Step 10 - Generics

Generics are a language independent concept (exist in C++ as well)

Let's learn it via an example

1. Problem Statement

Let's say you have a function that needs to return the first element of an array. Array can be of type either string or integer.

How would you solve this problem?

▶ Solution

What is the problem in this approach?

▶ User can send different types of values in inputs, without any type errors

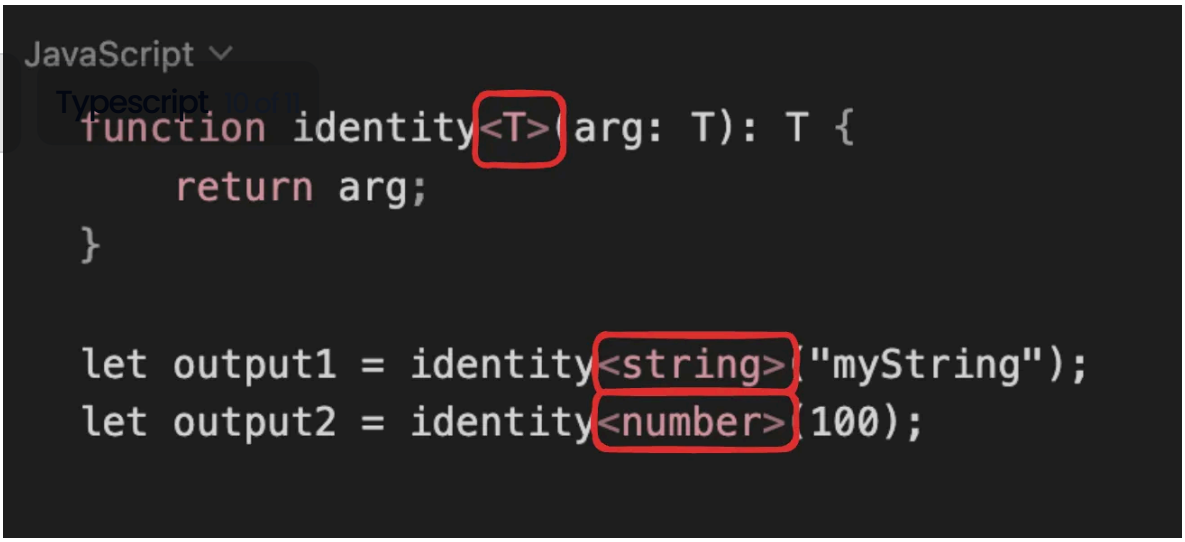
▶ Typescript isn't able to infer the right type of the return type

2. Solution - Generics

Generics enable you to create components that work with any data type while still providing compile-time type safety.

Simple example -

▶ Code



3. Solution to original problem

Can you modify the code of the original problem now to include generics in it?

```
function getFirstElement<T>(arr: T[]) {
  return arr[0];
}

const el = getFirstElement(["harkiratSingh", "ramanSingh"]);
console.log(el.toLowerCase())
```

Did the issues go away?

▶ User can send different types of values in inputs, without any type errors

▶ Typescript isn't able to infer the right type of the return type