1. javaScript：函数基础
   1. 定义方法

静态方法：function 函数名([形参]){

函数体；}

动态匿名方法：var 函数名 = new Function([“形参”]，”函数体”)；

直接量方法：函数名 = function([形参]){函数体;}

* 1. 调用方法

直接调用：函数名(实参)

在连接中调用：<a href= “javascript:函数名()”>内容</a>

在事件中调用：事件类型=”函数名()”

递归调用：在函数体内部调用函数自身

1. JavaScript一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。Netscape开发，用Unicode字符集编写
2. jQuery是一个兼容多浏览器的javascript库，核心理念是write less,do more(写得更少,做得更多)
3. javaScript内置对象<http://www.jb51.net/article/96194.htm>：

Object、String、Array、Number、Boolean、Date、RegExp、Math、Function、Arguments、Error、EvalError、RangeError、ReferenceError、SyntaxError、TypeError、URIError

1. 对于js不存在对象的实例化，每次新建的都是一个对象。同java一样，对象可以拥有属性，也可以拥有方法。

Var book = {

Title: “think in js!”,

Tag:true};

对于对象的=属性获取有两种方式：book.title; book[“title”];

1. Js是一个极为不严谨的语言，对象的属性可以直接在使用的地方新建并赋值。

所以构造函数和java区别：

Function Point(x,y){

This.x = x;

This.y =y;}

Var p1 =new Point(1,1);

Class Point(int x, int y){

Int x; int y;

This.x = x; this.y=y;}

Point p = new Point(1,1);

1. 数组的声明和方法：

Var arr = [0,2,3,4,6,23];

Arr.length();

Arr.push();

Arr.reverse()

Var point = [{x:0,y=1},{x=1,y=0}];

1. “2”+”3” --->”23”允许两个字符串直接相加
2. Js中没有类的概念,可以通过方法实现类的功能。
3. HTML5提供一系列Web应用API，包括网络，存储，图像。这些Web浏览器提供的操作系统级的服务，他们定义了全新的跨平台的应用环境。
4. Js区分大小写，HTML不区分大小写（XHTML区分）
5. Js基本数据类型：数字、字符串、布尔、null、undefind
6. Js对象：普通对象（名/值对）；特殊对象（函数）
7. 垃圾回收机制，自动回收
8. 数据类型本身可以定义方法，对a进行排序，不必将a传入sort()函数，而是调用a的一个方法sort();
9. 字符串是不可变的，js不提供修改已知字符串的文本内容的方法。
10. Js中的构造函数。
11. 数字是直接量，负号是一元运算符。Js不区分整数值和浮点数，所有数字均以浮点数表示。可以识别十六进制数
12. Js的上溢用infinity表示，js四舍五入会有问题，使用整数单位扩大倍数解决问题。
13. Js中任何值和对象都可以转换为布尔值。（其中：undefined、null、0、-0、NaN、””都是false；其他的对象、值都是true ）
14. 区分undefind和null可以用严格相等运算符“===”进行。
15. 一般可以认为undefind是系统级的、出乎意料的或类似错误的值的缺损。而null表示为程序级的、正常的或者在意料之中的值的缺省。
16. 全局属性：比如undefined、Infinity、NaN

全局函数：比如isNaN()、parseInt()、eval()

构造函数：比如Date()、RegExp()、String()、Object()、Array()

全局对象：Math、JSON

1. 在读取字符串、数字、boolean的属性值的时候，表现的和对象一样。但是如果你试图给其属性赋值，则会忽略这个操作：修改只是发生在临时对象身上，而这个临时对象并没有继续保留下来。
2. Date对象：