正课:

1. Function

2. OOP

1. Function

闭包:

原理:

作用域scope: 用法: 一个变量的可用范围

本质: 内存中一个存储变量的对象

闭包形成过程:

调用外层函数时，临时创建外层函数的作用域对象，保存外层函数的局部变量(受保护的变量)

外层函数，返回内层函数对象时，内层函数对象的scope属性引用了外层函数的作用域对象——形成了闭包

鄙视: 闭包形成的原因:

外层函数的作用域对象，被内层函数长期引用，无法释放。

何时主动使用闭包:

如果希望给一个函数，绑定一个专属的变量时

鄙视: 判断程序的输出结果: 画简图

1. 找受保护的变量: 外层函数的局部变量

2. 找外层函数抛出的内层函数对象:

2. OOP

什么是面向对象: 程序中都是用对象结构来描述现实中一个具体事物

什么是对象:

用途: 描述现实中一个具体事物的程序结构

本质: 内存中存储多个数据和函数的存储空间

为什么: 便于大量数据的维护

何时: 今后，几乎所有程序都是面向对象的

如何: 三大特点: 封装，继承，多态

封装(encapsulate): 创建一个对象，集中存储一个事物的属性和功能

为什么: 便于大量数据的维护

何时: 只要使用面向对象方式编程，都要先封装对象，再使用对象操作数据

如何: 3种:

1. 直接量: var obj={

属性名: 属性值,

... : ...,

方法名: function(){

... this.属性名 ...

},

...

}

问题: 对象自己的方法，要用自己的属性

错误: 直接写属性名

为什么: 因为不带任何前缀的普通变量，只能在作用域对象中找，无权擅自进入某个对象中查找。

正确: 即使对象自己的方法，想访问对象自己的属性，也必须加this.

this: 每个函数作用域对象中自带的，指向正在调用当前函数的.前的对象。

访问对象成员: 成员=属性+方法

对象.属性 其实就是保存在对象内的变量

对象.方法() 其实就会保存在对象内的函数

2. 用new: 2步:

1. var obj=new Object();

2. obj.属性名=值;

obj.方法名=function(){}

揭示: 其实js底层一切都是关联数组:

1. 可随时添加新属性

2. 如果访问不存在的成员，不报错，只返回undefined

3. obj.属性名=obj["属性名"]

如何选择:

1. 如果属性名是固定的，不需要动态生成，则首选obj.属性名

2. 如果属性名需要动态生成，则只能用obj["属性名"]

4. 都能用for in循环遍历

问题: 一次只能创建一个对象

反复创建多个结构功能相同的对象时，代码冗余

解决: 构造函数

3. 构造函数(constructor):

什么是: 描述一类对象统一结构的函数

何时: 如果需要反复创建多个相同结构的对象时

如何: 2步：

1. 定义构造函数:

function 类型名(属性参数列表){

this.属性名=属性参数;

... = ... ;

this.方法名=function(){

... this.属性名 ...

}

}

2. 调用构造函数反复创建多个新对象

var obj=new 类型名(属性值列表)

new: 4件事:

1. 创建新的空对象

2. 让新对象自动继承构造函数的原型对象

3. 调用构造函数为新对象添加新属性

new可将构造函数中的this->新对象

通过为新对象强行赋值新属性的方式，为新对象添加属性。

4. 返回新对象的地址保存在变量中

问题: 反复创建方法，浪费内存

解决: 继承

继承(inherits)

什么是: 父对象的成员，子对象无需重复创建，就可直接使用

为什么: 重用代码，节约内存

何时: 多个对象拥有相同的方法定义时

如何: 继承原型对象

原型对象: 集中存储多个子对象相同成员的父对象

何时: 只要希望有成员可被多个子对象继承使用时

如何:

创建: 不用自己创建原型对象

在定义构造函数时，附赠一个原型对象

如何继承: 不用手动设置继承关系

new创建子对象时，自动让子对象继承父对象

将共有的方法添加到原型对象中:

构造函数.prototype.方法名=function(){ ... }

强调: 在构造函数中不要定义方法

原型链: 由多级父对象逐级继承形成的链式结构

ES内置对象:

什么是: ES标准规定的，浏览器厂商已经实现的，我们直接使用的对象/类型

包括: 11个:

String Number Boolean

Array Date RegExp Math

Error

Function Object

global(全局作用域对象，在浏览器中被window代替)

特例:

Math不能new，因为不是构造函数

global也不能new，因为已经是全局作用域对象

其余类型都由2部分组成:

1. 构造函数: 创建该类型的子对象

2. 原型对象: 保存该类型所有子对象共享的API