█js数据类型

◎JS数据类型：

基本数据类型

undefined

null

number

string

boolean

复杂数据类型

object

◎对象的种类：

Js内置的（如Number）

宿主环境（如window）

自己创建的

█

对象的私有变量，以及通过get和set来访问和修改私有变量

<script>

        var p = {

            name: 'jojo',

            work: function () {

                console.log('working...')

            },

            \_age:18, //带有下划线开头的，是对象的私有变量，不能在外部直接访问（虽然可以，但不建议在外部访问）

            // 可以使用下面的get和set方法来从外面访问和修改私有变量\_age

            get age () {

                return this.\_age

            },

            set age (val) {

                if (val < 0 || val > 150) {

                    throw new Error('invalid value')

                } else {

                    this.\_age = val

                }

            }

        }

        // 访问私有变量\_age

        console.log(p.age) //18

        // 修改私有变量\_age

        p.age = 30

        console.log(p.age) //30

    </script>

█对象的访问

通常使用点和中括号来访问对象中的属性或方法，使用中括号来访问有如下两点特点：

第一

<script>

        var a = {

            1: 'haha'

        }

        // 如果用a.1来访问会报错，因为JS不建议用数字来做键名

        // console.log(a.1)

        // 使用[]来访问则可正常运行

        console.log(a[1]) //haha

    </script>

第二，使用中括号可以以变量的形式来访问

<script>

        var a = {

            b1: '11',

            b2: '22'

        }

        var b = 'b1'

        console.log(a[b]) //11

    </script>

█使用&&来访问不确定的属性

<script>

        var p = {

            np: {

                nnp: 'haha'

            }

        }

        // 当不确定p里面是否有某个属性时，可以用&&

        var r = p && p.np && p.np.nnp

        console.log(r) //haha

        var r2 = p && p.np && p.np.nnnp

        console.log(r2) //undefined

    </script>

█遍历对象

For in 来遍历对象是无序的

<script>

        var p = {

            a: 'aa',

            b: 'bb',

            c: 'cc'

        }

        for (x in p) {

            console.log(x + ':' + p[x]) //不能用p.因为x是变量所以要用中括号

        }

        // 输出：

        // a:aa

        // b:bb

        // c:cc

    </script>

如需有序遍历对象，可使用Object.keys方法

<script>

        var p = {

            a: 'aa',

            b: 'bb',

            c: 'cc'

        }

        // Object.keys方法会返回一个有序的数组

        var ap = Object.keys(p)

        console.log(ap) //["a", "b", "c"]

    </script>

█关于对象属性的一些操作

查看对象中是否有某个属性

删除属性

查看对象属性的特性

<script>

        var p = {

            a: 'aa',

            b: 'bb',

            c: 'cc'

        }

        console.log('a' in p) //true

        console.log(p.hasOwnProperty('a')) //true

        console.log(p.b) //bb

        delete p.b //删除P的b属性

        console.log(p.b) //undefined

        console.log(Object.getOwnPropertyDescriptor(p, 'c')) //{value: "cc", writable: true, enumerable: true, configurable: true}

    </script>