## Overview

This file serves as the main entry point for the code\_monitor package when executed as a module (e.g., python -m code\_monitor). It integrates with the click command-line interface library to launch the core analysis functionality. The script calls the analyze function from a sibling module (.main) and includes logic to gracefully handle exits triggered by click for informational commands like --help.

## Functions

### main()

This function acts as the primary entry point for the command-line application. It invokes the analyze function, which contains the core logic of the tool.

It specifically calls analyze with standalone\_mode=False. This is a common pattern when using click to indicate that the function is part of a larger command group and should not handle its own command-line parsing or execution flow independently.

* **Parameters**: None
* **Returns**: None

### Script Execution Block

The if \_\_name\_\_ == "\_\_main\_\_": block is the standard Python construct that runs when the script is executed directly.

It calls the main() function within a try...except block. This block is designed to catch SystemExit exceptions, which click raises after processing certain command-line arguments that terminate the program, such as --help. The code checks the exit code and only re-raises the exception if it’s not a clean exit (i.e., the code is not 0 or None), preventing normal help-related exits from being treated as errors.

#### Usage Example

To run the application, you would execute the package as a module from your terminal:

python -m code\_monitor [OPTIONS] [COMMAND]