

# 20 | 映射类型

周爱民 (Aimingoo)

# 目录

- 1 映射类型的基本语法
- 2 映射类型中 Key 的限制
- 3 映射类型中 Key 的筛选
- 4 总结

# 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



# 总结

## 1. 映射类型的基本语法

- ▶ 在语法 `{ [k in U]: T }` 中, `k` 是 `U` 中的每一个分量, `T` 可以是任意类型
- ▶ `U` 的类型是 `string | number | symbol | never` 以及它们的子类型的联合

## 2. 映射类型中的 `as ...`

- ▶ 在语法 `{ [x in X as T]: ... }` 中, `T` 是临时断言的, 它是最终被使用的类型, 但不影响 `x` 原有的类型
- ▶ `T` 的类型是 `string | number | symbol | never` 以及它们的子类型的联合

## 3. 映射类型中 `in` 运算符的右侧的 `keyof ...` 的结果存在特殊性

# 作业

➤使用映射类型的方法，来得到一个元组类型对应的 ArrayLike 对象类型。

```
type T = [1, 2, 3, string, 'a', false];  
type X = { ... } ; // 映射 T 类型，得到的X是一个ArrayLike对象类型
```

THANKS