

Introduktion till Generics i TypeScript

Vad är Generics?

- Generics gör det möjligt att skapa **återanvändbara funktioner och komponenter**.
- Istället för att låsa fast sig vid en specifik typ kan vi använda en **generisk parameter**.
- Detta gör koden **flexibel** och samtidigt **typsäker**.

Exempel 1: Enkel generisk funktion

Vi skapar en funktion `genericFunction` som tar ett värde och returnerar det.

```
const genericFunction = <T,>(value: T) => value;  
  
const numberValue = genericFunction(42); // Returnerar 42  
const stringValue = genericFunction("Hej!"); // Returnerar "Hej!"
```

Här används `<T,>` för att låta funktionen fungera med **vilken typ som helst**.

Exempel 2: Generisk funktion för arrays

Vi skapar en funktion `getFirstElementInArray` som returnerar det första elementet i en array.

```
const getFirstElementInArray = <T,>(array: T[]) => array[0];  
  
const firstNumber = getFirstElementInArray([1, 2, 3]); // Returnerar 1  
const firstWord = getFirstElementInArray(["äpple", "banan", "citron"]); // Returnerar "äpple"
```

Här används `T[]` för att ange att parametern är en **array av valfri typ**.