Windows对象

事件对象

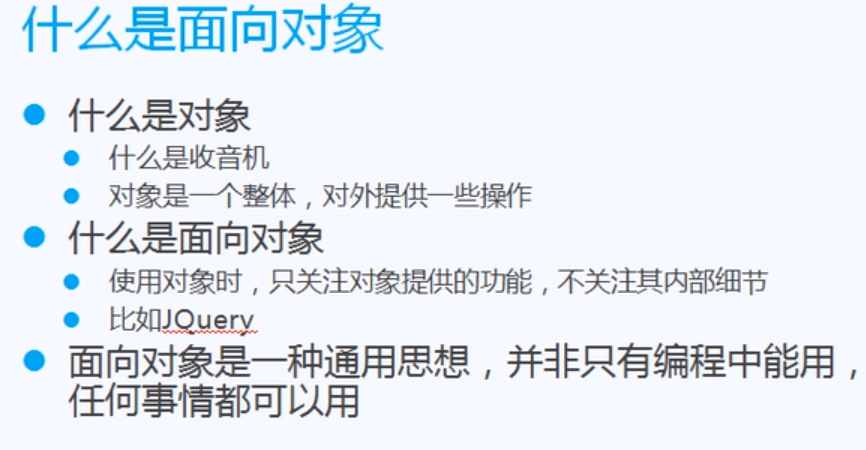
内置对象

浏览器对象

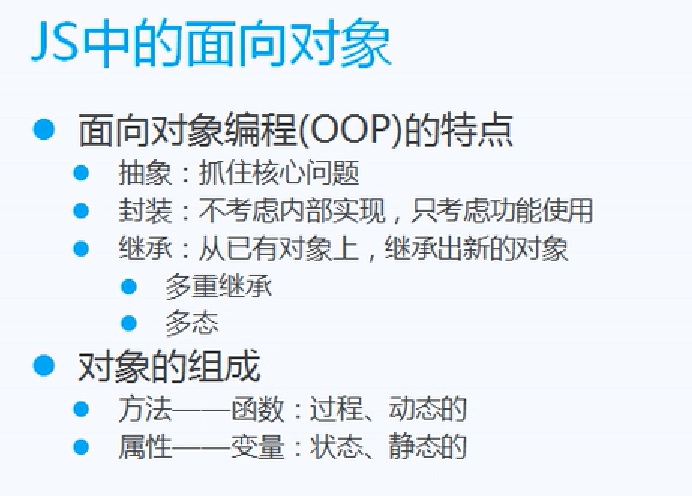
History对象

Location对象

Screen对象



关于变量和属性、函数和方法的关系



两者之间最大的区别是，一边是自由的，谁也不属于谁，另一边是附加在别人身上的，无法单独使用

This的定义：this就是当前方法的主人，这个方法是谁身上的，这个this就是谁

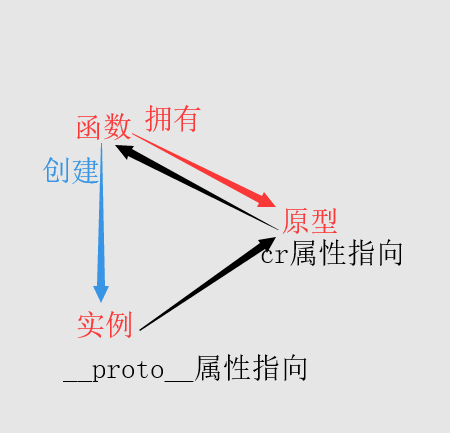
关于构造函数

它和其他函数无差别，只是他的功能不同，它用来创建并加工打造一个对象，如同工厂中的，原料，加工，出厂一般，这也正是所说的工厂模式

工厂模式的缺陷：每创造一个对象，他都会生产一种不同的方法，从而浪费系统资源

问题可以通过prototype解决(通过它可以为同一种对象批量添加方法) 

追加：prototype它是每一个函数对象都有的一个原型，原型之中有一个默认的constructor属性，想当于一个指针，指向于它对应的函数，从这个函数中创建出的实例，也有\_\_proto\_\_属性指向对应的原型



关于类与对象

例子：小熊蛋糕

类就是模具，对象就是小熊蛋糕（成品）

拿代码来说：

 arr就是成品，Array就是类、



非静态对象是需要new出来用的

比如：var a = new Object();

静态对象是不需要且不能new，通过直接调用上面的方法来使用

比如 Math.floor()