

```
1 // Punto 1
2 function plantarArboles(cantidad, tipo) {
3   console.log(`${cantidad} ${tipo} han sido plantados.`);
4 }
5
6 // Punto 2
7 function plantarArboles(cantidad: number, tipo: string): void {
8   console.log(`${cantidad} ${tipo} han sido plantados.`);
9 }
10
11 // Punto 3
12 function obtenerAleatorio(max: number, min?: number | undefined): number {
13   min ??= 0;
14   return Math.floor(Math.random() * (max - min + 1)) + min;
15 }
16
17 // Punto 4
18 const palabras: string[] = ["murcielago", "microondas", "superdomesticos"];
19
20 palabras.forEach(palabra => {
21   if (palabra.length % 2 === 0) {
22     console.log(palabra.toUpperCase());
23   } else {
24     console.log(palabra);
25   }
26 });
```

```
[LOG]: "MURCIELAGO"
[LOG]: "MICROONDAS"
[LOG]: "superdomesticos"
```

Customize

Site Colours:

Code Font:

Popular Documentation Pages

- [Everyday Types](#)

All of the common types in TypeScript
- [Creating Types from Types](#)

Techniques to make more elegant types
- [More on Functions](#)

How to provide types to functions in JavaScript
- [More on Objects](#)

How to provide a type shape to JavaScript objects
- [Narrowing](#)

How TypeScript infers types based on runtime behavior
- [Variable Declarations](#)

How to create and type JavaScript variables
- [TypeScript in 5 minutes](#)

An overview of building a TypeScript web app
- [TSConfig Options](#)

All the configuration options for a project
- [Classes](#)

How to provide types to JavaScript ES6 classes

Community

- [Get Help](#)

[@TypeScript](#)
- [Blog](#)

[Stack Overflow](#)
- [GitHub Repo](#)

[Web Repo](#)
- [Community Chat](#)

Using TypeScript

- [Get Started](#)


[TSConfig Ref](#)
- [Download](#)

[Why TypeScript](#)
- [Community](#)

[Design](#)
- [Playground](#)

[Code Samples](#)

Made with ♥ in Redmond, Boston, SF & Dublin


© 2012-2022 Microsoft
[Privacy](#)