**API documentation:**

## Pareto Execution API

The Pareto Execution API is a Python module that implements the 80/20 rule, also known as Pareto's Law.

Functions

#### pareto\_execute(task)

**Code:**  ***pareto\_execute(task: Task) -> None***

The pareto\_execute function takes a Task object as input and simulates the time allocation according to Pareto's Law. It calculates the time to spend on reading and understanding and executing the task based on the task's duration. It then calls the read\_and\_understand and execute\_task functions, passing the calculated durations.

**Parameters:**

task (Task): An instance of the Task class representing the task to be executed.

**Returns:**

None

read\_and\_understand(duration)

**Code: *read\_and\_understand(duration: float) -> None***

The read\_and\_understand function simulates the time spent on reading and understanding the task. In the provided code, it simply prints the duration that is passed as an argument.

**Parameters:**

duration (float): The duration of time to spend on reading and understanding.

**Returns:**

None

execute\_task(duration)

**Code: *execute\_task(duration: float) -> None***

The execute\_task function simulates the time spent on executing the task. In the provided code, it also prints the duration that is passed as an argument.

**Parameters:**

duration (float): The duration of time to spend on executing the task.

**Returns:**

None

**Classes**

Task

**Code: *class Task:***

***def \_\_init\_\_(self, duration: float)***

The Task class represents a task to be executed. It has a single attribute:

**Attributes:**

**duration (float):** The duration of the task, representing the total time it takes to complete.

**Code: *task = Task(10)***

***pareto\_execute(task)***

It’s creates a Task object with a duration of 10 units of time. It then calls the pareto\_execute function, which simulates the time allocation according to Pareto's Law and outputs the simulated time spent on reading and understanding and executing the task.