JavaScript 中的短路计算是什么?

作者：Vanessa
原文链接：https://ld246.com/article/1549350032554
来源网站：链滴

2019-02-05

回答

由于逻辑表达式的运算顺序是从左到右，并且他可以提前结束。使用这一规则便可进行短路计算。

```
true || false
```

以上使用逻辑或的示例中，由于表达式的计算结果为 true，JavaScript 并不会计算执行第二个操作 false。这就是短路计算。

这也适用于逻辑与 false && true

在逻辑运算过程中如果有一个表达式执行抛出异常，程序也不会受到影响，如：

```
true || nonexistentFunction()
false && nonexistentFunction()
```

由于从左到右的执行顺序，我们可以进行多个操作，如：

```
true || nonexistentFunction() || window.nothing.wouldThrowError
true || window.nothing.wouldThrowError
true
```

一个常用的场景是设置默认值。如果第一个对象是假值，将会使用第二个对象的值。如：

```
const options = {}
const setting = options.setting || "default"
console.log(setting) // default
```

另一个常用的场景是当对象为真值时，去执行该对象的相关操作，如：

```
addEventListener("click", e => {
  if (e.target.closest("button")) {
    handleButtonClick(e)
  }
})
// 以上代码片断等同于以下使用短路计算的代码片断
addEventListener(
  "click",
  e => e.target.closest("button") && handleButtonClick(e)
)
```

在上面的示例中，如果 e.target 不存在或他里面不包含和 “button” 相匹配的元素，后面的函数将会被执行。这是因为第一个对象如果是假值，那么第二个对象就不会被运行。

加分回答

- 逻辑操作不会产生布尔值，触发对象的值是一个布尔值。

返回总目录

每天 30 秒