝐨𝐫 𝟏:𝟏 - https://topmate.io/dataengineering/

💼𝐋𝐢𝐧𝐤𝐞𝐝𝐈𝐧 - https://www.linkedin.com/in/sbgowtham/

🌐𝐖𝐞𝐛𝐬𝐢𝐭𝐞 - https://codewithgowtham.blogspot.com

💻𝐆𝐢𝐭𝐇𝐮𝐛 - http://github.com/Gowthamdataengineer

💬𝐖𝐡𝐚𝐭𝐬 𝐀𝐩𝐩 - https://lnkd.in/g5JrHw8q

📧𝐄𝐦𝐚𝐢𝐥 - atozknowledge.com@gmail.com

📱𝐀𝐥𝐥 𝐌𝐲 𝐒𝐨𝐜𝐢𝐚𝐥𝐬 - <https://lnkd.in/gf8k3aCH>