12.13-js08

 一、对象属性的操作：

1、定义属性：

1.1 使用直接量定义：属性名和属性值之间使用冒号间隔，多个属性之间可以用逗号隔开，一个对象中属性不能重复。

**var *obj1*** = {

**name**:**"laowang"**,

**sex**:**"男"**

};

**var *obj2*** = **new *Object***({

**name**:**"xiaowang"**,

**sex**:**"nv"**

})

**var *obj3*** = ***Object***.create(**null**,{

**name**:{

**value**:**"dawang"**,

**writable**:**true**

},

**sex**:{

**value**:**"nan"**

}

})

1.2 点语法定义：直接使用对象名+.+要添加的属性或要修改的属性名 = 属性值。

**var *obj4*** = {

**name**:**"xiaozhang"**

}

***obj4***.**sex** = **"nan"**;

***console***.log(***obj4***);

**var *obj5*** = **new *Object***({

**name**:**"xiaowang"**,

**sex**:**"nv"**

})

***obj5***.**age** = 20;

***console***.log(***obj5***);

**var *obj6*** = ***Object***.create(**null**,{

**name**:{

**value**:**"dazhang"**,

**writable**:**true**,

}

})

***obj6***.**age** = 10;*//也可以直接对 create创建的对象设置属性，但是无法选择属性的特性，默认可以被修改,可以被枚举*

***console***.log(***obj6***);

***obj6***.**age** = 20;

***console***.log(***obj6***);

**for**(**var *i* in *obj6***){

***console***.log(***i***);

}

1.3 中括号语法：使用对象名+[书写字符串类型的属性名(要使用引号)，或者是代表属性名的变量] = 属性值。

*// 03.中括号语法***var *obj7*** = {  
 **name**:**"xiaowang"**}  
**var *a***= **"sex"  
*obj7***[***a***] = **"nv"**;  
***obj7***[**"sex"**] = **"nv"**;  
***console***.log(***obj7***);  
  
  
**function** *getMess*(obj,pro) {  
 **return** obj[pro]  
}  
**var *myself*** = {**name**:**"huahua"**,**sex**:**"nan"**,**age**:**"19"**};  
***console***.log(*getMess*(***myself***, **"sex"**));

1.4 Object.defineProperty(obj,pro,{})，obj是被设置的对象，pro是要设置的属性名，{}是设置属性的特性（包含value……等），使用defineProperty函数可以为对象添加属性，或者是修改，也可以为属性添加特性。

*// 04.Object.defineProperty（obj,pro,{}）***var *obj8*** = {  
 **name**:**"xiaoli"**}  
***Object***.defineProperty(***obj8***,**"sex"**,{  
 **value**:**"nv"**,  
});  
***Object***.defineProperty(***obj8***,**"name"**,{  
 *//如果修改原有的name属性值，它可以被修改和枚举* **value**:**"dali"**,  
});  
***console***.log(***obj8***);  
**for**(**var *i* in *obj8***){  
 ***console***.log(***i***);  
}

1.5 Object.defineProperties(object,description)，可以一次性定义多个属性，object是被设置的对象，description就是要设置的属性的一个集合。

*// 05.Object.defineProperties(object,description)***var *obj9*** = {  
 **like**:**"miantiao"**}  
***Object***.defineProperties(***obj9***,{  
 **color**:{  
 **value**:**"yellow"**,  
 **enumerable**:**true** },  
 **length**:{  
 **value**:**"10m"**,  
 }  
})  
***console***.log(***obj9***);

2、读取属性：

2.1 使用点语法：点语法可以快速的读取属性，点的左侧是要读取的对象变量，右侧是属性名。

2.2 中括号语法：--结构上分析，对象被称作关联数组，所以也可以通过中括号来获取属性值。

--中括号的语法中，里面书写变量，或是必须引号包围的属性名。

*// 遍历以下对象，然后在每一个属性值后添加一个 @ 符号***var *obj1*** = {  
 **name**:**"xiaowang"**,  
 **like**:**"唱跳rap篮球"**,  
 **time**:**"两年半"**}  
  
**for**(**var *item* in *obj1***){  
 ***console***.log(***item***);  
 ***obj1***[***item***] = ***obj1***[***item***] + **"@"**;  
}  
***console***.log(***obj1***);

2.3 Object.getOwnPropertyNames()函数能够返回对象的私有属性名称，私有属性是用户给当前对 象定义的属性，而不是继承的属性。返回的是一个数组（可以用这个数组来获取对象的length），无论属性的特性是什么样子，都可以被 获取。

***console***.log(***Object***.getOwnPropertyNames(***obj1***));*//["name","like","time"]*

*//对 create方法创建的对象 进行属性设置***var *obj2*** = ***Object***.create(**null**,{  
 **name**:{  
 **value**:**"新宝岛"**,  
 **enumerable**:**true** },  
 **time**:{  
 **value**:**"5min"**,  
 **enumerable**:**true** },  
 **like**:{  
 **value**:**"rap"** }  
})  
***console***.log(***Object***.getOwnPropertyNames(***obj2***));*//["name","like","time"]****console***.log(***Object***.keys(***obj2***));*//["name","time"]*

2.4 Object.keys(）函数仅能获取可枚举的私有属性，返回的是一个数组。

***console***.log(***Object***.getOwnPropertyDescriptor(***obj1***, **"like"**));*//{value: "唱跳rap篮球@", writable: true, enumerable: true, configurable: true}*

2.5 Object.getOwnPropertyDescriptor()方法获取对象属性的描述符。

***console***.log(***Object***.getOwnPropertyDescriptor(***obj2***, **"time"**));*//{value: "5min", writable: false, enumerable: true, configurable: false}*

 **3、删除属性：**

3.1 delete运算符可以 删除属性

**var *obj*** = {  
 **name**:**"laowang"**,  
 **sex**:**"nan"**,  
 **like**:**undefined**}  
***console***.log(***obj***.**like**);*//可以设置一个属性值为undefined****console***.log(***Object***.getOwnPropertyNames(***obj***));*//设置为undefined值的属性名 仍然可以获取到***delete *obj***.**sex**;*//删除一个属性****console***.log(***obj***);  
***console***.log(***Object***.getOwnPropertyNames(***obj***));*//当一个属性被删除的话，就枚举不到他的这个属性名了。*

3.2 当属性被删除以后，这个属性的值并不是变成undefined了，而是直接从对象中移出。

二、控制对象的状态

1、Object.preventExtensions()：阻止扩展新属性

2、Object.seal()：（封装,封存）阻止扩展且不能删除旧属性。

3、Object.freeze()：阻止扩展且不能删除旧属性也不允许修改属性值。

4、3个辅助检查函数：

4.1 Object.isExtensible()：检测对象是否可以被添加新属性。

4.2 Object.isSealed()：检测对象是否使用了seal方法。

4.3 Object.isFrozen()：检测对象是否使用了freeze方法。

三、引用类型值和基本类型值

1、引用类型值

1.1 定义：在栈区存储的是变量的标识符和对象的引用地址，在堆区储存的是对象本身。当访问一 个对象的时候，要先访问到栈区的地址，然后通过地址去找到对象。

 1.2 特点：

A、引用类型值是可变的，我们可以给引用类型值设置属性和方法。

B、引用类型值是同时保存在堆区和栈区的。

C、引用类型值的比较是地址的比较。

2、基本类型值

2.1 分类：null、undefined、number、Boolean、string

 2.2 特点：

A、基本类型值不可变（除非赋值后直接替换新值）。

B、不能给基本类型值设置自定义的方法和属性。

C、基本类型值的比较是值和值的比较。

D、基本类型值的变量存储在栈区，栈区包括变量的标识符和变量的值。

|  |  |  |
| --- | --- | --- |
| 特点 | 基本类型值 | 引用类型值 |
| 1、值是否可变？ | 否（除非赋值后直接替换新值） | 是 |
| 2、是否可以给值设置自定义的属性及方法？ | 否 | 是 |
| 3、存储位置？ | 栈（包括变量的标识符及值） | 同时保存在堆与栈中 |
| 4、比较方式？ | 值与值的比较 | 地址与地址的比较 |

四、数组

1、数组是一个有序的数据集合，数组中每一个成员都被称作元素，每个元素的键名被称作下标 （index），数组中不同的元素的值可以是不同类型的。数组的长度也是弹性的，可以续写。

 **2、定义数组：**

2.1 构造数组：直接使用new Array()函数创建；如果不传值，则构造一个空数组；如果穿多个值， 可以创建一个实数组；如果传递一个参数，是定义数组的长度，每个值都是undefined，如果传 的不是数字类型的，则会生成一个长度为1的数组。

2.2 数组直接量：直接在中括号中书写多个值的列表，值与值之间使用逗号隔开。（推荐使用）

2.3 多维数组：JS不支持多维数组，但是可以通过将元素设置成数组（嵌套）来模拟多维数组。

2.4 空位数组：数组中包含空元素（即两逗号间没有任何值）；空位的长度也是可以被获取的，并 且for循环等遍历操作也是可以读取到出现空位数组情况。注意空位数组的空位返回undefined。

2.5 关联数组：如果数组的下标超出范围，如负数、浮点数、布尔值、对象、字符串等，JS会把这 些值定义为字符串，并生成一个关联数组。定义的字符串的下标就是关联数组对象的属性，如果 是数字则需要使用[]获取。

2.6 伪类数组：也称为伪数组或类数组，即类似数据结构的对象。下标从0开始依次递增，同时包 含了length属性，并且length属性还要动态的改变。因为这个对象键名都是数字，数字不是合 法的标识符，所以需要同中括号来获取。

3、读取数组：

3.1 使用中括号可以读取数组，中括号左边是数组名称，中括号中是数组的下标。

3.2 练习：将n<100的斐波那契数列放在数组中。

*/\*  
\* 练习：斐波那契数列  
\* 前两个值是1,2 后边每一个值都是前边两个值的和  
\* \*/* **var *arr*** = [1,2];  
 **for** (**var *i*** = 2; ***i*** < 100; ***i***++) {  
 ***arr***[***i***] = ***arr***[***i*** - 1] + ***arr***[***i*** - 2];  
 }  
 ***console***.log(***arr***);

**4、数组的长度：**每个数组都有自己的length属性，该属性返回数组的长度。length是可读可写的 属性，是一个动态的属性，是可以随着数组的长度变化的；同样设置了长度以后也会发生改变。

 **五、栈**

1、push()方法，能把一个或多个参数附加到数组的尾部，并返回附加元素之后数组的长度（所以用 for循环的时候要注意数组长度可能已经被改变了）。

2、pop()方法，能删除数组中的最后一个元素，并返回删除的值（= 拎出末尾那个元素）。

3、栈操作练习-进制转换。

**function** *to2b*(num) {  
 **var** arr = [];*//一个新栈，把每次生成的余数压到栈里* **var** r = **null**;*//用来保存每次生成的余数* **var** s = **""**;*//最后返回的二进制字符串* **while**(num > 0){  
 r = num % 2;*//求出每次的余数* arr.push(r);*//把余数入栈* num = *parseInt*(num / 2);*//求出num计算后的值 并向下取整* }  
 **while**(arr.**length** > 0){  
 s += arr.pop();*//pop是把最后一个值删了 并返回出来 和s拼接 放在了s的首位* }  
 **return** s;  
}  
***console***.log(*to2b*(100));

六、晨测

1、引用类型值的特点和基本类型值的特点

2、如何定义数组

3、数组的栈操作是什么，并说明返回值

4、对象属性的操作

5、删除对象的属性