

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	<code>functools</code>
Method	<code>partial()</code>
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)
```

Gowtham SB

www.linkedin.com/in/sbgowtham/

Instagram - @dataengineeringtamil

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/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>

Gowtham SB

www.linkedin.com/in/sbgowtham/

Instagram - @dataengineeringtamil

💬 WhatsApp - <https://lnkd.in/g5JrHw8q>

✉️ Email - atozknowledge.com@gmail.com

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