一、2月25日下午（13:00—17:30）：JS基本语法（变量、数据类型、条件和循环、函数）、对象（数组、字符串、Date）、window的使用：

1、JavaScript（简称JS）简介：

1）说明：JS是基于对象的脚本语言；解释执行，嵌入到HTML中执行；代码以纯文本方式存储；类型宽松；

2）JS的主要功能：表单验证、页面特殊效果、动态页面生成；

2、JS的三种使用方式：

1）页面中嵌入，建议在<head></head>里；

2）标记中嵌入，用于响应标记的事件；

3）引入外部的js文件，推荐的使用方式，适合多个页面共用；

示例：js\_01\_js的三种使用方式.html

3、JS的基本语法；

1）区分大小写，变量是弱类型（使用var声明变量）；每行结尾加分号。

2）标识符命名、基本运算符等与Java语言类似；

a、标识符以字母、下划线或美元符号开头，余下字符可以是前3者+数字；

3）注释：//表示单行注释，/\*…\*/表示多行注释

4）由浏览器逐条解释执行，某条语句出错后，后续语句不再执行；

4、ECMAScript中规定6种数据类型（5种基本的、1种复杂的），使用typeof检查：

1）null：空、无，表示不存在；

2）undefined：未定义。当声明变量却没有赋值时会显示该值；

3）number：数值。最原始的数据类型，表达式计算的载体；

4）string：字符串。最抽象的数据类型，信息传播的载体；

5）boolean：布尔值。最机械的数据类型，逻辑运算的载体；

6）object：对象。面向对象的基础；注意：function和null属于object的一种特殊形式；

示例：js\_02\_js的数据类型.html

5、类型转换：

1）自动转换（优先级由高到低：字符串、数值、布尔值）：

a、数值与字符串相加：数值转换为字符串；

b、布尔值与字符串相加：布尔值转为字符串；

c、数值与布尔值相加：布尔值转为数值；

2）通过函数把字符串转换为数值：

a、parseInt(var)、parseFloat(var)：只对string类型有效，并且应该是数字开头；

b、Number(var)：可以把boolean也转为数字；如果是字符串，需要是全数字；

示例：js\_03\_js类型转换.html

6、运算符：与Java一样；

7、流程控制：

1）条件语句if：

2）条件语句switch：

3）循环语句for：

4）循环语句while和do…while：

5）可以使用continue和break中断循环：

示例：js\_04\_流程控制.html

8、函数：

1）怎么定义函数；

function 函数名(参数1, 参数2, …){

函数体；

return;

}

2）arguments的使用；--通常不建议使用

function doTest(){

if(arguments[0] > 5){

console.log(“第一个参数大于5”);

}

}

doTest(5, 6, 7);

3）函数的嵌套定义；--通常不建议嵌套

4）函数的使用：普通调用、超链接调用、事件触发（最常用的）；

示例：js\_05\_函数的3种使用方式.html

5）isNaN(var)和eval(string)的使用；

a、isNaN(var)：判断参数是否不是数值，如果不是数值则返回true；

b、eval(string)：可以把参数字符串当做js表达式去执行；

6）练习：使用函数，实现在页面中输出九九乘法表（document.write(string)可以将参数输出到页面，并且可以在参数中使用HTML的标记，在页面可以正常解析HTML的标记；如：document.write( “1\*1=1” + “<br/>”) ）；

9、对象：主要掌握JS的4个内部对象

1）Array：

a、数组是一组变量的有序集合，允许一个数组中有多种数据类型；通过索引来使用数组中的元素，索引从0开始；JS中数组长度不固定，可以任意改变；

b、创建数组；var a = new Array(5); var b = [1, 2, 3, “ab”, “dcc”];

c、遍历数组；通常使用普通for来遍历，也可以使用for each遍历；

示例：js\_06\_数组的使用.html

d、练习：创建一个字符串数组，每个元素是颜色；如var a = new Array("red", "blue", "pink", "yellow", "green")，遍历该数组，并以每个元素值的颜色来显示该元素值（使用document.write()）；

示例：js\_07\_数组颜色练习.html

e、数组常用的方法：join(分隔符)、reverse()、sort()、toString()；

示例：js\_08\_数组的常用方法.html

2）Date：

a、Date对象可以表示任意的日期和时间、获取当前系统日期、计算两个日期的间隔（两个Date对象相减的结果是两者之间相差的毫秒数）；

b、可以获取日期中的年、月、日、小时、分钟、秒、当前毫秒、总毫秒、星期几的字段值；

getFullYear() 返回4位的年

getMonth() 返回月份值，比实际小1；如：八月份的值为7

getDate() 返回日期的值

getDay() 返回星期几，从0开始，星期日为0；

getHours() 返回小数数

getMinutes() 返回分钟数

getSeconds() 返回秒数

getMilliSeconds() 返回当前时间的毫秒数（是1秒之内的值）

getTime() 返回毫秒数，是从1970年1月1日0点0分0秒到现在的毫秒数

c、构造方法：Date()、Date(毫秒数)、Date(年, 月, 日) 、Date(年, 月, 日, 时, 分, 秒)等；

示例：js\_09\_创建Date对象.html

d、练习：获取110254671055毫秒数表示的日期，以“yyyy-mm-dd hh24:mi:ss”格式输出到页面；

e、练习：计算2017年圣诞节是星期几？据今天还有多少天？据现在还有多少毫秒？

3）Math：

a、Math是全局对象，不需要创建对象，直接使用；

b、常用方法：abs()—取绝对值、ceil()—向上取整、floor()—向下取整、round()—四舍五入、sqrt()—取平方根、random()—取随机数（范围0.0到1.0之间）等；

示例：js\_10\_Math的使用.html

c、练习：获取1到500之间的随机整数，直到获取1或者500为止；输出每次取到的随机整数，并输出一共取多少次才取到1或者500；

示例：js\_11\_取随机数.html

4）String：

a、表示字符串，通常直接赋值；

b、常用方法：charAt(index)、indexOf(str)、replace(str1, str2)、split(分隔符)、substr(start, length)、substring(start, end)、toLowerCase()、toUpperCase()等；

示例：js\_12\_字符串的方法.html

c、练习：定义一个函数，功能是返回去除参数字符串开头和结尾空格后的新字符串；写代码测试该函数的功能；

d、练习：定义一个函数，将参数字符串反转后返回；写代码测试；

e、练习：定义一个函数，计算参数字符串中各个元音字母出现的次数，输出到控制台；写代码测试；

f、练习：定义一个函数，参数是一个单词；把参数单词的第一个辅音音素的字母移动到词尾并且加上后缀-ay（譬如“banana”会变成“anana-bay”）返回；写代码测试；

5）自定义对象：

a、工厂函数方式（函数中var temp=new Object()）：直接调用；

b、构造函数方式（函数中使用this）：需要new调用；

c、通过{ } 直接创建对象；

示例：js\_13\_自定义对象.html

10、window对象：

1）打开新窗口（掌握）：open(url, name, 特性字符串)；

2）对窗口的操作（可能设计到安全性问题，需要发布到Tomcat中测试）：resizeBy(x, y)、resizeTo(x, y)、moveBy(x, y)、moveTo(x, y)；

3）与用户交互的两个方法：alert(str)、confirm(str)；

4）练习：在页面中提供5个按钮并实现如下功能：第一个按钮用于打开一个新窗口（不要菜单栏、状态栏；大小默认500\*300）；后面4个按钮分别是resizeBy(20, 20)、resizeTo(300, 300)、moveBy(20, 20)、moveTo(20, 20)；

11、获取页面中表单域的值：

1）两个常用方法用于获取页面中的对象：

a、var nodes = document.getElementsByName(name属性值); //返回集合

b、var node = document.getElementById(id属性值); //返回单个对象

2）通过对象的value属性获取值和修改值；

12、课后作业：做节日倒计时页面，在下拉列表框中选择下一个“元旦、情人节、劳动节、儿童节、国庆节、圣诞节”后，在后面的文本框中输出：XX节距离今天还有X天X小时X分X秒；（提示：要判断当前日期，才能获取正确的下一个节日；如：当前为2017年2月10日，则下一个元旦是2018年1月1日，下一个劳动节是2017年5月1日。）

