

22 | 条件类型概要

周爱民 (Aimingoo)

目录

- 1 条件类型的基本语法
- 2 条件类型连用，以及它与三元表达式的语义区别
- 3 条件类型中左侧操作数（L）的特殊性
- 4 总结

①- 22 条件类型的概要

23 赋值兼容性的再说明（应用：统一“兼容性”的具体做法）

24 条件类型用作其它表达式类型的操作数（X、Y）

25 在条件表达式的操作数中使用其它表达式类型（L、R）

26 分布式条件类型的应用

①- 27 推断（infer）

①- 28 约束（constraint）

E：条件类型与赋值兼容性

表达式类型

- 1、作为表达式，自身要求值（表达式求值）
- 2、作为表达式类型，要参与其它运算符的运算（操作数）

类别	名称	运算符	(注)	求值结果	优先级
语法 (上下文受限)	✓ 分组/括号	(...)			10
	✓ 模板变量	\${T}	(注8)		10
	✓ 展开	...T			10
	✓ 映射	x in X as T			10
	✓ 模板字面字符串类型	` ... `	(注7)		9
	✓ 元组类型	[...]			9
	✓ 映射类型 (注3)	{ ... }			9
表达式	✓ 类型查询	typeof V	(* V是变量名)	任意类型	8
	✓ 索引访问 (类型)	T[K]	(* T和K都被立即求值)	联合 (包括任何单类型)	7
	✓ 键名查询	keyof T	(* T被立即求值)	联合 (包括任何单类型)	6
	✓ 交叉类型	A & B		单类型	5
	✓ 联合类型 (注2)	A B		联合 (包括单类型或交叉得 到的单类型)	4
	条件类型	extends		X, Y, X Y	3
		(保留)			2
语法 (上下文受限)	断言、标注等	as, <>, :			1
	约束	extends ...			1
	✓ 别名/缺省值 (注6)	= ...			1
		(保留, 例如, 号等)			0

表达式类型的三种运算语义

- 1、作为表达式，自身要求值（表达式求值）
- 2、作为表达式类型，要参与其它运算符的运算（操作数）
- 3、作为变量的类型声明，要接受其它类型的赋值（赋值兼容性）

条件类型

L extends R ? X : Y;

L : 左操作数 (*Source*)

R : 右操作数 (*Target*)

X : *True* 分支

Y : *False* 分支

总结

1. 条件类型的基本语法及其求值结果

- ▶ 基本语法 ``L extends R ? X : Y``
- ▶ 求值结果包括: `never`、`X`、`Y`、`X | Y`、`X = L = L & R`

2. 条件类型的特殊之处在于它的 `True/False` 分支都是被作为类型表达式求值的

- ▶ 这意味着结果可能是惰性的或未完全求值的
- ▶ 没有取反运算, 但是可以通过交换条件类型的两个分支来获得相同效果

3. 裸类型参数与分布式条件类型

- ▶ Union types 与 `never` type
- ▶ `any`

作业

＞使用条件类型来分别得到一个接口的签名和成员列表

// 可以写成泛型工具Signs<T>和Keys<T>

```
type T = {  
  [k: string|symbol|number]: string | number;  
  a: string;  
  b: number;  
  c: 1;  
}  
type keys = ...  
type signs = ...
```


THANKS