

JavaScript 错误处理

课程目标

通过本节课程，我们将学到如下内容

- JS中的语法错误/运行时错误/以及其解决方案
- JS中的逻辑错误及debugger的使用
- JS中try catch finally语句的使用

语法错误

语法错误什么

简单来说，语法错误就是不符合JS语法的错误，这么说可能有点绕，来看几个具体的例子。

语法错误（续）

当出现语法错误的时候，控制台一般会报相应的错误并告知出错的行号（但行号不一定准确），常见的语法错误如下：

- 符号漏打，多打，少打，错打
- 使用了不合语法的变量名
- 语句写错，没写完等

运行时错误

运行时错误（Runtime Error），是指代码没有语法错误，而在运行的时候才发生的错误。运行时错误是一个统称

常见的运行时错误有如下几种：

-ReferenceError，变量引用异常触发

-TypeError，类型使用错误时触发

-RangeError，不太常见，一般会在递归爆栈时触发，即递归深度太深

逻辑错误

逻辑错误一般指的是计算结果不符合预期

比如说计算1到10的和，使用如下代码：

```
var sum = 0, i;  
for(i = 1; i<10; i++) {  
    sum += i  
}
```

结果是45而不是55

try catch finally语句

try catch语句

当一段代码有潜在的错误时，我们该如何处理这种情况呢？

- 使用try catch语句，捕捉程序在运行时可能出现的异常
- 它能保证处理运行时错误但不能保证处理语法错误

先看一个小例子

两个数字相乘的函数：

```
function multiple(a,b){  
    return a * b ;  
}
```

如果传入的参数不是数字，怎么办？

主动触发错误

对于前面的情况，可以主动触发一个错误

当发生一些不可避免的错误时，可以使用下面的语句来抛出这个错误

throw语句

发生不可避免的错误时怎么办？

某些运行时错误不可避免

使用try catch语句

课程总结

- 语法错误
- 运行时错误
- 逻辑错误
- 代码的调试：debugger，断点与单步跟踪
- try catch finally 语句