JavaScript面向对象思想

2019年8月23日

10:16

一、面向对象和面向过程思想的理解

* 案例：把大象装进冰箱
  + 面向过程
    - 开门（冰箱）、装进（冰箱、大象）、关门（冰箱）
  + 面向对象
    - 冰箱.开门、冰箱.装进（大象）、冰箱.关门

二、类和对象

1、JavaScript是基于对象的弱语言

* Javasricpt只有对象没有类，其面向对象的思想是基于对象实现的

2、什么是类

* 类是具有相同属性和方法的一组对象的集合

3、什么是对象

* 对象具有确定属性和方法的一个个体

案例：变量和属性

var a=10;//变量，是自由的，不属于谁

alert(a);

var arr=newArray(1,2,3);

arr.a=20;//属性，不自由，有归属

alert(arr.a);

* 5、方法
* 方法：
* 案例：理解函数和方法
* function fun(){//函数,自由的，不属于谁
* alert("hello")
* }
* arr.fun = function(){//方法，是某个对象的行为
* alert("你好");
* }
* 三、this 的本质
  + this:当前方法属于谁，this就代表谁。

var arr=[1,2,4];

arr.a=12;//给arr添加一个车属性

arr.show=function(){//给arr添加一个方法

alert(this.a);//这里arr指arr

};

* 全局函数属于window

四、第一个面向对象的程序

1、Object

* 之前已经接触过很多的对象
  + 比如说，Date、Array，每个对象都有自己特定的功能
* Object是一个空对象，没有功能，所以我们在实现面向对象的编程时，多是使用Object；

代码实现：

var obj=newObject();

obj.name="王浩";//属性

obj.age=18;//属性

obj.showName=function(){

alert("我的名字叫"+this.name)

}

obj.showAge=function(){

alert("我的年龄是："+this.age)

}

obj.showAge();

obj.showName();

2、利用Object创建对象的弊端

* 一个程序中有很多的同类对象的话，如果我们每用一个对象都new，再添加属性和方法，代码会很多，根本没法写。

五、利用构造函数来创建对象。

* 构造函数其实就是一个普通函数，只不过它是用来创建对象的
* 代码得到了大幅度的简化，创建对象时调用这个函数即可

function createUser(name,age){//写一个方法，专门用来创建对象

var obj=new Object();

obj.name=name;//属性

obj.age=age;

obj.showName=function(){//方法

alert("我的名字叫："+this.name)

}

obj.showAge=function(){

alert("我的年龄是："+this.age)

}

return obj;

}

var obj1=createUser("王浩",18);

obj1.showName();

var obj2=createUser("航空",17);

obj2.showAge();

六、工厂模式

* 什么是工厂
  + 批量生产某种东西的地方
  + 生成一个东西的步骤
    - 原料
    - 加工
    - 出厂

function createUser(name,age){//写一个方法，专门用来创建对象

//创建一个空白的对象就是原料

var obj=new Object();

//添加属性和方法就是加工

obj.name=name;//属性

obj.age=age;

obj.showName=function(){//方法

alert("我的名字叫："+this.name)

}

obj.showAge=function(){

alert("我的年龄是："+this.age)

}

//返回出去加工号的对象就是出厂

return obj;

}

* 我们用这种构造函数创建对象的过程类似于工厂的生产过程，所以称之为工厂模式。

利用this改造代码

function createUser(name,age){//写一个方法，专门用来创建对象

//创建一个空白的对象就是原料

//系统会偷偷地帮我们创建一个叫this的空对象

//var this = newObject;

//var obj=newObject();

//添加属性和方法就是加工

this.name=name;//属性

this.age=age;

this.showName=function(){//方法

alert("我的名字叫："+this.name)

}

this.showAge=function(){

alert("我的年龄是："+this.age)

}

//返回出去加工号的对象就是出厂

return obj;

}

七、原型

**js中函数加括号和不加括号的区别**

加括号：执行函数

不加括号：返回函数本身

**什么是原型**

css中的class可以被多个元素引用，用于一次给多个元素添加样式

class在js中被称之为原型

**案例：利用原型求数组和**

function CreateUser(name,age){

this.name=name;

this.age=age;

return this;

}

CreateUser.prototype.showName=function(){

alert("我的名字叫"+this.name);

};

CreateUser.prototype.showAge=function(){

alert("我的名字叫"+this.age);

};

var obj1=CreateUser("王浩",23);

var obj2=CreateUser("账号",234);

alert(obj1.showName==obj2.showName)

alert(obj1.showName())

**混合式创建对象的方式**

构造函数中写属性

用原型添加方法

function CreateUser(name,age){

this.name=name;

this.age=age;

return this;

}

CreateUser.prototype.showName=function(){

alert("我的名字叫"+this.name);

};

CreateUser.prototype.showAge=function(){

alert("我的名字叫"+this.age);

};

var obj1=CreateUser("王浩",23);

var obj2=CreateUser("账号",234);

alert(obj1.showName==obj2.showName)

alert(obj1.showName())

**通过一个对象获取新对象: create()**

例子: var people2 = Object.create(people1)