

# Decorators and Generators

---

# Decorators

---

What is a Decorator?

- A **decorator** is a function that modifies or enhances another function without changing its structure.
- Think of it as a wrapper that adds additional functionality to the original function.
- Decorators are widely used in Python to extend the behaviour of functions or methods.

# How Does a Decorator Work?

---



**Takes a Function as Input:** A decorator function receives another function as an argument.



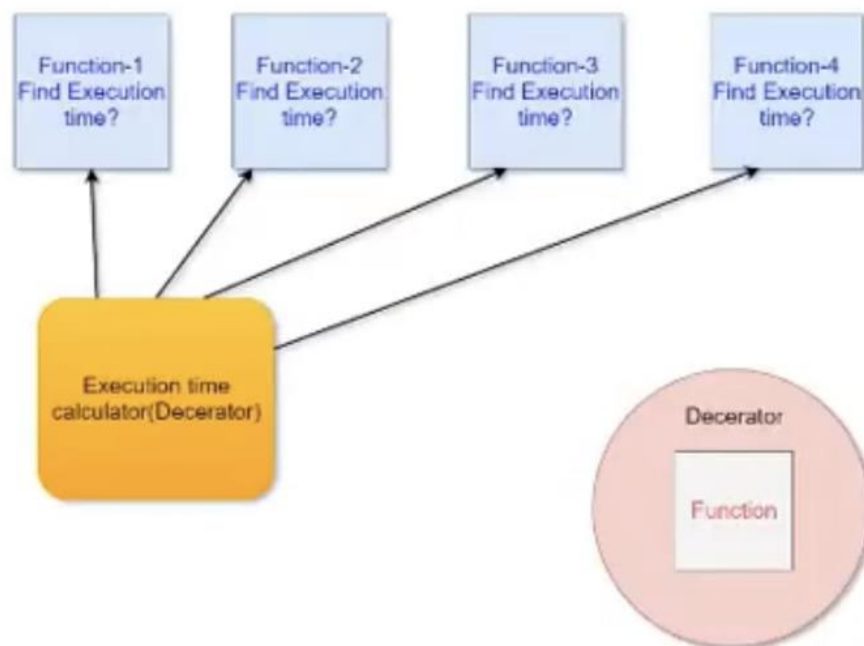
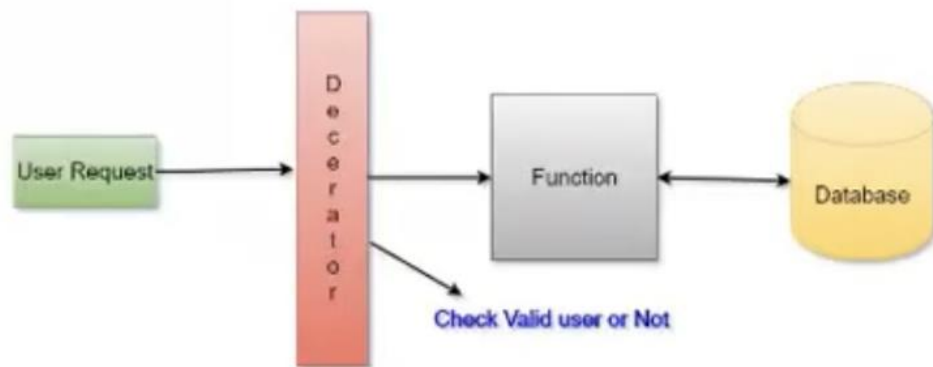
**Defines a New Function:** Inside the decorator, a new function (let's call it 'addon') is defined.



**Calls the Original Function:** The new function calls the original function and then adds some extra features.



**Returns the New Function:** Finally, the decorator returns the new function, which can be called as if it were the original.



# Generators

---



## What is a Generator?



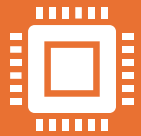
A **generator** is a special type of function that returns an iterator, which we can loop through (iterate over) one value at a time.



Unlike regular functions that return a single value and exit, generators use the `yield` keyword to produce a series of values, pausing between each.

# Why Use Generators?

---



**Memory Efficiency:** Generators are memory-efficient because they produce items one at a time and don't store the entire sequence in memory.



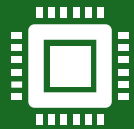
**Lazy Evaluation:** They compute the values only when needed, which is useful for large datasets or infinite sequences.

# Key Points:

---



**Decorators** are powerful tools for extending the functionality of functions without modifying their structure. They are widely used in frameworks and libraries.



**Generators** offer a memory-efficient way to iterate over sequences, especially when dealing with large data sets or streams of data.

## Conclusion:

- Understanding these concepts is crucial for writing efficient and clean Python code, particularly in advanced applications like web development, data processing, and more.
- Decorators and generators serve different purposes: decorators are used to modify the behaviour of functions, while generators are used to iterate over sequences lazily.





**Thank you**