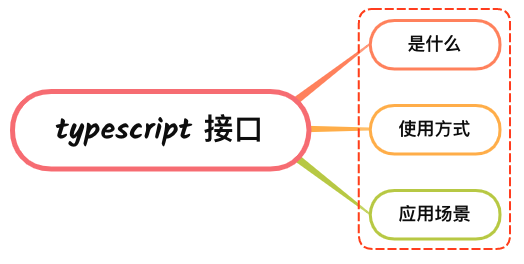
# 面试官：说说你对 TypeScript 中接口的理解？应用场景？



## 一、是什么

**接口**是一系列抽象方法的声明，是一些方法特征的集合，这些方法都应该是抽象的，需要由具体的**类**去实现，然后第三方就可以通过这组抽象方法调用，让具体的类执行具体的方法

简单来讲，一个接口所描述的是一个对象相关的属性和方法，但并不提供具体创建此对象实例的方法

typescript的核心功能之一就是对类型做检测，虽然这种检测方式是“鸭式辨型法”，而接口的作用就是为为这些类型命名和为你的代码或第三方代码定义一个约定

## 二、使用方式

接口定义如下：

interface interface\_name {  
}

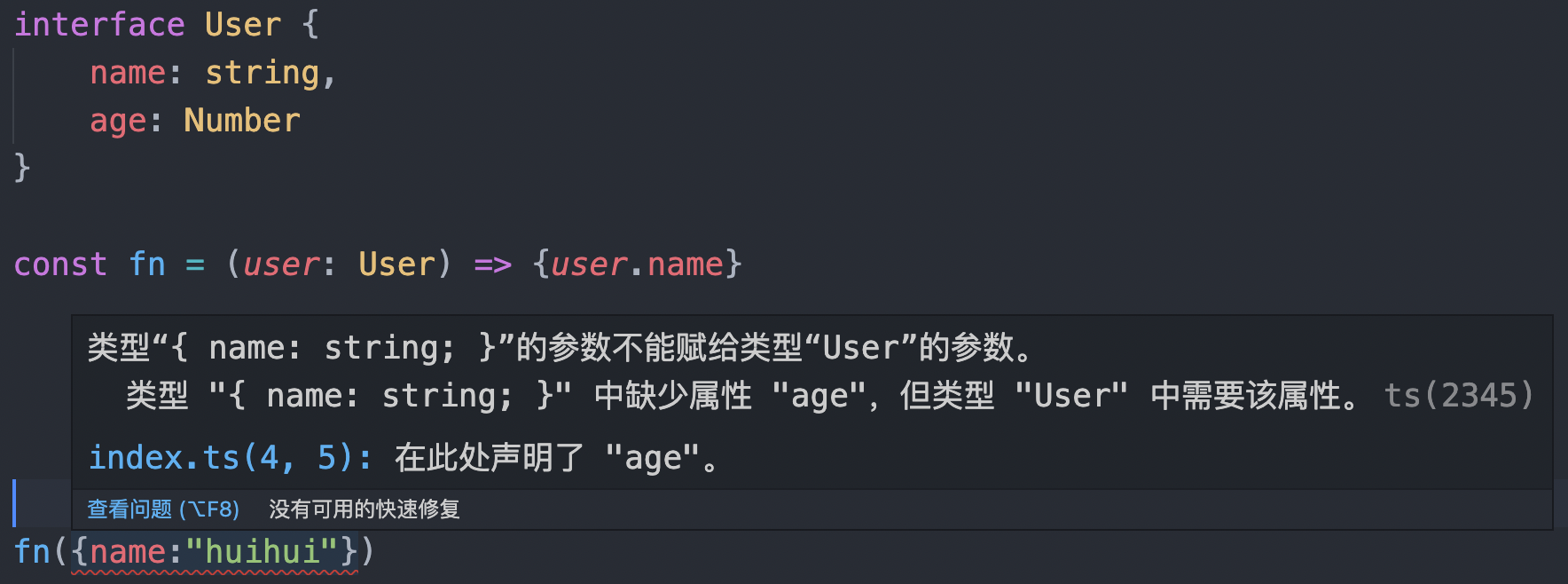
例如有一个函数，这个函数接受一个 User 对象，然后返回这个 User 对象的 name 属性:

const getUserName = (user) => user.name

可以看到，参数需要有一个user的name属性，可以通过接口描述user参数的结构

interface User {  
 name: string  
 age: number  
}  
  
const getUserName = (user: User) => user.name

这些属性并不一定全部实现，上述传入的对象必须拥有name和age属性，否则typescript在编译阶段会报错，如下图：



如果不想要age属性的话，这时候可以采用**可选属性**，如下表示：

interface User {  
 name: string  
 age?: number  
}

这时候age属性则可以是number类型或者undefined类型

有些时候，我们想要一个属性变成只读属性，在typescript只需要使用readonly声明，如下：

interface User {  
 name: string  
 age?: number  
 readonly isMale: boolean  
}

当我们修改属性的时候，就会出现警告，如下所示：



这是属性中有一个函数，可以如下表示：

interface User {  
 name: string  
 age?: number  
 readonly isMale: boolean  
 say: (words: string) => string  
}

如果传递的对象不仅仅是上述的属性，这时候可以使用：

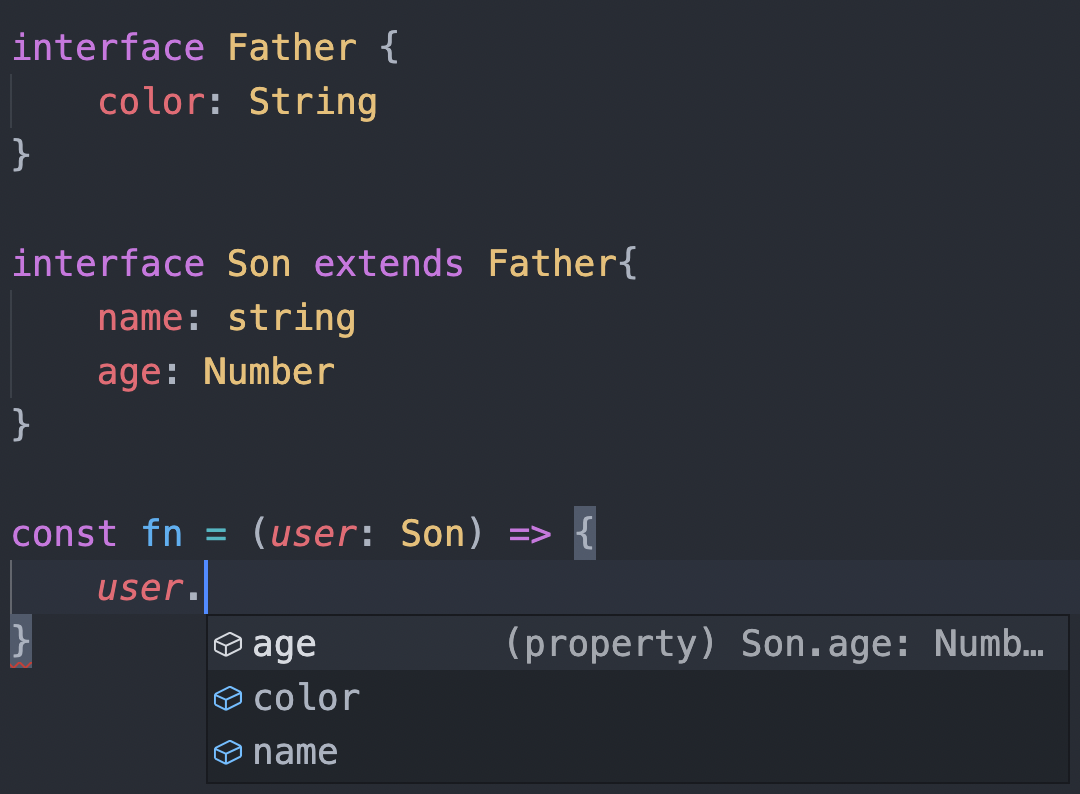
* 类型推断

interface User {  
 name: string  
 age: number  
}  
  
const getUserName = (user: User) => user.name  
getUserName({color: 'yellow'} as User)

* 给接口添加字符串**索引签名**

interface User {  
 name: string  
 age: number  
 [propName: string]: any;  
}

接口还能实现继承，如下图：



也可以继承多个，父类通过逗号隔开，如下：

interface Father {  
 color: String  
}  
  
interface Mother {  
 height: Number  
}  
  
interface Son extends Father,Mother{  
 name: string  
 age: Number  
}

## 三、应用场景

例如在javascript中定义一个函数，用来获取用户的姓名和年龄：

const getUserInfo = function(user) {  
 // ...  
 return name: ${user.name}, age: ${user.age}  
}

如果多人开发的都需要用到这个函数的时候，如果没有注释，则可能出现各种运行时的错误，这时候就可以使用接口定义参数变量：

// 先定义一个接口  
interface IUser {  
 name: string;  
 age: number;  
}  
  
const getUserInfo = (user: IUser): string => {  
 return `name: ${user.name}, age: ${user.age}`;  
};  
  
// 正确的调用  
getUserInfo({name: "koala", age: 18});

包括后面讲到类的时候也会应用到接口

## 参考文献

* https://www.tslang.cn/docs/handbook/interfaces.html