

32 | 泛型中的推断、约束和缺省

周爱民 (Aimingoo)

目录

1 泛型中的推断、约束和缺省

2 总结

两种约束

1. 在模板的类型参数中的约束包括两种语义：“必须是”和“将会是”
2. 在泛型参数中的约束只有一种语义：“必须是”
 - ▶ 缺省不能替代约束（不具有约束的语义）

四种推断

1. TS的一般推断：如果没有声明类型，那么就推断类型

▶ `let x = 100; // 要么是推断类型，要么是断言类型，要么就标注类型`

2. 使用 `infer` 的显式推断：总是在 `extends` 的右侧

▶ `type T = U extends infer X ? X : never; // 可以推断联合分量、参数与结构成员`

▶ `type T = S extends `${infer X extends number}` ? X : never; // 可以推断格式化字符串的片断（模板变量）`

3. 在函数界面上的隐式推断：在泛型函数和泛型类的调用过程中推断

▶ `function foo<T>(a: T): T { return T };`

4. 阻止推断：`NoInfer<X>`

▶ `function foo<T>(a: T, b: NoInfer<T> | string): T { return T };`

总结

1. 推断、约束和缺省

- ▶ 类型参数与泛型参数中的两种约束
- ▶ 在两种泛型参数中的缺省
- ▶ TypeScript 中的四种推断

THANKS