今天课程：JavaScript（js）拓薪教育

JavaScript部分（重点） 简称：js

js的简介

```
1.js基于对象和事件驱动的脚本语言，作用在客户端（浏览器）上。
2.js的特点
    * 交互性好。 人与服务器之间进行沟通。例子：注册的功能，输入特殊字符。浏览器直接给我提示。
    * 安全性          不可以直接访问本地的硬盘。
    * 跨平台性   浏览器就可以解析JS的文件。
3.JavaScript和Java是不同（一点关系都没有）          雷锋与雷峰塔
    * js语言的历史：网景      牛X的公司     开发LiveScript动态效果。      微软巨头。开发 浏览器大战     撕X大战
    * LiveScript改名称JavaScript         微软开发的语言JScript        弹劾（SUN 微软     网景   ECMA欧洲计算机制造协会）
    * 标准：开发一套语言标签     ECMAScript标准。
    * 如果想开发js     底层遵循ECMAScript标准，然后在基础上进行扩展。
    * 如果JScript 底层遵循ECMAScript标准，然后在基础上进行扩展。
    * PHP
4.js和Java的不同点（学习）
    * js基于对象，Java面向对象
    * js解析就可以执行，Java先编译，再执行。
    * js是弱类型的语言，Java是强类型的语言。
5.js语言的组成
    * ECMAScript标准             定义语法
    * BOM（浏览器对象模型）   代表整个浏览器（对象和API）
    * DOM（文档对象模型）          代表整个文档
```

js需要和HTML结合到一起（使用）

```
1.HTML用标签封装数据，CSS通过标签设置样式。js通过对象和语句来操作标签。
2.js和HTML的结合（2种）
    *.HTML的文件提供了一个标签<script type="text/javascript">  js代码    </script>
        特点：<script>标签可以放在页面的任意位置。
    *.引入外部文件的方式：<script type="text/javascript" src="引入JS的文件的地址" ></script>
        注意：如果src的属性引入了外部的文件，在<script>标准中间定义js代码就不会执行。
    *.扩展的问题：
        特点：<script>标签可以放在页面的任意位置。
        一般情况下：不是必须的。建议。
        如果引入了外部的文件（外部文件没有直接操作HTML的标签，一般的情况下放在<head>中间）
        如果在本HTML文件上编写JS的代码，如果也没有直接操作HTML的标签，推荐放在</body>标签之后。
```

js的语法

JS的关键字 标示符 注释 js的基本数据类型

```
1.关键字：用到哪个记住关键字
2.标示符：在js和java是一样的。
3.注释：         // 单行注释       /* 多行注释 */            /** 文档注释 */
```

```
4.变量：声明变量。
    如果Java中：int a = 10; String str = "abc";
    如果是在js中声明变量的话：就使用一个关键字var
5.js的数据类型
    5种基本数据类型：Undefined、Null、Boolean、Number 和 String
    * String         字符串
        在js中单引号和双引号都代表的是字符串 var str = "abc";    var str='abc';
    * Number         数字类型
        没有整数和小数之分。
    * Boolean        布尔类型
        和java一样的
    * Null       空，一般给引用赋值
    * Undefined 未定义（声明变量，没有赋值）

    js中声明变量使用    var
    typeof(变量)   ：判断当前的变量是属于什么类型的
    js语言弱类型的语言，声明任意类型的变量，都可以进行赋值。
```

js的运算符

```
* 算术运算符
    alert(num / 1000 * 1000);   //3710
    数字类型不区分整数和小数
    * 字符串中间数字，做减法js默认字符串解析，再进行运算。
    alert("abc"-1);      // NaN：非法的数字      弹出NaN（非法的数字）
    * 0或者null代表是false，非0或者非null代表是true，默认用1来表示。
* 比较运算符
    ==   ：只比较值是否相等
    ===  ：即比较值又类型是否相等
* 赋值元素   逻辑运算符    三元运算符    和java是一样的
```

js的语句

```
判断语句
循环语句
判断语句
if(true){

}else{

}
循环语句：使用var的关键字
    for(var i=0;i<10;i++){

    }
while(){

}
// 向浏览器的页面输出内容
window.document.write("i="+i+"<br />");
* 编写一个99乘法表（你们写）
```

js的数组（重要）

```
1.var arrs = ["abc","cba",1,3];
2.var arrs = new Array("abc","cba",..);
    var arrs = new Array(5);          声明数组，长度是5
    var arrs = new Array(5,6,7);      声明数组，元素是5,6,7
3.java数组长度不可变，js的数组长度可变。
```

js定义方法（函数）

```
* js中编写方法，需要使用关键字    function声明方法。
* java定义方法
    public String run(参数列表 int x,String y){
        ....
        return null;
    }
* js中定义方法
    function 方法名称(参数列表（没有var的关键字） x,y){
        ... 方法体
        return 返回值; 如果方法有返回值，直接写返回值，如果没有返回值，return可以省略不写。
    }
* js没有重载的方式
* arguments数组存入传入进来的参数
* 匿名函数：没有名称的函数。起个名称。
        xxxx.onclick = function(){
            // 执行
        }
```

js的全局和局部变量

```
* js全局变量：定义在<script>标签中间的变量，都是全局变量。
    不仅可以在<script>标签中间使用，在其他的<script>标签也可以使用。
* js局部变量：定义在方法内部的变量，是局部变量。
```

js的对象和api js的String对象

```
两种声明字符串的方式：
    var str = "abc";    var str ='abc';
    var str = new String("abc");
属性：length    -- 字符串长度。（截取字符串操作）
方法：
    1）和HTML的标签相关的一些方法
        bold()          把字符串显示粗体。
        fontcolor()     设置字体演示
        fontsize()      字体的大小
        italics()       斜体显示字符串
        link()          设置字符串为超链接
        small()         小字体
        sup()           上标显示字符串
    2）和Java中String相关的方法

        charAt()                                返回指定位置的字符
```

```
        indexOf(searchvalue,fromindex)      检索字符串
        lastIndexOf()                        从后向钱检索字符串
        replace()                            替换字符串
        substring(start,stop)               截取字符串（包含开始，不包含结束）
        substr(start,length)                截取字符串（从哪开始，截取多长）（包含开始）
        toLowerCase()                        小写
        toUpperCase()                        大写
```

js的Array对象

```
var arr = [];
var arr = new Array(5,6);
属性：length    -- 数组的长度
方法：
    concat(元素或者数组)
    join(分隔符)         把数组分隔字符串
    pop()              删除数组最后一个元素并且返回
    push(x)            向数组的末尾去添加一个元素
```

js的Date对象

```
Date对象
    * 获取当前的时间    var date = new Date();
方法：
    toLocaleString()         根据本地时间格式，把date转换字符串
    toLocaleDateString()     只有日期
    toLocaleTimeString()     只有时间
    getTime()                 获取1970-1-1号至今的毫秒数
    setTime(毫秒数)             // 通过毫秒数变成当前的时间
    parse(str)                静态方法，使用Date.parse();解析字符串，返回毫秒数
        // 2015-1-10 js中不能解析
        // 可以解析类型    2015,1,10    1/10/2015
```

js的Math对象

```
* 都是和数学相关的
* 静态的方法 Math.xxx()
    ceil(x)      对一个数进行上舍入
    floor(x)     对一个数进行下舍入
    round(x)     对一个数进行四舍五入
    random()     获取随机数    0-1之间小数
```

js的RegExp对象

```
* 作用：完成表单的校验。
* 声明
    var reg = new RegExp(表达式);
    var reg = /表达式/;
    特殊的写法：var reg = /^表达式$/;     （记住）
* 方法
    exec(str)    ：如果匹配成功，返回是匹配的结果
    test(str)    ：如果匹配成功，返回true，如果不成功，返回false     （记住）
```

js的全局函数

```
* 全局函数游离的状态，函数直接拿过来使用。
eval()  解析字符串，可以执行字符串中的方法
isNaN() 检测是否是非法的数字  如果是true代表是非法的数字

下面这些方法都和编码和解码有关。
想传输中文乱码的问题，先把中文编码（UTF-8），进行传输。再把内容进行解码。
encodeURI()      编码
decodeURI()      解码

encodeURIComponent()     编码
decodeURIComponent()     解码

escape()         编码
unescape()       解码
```

BOM（浏览器对象模型）

```
BOM：浏览器对象模型。
BOM Window                -- 窗口对象（*****重点）
BOM Navigator             -- 和浏览器版本相关的（**）
BOM Screen                -- 和浏览器屏幕相关的（忘了吧）
BOM History               -- 和浏览器历史相关的（**）
BOM Location              -- 和浏览器地址栏相关的（***）
```

BOM对象一

```
BOM Navigator             -- 和浏览器版本相关的（**）
BOM Screen                -- 和浏览器屏幕相关的（忘了吧）
BOM History               -- 和浏览器历史相关的（**）
    forward()    去下一页
    back()       去上一页
    go()         传入值，如果1等于forward()  如果传入-1代表是back()
BOM Location              -- 和浏览器地址栏相关的（***）
    href         当前页面的地址的链接  获取和设置当前网页的地址

window可以省略不写。
```

BDOM对象window对象

```
alert()                                  弹出提示框
confirm(显示的内容)                       弹出询问框，询问框上有两个按钮，一个是确定一个是取消。点击确定，
默认返回true，点击取消，返回false
setInterval("函数名称",毫秒数)       每隔多少毫秒就执行一次（执行n次）     默认返回id值
setTimeout("函数名称",毫秒数)            到多少毫秒后执行一次（执行一次）       默认返回id值
clearInterval(唯一的id值)              清除定时器
clearTimeout(唯一的id值)                   清除定时器
open(url,name,features)                弹出一个新的窗口
```

DOM（介绍）