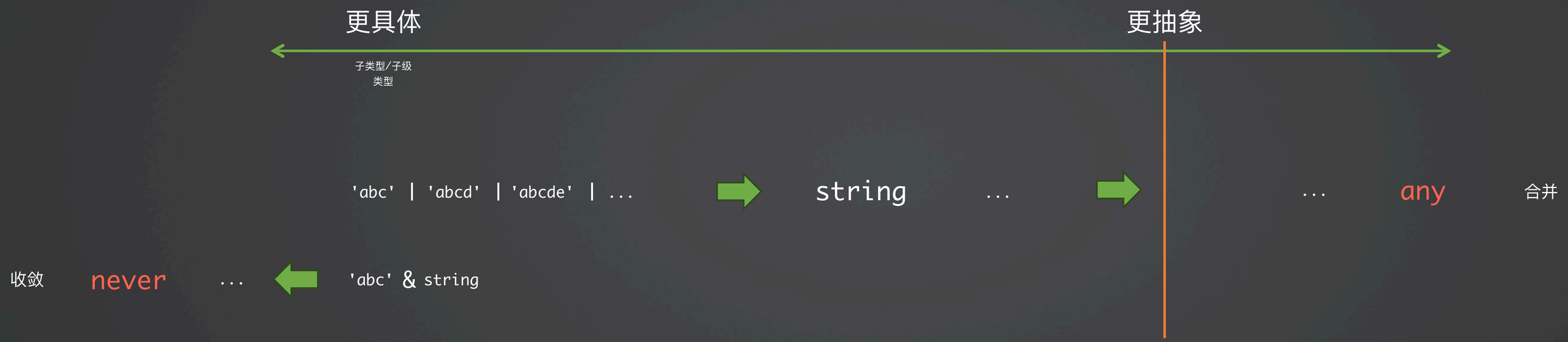
13 | 类型守护与类型收窄

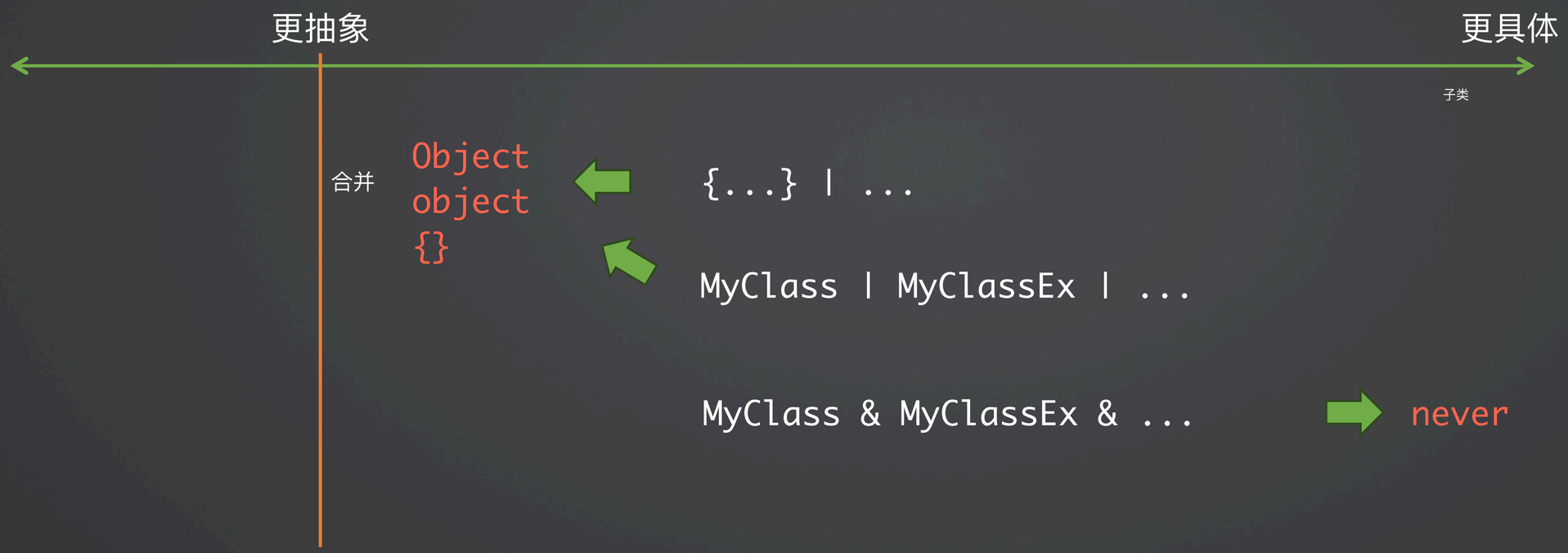
周爱民（Aimingoo）

目录

联合与交叉


更具体
更抽象
子类型/子级
类型
'abc' | 'abcd' | 'abcde' | ...
string
...
...
any
合并
收敛
never
...
'abc' & string

接口的联合与交叉


更抽象
更具体
子类
合并
Object
object
{}
{...} | ...
MyClass | MyClassEx | ...
MyClass & MyClassEx & ...
never
收敛

总结

1. 所谓类型收窄，就是在一定的作用域中，变量的类型比它原有的声明要小
   - 是原类型的子类型、更具体的类型
   - 赋值（推断）、断言，以及识别（守护）

2. 可辨识联合类型

3. 守护函数是特定语法，它通过额外的签名来为函数提供编译期语义。
   - 可以理解为谓词签名与断言签名的统称

名词/概念

名词、术语

谓词签名、断言签名：Predicate signatures / type predicates, Assertion signatures
类型加宽、类型收窄、类型推断：Type Widens, Type narrowing, Type inference
类型标注、类型断言：Type Annotation, Type assertion
类型守护、类型推断、推断：Type guards / Guards, Type inference, infer
可辨识联合：Discriminated unions
控制流分析：Control flow analysis

概念

- 类型加宽（Widens, *converts to a supertype*）和收窄（Narrowing, *refining types to more specific types than declared*），是类型推断行为的结果表现。谓词签名与断言签名是通过用户自定义的函数对其中的控制流分析过程进行人工干预，以显式地影响TypeScript内部类型推断结果。
  @see: https://www.typescriptlang.org/docs/handbook/2/narrowing.html
  @see: https://www.typescriptlang.org/docs/handbook/2/narrowing.html#control-flow-analysis
  @see: https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes-func.html#unit-types
- 断言函数（Assertion functions）和守护函数（User-defined type guard）
  @see: https://www.typescriptlang.org/docs/handbook/2/narrowing.html#using-type-predicates
  @see: https://www.typescriptlang.org/docs/handbook/release-notes/typescript-3-7.html#assertion-functions
- 类型推断（Type inference）针对的是具体变量，因此结果总是“某个变量的当前类型”。注意这在概念上有别于类型计算（Evaluation），后者的结果是一个“类型”。
  @see: https://typescriptlang.org/docs/handbook/typescript-from-scratch.html
- 推断（infer）作为TypeScript中的关键字，还用于在类型计算过程中表示（或返回、得出）新的类型。通常在语法上记为“infer X”，表明新类型在当前上下文中被暂时命名为X，这类似于创建了新类型的别名（*helper type aliases*）或泛型变量（*new generic type variable*）。
  @see: https://www.typescriptlang.org/docs/handbook/2/conditional-types.html#inferring-within-conditional-types

Q&A

Q: 有哪些运算会导致收窄?

A: 在TypeScript手册中，明确发生收窄效果的 JavaScript 表达式（运算）：赋值（=, +=等）、等值（===）、typeof、in、instanceof，以及用户定制的守护（谓词和断言签名）。

Q: 在课程中提及的“++、&&”等会导致收窄吗?

A: 基于上述的解释，++ 和 && 这些一般运算是不会导致类型收窄的，这是因为它们不会“关联到”某个变量名字。但是，这些运算是会影响类型识别或类型推断结果的，例如“`x = a && b`”，这会使得 x 收窄到布尔值（“&&”用于影响推断，“=”号导致收窄）。

Q: 可辨识联合要求对成员类型有怎样的限制

A: 在课程中说的是“可常量化的字段”，这有点口误。事实上“字段”指的是接口成员，而“可常量化”指的是成员类型必须是“常量或可计算的（*constant or computed*）”，这一要求与上一讲中的常值枚举表达式（Constant enum expression） 一致。事实上，所谓“可计算的（computed）”在 TypeScript 中是一个类型子集，包括字面的类型，还有对这些字面类型的引用（例如使用“T"xxx"]”语法访问或引用的接口、枚举等类型的成员类型），以及基于这些类型的表达式，例如“`1 + 2`”，或“`1 + AEnum.X`”。


极客时间
THANKS