Everyday Types in Angular

<https://www.typescriptlang.org/docs/handbook/2/everyday-types.html>

In this chapter, we’ll cover some of the most common types of values you’ll find in JavaScript code, and explain the corresponding ways to describe those types in TypeScript. This isn’t an exhaustive list, and future chapters will describe more ways to name and use other types.

Types can also appear in many more *places* than just type annotations. As we learn about the types themselves, we’ll also learn about the places where we can refer to these types to form new constructs.

We’ll start by reviewing the most basic and common types you might encounter when writing JavaScript or TypeScript code. These will later form the core building blocks of more complex types.

# The primitives:string, number, and boolean

JavaScript has three very commonly used [primitives](https://developer.mozilla.org/en-US/docs/Glossary/Primitive): string, number, and boolean. Each has a corresponding type in TypeScript. As you might expect, these are the same names you’d see if you used the JavaScript typeof operator on a value of those types:

* string represents string values like "Hello, world"
* number is for numbers like 42. JavaScript does not have a special runtime value for integers, so there’s no equivalent to int or float - everything is simply number
* boolean is for the two values true and false

The type names String, Number, and Boolean (starting with capital letters) are legal, but refer to some special built-in types that will very rarely appear in your code. Always use string, number, or boolean for types.