

# Using the never type

In this lesson, we'll learn about the 'never' type and how it is different from the 'void' type.

## WE'LL COVER THE FOLLOWING ^

- An example
- Wrap up

## An example #

The code below is an arrow function expression that contains an infinite loop where a message is output to the console:

</> TypeScript

```
const keepLogging = (message: string) => {  
  while (true) {  
    console.log(message);  
  }  
}
```



What do you think the return type is? Could the return type be the `void` type we have just learned about? Hover over `keepLogging` to discover its inferred type.

 Show Answer

## Wrap up #

So, the `never` type is used to represent a type of value that will never occur.

The above example is contrived, but there are some use cases for `never` when we want to flag areas of code that shouldn't be reached explicitly. We'll use

the `never` type [later in this course](#) when we implement a super type-safe reducer function.

More information can be found about the `never` type in the [TypeScript handbook](#).

In the next lesson, we'll learn about a type that we can use as an alternative to `any`.