

- Functional abstraction operates at different levels of granularity. Functions abstract away details within a single operation, while coroutines abstract away the management of control flow and asynchrony.

Certainly! Let's explore the differences between these concepts with examples:

## 1. Functions:

A function is a self-contained block of code that performs a specific task and can return a value. Functions help modularize code and promote reusability.

Example (Python):

```
def add(a, b):  
    return a + b  
  
result = add(3, 5)  
print(result) # Output: 8
```

## 2. Procedures:

A procedure is a type of function that doesn't return a value. It performs actions or operations without producing a result