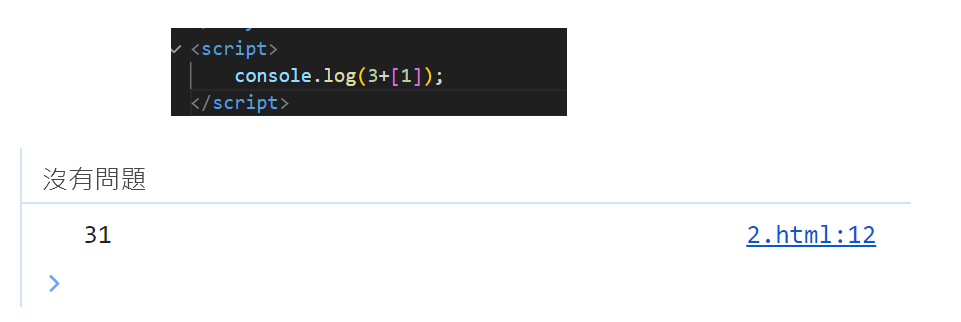
背景：

在学习Js之前，我在大学期间学过学校开设的Java课程，了解过一点点Java的基础知识。就入门而言，Js与Java是两个极端的语言，相较于Java，Js不需要在变量声明之初就指定它的类型，也不需要安装和配置各种JDK就可以直接在浏览器的控制台进行测试。而且相较于Java的Eclipse编辑器里的那种复杂且不易上手的调试程序而言，Js的调试就相对来说易上手很多。但是这就说明Js比Java简单吗？不不不，一点都不。Js这们语言无论是从广度和深度上现在都绝对不输Java。实际上它们是两种不同类型系统的语言而已，Java和C++需要显式的注解所有类型然后在编译时检查，而Js和Python则是在运行时推导类型；也就是说:Js内部或者说底层并不是没有类型，只不过是Js在内部帮你推导了而已。为了方便理解我举一个简单的例子：



在Js中两种不同类型的数据可以放在一个表达式中参与运算，而不需要提前做类型转换

像上图这种运算在Java中是根本不可能有结果的，但是在Js中，它可以运算得出“31”这个结果。为什么呢？我们来简单分析一下Js是如何得到这个结果的：

1. Js发现3是一个数字，而[1]是一个数组。
2. 中间你使用的+号运算符，Js假定你是想拼接两者
3. Js把3隐式转换为字符串“3”，把[1]隐式转换为“1”
4. 拼接两者得到字符串“31”

相信通过这个例子你应该能了解Js这门语言的类型系统的特性：

1. Js的类型是动态绑定的。
2. 在运算过程中若遇到不相容的类型，Js会帮你自动做类型转换，以完成你指定的计算。
3. Js只有在运行这个表达式的时候才会真正的检查表达式中的数据类型是否能够相容。
4. Js是在程序运行时报告错误，这与Java有非常大的区别，Java是在编译时直接不通过。

我们说这样对于一个刚刚学习编程语言的小白来说是友好的，因为在学习的初期小白是无法写出复杂的程序的。也无需记忆”相对来说比较繁琐的数据类型之间的运算规则”，就可以尝试写一些代码在浏览器的控制台中测试。总之来说，Js在尽可能的避免抛出异常。这对于调动初学程序的人的学习积极性来说是好事，但是这样做并不利于快速的捕获问题；且往往会导致程序计算出意想不到的结果，这样就会让程序做了无用功（是指程序虽未报错，但做的事却毫无意义，因为它没有得出你期待的结果）。

解决由Js动态类型系统造成的种种问题

我们写一个程序总是希望在用户发现错误之前，自己先发现错误。因为作为程序员来说谁也不想收到来自用户或者老板的指责，这有可能让你丢到饭碗！所以一个训练有素的程序员，在将自己所写的程序交给领导或用户使用之前，应当做足够的单元测试和端对端测试，甚至需要在内部试运行一段时间，以期在用户发现错误之前，自己先把错误找到并解决；以提升用户的体验。在以往,要做到让Js的程序尽可能少做无用功，需要一个Js程序员对Js的编译及执行机制有比较深的了解，同时需要Js程序员在程序的内部写一些代码来人为的控制如果参与运算的数据类型不是程序所期待的数据类型应该怎么处理？例如下图，我写了一个非常简单的求和函数，求和函数的功能当然是计算数值类型的和了。但是由于Js动态类型系统的特性，可能会导致程序计算出与你期待的结果不一致的情况，所以需要我们在函数内部处理一些特殊情况。



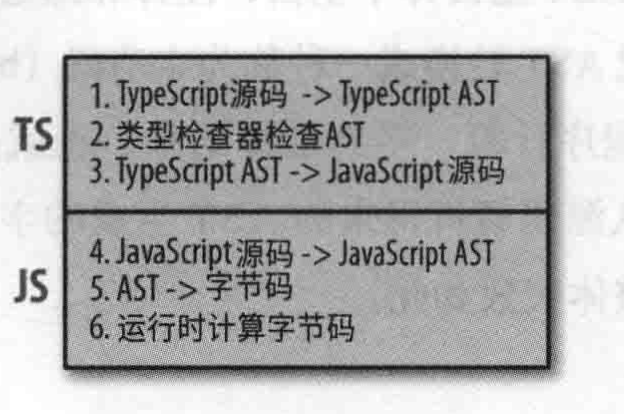
简单的演示在使用Js开发程序时应如何处理一些有关类型问题导致的意外结果。

想象一下，一个程序可能有数千个函数，这些函数之间还牵扯到相互调用传参以配合起来完成一项工作，如果每个函数或者所每段程序，都需要Js程序员手写处理这些特殊情况的代码，工作量无疑是巨大的。且要求Js的程序员要了解Js的编译器是如何编译并执行Js代码的，这是相当不容易的事！（例如chrome的V8引擎是C++写的，要深入了解Js的编译及执行机制又需要有一定的C++知识，这肯定不是一件轻松的事）自Js诞生以来，就是一门具有非常大的争议的语言，很多人对它是又爱又恨，爱就爱在在开发浏览器页面的时候它是一个非常好用的语言，但又由于它动态类型的特性，往往又很难用它写出功能强大且复杂的应用（其实可以写，但很费劲，也有一定的性能问题），所以许多人对它非常深恶痛绝。可是这丝毫不影响它被各个互联网巨头捧在手心当中，想一下你就知道了，谷歌为了提升Js这门语言的性能亲手为其打造了V8执行引擎，为了解决Js动态类型产生的问题，微软亲自操刀为其打造了静态类型支持系统“Typescript”，再比如facebook等大厂，参与维护Js这门语言的开发者更是不计其数！接下来介绍Ts是如何让Js具有静态类型系统的（像Java和C++那样），这让Js的程序运行的更加平稳高效。

Ts是如何实现让Js具有静态类型系统的

一般的强类型语言像Java都是在编译时检查类型，如果有问题直接编译不通过。而Typescript同样需要通过编译器，只不过是将Typescript语言编译成Js语言，最终再交由Js的编译器编译成AST(abstract syntax tree)再将AST转化为计算机可执行的字节码并由计算机执行。但是与Java不同的是，Typescript并不在将Ts编译成Js的时候检查类型，而是在将Typescript编译成Typescript AST的时候使用类型检查器检查AST。注意：这与Java的编译并检查类型的机制完全不同，因为这并不会阻止TS最终生成Js代码！说到这你可能就懵了？没关系，为了让你更加轻松的理解上面的内容，我先介绍一门编程是如何经过编译器编译最终交由计算机执行的。

1. 概括的说程序由一些文件构成，文件中就是程序员写的文本（代码）
2. 这些文本由一个特殊的程序（编译器）解析生成AST
3. AST是去掉了空白注释缩进制表符之后的一种数据结构（abstract syntax tree）
4. 最后编译器再将AST处理成计算机可以执行的字节码（bytecode）交由计算机执行



Ts转化为Js并最终交由计算机执行的流程图

我们来写一段非常简单的程序来说明上图：

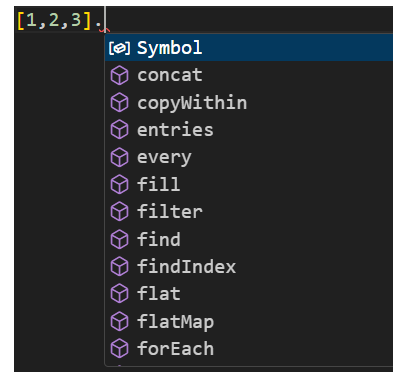


实例演示ts是如何编译成Js代码的

也就是说Typescript会对代码做静态分析，找出这类错误，在运行之前反馈给你。对的，你没看错，这其实已经完成了像Java那样的类型检查，Typescript所能做的就是把纯Js代码中那些运行时错误提前到编译时报告，但并不会阻止ts编译成js文件。对于用户在恶意输入这种情况，我想就算是是Java也没办法阻止吧！你能做的只能是在前端做好表单验证，或者在程序内部做好数据类型转换。所以Ts的作用就是帮助你完成更稳定Js代码的书写。

使用TS的另个好处：完善的代码提示

由于TS是微软开发的，Vscode也是微软开发的，所以Ts在Vscode中有非常完美的支持，在Vscode中书写Ts代码有非常好的代码提示，同时支持你自定义代码片段，让你快速高效的完成Js的代码构建。



在vscode中使用ts时的代码提示

这部分只是非常浅显得介绍了使用ts得好处，实际上ts的类型操作也比较复杂，但这些细节问题，我们在项目的实际代码中讨论。

本节资料

图书《typescript编程》----boris cherny著