

GENERICICS

In TypeScript, i generics sono uno strumento che consente di creare componenti riutilizzabili e flessibili nel tipo di dati che possono gestire. I generics consentono di definire funzioni, classi e interfacce che possono lavorare con diversi tipi di dati senza specificarli in modo rigido.

La sintassi per dichiarare un generico in TypeScript è utilizzando delle parentesi angolari < >. Puoi utilizzare qualsiasi nome per il generico, anche se solitamente si utilizzano nomi come T, U, V, ecc. Ad esempio, ecco come puoi dichiarare una funzione generica che restituisce l'elemento di un array in base all'indice specificato:

typescriptCopy code

```
function getElement<T>(array: T[], index: number): T {  
  return array[index];  
}
```

Nell'esempio sopra, T rappresenta il tipo generico. La funzione getElement accetta un array di tipo T[] e restituisce un valore di tipo T. Questo significa che la funzione può essere utilizzata con qualsiasi tipo di array, come ad esempio number[], string[], boolean[], ecc.

Ecco un esempio di utilizzo della funzione getElement:

typescriptCopy code

```
const numbers = [1, 2, 3, 4, 5];  
const strings = ['apple', 'banana', 'orange'];  
  
const numberElement = getElement(numbers, 2); // Restituisce 3 di tipo number  
const stringElement = getElement(strings, 1); // Restituisce 'banana' di tipo string
```

In questo esempio, i generics consentono alla funzione getElement di adattarsi dinamicamente ai tipi di dati degli array passati come argomenti, fornendo un controllo statico del tipo durante la compilazione.

È possibile utilizzare più generics contemporaneamente. Ad esempio, ecco come potresti dichiarare una funzione generica che scambia la posizione di due elementi in un array:

typescriptCopy code

```
function swapElements<T>(array: T[], index1: number, index2: number): void {  
  const temp = array[index1];  
  array[index1] = array[index2];  
  array[index2] = temp;  
}
```

In questo caso, il generico T viene utilizzato per rappresentare il tipo degli elementi dell'array. La funzione swapElements prende un array di tipo T[] e due indici, e scambia la posizione degli elementi corrispondenti agli indici specificati.

```
const numbers = [1, 2, 3, 4, 5];  
const strings = ['apple', 'banana', 'orange'];  
  
swapElements(numbers, 1, 3); // Scambia gli elementi 2 e 4 nell'array dei numeri  
swapElements(strings, 0, 2); // Scambia gli elementi 'apple' e 'orange' nell'array delle stringhe
```