复习：

1. 创建自定义对象的4种方法

var obj1 = new Object();

var obj2 = {};

function Emp(){}; var obj3 =new Emp();

var obj4 = Object.create(proto);

1. 基于原型的对象创建和继承

任何一个js对象都有一个原型对象，即它的父对象；所有js对象的最终父对象都是Object.prototype.toString = function(){

return ‘[object Object]’;

}

补充知识：

1.ECMAScript5中属性的分类

(1)数据属性(Data Property)

(2)访问器属性(Accessor Property)

var obj = {

**get 属性名(){**

**return 属性值;**

**},**

**set 属性名(value){**

**//需要接收到的value作处理**

**}**

};

注意：(1)访问器属性的本质是两个函数，若向读取访问器属性的值，会自动调用get访问器；若想为访问器属性赋值，会自动调用set访问器，并把等号右边的值传递给set访问器的形参。(2)访问器属性自己是不能存储数据的，所以访问器属性的值往往要依赖于另一个数据属性。(3)访问器属性一般用于两个场合：用于冗余属性 和 有意控制属性的只读或只写。

练习：创建一个emp对象，其中有两个属性：月薪、年薪，恰当的使用数据属性和访问器属性。

创建一个stu对象，其中保存语文、数学、总成绩三个属性，恰当的使用数据属性和访问器属性。

2.ECMAScript5中新添内容：属性(property)的特性(attribute)

属性：对象中可以保存数据的变量，如stu.math / stu.total

属性的特性：

1. 数据属性的特性：value、writable、enumerable、configurable
2. 访问器属性的特性：get、set、enumerable、configurable

练习：创建一个square对象，其中有三个属性：width、perimeter、size，要求可以精确的控制这三个属性的可枚举性。

作业整理：

(1)创建一个函数，可以去除字符串参数左侧的所有空白字符

(2)创建一个函数，可以去除字符串参数右侧的所有空白字符

(3)创建一个函数，可以去除字符串参数前导/后导的所有空白字符

上述三种方式，是“面向过程”的设计方式

(4)采用面向对象的方式，裁剪掉一个字符串对象的前导/后导的所有空白字符

(5)采用面向对象的方式，为所有的Date对象添加一个format，进行日期格式化

JavaScript中的数据类型：

1. 值类型
2. 特殊类型
3. 引用类型
   1. 原生对象——ECMAScript标准定义
   2. 宿主对象——由浏览器提供的对象——浏览器兼容性问题——W3C组织定义
   3. 自定义对象

3.当前主流的动态Web开发技术

动态：**网页的内容**，可以在不同的时间、针对不同的客户呈现出不同的内容。

1. JSP = HTML + Java
2. ASPX = HTML + C#
3. PHP = HTML + php

4.DHTML

Dynamic HTML，动态HTML、动态网页，**指的是页面的表现、样式**是可以随着用户的操作而发生不同的变化。DHTML = HTML + CSS + JavaScript，即DHTML是把已经存在的三项技术整合起来进行组合应用，就是使用JavaScript来“增删改查”HTML元素和CSS样式，最终使得页面呈现一个更友好的交互效果。

DHTML对象：

1. BOM：window、history、location、document、screen、event、navigator

用于JavaScript脚本与浏览器进行交互；

1. DOM：HTML页面中的每个元素都被JavaScript理解为一个DOM对象

用于JavaScript脚本与当前显示HTML文档进行交互；

5.DOM树

(1)**DOM元素树**：以document对象为根，每个HTML标签都是元素树的一个节点。

(2)**DOM节点树**：以document对象为根，每个标签、文本、属性、注释等都是节点树上的一个节点。

Node对象的几个属性：

1. parentNode: Node类型，当前节点的那个唯一的直接父节点对象
2. childNodes：NodeList类型，当前节点的所有子节点，组成一个类数组对象
3. firstChild：Node类型，当前节点的第一个子节点
4. lastChild：Node类型，当前节点的最后一个子节点
5. nextSibling：Node类型，当前节点的下一个兄弟节点
6. previousSibling：Node类型，当前节点前一个兄弟节点

---------------

1. nodeName：String类型
2. nodeType：Number类型
3. nodeValue：String类型

6.获取元素节点中的内容

var elementNode = ...;

elementNode.innerHTML; //获取/设置元素中的HTML内容

elementNode.textContent; //获取/设置元素中的纯文本——IE8-不支持

elementNode.innerText; //获取/设置元素中的纯文本——FF不支持

练习：

1、一个显示“读取报表”的按钮，被单击后，显示为“读取中，请稍候...”

2、实现“开关门”效果：一个span中，默认显示">>"，当被点击一次后，显示"<<"；再被点击一次后，显示">>"，依此往复

7.与元素节点的属性相关的DOM属性

(1)element.attributes //获取元素节点所有属性节点——类数组对象

(2)element.attributes[‘属性名’] //获取元素身上指定的属性的值

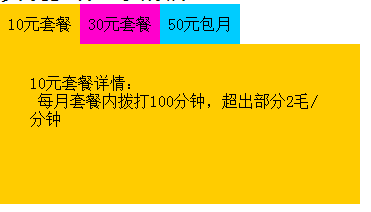
(3)element.getAttribute(‘属性名’) //获取元素身上指定的属性的值

(4)element.setAttributeNode(attrNode); //修改某个属性的值

(5)element.removeAttribute(‘属性名’) //删除指定的属性

练习：

使用“元素属性的添加和删除”方法，实现多标签页效果：



使用node.getElementsByTagName()方法实现表格隔行变色效果：

