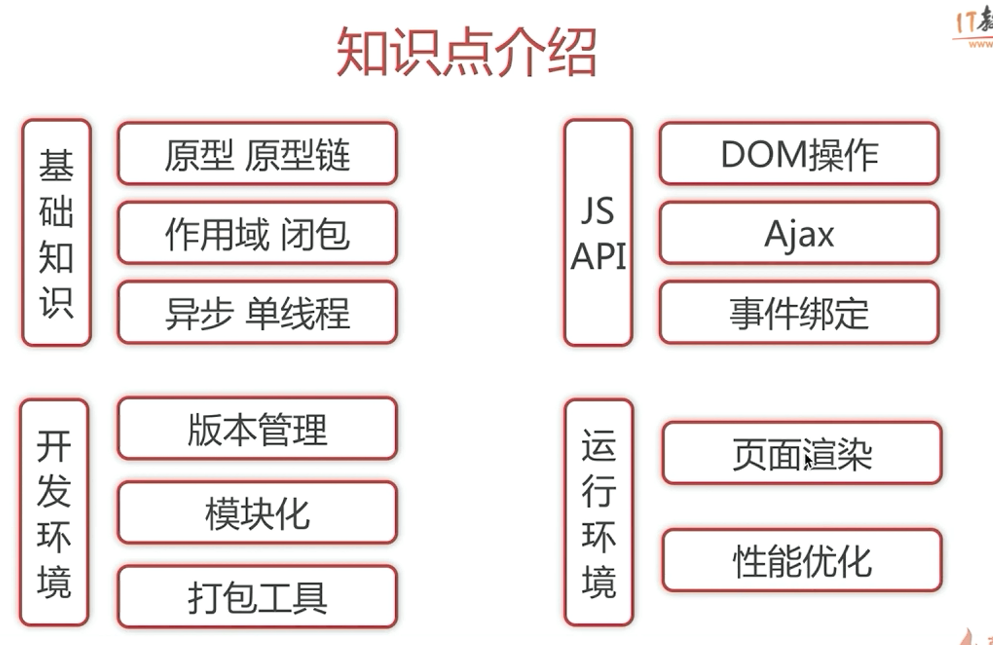
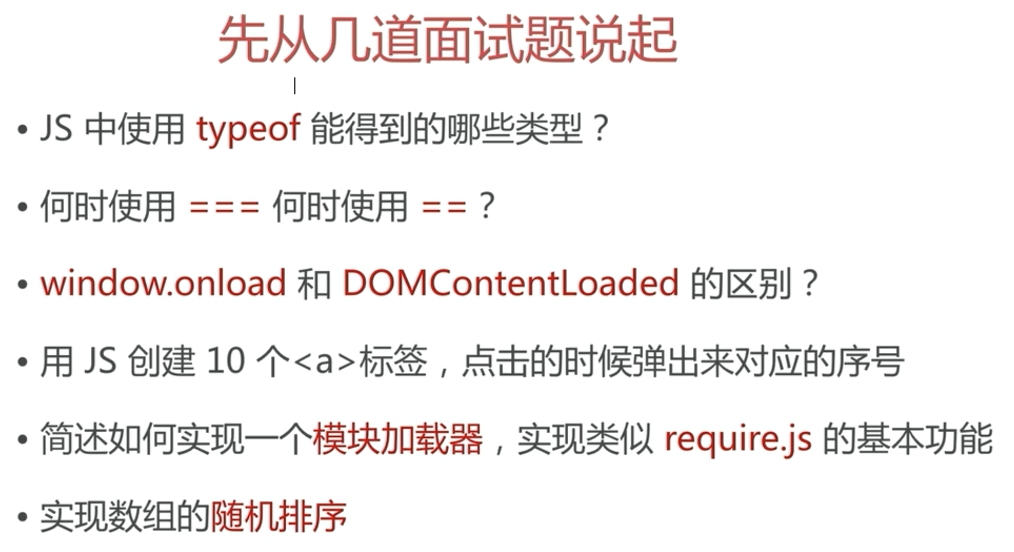
第一章：课程简介



# **第二章：JS基础知识（上）**

## 1.typeof返回的值？

Number，string，boolean，object，undefined，function。

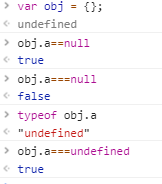
值类型可以区分，引用类型中的数组，对象（包括null），函数只能区分函数，其他都是object。



## 2.何时用===和==？



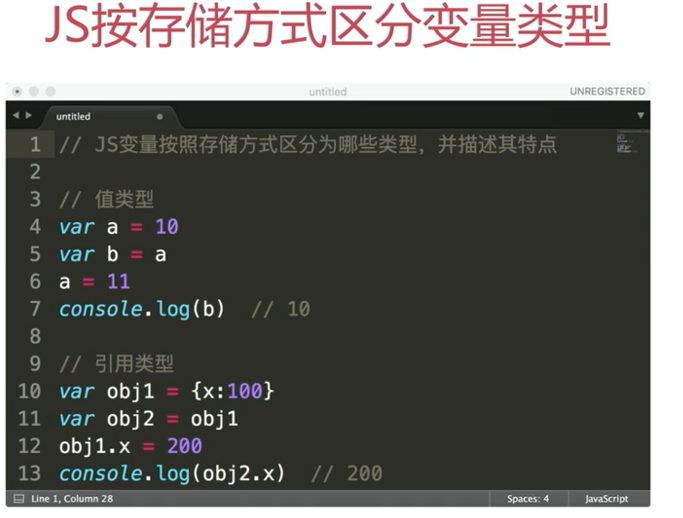
看这个对象的属性 是否存在，或者看函数的参数是否存在。除此之外，全部都用===。



## 3.JS的内置函数



## 4.JS按存储方式区分变量类型



值类型每个都开一个内存块，引用类型共用一个内存，节省内存空间。

## 5.如何理解json？

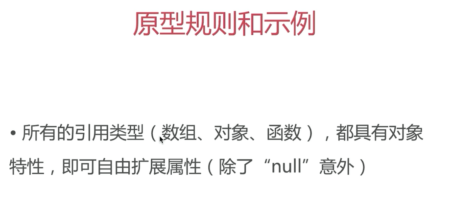


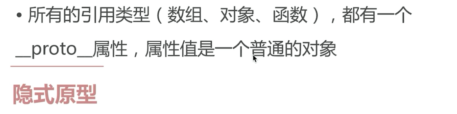
是一个js对象，有两个API（方法）；也是一种数据格式。

6.Falsy值（6种）：0，null，undefined，’’,NaN,false

## 视频2-5：原型和原型链

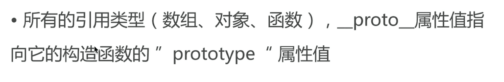




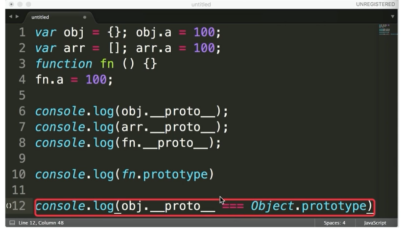


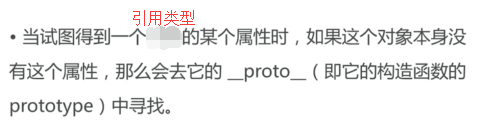


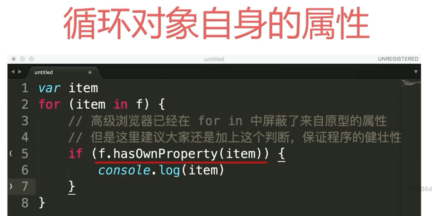
普通对象的隐式原型 就是Object了。



指向 就是 完全等于的意思。 ===

 //true

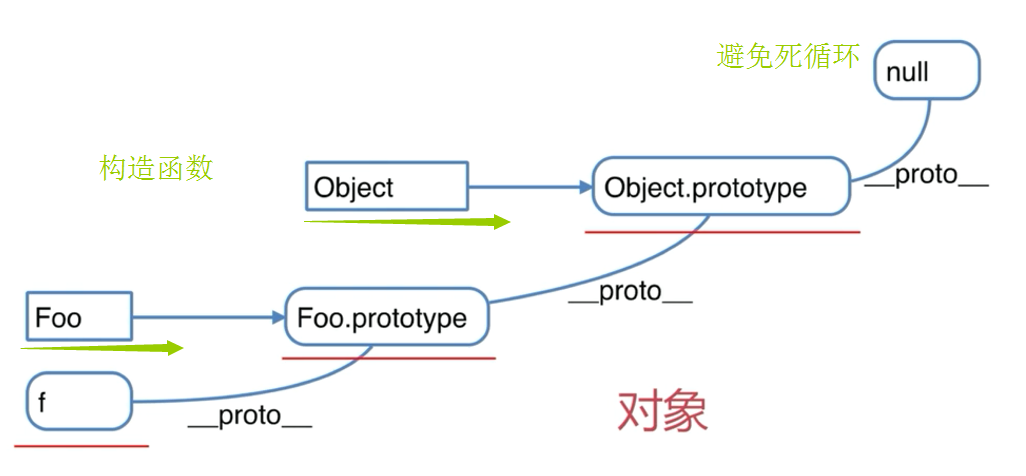




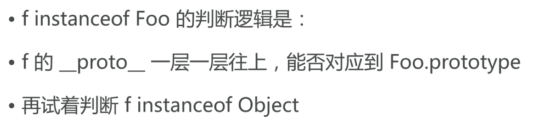


F中没有toString()方法，所以去\_\_proto\_\_（隐式原型）中去找，隐式原型即构造函数的显示原型prototype，即Foo.prototype，而Foo.prototype对象中也没有toString(),并且Foo.prototype是一个普通对象，再继续找这个普通对象的\_\_proto\_\_（隐式原型）,即Object的prototype。

也就是f.\_\_proto\_\_.\_\_proto\_\_。







## 6.如何准确判断一个变量类型是数组类型？ instanceOf Array、



## 7.写一个原型链继承的例子



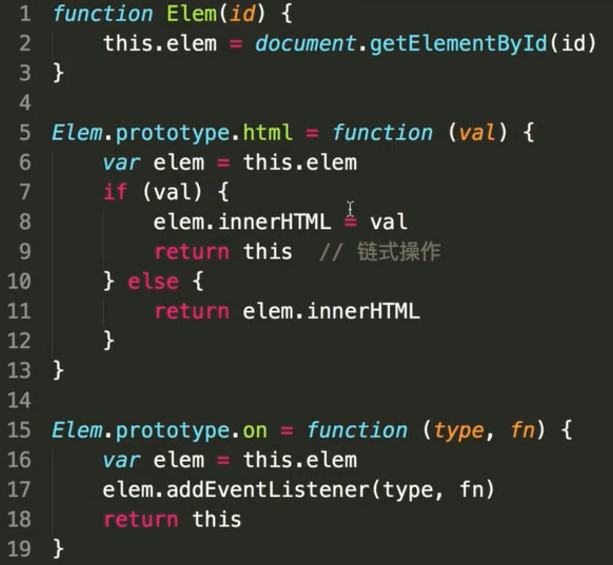
这个例子就是用来理解，面试的时候 low。

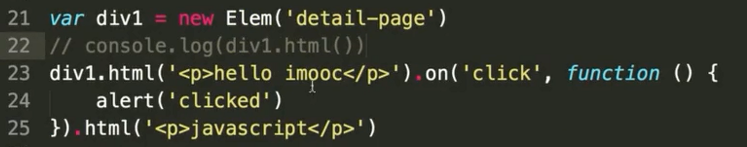
## 8.描述new一个对象的过程。

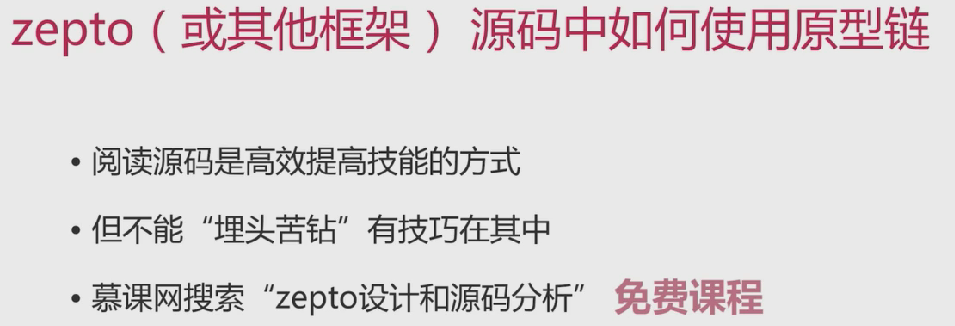


写一个贴近实际开发原型链继承的例子。（见下）

封装DOM查询的例子：



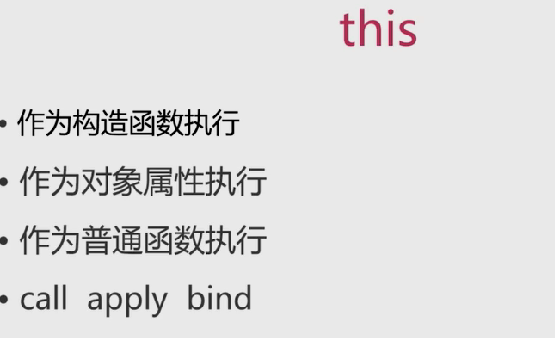




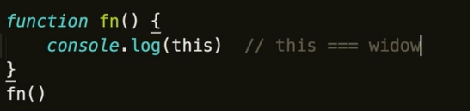
# 第三章：JS基础知识（中）

## 作用域与闭包

this要在执行时才能确定值，定义时无法确认。

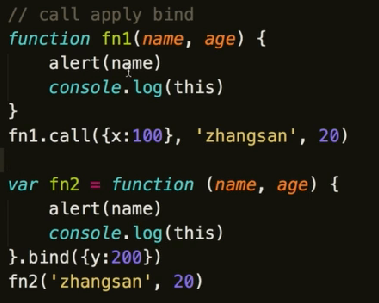


作为普通函数执行时，this===window。



Call()/Apply()：函数的方法，call(对象，传参)；apply(对象，[数组参数])

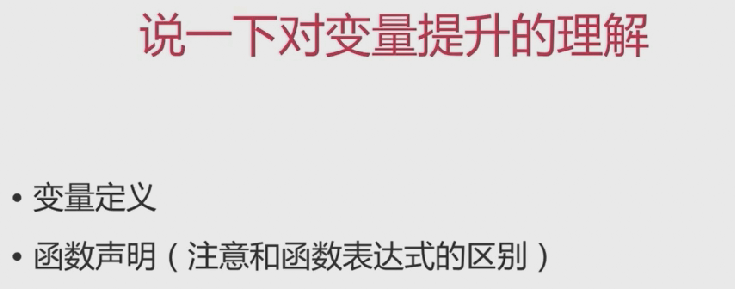
Bind():用函数表达式 调用（函数声明之后调用会出错）。参数为 对象。

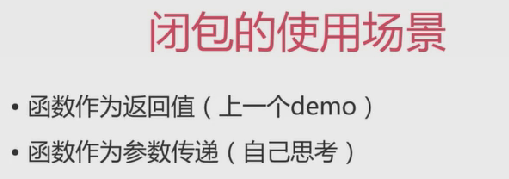


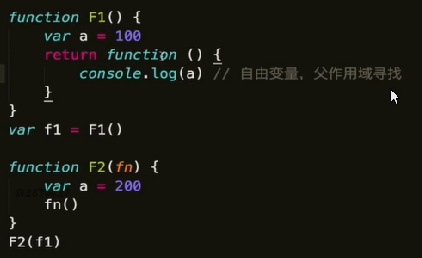
闭包：自由变量找父级作用域，定义的时候。而不是执行的时候。



返回100。

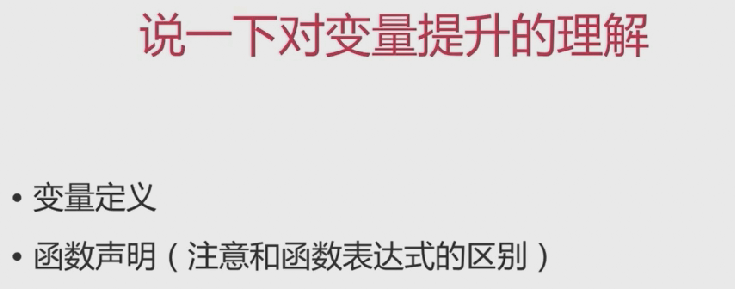






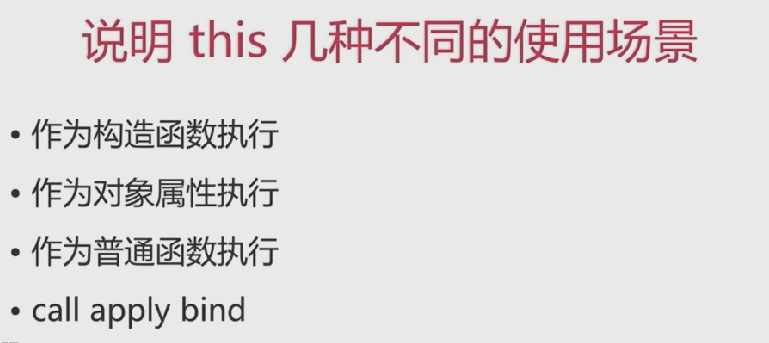
返回 100。

## 1.说一下对变量提升的理解



变量和函数的声明会被提前。

## 2.说明this几种不同的使用场景

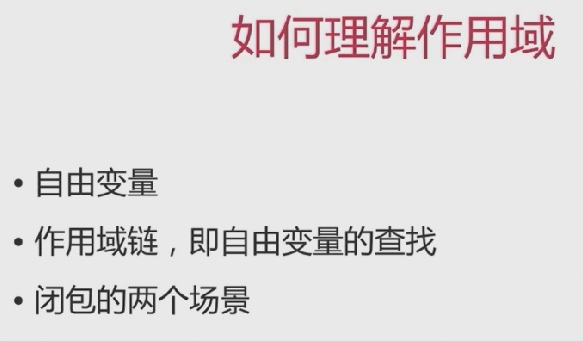


## 3.创建10个<a>标签，点击的时候弹出来对应的序号。





## 4.如何理解作用域？





# 第四章：JS基础知识（下）

## 异步和单线程

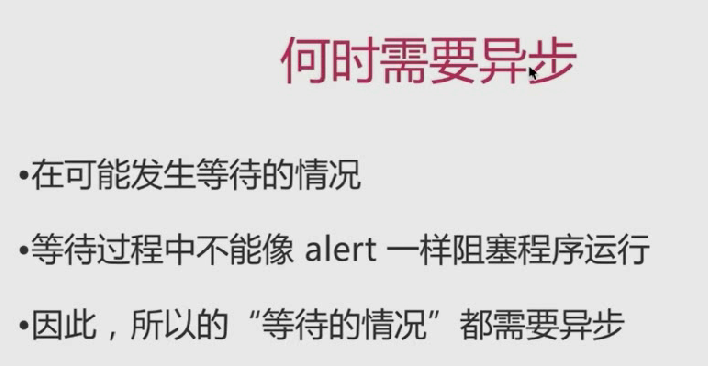


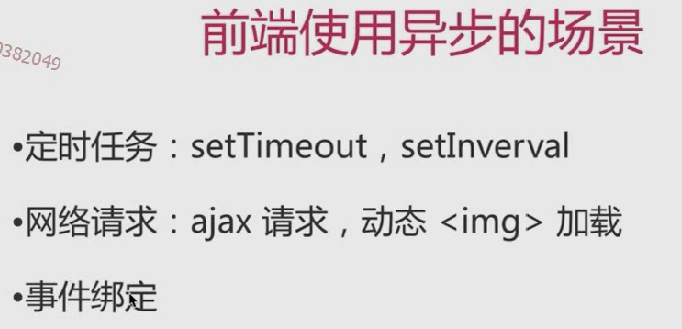
100 ，300 ，200



100 弹出200 点确认之后才会 出现300。

对比：异步不会阻塞，同步会阻塞。







Start ，end，请求道data1.json的数据之后，在打印数据。



Onload函数不会阻塞。



不会一直等着 人点击。