# **🧠 Python Guide: Partially Applied Functions**

Using functools.partial from the functools module

## **✅ What is a Partially Applied Function?**

A **partially applied function** is a new function created by **pre-filling some arguments** of an existing function — so you only pass the remaining arguments later.

🧠 It’s like making a shortcut version of a function with default values baked in.

✅ Done using functools.partial

## **🔧 Import First**

from functools import partial

## **📦 Example 1: GST Billing System**

### **💡 Problem:**

You're repeatedly calculating prices with 18% tax.

def calculate\_price(base\_price, tax\_rate):

return base\_price \* (1 + tax\_rate)

### **✅ Partial Function:**

from functools import partial

# Create GST-specific billing function

price\_with\_gst = partial(calculate\_price, tax\_rate=0.18)

# Use it with just base price

print(price\_with\_gst(1000)) # 1180.0

print(price\_with\_gst(500)) # 590.0

🟢 **Why it’s useful**: You don’t need to pass 0.18 every single time.

## **📦 Example 2: Gmail Email Generator**

### **💡 Problem:**

You're always building emails ending with @gmail.com.

def build\_email(username, domain):

return f"{username}@{domain}"

### **✅ Partial Function:**

from functools import partial

# Lock domain to gmail.com

create\_gmail = partial(build\_email, domain="gmail.com")

# Now only provide the username

print(create\_gmail("gowtham")) # gowtham@gmail.com

print(create\_gmail("rahul")) # rahul@gmail.com

🟢 **Why it’s useful**: Cleaner code for apps where domain is fixed (e.g., corporate or Gmail signup forms)

## **🧠 Where to Use partial in Real Projects?**

| **Use Case** | **Benefit** |
| --- | --- |
| API wrappers | Pre-fill headers or tokens |
| Email / username generators | Fix domain, pass username only |
| Discount/tax calculators | Lock rate, pass only base price |
| File handling | Fix encoding/flags, pass file only |
| Loop-based processing functions | Avoid repeating common args |

## **🏁 Summary**

| **Feature** | **Description** |
| --- | --- |
| Module | functools |
| Method | partial() |
| Returns | A new function with fixed values |
| Use when... | You repeat same arguments again and again |

### **📌 Syntax Recap:**

from functools import partial

new\_func = partial(original\_func, fixed\_arg=value)

## **✅ TL;DR (1-liner):**

Use functools.partial when you want to **pre-fill some arguments** of a function and create a **simplified version**.

### **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>