

**CHRISTMAS IS COMING
TIME TO DECORATE YOUR TYPESCRIPT FUNCTIONS**

GOAL OF THIS TALK

GOAL OF THIS TALK

- ◆ Increase awareness of decorators
- ◆ Show how to write and use them
- ◆ Show some (hopefully) interesting examples
- ◆ Spark curiosity to try them on your own

WHO AM I?

WHO AM I?



Marco Sieben

- ◆ Fullstack developer with a heart for frontend
- ◆ TypeScript enthusiast
- ◆ Loves annual Advent of Code (=> Sleep deprived in december)
- ◆ Discovered a nice usage for decorators last year

WHO AM I?



Marco Sieben

- ◆ Fullstack developer with a heart for frontend
- ◆ TypeScript enthusiast
- ◆ Loves annual Advent of Code (=> Sleep deprived in december)
- ◆ Discovered a nice usage for decorators last year

Shoutout to my employer andrena objects ❤

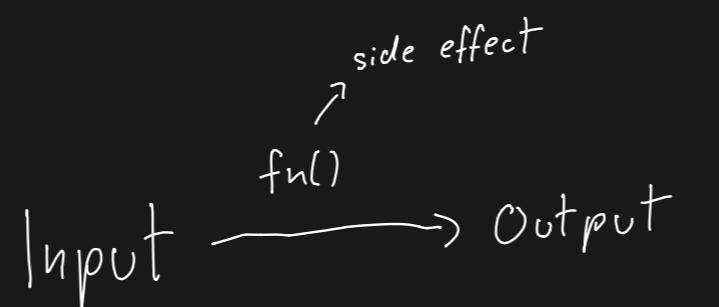
DECORATORS

DECORATORS

THEORY

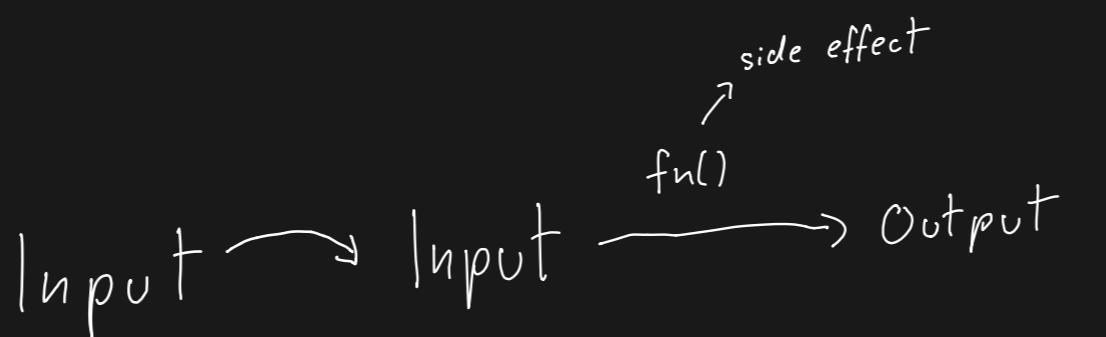
DECORATORS

THEORY



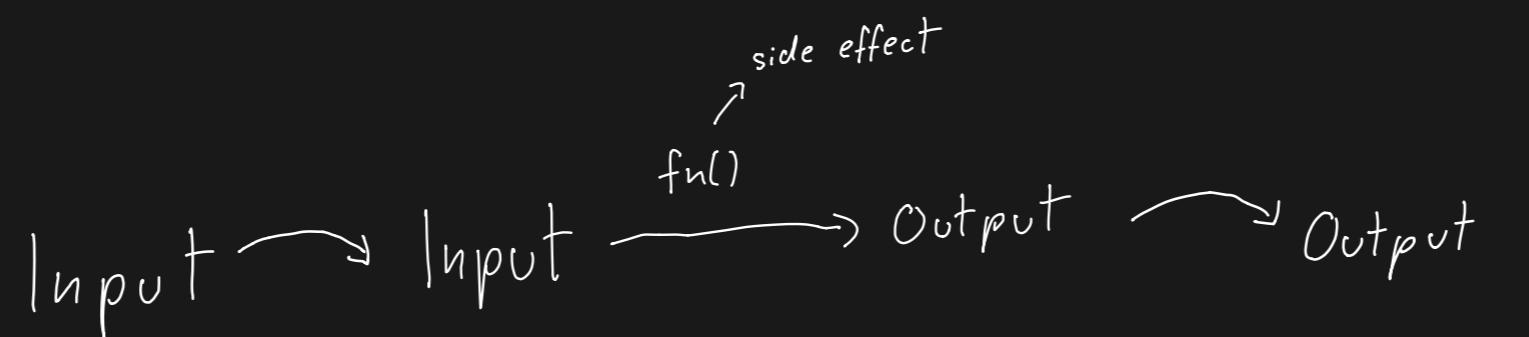
DECORATORS

THEORY



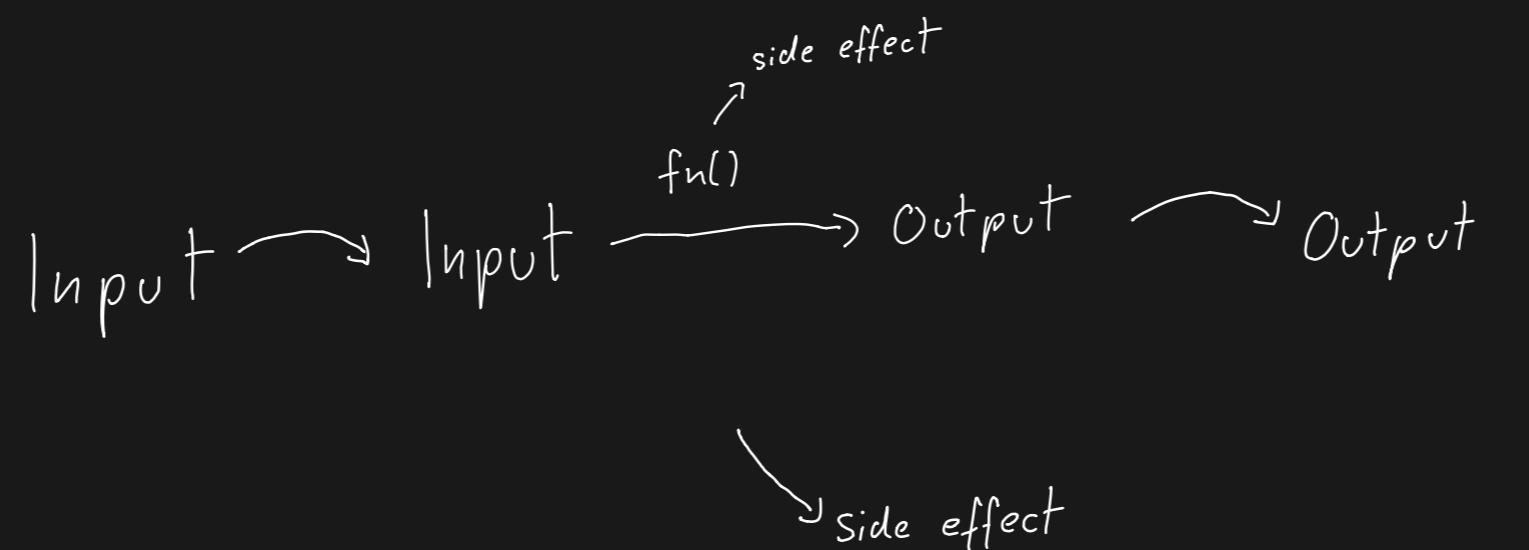
DECORATORS

THEORY



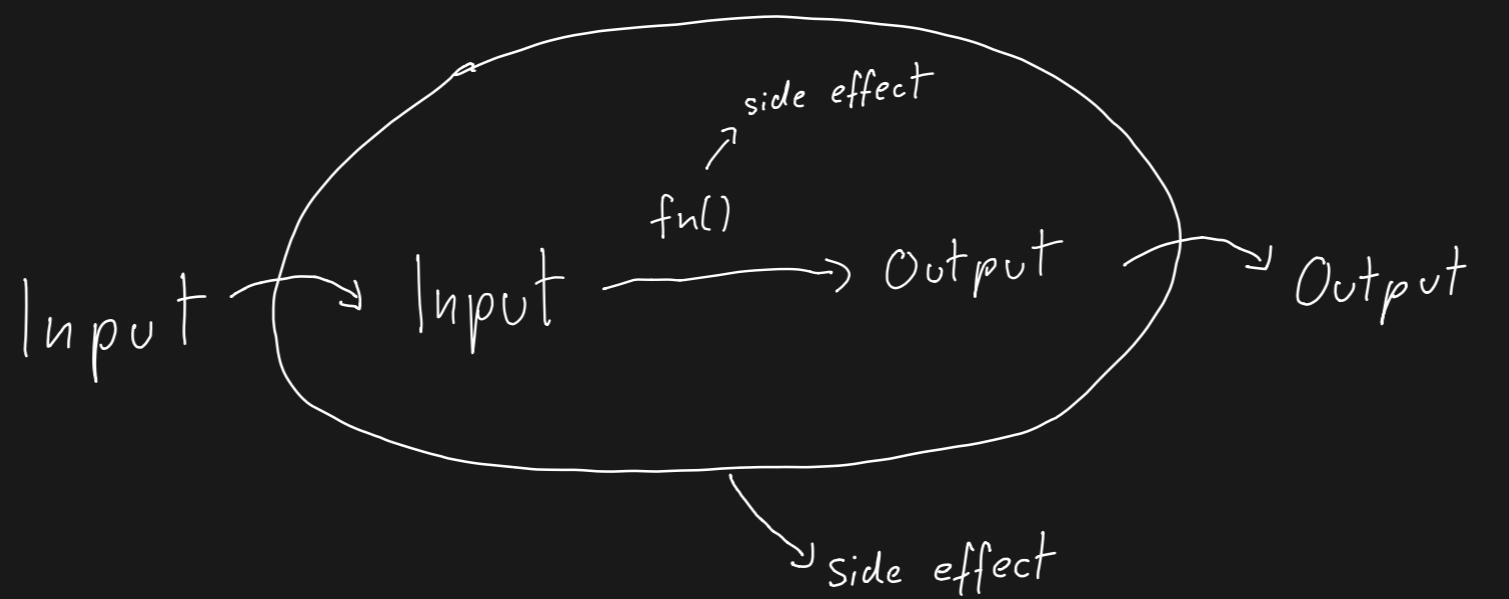
DECORATORS

THEORY



DECORATORS

THEORY



DECORATORS

DECORATORS IN TYPESCRIPT

DECORATORS IN TYPESCRIPT

- ◆ Class method decorators
- ◆ Class decorators
- ◆ Class getter/setter decorators
- ◆ Class field decorators
- ◆ Class auto-accessor decorators

DECORATORS IN TYPESCRIPT

- ◆ Class method decorators
- ◆ Class decorators
- ◆ Class getter/setter decorators
- ◆ Class field decorators
- ◆ Class auto-accessor decorators

CLASS METHOD DECORATORS

CLASS METHOD DECORATORS

EXAMPLE: LOGGED

CLASS METHOD DECORATORS

INTERFACE

```
type ClassMethodDecorator = (
  value: Function,
  context: {
    kind: 'method';
    name: string | symbol;
    metadata: object;
    static: boolean;
    private: boolean;
    access: { get: () => unknown };
    addInitializer(initializer: () => void): void;
  }
) => Function | void;
```

ADVENT OF CODE

Advent of Code [About] [Events] [Shop] [Settings] [Log Out] StNimmerlein (AoC++)
y (2024) [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

Hi! I'm Eric Wastl. I make Advent of Code. I hope you like it! I also made Vanilla JS, PHP Sadness, and lots of other things. You can find me on Twitter, Mastodon, and GitHub.

Advent of Code is an [Advent calendar](#) of small programming puzzles for a variety of skill sets and skill levels that can be solved in [any](#) programming language you like. People use them as [interview prep](#), [company training](#), [university coursework](#), [practice problems](#), a [speed contest](#), or to [challenge each other](#).

You don't need a computer science background to participate - just a little programming knowledge and some [problem solving skills](#) will get you pretty far. Nor do you need a fancy computer; every problem has a solution that completes in at most 15 seconds on ten-year-old hardware.

If you'd like to support Advent of Code, you can do so indirectly by helping to [\[Share\]](#) it with others or directly via [AoC++](#).

--- General Tips ---

If you get stuck, try your solution against the [examples](#) given in the puzzle; you should get the same answers. If not, re-read the description. Did you misunderstand something? Is your program doing something you don't expect? After the examples work, if your answer still isn't correct, build [some test cases](#) for which you can verify the answer by hand and see if those work with your program. Make sure you have the entire puzzle input. If you're still stuck, maybe ask a friend for help, or come back to the puzzle later. You can also ask for hints in the [subreddit](#).

--- Frequently Asked Questions ---

Is there an easy way to select entire code blocks? You should be able to triple-click code blocks to select them. You'll need JavaScript enabled.

```
#!/usr/bin/env perl  
use warnings;
```

andrena
OBJECTS

- ◆ Daily challenges (Dec 1 until Dec 25)
- ◆ Embedded in a nice story
- ◆ No specified language
- ◆ Global and private leaderboards
- ◆ Completely free

Check it out at
<https://adventofcode.com>

CLASS METHOD DECORATORS

EXAMPLE 2: MEMOIZATION

CLASS DECORATORS

CLASS DECORATORS INTERFACE

```
type ClassDecorator = (
  value: Function,
  context: {
    kind: 'class';
    name: string | undefined;
    metadata: object;
    addInitializer(initializer: () => void): void;
  }
) => Function | void;
```

CLASS DECORATORS

EXAMPLE: AUTOBIND

CLASS DECORATORS

EXAMPLE 2: SINGLETON

CLASS DECORATORS

EXAMPLE 3: DEPENDENCY INJECTION

MORE

RELATED TOPICS

- ◆ Other decorator types (field, getter/setter, auto accessor)
- ◆ Decorator metadata
- ◆ Auto accessors

FURTHER READING

- ◆ [Release notes of TypeScript 5.0 \(Decorators\)](#)
- ◆ [Release notes of TypeScript 5.2 \(Decorator metadata\)](#)
- ◆ [Extensive blog post from Axel Rauschmayer about TypeScript decorators](#)

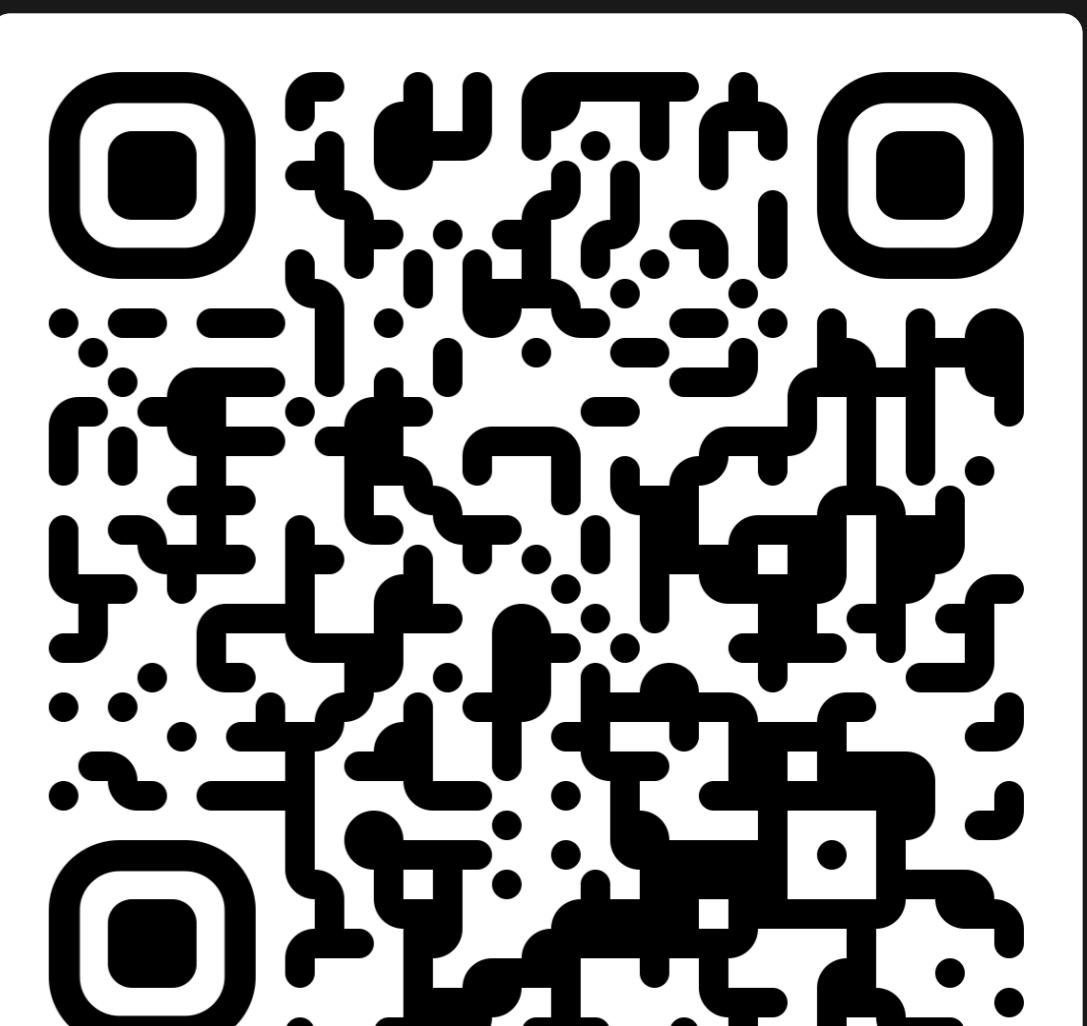
Marco Sieben

 marco.sieben@andrena.de

 <https://www.linkedin.com/in/stnimmerlein>

andrena
OBJECTS

Repository with slides and examples:

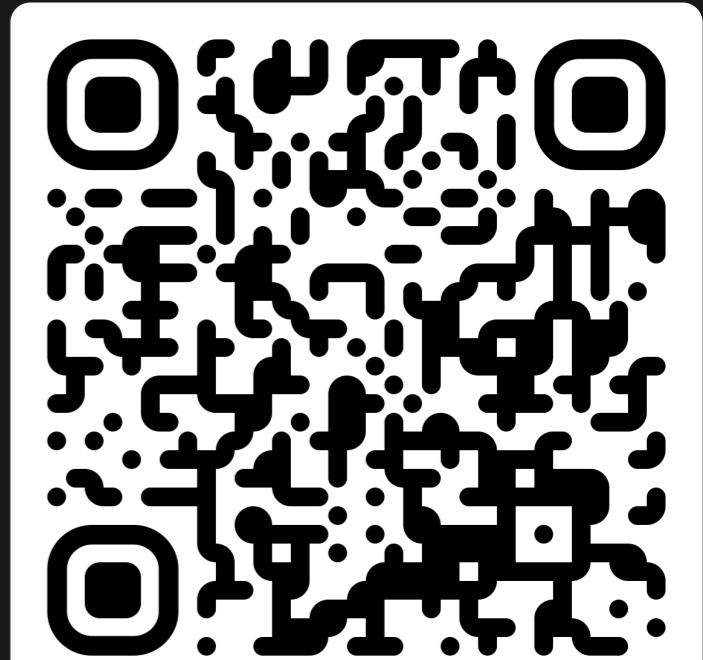


THANK YOU

AND HAVE FUN DECORATING



Repository with slides and examples:



Marco Sieben

marco.sieben@andrena.de

<https://www.linkedin.com/in/stnimmerlein>