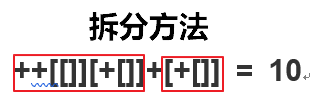
# [ValueOf()和toString()的区别](https://www.cnblogs.com/diantao/p/6214203.html)

# 为什么 ++[[]][+[]]+[+[]] = 10

这是一个[Javascript 语言题目](https://link.zhihu.com/?target=http%3A//justjavac.com/javascript/2012/04/05/you-dont-know-javascript.html)，一个完全有效的等式，不信自己可以试一下 在控制台输出console.log(++[[]][+[]]+[+[]]) 看看结果是否为10

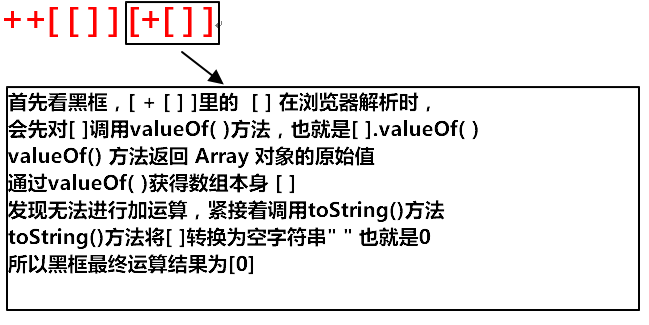
将这个表达式 拆分成如下

**++[ [ ] ] [+[ ] ]**

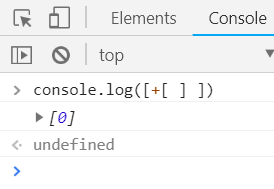
**+**

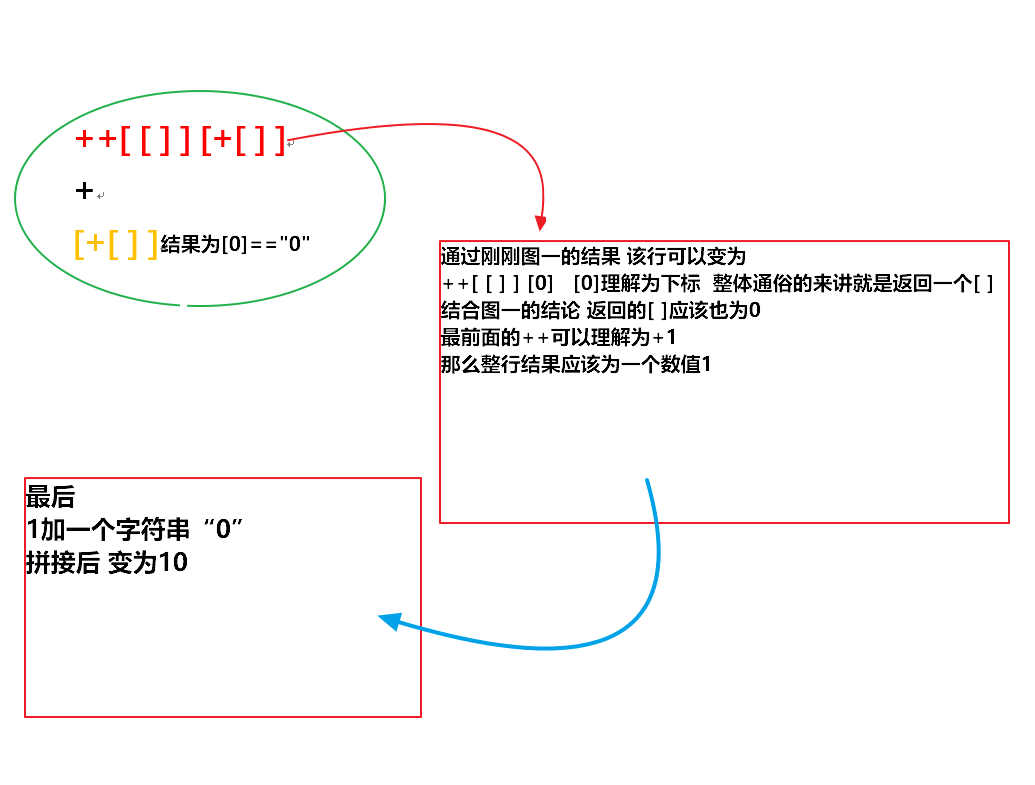
**[+[ ] ]**

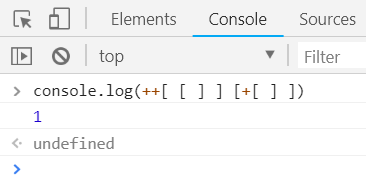
图1

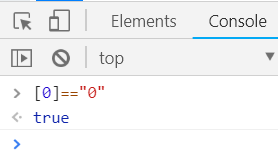


**看第一行**

****



****

****

这两个方法有意思的地方在于什么时候使用，总结如下：

1、valueOf()偏向于运算，toString()偏向于显示

2、对象转换时，优先调用toString()

3、强转字符串的情况下，优先调用toString()方法；强转数字的情况下优先调用valueOf()

4、正常情况下，优先调用toString()

5、在有运算操作符的情况下valueOf()的优先级高于toString()，这里需要注意的是当调用valueOf()方法无法运算后还是会再调用

**JavaScript的许多内置对象都重写了该函数，以实现更适合自身的功能需要。因此，不同类型对象的valueOf()方法的返回值和返回值类型均可能不同。toString()方法同理**

valueOf()方法：返回指定对象的原始值。

|  |  |
| --- | --- |
| **对象** | **返回值** |
| Array | 数组实例对象。 |
| Boolean | 布尔值。 |
| Date | 以毫秒数存储的时间值，从 UTC 1970 年 1 月 1 日午夜开始计算。 |
| Function | 函数本身。 |
| Number | 数字值。 |
| Object | 对象本身。这是默认设置。 |
| String | 字符串值。 |

toString()方法：返回对象的字符串表示。

|  |  |
| --- | --- |
| **对象** | **操作** |
| Array | 将 **Array** 的元素转换为字符串。结果字符串由逗号分隔，且连接起来。 |
| Boolean | 如果 Boolean 值是 **true**，则返回 “true”。否则，返回 “false”。 |
| Date | 返回日期的文字表示法。 |
| Error | 返回一个包含相关错误消息的字符串。 |
| Function | 返回如下格式的字符串，其中 functionname 是被调用 **toString** 方法函数的名称：  function functionname( ) { [native code] } |
| Number | 返回数字的文字表示。 |
| String | 返回 **String** 对象的值。 |
| 默认 | 返回 “[object objectname]”，其中 objectname 是对象类型的名称。 |

当函数fn用+连接一个字符串或者是数字的时候，如果我们没有重新定义valueOf和toString，其**隐式转换会调用默认的toString()**方法，将函数本身内容作为字符串返回；   
如果我们自己重新定义toString/valueOf方法，那么其转换会按照我们的定义来，其中**valueOf比toString优先级更高**

//当该对象为一个函数时

function fn(){

return 1

}

console.log(fn+10)

console.log(fn+'a')

//第一个输出结果

function fn(){

return 1

}10

//第二个输出结果

function fn(){

return 1

}a

//当该对象被定义后

fn.toString=function(){

return 1;

}

console.log(fn+10)

console.log(fn+'a')

//第一个输出结果

11

//第二个输出结果

1a

fn.valueOf=function(){

return 1;

}

console.log(fn+10)

console.log(fn+'a')

//第一个输出结果

11

//第二个输出结果

1a