

src\typeguard.ts

```
1  /** A typeof guard in TypeScript lets you narrow down the type of a variable based on
   its runtime value. In TypeScript, type narrowing allows you to write type-safe code by
   ensuring you only operate on the correct type under certain circumstances. This is
   particularly useful with union types and generic types.
2
3  const favHobbies = (hobby: string | string[]) => {
4      // return hobby.map(() => {});
5      if (typeof hobby === "object" && Array.isArray(hobby)) {
6          return hobby.map(() => {});
7      } else {
8          console.log(hobby);
9      }
10 };
11
12 favHobbies("coding");
13 favHobbies(["coding", "games"]);
14
```