**JS面向对象及组件开发**

第一章 面向对象基础

一、什么是面向对象

1、面向对象的编程

面向对象的编程就是使用面向对象的思想去写代码。以往我们学习的代码编写方法基本上都是过程式写法，但是我们一直在使用对象，如date 、array等，系统自带的那些对象是属于系统对象，我们自己定义的那些对象的属于自定义对象。

2、面向对象的编程（oop）的特点

抽象：抓住核心问题

封装：只通过对象来访问方法

继承：从已有的对象中继承出新的对象；（继承的关键问题就是如何复用代码）

多态：多对象的不同形态；

3、对象的组成

对象有属性和方法两部分组成。

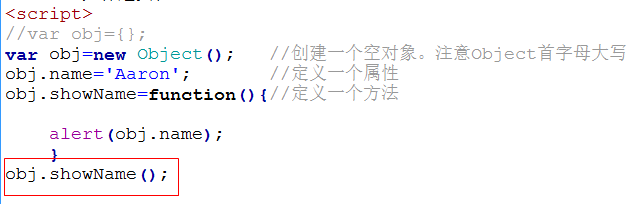
**方法**（行为、操作）——函数：指的是某些过程，是动态的；

**属性**——变量：指的是状态，是静态的；

对象下面的变量，叫做对象的属性。

对象下面的函数，叫做对象的方法。

有一种判断方法就是：如果后面加了括号的话，那就是方法，没加括号就是属性。就像数组的属性length和数组的方法sort（）一样。（在new Object的时候要注意O是大写的）。



面向对象中**this的指向**是很重要的：如上图，在调用obj的showName（）方法的时候，如果前面有obj这个对象，那么this就是指向的是这个对象，如果没有obj这个对象的话就是指的是window。

4、工厂方式与构造函数

（1）工厂方式——封装函数

实际上，所谓工厂方式就是我们通常所说的“封装函数”。

当new去调用一个函数：这个函数中的this就指的是创建出来的对象，并且函数的返回值直接就是this（这个创建出来的对象）（这叫做隐式返回）。

（2）**New后面的函数叫做构造函数**。

（3）对象的引用

基本类型：赋值的时候只是进行值的复制；

引用类型：赋值不仅是值得复制，而且还是引用的传递；

在程序中，只要是涉及到赋值，那么就一定会在内存中生成一个新的地址来存放这个赋值的内容。如果这个被赋值的变量是与其他变量同名的，那么它在之前的那些操作就会被全部清除，只留下现在这个赋值的内容。

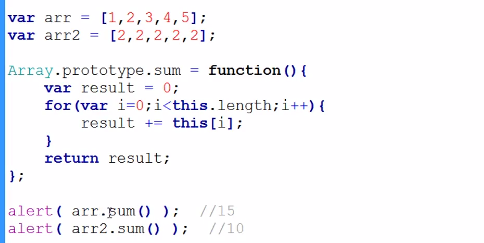
基本类型的比较只要值相等就行，但是在引用类型的中，比较不仅需要值相等，还需要引用也相等。

5、原型-prototype

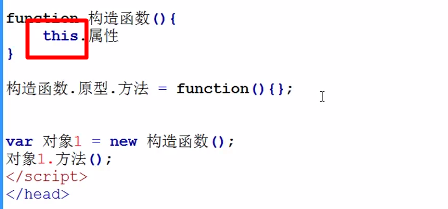
改写对象下面公用的方法或者属性，让这些公用的方法或者属性在内存中存在一份。

好处：可以提高性能。

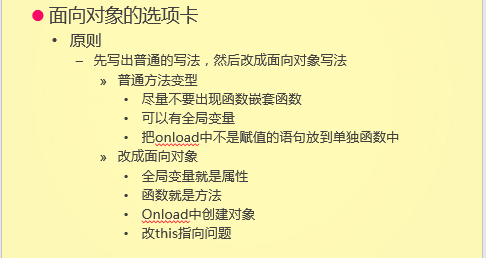
（1）prototype要写在构造函数下面。



面向对象的写法：



实例：选项卡



实际上，面向对象适合用于复杂的开发，对于一些比较简单的操作是没必要使用面向对象的方法来开发的。

**要点**：

**变形**的原则：

1、尽量不出现函数嵌套函数，将原先嵌套的函数，从内向外一层层剥离；

2、可以有全局变量；

3、把onload中不是赋值的语句放到单独函数中；

改成**面向对象**原则：

1、全局变量就是属性；

2、函数就是方法；

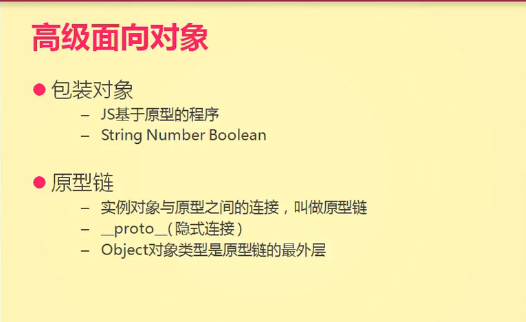
3、onload中创建对象；

4、更改this的指向；

5、event对象只能出现在事件函数中，如果不行那就是用参数的方式迂回。

**注意**：event对象只能出现在事件函数中，return false也一样。因此，在用的时候如果出现了event对象不再事件函数中的情况要注意解决。

第二章 高级面向对象



一、包装对象

1、在js源码中，系统对象本身就是基于原型的程序。因此，我们在使用的时候尽量不要去添加或者是修改系统对象下的方法和属性。

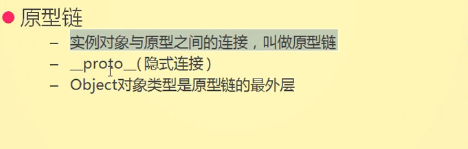
2、包装对象

基本类型都有自己对应的包装对象，包装对象的类型有：String、Number以及Boolean，undefined和null没有对应的包装对象。包装对象就类似于那些构造函数。

（1）对于基本类型在调用方法的时候，基本类型会找到对应的包装对象类型，然后包装对象把所有的属性和方法全部给基本类型，然后包装对象消失。

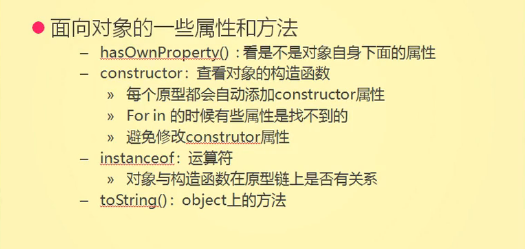
3、原型链

实例对象与原型之间的连接就叫做原型链。



原型链的最外层是object，在实体调用方法的时候会从内向外查找需要的信息，如果在内层没有查找到需要的信息，那么解析器就会沿着原型链向上一层查找，直到最外层为止。

4、

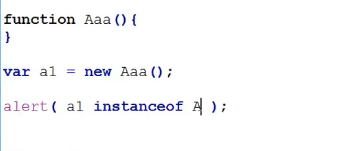


（1）hasOwnProperty（）如果是只属于当前元素自身的属性，那就返回true，否则如果是某一类元素都有那个属性的话就返回false。

（2）constructor：

查看对应的构造函数。

（3）instanceof：运算符——对象与构造函数在原型链上是否有关系。如果有就返回true，否则返回false。



判断数组：arr instanceof Array进行判断。还可以使用toString判断是否是数组。

不是面向对象的属性，但是和面向对象关系紧密。

（4）toString（）

系统对象下面的都是自带的，自己写的对象都是通过原型链找object下面的。

**作用**

（1）把对象转成字符串（ arr.toString（） ），内容还是和数组一样，也可以自己更改转换规则；

（2）进制转换：toString（进制），就是将一个数字类型的对象进行进制转换。可以用来颜色值转换。

（3）利用toString（）进行类型判断。

