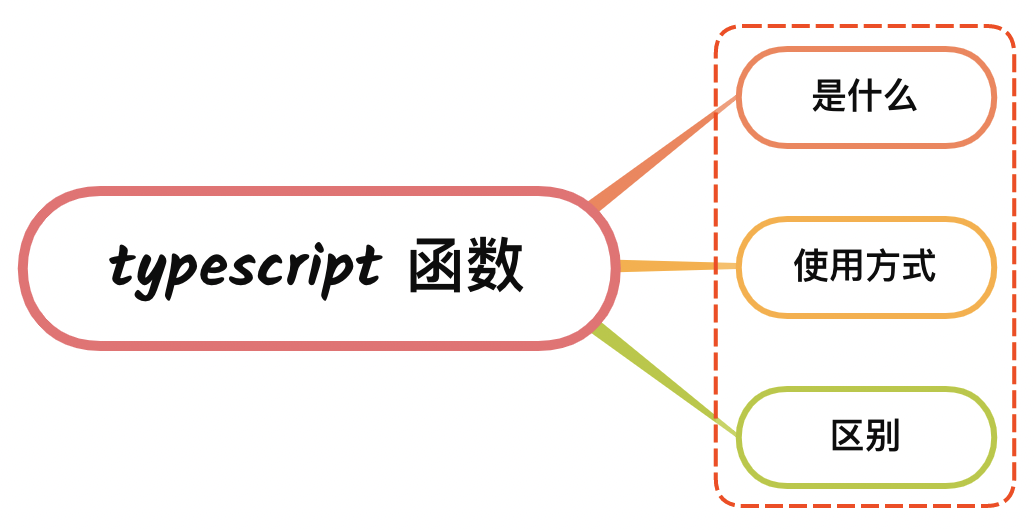
# 面试官：说说你对 TypeScript 中函数的理解？与 JavaScript 函数的区别？



## 一、是什么

函数是JavaScript 应用程序的基础，帮助我们实现抽象层、模拟类、信息隐藏和模块

在TypeScript 里，虽然已经支持类、命名空间和模块，但函数仍然是主要定义行为的方式，TypeScript 为 JavaScript 函数添加了额外的功能，丰富了更多的应用场景

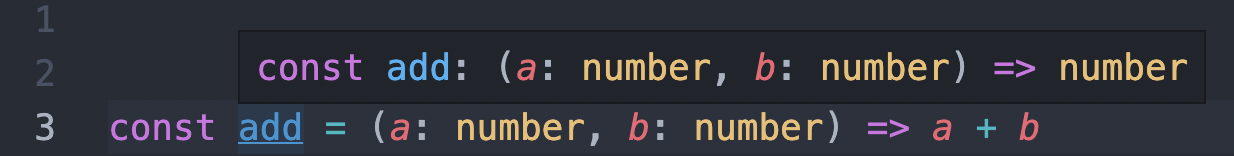
函数类型在 TypeScript 类型系统中扮演着非常重要的角色，它们是可组合系统的核心构建块

## 二、使用方式

跟javascript 定义函数十分相似，可以通过funciton 关键字、箭头函数等形式去定义，例如下面一个简单的加法函数：

const add = (a: number, b: number) => a + b

上述只定义了函数的两个参数类型，这个时候整个函数虽然没有被显式定义，但是实际上TypeScript 编译器是能够通过类型推断到这个函数的类型，如下图所示：



当鼠标放置在第三行add函数名的时候，会出现完整的函数定义类型，通过: 的形式来定于参数类型，通过 => 连接参数和返回值类型

当我们没有提供函数实现的情况下，有两种声明函数类型的方式，如下所示：

// 方式一  
type LongHand = {  
 (a: number): number;  
};  
  
// 方式二  
type ShortHand = (a: number) => number;

当存在函数重载时，只能使用方式一的形式

### 可选参数

当函数的参数可能是不存在的，只需要在参数后面加上 ? 代表参数可能不存在，如下：

const add = (a: number, b?: number) => a + (b ? b : 0)

这时候参数b可以是number类型或者undefined类型，即可以传一个number类型或者不传都可以

### 剩余类型

剩余参数与JavaScript的语法类似，需要用 ... 来表示剩余参数

如果剩余参数 rest 是一个由number类型组成的数组，则如下表示：

const add = (a: number, ...rest: number[]) => rest.reduce(((a, b) => a + b), a)

### 函数重载

允许创建数项名称相同但输入输出类型或个数不同的子程序，它可以简单地称为一个单独功能可以执行多项任务的能力

关于typescript函数重载，必须要把精确的定义放在前面，最后函数实现时，需要使用 |操作符或者?操作符，把所有可能的输入类型全部包含进去，用于具体实现

这里的函数重载也只是多个函数的声明，具体的逻辑还需要自己去写，typescript并不会真的将你的多个重名 function的函数体进行合并

例如我们有一个add函数，它可以接收 string类型的参数进行拼接，也可以接收 number 类型的参数进行相加，如下：

// 上边是声明  
function add (arg1: string, arg2: string): string  
function add (arg1: number, arg2: number): number  
// 因为我们在下边有具体函数的实现，所以这里并不需要添加 declare 关键字  
  
// 下边是实现  
function add (arg1: string | number, arg2: string | number) {  
 // 在实现上我们要注意严格判断两个参数的类型是否相等，而不能简单的写一个 arg1 + arg2  
 if (typeof arg1 === 'string' && typeof arg2 === 'string') {  
 return arg1 + arg2  
 } else if (typeof arg1 === 'number' && typeof arg2 === 'number') {  
 return arg1 + arg2  
 }  
}

## 三、区别

从上面可以看到：

* 从定义的方式而言，typescript 声明函数需要定义参数类型或者声明返回值类型
* typescript 在参数中，添加可选参数供使用者选择
* typescript 增添函数重载功能，使用者只需要通过查看函数声明的方式，即可知道函数传递的参数个数以及类型

## 参考文献

* https://www.tslang.cn/docs/handbook/functions.html
* https://zh.wikipedia.org/wiki/%E5%87%BD%E6%95%B0%E9%87%8D%E8%BD%BD
* https://jkchao.github.io/typescript-book-chinese/typings/functions.html#%E9%87%8D%E8%BD%BD