Code Understanding Report

Generated: 2025-05-05 23:52:57

This report presents automated insights based on large language models and code analysis tools.

File: pasted_code.js

Summary

- This Python code defines a function decorator `repeat` that repeats a function `num_times` number of times.

Here's how the code works:

First, `repeat` function accepts a number of repetitions `num_times`. Then, `decorator_repeat` function is a closure around the original function, this closure maintains the state of the original function, including its name, its arguments, and its docstring. In this closure, we create a new function `wrapper` which executes the original function `func`, repeated `num_times` times

Docstring

- ### Code: from functools import wraps

def repeat(num*times): def decorator*repeat(func): @wraps(func) def wrapper(*args,* *kwargs) for _ in range(num*times): result = func(*args, **kwargs) return result return wrapper return decorator*repeat

@repeat(num_times=3) def greet(name): print(f"Hello, {name}!")

greet("Alice")

Docstring:

The `repeat` function is a decorator that takes a number of times to repeat a function.

The `@repeat(num_times=3)` syntax is a way to apply the decorator to a function. The

Code Quality

Tool: `eslint`
Issues: 0`

```
text [ESLint Not Found] [WinError 2] The system cannot find the file specified — assuming valid JS.
```

Conclusion

This code defines a Python function decorator `repeat` that repeats a function `num_times` number of times. The purpose of this decorator is to provide a way to apply certain actions repeatedly in order to enrich a program, or to increase code readability, while keeping the specific logic separate from the main function. It allows for more readable and maintainable code in certain situations.