## Overview

This file serves as the main entry point for a command-line interface (CLI) tool designed to analyze a Git repository. It identifies changes at the function and class level within Python files. The tool can analyze all uncommitted changes or only staged changes, making it suitable for use in pre-commit hooks. It uses the click library for its CLI and colorama to provide color-coded output for better readability.

## Functions

### add\_numbers()

Adds two integers and returns the result.

* **Parameters:**
  + a (int): The first number.
  + b (int): The second number.
* **Returns:**
  + int: The sum of a and b.
* **Example:**
* >>> add\_numbers(3, 5)  
  8

### sub\_numbers()

Subtracts the second integer from the first and returns the result.

* **Parameters:**
  + a (int): The first number.
  + b (int): The second number.
* **Returns:**
  + int: The difference between a and b.
* **Example:**
* >>> sub\_numbers(5, 3)  
  2

### print\_analysis()

Prints the analysis results to the console in a structured and color-coded format.

* **Parameters:**
  + results (dict): A dictionary where keys are file paths and values are dictionaries containing details about the changes. The inner dictionary includes the file’s status (‘A’ for added, ‘M’ for modified, ‘D’ for deleted) and lists of added, removed, or modified code structures (functions/classes).
* **Returns:**
  + None.

### analyze()

The main CLI command function that orchestrates the repository analysis. It initializes the RepoAnalyzer, performs the analysis of changes, and then calls print\_analysis() to display the results. This function is decorated to handle command-line arguments.

* **Parameters:**
  + path (str): The file path to the Git repository to be analyzed. Defaults to the current directory (.).
  + staged\_only (bool): A flag that, when set, restricts the analysis to only staged files. Defaults to False.
* **Returns:**
  + None.
* **Usage Example:**
* Analyze all uncommitted changes in the current directory:
* python main.py
* Analyze only staged changes in a specific repository path:
* python main.py --path /path/to/your/repo --staged-only