

Improve your code with

Type Guards



I am Benny, nice to meet you!



[@bennycode](https://twitter.com/bennycode)



[@bennycode](https://github.com/bennycode)



bennycode.com



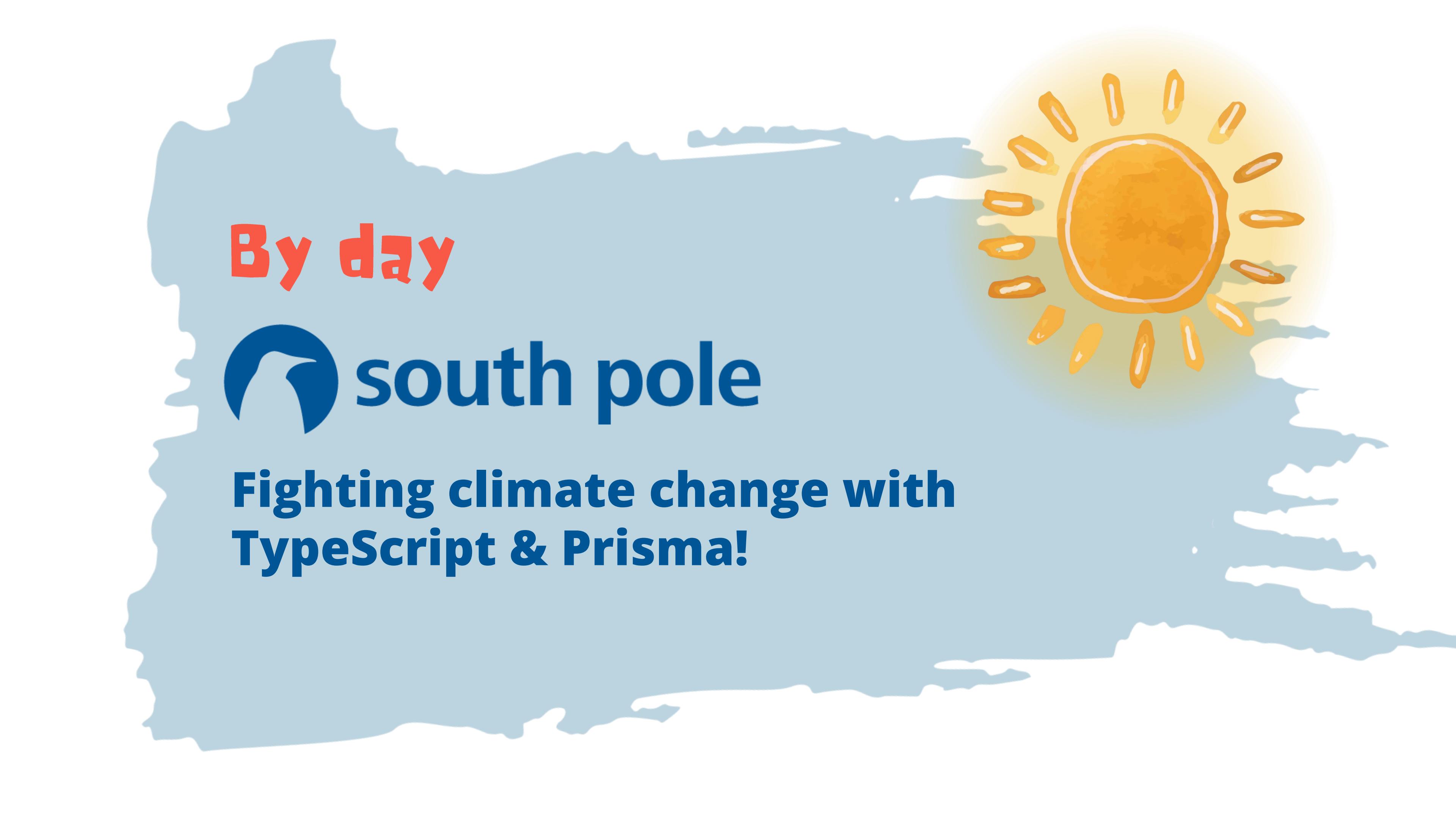


By day



south pole

**Helping organizations and individuals
to calculate, offset and reduce their
emissions (through software).**



By day



south pole

Fighting climate change with
TypeScript & Prisma!

At night



What is a Guard?

Guard (computer science)

From Wikipedia, the free encyclopedia

In computer programming, a **guard** is a boolean expression that must evaluate to true if the program execution is to continue in the branch in question.

What is a Guard?

```
def func(val: Union[str, float]):  
    # "isinstance" type guard  
    if isinstance(val, str):  
        # Type of val is narrowed to str  
        ...  
    else:  
        # Type of val is narrowed to float  
        ...
```

What is a Guard?

 TypeScript

[Download](#) [Docs](#) [Handbook](#) [Community](#) [Playground](#) [Tools](#)

It just so happens that TypeScript has something called a *type guard*. A type guard is some expression that performs a runtime check that guarantees the type in some scope.

Built-in Type Guards

`typeof`

Check for
primitive
types

`in`

Check for
object
properties

`instanceof`

Check for
class
instances

Built-in Type Guards



`typeof`



`in`



`instanceof`

User-Defined Type Guards



Credits

Guard (computer science),
Wikipedia

Widening and Narrowing in TypeScript,
Nathan Shively-Sanders

How to use type guards in TypeScript,
Oyinkansola Awosan

User-Defined Type Guards,
TypeScript Handbook

@tool-belt/type-predicates,
Na'aman Hirschfeld