Analyzes php-code for side-effects.

When code has no side-effects it can e.g. be used with eval($code) in the same process without interfering. [Side-effects are classified](https://github.com/staabm/side-effects-detector/blob/main/lib/SideEffect.php) into categories to filter them more easily depending on your use-case.

## Install

composer require staabm/side-effects-detector

## Usage

Example:

use staabm\SideEffectsDetector\SideEffectsDetector;

$code = '<?php version\_compare(PHP\_VERSION, "8.0", ">=") or echo("skip because attributes are only available since PHP 8.0");';

$detector = new SideEffectsDetector();

// [SideEffect::STANDARD\_OUTPUT]

var\_dump($detector->getSideEffects($code));

In case functions are called which are not known to have side-effects - e.g. userland functions - null is returned.

use staabm\SideEffectsDetector\SideEffectsDetector;

$code = '<?php userlandFunction();';

$detector = new SideEffectsDetector();

// [SideEffect::MAYBE]

var\_dump($detector->getSideEffects($code));

Code might have multiple side-effects:

use staabm\SideEffectsDetector\SideEffectsDetector;

$code = '<?php include "some-file.php"; echo "hello world"; exit(1);';

$detector = new SideEffectsDetector();

// [SideEffect::SCOPE\_POLLUTION, SideEffect::STANDARD\_OUTPUT, SideEffect::PROCESS\_EXIT]

var\_dump($detector->getSideEffects($code));

## Disclaimer

Non goals are:

* find the best possible answer for all cases
* add runtime dependencies

If you are in need of a fully fledged side-effect analysis, use more advanced tools like PHPStan.

Look at the test-suite to get an idea of [supported use-cases](https://github.com/staabm/side-effects-detector/blob/main/tests/SideEffectsDetectorTest.php).