事件；用户的操作

div1.style.display ='block';（把div1样式里面的display样式变成block；

.就是“的”的意思；

js里面的等号就是赋值的意思

编写js的流程：

布局：html和css

属性：确定要修改哪些属性

事件：确定用户做哪些操作

编写js：在事件中，用js来修改页面元素的样式

函数：跟css的class和id差不多

函数定义和调用：

function show() { //定义

alert('abc');

}

show(); //调用

任何标签都可以加id，包括link

if判断

赋值（=改变）

= = ：判断

函数传参：

参数就是占位符；当函数里面有不确定的东西的时候，就可以使用传参。

操作属性的第二种方式，当修改的属性不固定的时候，

字符串和变量

除了变量和参数之外都要加引号“”

如果不带引号，系统就会默认这是一个变量，会提示变量没有定义

提取行间事件：

style加样式是加在行间的。

为元素添加事件

事件和其他属性一样，都可以用js添加

window.onload:页面加载完成时候发生

循环：

选项卡：

innerhtml：

JavaScript组成

ecmascript：翻译 核心 解释器

dom：document object model 文档对象模型 就是操作html的能力

bom browser object model 跟浏览器打交道的操作

各组成部分的兼容性：

ecma： 几乎没有兼容性问题

dom： 有一些操作不兼容

bom： 完全不兼容

变量类型：

一个变量应该只存一种类型的数据

类型转换：

显示类型转换

隐式类型转换

作用域：

全局变量（在任何地方都可以使用）和局部变量（只能在定义他的函数里面使用）有一种情况列外：子函数可以使用父函数长度局部变量。

闭包：

闭包实际总结一句话：子函数可以使用父函数的局部变量

命名规范

可读性和规范性

命名方式：类型前缀，首字母大写



括号可以改变运算符的优先级

程序流程控制

if的高级用法：

if（条件）{}

else if（条件）{}

else（条件）{}

另外一种：switch

switch（变量）{

case 值1：

break；

case 值2：

break；

default；

}

三元运算符 ？：

*var* n = 24;  
n % 2 == 0 ? alert('这是双数'):alert('这是单数');

break 打破、中断（整个循环结束）

continue 结束本次循环

真：有东西的时候 true 非零数字 非空字符串 非空对象

假：没有东西的时候 false 数字0 空字符串 空对象 undefined

json

*var* json = {a : 12,b : 23,c : 'acv'};  
json.b ++;  
alert(json.b);

json（写了的才会有）和数组

json的下标是字符串，而数组的下标是数字

数组有length属性，而json没有length属性

数组的length可以循环

json和for in

数组还有一种循环for in

但是如果用数组的话：一般用for 0 – length

如果是用json的话，一般用for in循环

函数返回值：

一个函数应该只返回一种类型的值

函数传参

不定参：arrguments

参数的个数是可变的

取非行间的样式：

chrome下获取非行间的样式：（不兼容）

window.onload = *function* () {  
 *var* oDiv = document.getElementById('div1');  
 alert(getComputedStyle(oDiv).width);  
}

数组：

数组的属性，length

数组的length即可以获取也可以设置

让length = 0；可以快速清除数组里面的内容

数组的方法：尾部添加push和尾部删除pop

头部添加shift和头部删除unshift。

数组的排序：

数组中存储的字符串的时候用arr.sort();

数组中存储的是数字的时候：

*var* arr = [4,2,9,25,5,6,56,8];  
arr.sort(*function* (*n1*,*n2*) {  
 *return n1* - *n2*;  
});  
alert(arr);

数组的插入删除：

删除：splice；（起点，长度）

*var* arr = [1,2,3,4,5];  
//删除从第三位开始的后一位  
arr.splice(3,1);

splice插入：相当于替换

splice（起点、长度、元素）

数组的连接：

*var* a = [1,2];  
*var* b = [3,4];  
alert(a.concat(b));

得到1，2，3，4

数组的join分隔符：

用字符串内的符号或者数字将数组中的内容连接到一起

*var* a = [1,2,3,4,5,6,7,8];  
alert(a.join('-'));

定时器的作用：

setinterval 间隔型 （无限执行）

settimeout 延时型 （只执行一次）

数码时钟：

效果思路：

获取系统时间

date对象

getHours,getMinutes,getSeconds

显示系统时间

字符串连接

空位补零

设置图片路径

charAt方法charAt(i)兼容

获取元素当前的宽度、高度、左边距和右边距分别用offSetLeft\offSetTop\offSetWidth\offSetHeight

这个函数得到的是一个数字值，所以可以直接相加，改变元素的位置和宽高。

将oDiv对象的left左边距增加10px；可以这样写：

oDiv.style.left = oDiv.offSetLeft + 10 + ‘px’;

DOM：document

让js能够操作页面的元素

子节点：（只将直系的节点，作为父节点的子节点，孙子辈的节点不算做父节点的子节点）

childnodes和nodetype配合就可以循环节点

还有一个可以循环所有标签节点的就是children

window.onload = *function* () {  
 *var* oUl = document.getElementsByTagName('ul')[0];  
 console.log(oUl.children);  
 *for* (*var* i = 0;i < oUl.children.length; i ++){  
 oUl.children[i].style.backgroundColor = 'red';  
 }  
};

css选择器优先级：

！important： infinity

行间样式： 1000

id选择器：100

class选择器、属性选择器、伪类（：hover）： 10

标签和伪元素（：：）：1

通配符\*：0

利用dom操作元素属性

设置：setAttribute(名称，值)

获取：getAttribute（名称）

删除：removeAttribute（名称）

用className选择元素：

用dom插入一个新的元素：

appendChild：

父级.appendChild(子节点)

创建、插入、删除元素

创建

createElement（标签名） 创建一个节点

appendChild（节点） 追加一个节点

在父节点上插入子节点

父级.appendChild(子节点)

插入：insertBefore(节点，原有节点) 在已有元素前插入

文档碎片createDocumentFragment

表格应用