Js学习笔记

1. About “this”

从看到这篇文章起，你一定要记住，第三种调用形式，才是正常调用形式：

func.call(context, p1, p2)

其他两种都是语法糖，可以等价地变为 call 形式：

func(p1, p2) 等价于

func.call(undefined, p1, p2)

obj.child.method(p1, p2) 等价于

obj.child.method.call(obj.child, p1, p2)

var obj = {

foo: function(){

console.log(this)

}

}

obj.foo()

=obj.foo.call(obj)

1. JavaScript 语言的标识符对大小写敏感，所以a和A是两个不同的标识符。
2. JavaScript 使用大括号，将多个相关的语句组合在一起，称为“区块”（block）。

对于var命令来说，JavaScript 的区块不构成单独的作用域（scope）。

1. switch语句内部采用的是“严格相等运算符”
2. 如果存在多重循环，不带参数的break语句和continue语句都只针对最内层循环。
3. 由于数值正向溢出（overflow）、负向溢出（underflow）和被0除，JavaScript 都不报错，而是返回Infinity，所以单纯的数学运算几乎没有可能抛出错误。
4. 对于那些会自动转为科学计数法的数字，parseInt会将科学计数法的表示方法视为字符串，因此导致一些奇怪的结果。
5. var f3 = function myName() {};

f3.name // 'myName'

上面代码中，f3.name返回函数表达式的名字。注意，真正的函数名还是f3，而myName这个名字只在函数体内部可用。

1. 注意，对于var命令来说，局部变量只能在函数内部声明，在其他区块中声明，一律都是全局变量。

if (true) {

var x = 5;

}

console.log(x); // 5

上面代码中，变量x在条件判断区块之中声明，结果就是一个全局变量，可以在区块之外读取。