1. ASCII

1--49 A--65 a--97

2.

<input type="text" id="inp01"><br><br>  
<div id="id01" style="width:100px;height:100px;background:red;position:absolute">hi....</div>

<script>  
 //DOM  
 //var 定义变量  
 var v1=document.getElementById("id01");//查找元素  
  
 //事件处理--点击  
 v1.onclick=function(){  
 alert("hello world!");  
 }  
  
 //键盘事件  
 var v2=document.getElementById("inp01");  
 /\*键盘按下触发\*/  
 v2.onkeydown=function(e){  
 alert(e.keyCode);  
 if(e.keyCode==39){  
 v1.style.left=v1.offsetLeft+100+"px";  
 }  
 }  
</script>

1. javascript简介  
   jquery.js [ angular.js node.js--express react.js zepto]移动端 boostrap backboon  
   1)js 1995  
   2)ECMA(equropean computer manufactures association)欧洲计算机制造商协会  
    1997 ECMA组织定义新的脚步语言标准：ECMAScript(ECMA-262)  
   3)ECMAScript规定了：语法、类型、语句、关键字。。。。等  
   4)2009 ECMAScript第五版 现在已有第六版 第七版只在编译未发布  
    ie5-7 ECMAScript第三版  
   5)js组成  
    ECMAScript、DOM、BOM  
     
   4.js基础  
   1)<script>可以放到文档中的任意位置  
   2)属性  
    src:引用的外部文件  
    type:指定脚本语言的内容类型  
    默认:text/javascript  
    application/javascript 非ie的早期浏览器  
     
    只适用于外部引用文件（defer async）：  
    defer:延迟脚本  
    1、脚本会被延迟到整个页面都解析完毕后再运行  
    2、采用defer属性的脚本不一定会按顺序执行，所以最好只写一个defer属性的脚本  
    async:异步脚本  
    该属性的目的是不让页面等待脚本下载完后再执行，而是异步加载页面其他内容

<!--引入外部脚本-->  
<script src="js/test2.js" type="text/javascript" defer="defer"></script>

<script src="js/test.js" type="text/javascript" async="async"></script>

3)分号  
 （1）每条语句在独立行时，分号可以省略  
 （2）多条语句一行时，必须写分号，最后一条语句的分号可以省略  
 最好养成写分号的习惯  
4)常量 const  
const pi=3.14; 不允许修改

<script>

**const** pi=3.14;  
 pi=pi+10; //不允许修改  
 alert(pi);  
</script>

1. 数据类型  
   一、数值型  
    var v1=12;  
    v2=12.12;  
    v3=0.009;  
    v4=6e9;/\*科学计数法6\*10^9 0.5e-5\*/  
    v5=0x56;/\* 5\*16^1+6\*16^0\*/  
    v6=0X67;/\*0x 0X表示16进制 568--2进制1000111000 8进制1070 16进制238\*/  
    v7=023;/\*0 八进制\*/  
    v8=090;/\*八进制，超出范围，原样输出\*/  
    v9=50e500;/\*infinity 正无穷 正数超出范围\*/  
    v10=-50e500;/\*-infinity 负无穷 负数超出范围\*/  
    alert(Number.MIN\_VALUE);  
    alert(Number.MAX\_VALUE);/\*超出该最大值，infinity 正无穷 正数超出范围\*/  
    /\*大整数的四则运算\*/

1. var v1=prompt("请输入年龄");//输入提示框  
    alert(v1);//消息提示框  
    document.write("您的年龄："+v1);//网页中显示  
     
   三、类型转换  
   1)显示类型转换

(1)number()  
NaN:not a number 非数值 表示一个本来要返回数值的操作数而没有返回数值的情况  
number规则：  
 1）布尔型 true--1 false--0  
 2）var v14=Number(null); 0  
 var v15=Number(NaN); NaN  
 var v16=Number(undefined); NaN  
 3）对数值正常转换  
 4）八进制、十六进制正常转换  
 5）字符串：1、字符串为空，转换为0  
 2、字符串中只含数字('023')，将其转为十进制，并忽略前导0  
 3、字符串中含有效的十六进制格式，将其转为对应的十进制数值  
 4、除上述格式以外，全部转换为NaN

例：

var v1=Number(12); 12  
var v2=Number('23'); 23  
var v3=Number(12.5); 12.5  
var v4=Number(12.9); 12.9  
var v5=Number(0.12); 0.12  
var v6=Number(-12); -12  
var v7=Number('hello'); NaN  
var v8=Number('56aa'); NaN  
var v9=Number(56aa); 报错  
var v10=Number(true); 1  
var v11=Number(false); 0  
var v12=Number(""); 0  
var v13=Number(''); 0  
var v14=Number(null); 0  
var v15=Number(NaN); NaN  
var v16=Number(undefined); NaN  
var v17=Number(0x45); 69  
var v18=Number('0x45'); 69  
var v19=Number(023); 19//八进制转十进制  
var v20=Number('023'); 23  
var v21=Number(000023); 19  
var v22=Number('000023'); 23  
var v23=Number(23e4); 230000  
var v24=Number('23e4'); 230000  
var v25=Number(0.2e-5); 0.000002  
var v26=Number('0.2e-5'); 0.000002

(2)parseInt()  
 1.对数值型取整数部分  
 2.对字符型数据，如第一个字符不是数字或正负号,那么输出NaN 空字符--NaN  
 3.如第一个字符是数字，会继续解析第二个，直到遇到非字符为止  
 4.忽略字符前面的空格  
// 其他进制转十进制：  
// var v1=parseInt(25,16); 37  
// var v1=parseInt('25',16); 37  
// var v1=parseInt(25,8); 21  
// var v1=parseInt('25',8); 21  
// var v1=parseInt(1100,2);12  
// var v1=parseInt('1100',2);12  
// var v1=parseInt(25,10);25//解释（数值，进制）  
// var v1=parseInt('25',10);25  
  
// var v1=parseInt('073');//73  
// ie低版本 59 ie6、7 ECMAScript 3  
// 高级浏览器 73 ECMAScript 5

var v1=parseInt(12); 12  
var v1=parseInt("12");12  
var v1=parseInt("-12"); -12  
var v1=parseInt("12.3");12  
var v1=parseInt("12.9");12  
var v1=parseInt("0.12");0  
var v1=parseInt("-12");-12  
var v1=parseInt("-0.12");0  
var v1=parseInt('hello');NaN  
var v1=parseInt(23aa); 错误  
var v1=parseInt('23aa'); 23  
var v1=parseInt(true);NaN  
var v1=parseInt(false);NaN  
var v1=parseInt("");NaN  
var v1=parseInt('');NaN  
var v1=parseInt(0x56);86  
var v1=parseInt('0x56');86  
var v1=parseInt(076);62  
var v1=parseInt('076');76  
var v1=parseInt(089);89  
var v1=parseInt('089');89  
var v1=parseInt(000123);83  
var v1=parseInt('000123');123  
var v1=parseInt(30e5);3000000  
var v1=parseInt('30e5');30  
var v1=parseInt(0.2e-5);0  
var v1=parseInt('0.2e5');0  
var v1=parseInt('aa23');NaN  
var v1=parseInt('23aa45');23  
其他进制转十进制：  
var v1=parseInt(25,16); 37  
var v1=parseInt('25',16); 37  
var v1=parseInt(25,8); 21  
var v1=parseInt('25',8); 21  
var v1=parseInt(1100,2);12  
var v1=parseInt('1100',2);12  
var v1=parseInt(25,10);25  
var v1=parseInt('25',10);25

(3)parseFloat()  
 1.对浮点型正常转换  
 2.parseFloat()不支持两个参数用法（进制转换）

var v1=parseFloat(12);12  
var v1=parseFloat("12"); 12  
var v1=parseFloat("-12");-12  
var v1=parseFloat("12.3"); 12.3  
var v1=parseFloat("12.9");12.9  
var v1=parseFloat("0.12");0.12  
var v1=parseFloat("-12");-12  
var v1=parseFloat("-12.9");  
var v1=parseFloat('hello');-0.12  
var v1=parseFloat(23aa);  
var v1=parseFloat('23aa');23  
var v1=parseFloat(true); NaN  
var v1=parseFloat(false);NaN  
var v1=parseFloat("");NaN  
var v1=parseFloat('');NaN  
var v1=parseFloat(0x56);86  
var v1=parseFloat('0x56');0  
var v1=parseFloat(076);62  
var v1=parseFloat('076');76  
var v1=parseFloat(089);89  
var v1=parseFloat('089');89  
var v1=parseFloat(000123);83  
var v1=parseFloat('00123');123  
var v1=parseFloat(30e5);3000000  
var v1=parseFloat('30e5');3000000  
var v1=parseFloat(0.2e-5);0.000002  
var v1=parseFloat('0.2e5');20000  
var v1=parseFloat('aa23');NaN  
var v1=parseFloat('23aa45');23  
var v1=parseFloat(25,16); 25  
var v1=parseFloat('25',16); 25  
var v1=parseFloat(25,8); 25  
var v1=parseFloat('25',8); 25  
var v1=parseFloat(1100,2); 1100  
var v1=parseFloat('1100',2); 1100  
var v1=parseFloat(25,10); 25  
var v1=parseFloat('25',10); 25

1. 隐示类型转换

(1) var v1=prompt("请输入年龄");  
 if(v1>20){  
 alert("成人了，长大了");  
 } //隐示转换  
  
(2) var a='3';  
 alert(a\*8);  
 alert(a+8);//38 + 字符串的连接符  
 alert(a\*1+8);//11 \* 隐示转换

1. 函数

<input type="text" id="id01" value="">  
<button onclick="fun1()">确定</button>  
<div id="show">

//函数  
function fun1(){

//取文本输入控件中的值  
 var v1=document.getElementById("id01").value;

//查找元素  
 var v\_show=document.getElementById("show");  
 //div中加入数据  
 v\_show.innerHTML="<span style='color: red;'>数据为：</span>"+v1;

v\_show.innerText="<span style='color: red;'>数据为：</span>"+v1;

//innerText不支持“”里嵌套html innerHTML支持“”里嵌套html  
}

7.结算

<div id="danjia">34.5</div>  
<input type="text" id="id02" value="1">  
<span onclick="fun2()" style="width: 30px;height: 30px;background: gainsboro;display: inline-block;">+</span>  
<div id="zongjiage"></div>  
<input type="checkbox" id="check1" onclick="fun3()">  
<div id="jiesuan"></div>

//函数

function fun2(){  
 var v1 = document.getElementById("id02").value; //取值  
 //v1+=1; v1++ “2”\*1+1=3

var v2 = Number(v1) + 1;

document.getElementById("id02").value = v2; //赋值  
 //取div中的值  
 var v\_dj = document.getElementById("danjia").innerText;  
 document.getElementById("zongjiage").innerText = v\_dj \* v2;//赋值  
}

function fun3(){  
 var v3=document.getElementById("check1");  
 //判断复选框是否被选中  
 //if(v3.checked==true){}  
 if(v3.checked){

document.getElementById("jiesuan").innerText=document.getElementById("zongjiage").innerText;  
 }else{  
 document.getElementById("jiesuan").innerText=0;  
 }  
 }

8.舍入误差

var a=0.1;  
var b=0.2;  
if((a+b)==0.3){  
 alert("yes");  
}else{  
 alert("no");  
}  
a+b=0.30000004  
浮点数值计算会产生“舍入误差”问题  
解决：Math.round tofix()

9.

一、null

Number(null)---0  
Number(undefined)---NaN  
Number(NaN)---NaN  
alert(null\*8)---0;

二、NaN

（1）任何涉及NaN的操作，都会返回NaN  
（2）NaN与任何值都不相等，包括本身  
alert(NaN\*5)---NaN  
alert(NaN+5)---NaN  
alert(NaN/8)---NaN  
alert(Nan==NaN);---false

（3）isNaN()判断是否为非数值  
任何不能被转换为数值的值都会返回true  
var v1='8';  
if(isNaN(v1)){  
 alert("a1");//输出  
}else{  
 alert("b1");  
}  
isNaN(NaN); true 是非数值  
isNaN(10); false 不是非数值，是数值  
isNaN('8'); true  
isNaN('hello'); true  
isNaN(true); false  
isNaN(false ); false  
（4）alert(0/0)---NaN  
（5）alert("hello"\*3);---NaN

三、

（1）'23'+9 ---239 + 字符串的连接符  
 '23'\*2 ---46 \* 隐示转换  
（2）34/0 ---infinity 正无穷  
 -34/0 --- -infinity 负无穷

四、isFinite();判断是否为无穷 Finite--有穷

if(isFinite(34/0)){  
 alert("a1");  
}else{  
 alert("b1");//输出  
}

1. 包装类型 new Number()此Number非彼Number()  
   var a=10;  
   a1=new Number(10); //new Number()定义object（对象）  
   alert(a1);//输出10  
   alert(typeof a);--number  
   alert(typeof a1);--object

11.

toFixed//保留几位小数，自动四舍五入  
toExponential//科学计数法  
toPrecision//即可返回固定大小的格式，也可返回指数（e）格式，自动匹配规则  
  
var v1=199;  
var v2=89.008;  
v1.toFixed(2);---199.00  
v1.toExponential(2);---1.99e+2  
v2.toFixed(2);---89.01  
v2.toExponential(2);---8.90e+1 //科学计数法 2位小数  
  
v1.toPrecision(1);---2e+2  
v1.toPrecision(2);---2.0e+2  
v1.toPrecision(3);---199  
v1.toPrecision(4);---199.0  
v2.toPrecision(3);---89.008  
  
alert(v1.toPrecision(3));  
alert(v2.toPrecision(3));

1. 字符型  
   （1）单双引号都可以  
   （2）十进制转多进制  
    var v2=23;  
    v2.toString(2);---10111  
    v2.toString(8);---27  
    v2.toString(10);---23  
    v2.toString(16);---17  
   （3）转义字符  
    \'  
    \"  
    \\  
    部分转义字符在输出为html文本时不会发生作用  
    \n回车换行  
    \r回车  
    \t tab制表符  
    document.write \n\r失效 <br>代替  
     
   （4）字符串操作  
    1）v1.length字符串长度  
    var v1="abcd";  
    alert(v1.length);//4  
     
    2）chatAt() 读取指定位置的字符 下标由0开始  
    var v1="abcd";  
    v1.charAt(0)  
    v1.charAt(3)  
    alert(v1.charAt(0));---a  
    alert(v1.charAt(3));---d  
    3）charCodeAt() 返回单个字符的编码  
    中文：4E00~~9FCF UNICIDE码 //5f2 5B5d 60a6 张孝悦  
    var v4="孟祥东总迟到";  
    alert(v4.charCodeAt(2));// 东 19996 十进制  
     
    var v5=19996;  
    alert(v5.toString(16));//4e1c 十六进制  
      
    var v6=v4.charCodeAt(0);// 孟  
    if((v6>=parseInt("4e00",16)) && (v6<=parseInt("9fcf",16))){  
    alert("中文");  
    }else{  
    alert("不是中文");  
    }

4)查找指定字符串的索引值 //索引值从0开始  
 indexof() 从前找  
 lastIndexOf() 从后找  
 如找不到返回-1

例：

var v1="hello world";

v1.indexOf("o");4  
 v1.lastIndexOf("o");7  
 v1.indexOf("o",6); 7  
 v1.lastIndexOf("o",6); 4

v1.indexOf("aa"); -1

//定义数组

var arr1=new Array();

//添加数组元素

arr1.push(flag);

var v2="hello world";  
 //数组循环输出  
 for(var i=0;i<v2.length;i++){  
 alert(v2[i]);  
 }  
 5)replace()替换  
 1)v1.replace("o","a");  
 alert(v1.replace("o","a"));//只替换第一个  
 2)//正则表达式 全部替换 g表示全局  
 var reg=/o/g;  
 var reg=/\d/g; //\d表示数字  
 alert(v1.replace(reg,'a'));  
 6)toLowerCase()//转小写  
 toUpperCase()//转大写  
 7)split --分割 --数组格式返回

var v1="heomoLLo";

var v2=v1.split("o"); //he,m,LL,  
 var v3=v1.split("o",2);//he,m  
 8)截取子字符串  
 slice() substr() substring()

var v1="hello world";  
alert(v1.slice(3)); lo world  
alert(v1.slice(3,8)); lo wo //[3,8)，位置从0开始  
alert(v1.slice(-3)); rld //从后取3位，位置从-1开始  
alert(v1.slice(3,-2)); lo wor //[3,-2)  
alert(v1.slice(0)); hello world  
alert(v1.slice(8,3)); ""  
alert(v1.slice(-3,-4)); ""  
  
alert(v1.substr(3));lo world  
alert(v1.substr(3,8));lo world //从3开始8个单位（截取长度）  
alert(v1.substr(-3));rld  
alert(v1.substr(3,-2));""  
alert(v1.substr(0));hello world  
alert(v1.substr(8,3));rld //从8开始3个单位  
alert(v1.substr(-3,-4));""  
  
alert(v1.substring(3));lo world  
alert(v1.substring(3,8));lo wo //[3,8)  
alert(v1.substring(-3));hello world //将所有负数转换为0  
alert(v1.substring(3,-2));hel //等价于substring(3,0) [0,3)  
alert(v1.substring(0));hello world  
alert(v1.substring(8,3));lo wo //第一个参数大于第二个参数，自动调换 [3,8)  
alert(v1.substring(-3,-4));""

9)trim() 删除前缀后缀空格  
 10)concat() 字符串的拼接 等价于加号

var v1="hello";

alert(v1.concat("world")); //hello world  
 11)localeCompare() 比较  
 var v1="hello";  
 alert(v1.localeCompare("abc")); 1 h>a 值为1  
 alert(v1.localeCompare("hello")); 0 hello=hello 值为0  
 alert(v1.localeCompare("zdf")); -1 h<z 值为-1

v1="江";//jiang  
 alert(v1.localeCompare("于"));//yu -1  
 alert(v1.localeCompare("阿"));//a 1  
 alert(v1.localeCompare("句"));//ju -1

（5）for() 可以嵌套  
  
（6）清空 获得焦点  
 function fun1(){  
 var inp=document.getElementById("id01");  
 v1=inp.value  
 if(v1.length>5){  
 alert("error!!");  
 //清空  
 inp.value="";  
 //获得焦点  
 inp.focus();  
 }  
 }

1. 流程控制  
   1)if(){  
    }else{  
    }  
   2)if(){  
    }else if(){  
    }else if(){  
    }else{}  
   3)分支语句  
   switch(){  
   case ...: ..;break;  
   case ...: ..;break;  
   default: ..;break;  
   }  
   4)for(初始值；判断语句；递增或递减){  
    循环体  
    }  
   5)while(){  
    循环体  
    }  
   6)do{  
    循环体  
    } while();//至少执行一次  
   7)结束循环  
    return  
    break 立即跳出循环  
    continue 停止当前循环 进入下一次循环

14.

1. 布尔型Boolean()  
   true false 小写
2. 其他类型转为布尔型：  
    true false  
   数值型 非0 0  
   字符型 非空字符串 ""  
   undefined 返回false  
   null 返回false  
   object 任意对象返回都是true

var v1=true;y  
 var v1=false;n  
 var v1="";n  
 var v1="hello";y  
 var v1=1;y  
 var v1=0;n  
 var v1=undefined;n  
 var v1=null;n  
 if(v1){  
 alert("yes");  
 }else{  
 alert("no");  
 }  
  
2)基本类型：数值型 字符型 布尔型 undefined null object  
包装类型：new Number() new String() new Boolean()  
区别：  
 typeof 对基本类型返回对应的类型，只存在于代码执行瞬间，之后立即被销毁  
 typeof 对包装类型返回object，在离开当前作业域之前一直保存在内存中

var a='23';  
 var b=Number(a);  
 alert(typeof b);//number  
 var c=new Number(a);  
 alert(typeof c);//object  
  
 var a1="hello";  
 alert(typeof a1);//string  
 var a2=new String(a1);  
 alert(typeof a2);//object  
  
 var b1=true;  
 alert(typeof b1);//boolean  
 var b2=new Boolean(true);  
 alert(typeof b2);//object

var v1=new Boolean(false);---对所有对象都会转换为true  
 var v1=new Boolean(0);  
 var v1=new Boolean("");  
 var v2=v1&&true;  
 alert(v2);//true  
 var v1=false;  
 var v2=v1&&true;  
 alert(v2);//false

1. 四、undefined  
   表示变量被声明 但没有被赋值

var a;  
alert(a);//undefined  
alert(typeof a);//undefined

1. 五、null 空类型

var v1=null;  
alert(typeof v1);//object  
alert(undefined==null);//true 原因：undefined派生自null

1. 六、object  
   typeof:  
    alert(typeof(1));//number  
    alert(typeof(NaN));//number  
    alert(typeof(Number.MAX\_VALUE));//number  
    alert(typeof(Infinity));//number  
    alert(typeof("23"));//string  
    alert(typeof("hello"));//string  
    alert(typeof(true));//boolean  
    alert(typeof(null));//object  
    alert(typeof(new Array()));//object  
    alert(typeof(document));//object  
    alert(typeof(undefined));//undefined  
    alert(typeof(abc));//undefined  
    alert(typeof(Date));//function  
    alert(typeof(function fun1(){}));//function
2. 内置对象  
   1)数学对象--Math  
    var v1=-23;  
    alert(Math.abs(v1));//23 绝对值  
    var v2=23.6;

alert(Math.ceil(23.4));//24 向上取整  
 alert(Math.floor(v2));//23 向下取整  
 alert(Math.round(v2));//24 四舍五入  
 alert(v2.toFixed(2));//四舍五入保留两位小数  
 var v3=12;  
 alert(Math.sqrt(v3));//3.4641016151377544 除方  
 alert(Math.pow(v3,2));//144 平方  
 alert(Math.PI);//3.1415926  
 alert(Math.random());//范围0-1之间随机数

//输入两个数，取中间随机值  
 r\_b:开始值  
 r\_e:结束值  
 var v3=Math.round(Math.random()\*(r\_e-r\_b)+r\_b);

2)日期对象--Date

1. java.util.Date 基础上构建

一、

var d1=new Date();  
document.write(d1);  
输出：Wed May 25 2016 09:06:23 GMT+0800 (中国标准时间)

二、

document.write(d1.getDay());//星期 1 .... 6 0  
document.write(d1.getDate());//几号（第几天）  
document.write(d1.getHours());//小时 1 ....23 0  
document.write(d1.getMinutes());//分钟 1 ..... 59 0  
document.write(d1.getMonth());//月份+1=当前月份 0.....11  
document.write(d1.getSeconds());//秒 1.....59 0

document.write(d1.getYear());//116 年份 从1900开始算  
document.write(d1.getFullYear());//2016 年份

document.write(d1.getTime());//毫秒数 从1970.1.1零时开始到当前时间的毫秒数

三、

//设置日期

方法一：<主要>  
var d2=new Date("1/20/1995 12:00:00");  
d2.getTime();//1995.1.20-现在 的毫秒数

方法二：

var d3=new Date();  
d3.setFullYear("1995");  
d3.setDate("20");  
d3.setMonth("1");

四、

//BOM--Browser Object Model  
//弹出新窗口  
//top=0,left=0(不常用toolbar=no,menubar=no,scrollbars=no,resizable=no)  
window.open("images/1.jpg","新窗口","width=200,height=200");  
  
//5秒钟后弹出广告

//关于时间的两个属性：  
//setInterval()间歇调用  
//setTimeout()延时调用  
function fun1(){  
 window.open("images/1.jpg","新窗口","width=200,height=200");  
}  
setTimeout(fun1,5000);  
  
//关闭窗口  
var win=window.open("images/1.jpg","新窗口","width=200,height=200");  
function fun2(){  
 //关闭子窗口  
 win.close();  
}

<input type="button" value="关闭子窗口" onclick="fun2()">

五、

//常见日期格式：  
//(1)月/日/年  
//(2)英文 月 日，年 January 23,2016  
//(3)英文日期 英文 月 日 年 时：分：秒：时区 Wed May 25 2016 09:06:23 GMT+0700  
//(4)Iso8601扩展：YYYY-MM-DD-HH-mm-SS

var d3=new Date();  
document.write(d3.toDateString()); // Thu May 26 2016  
document.write(d3.toLocaleString()); // 2016/5/26 上午9:36:15

document.write(d3.toLocaleDateString()); // 2016/5/26  
document.write(d3.toLocaleTimeString()); // 上午9:36:46  
document.write(d3.toTimeString()); // 09:37:04 GMT+0800 (中国标准时间)  
document.write(d3.toUTCString()); // Thu, 26 May 2016 01:37:23 GMT

六、

// Date.parse()  
// var d4=new Date(Date.parse('May 23,2016'));  
// var d4=new Date('May 23,2016');//等价于  
  
// Date.UTC()  
// var d4=new Date(Date.UTC('2016,3,23'));  
// var d4=new Date('2016,3,23);//等价于  
  
// Date.now()//毫秒数  
// d1.getTime()//等价于

1. 指针转动

<script>  
function fun3(){  
 var v1=document.getElementById("id01");  
 v1.style.transform="rotate(10deg)";  
 //transformOrigin基准点设置  
 //v1.style.transformOrigin="left top";  
 v1.style.transformOrigin="40px 0";  
}

</script>

//transform-origin基准点设置属性

<style>  
 #id01{  
 width: 200px;  
 background: red;  
 height: 3px;  
 transform: rotate(10deg);  
 transform-origin:left top;

transform-origin:40px 0;  
 position: absolute;  
 top:100px;  
 left:200px;  
 }  
</style>

<div id="id01"></div>  
<input type="button" value="转" onclick="fun3()">

19.3D彩票  
//定义数组  
var arrl=[0,1,2,3,4,5,6,7,8,9];  
var str="";  
for(var i=0;i<3;i++){  
 //随机产生0-9整数，作为数组的下标值  
 var n=Math.floor(Math.random()\*9);  
 str=str+arrl[n];  
}  
document.write("3D彩票："+str);

20.旋转

<div style="width: 100px;height: 100px;background: blue;" id="id01">sssss</div>

#id01{  
 /\*css3\*/  
 transform: rotate(45deg);  
 -moz-transform: rotate(45deg);火狐  
 -o-transform: rotate(45deg);opera  
 -webkit-transform: rotate(45deg);谷歌 safir  
 -ms-transform: rotate(45deg);微软  
}

21.

<style>  
 .a1{  
 font-size: 20px;  
 }  
</style>

<script>  
 var v1=document.getElementById("id01");  
 v1.style.transform="rotate(45deg)";  
  
 var v1="A";  
 //DOM创建元素节点  
 var v\_span=document.createElement("span");/\*表现为<span></span>\*/  
 //DOM创建文本节点  
 var v\_text=document.createTextNode(v1);  
 //将文本节点追加到span元素节点中??????  
 var v\_st=v\_span.appendChild(v\_text);/\*表现为<span>A</span>\*/  
 //动态加载属性 <span class="a1">A</span>  
 v\_st.setAttribute("class","a1");  
 //显示到body里  
 document.body.appendChild(v\_st);

//DOM创建元素节点  
var v\_span= document.createElement("span");/\*<span></span>\*/

//将文本追加到span元素节点中  
v\_span.innerText="ccc";

//动态加载属性 <span class="a1">ccc</span>  
v\_span.setAttribute("class","a1");

//显示到body里  
document.body.appendChild(v\_span);  
  
 //间歇调用  
 var i=0;  
 var timer;  
 function fun1(){  
 i++;  
 document.write(i);  
 if(i==20){  
 //清除定时器  
 clearInterval(timer);  
 }  
 }  
 timer=setInterval(fun1,500)  
</script>

1. 操作符  
   一、算数操作符  
   1)+ - \* / %  
    减法中遇到非数值型,会用Number()进行类型转换

alert(24/11);//2.1818181818181817  
alert(24%11);//2  
alert(Infinity+Infinity);//Infinity  
alert(-Infinity+(-Infinity));//-Infinity  
alert(+0+(+0));//0  
alert(-0+(-0));//0  
alert(+0+(-0));//0  
alert(NaN+2);//NaN  
alert('6'+3);//63  
alert(6+3);//9

alert(5-true);//4  
alert(NaN-3);//NaN  
alert(8-"");//8  
alert(8-null);//8  
alert(8-'2');//6  
alert(Infinity-Infinity);//NaN  
alert(-Infinity-(-Infinity));//NaN  
alert(Infinity-(-Infinity));//Infinity  
alert(-Infinity-Infinity);//-Infinity  
  
2)++ --  
 a++ a=a+1 先运算，后加减  
 ++a a=a+1 先加减，后运算

var a= 1,b=2;  
//a++ a=a+1 先运算，后加减  
//++a a=a+1 先加减，后运算  
b=a++;  
alert(b);//1  
alert(a);//2  
b=++a;  
alert(b);//2  
alert(a);//2  
  
var a=8;  
b=(++a)+(a++)-(--a)+(a--)+(--a)+(a--);  
alert(a+" "+b);//6 32  
  
var m= 3,n=4;  
k=(++m)-(--n)+(n++)-(m++)-(n++)+8-(--n)+9;  
alert(m+" "+n+" "+k);//5 4 9  
  
var a= 3,b= 4,c=5;  
d=a-((b++)-(--c)-8+(--a))+a\*(b--)+(c++)-b\*((--a)+(b--));  
alert(a+" "+b+" "+c+" "+d);//1 3 5 3

a='2'; a='A'; a=true; a=false;  
a++; a++; a++; a++;  
alert(a);//3 alert(a);//NaN alert(a);//2 alert(a);//1  
  
a=1.3;   
a++;   
alert(a);//2.3   
a--;  
alert(a);//0.300000000000000000004  
  
二、算数赋值操作符  
 += -= \*= /= %=

var a=3;  
a+=4;//a=a+4  
alert(a);//7  
  
var a= 3,b=4;  
c=a\*=b+5;  
alert(a+" "+b+" "+c);//27 4 27  
c=a/=b+6\*(b+=5);  
alert(a+" "+b+" "+c);//0.05172414 9 0.05172414 (3/58=0.05172414)  
  
var a= 3,b=4;  
c=++a +(a-=b-- -3)\*--b %(a%=b++ +3)+ --a;  
alert(a+" "+b+" "+c);//2 3 6  
  
三、比较操作符  
 > < >= <= != == === !==

alert(23<4);//false

//若比较两个字符型，则按第一个字母的字符编码进行比较  
alert('23'<'4');//true 4-52 2-50  
alert('a'<'4');//false a-97 4-52  
alert('Bcd'<'ann');//true B-66 a-97  
alert('Bcd'.toLocaleLowerCase<'ann'.toLocaleLowerCase);//false b-98 a-97  
  
//如果一个操作数是数值，就将另一个操作数转换为数值，再进行比较  
alert('a'<4);//false Number('a')---NaN  
alert('23'<4);//false Number('23')---23  
alert(""<4);//true  
alert(null<4);//true  
alert(NaN<4);//false ---任何操作数与NaN比较，返回都是false  
  
alert(null==undefined);//true

//如一个是NaN，用相等操作符时，返回false  
 用不相等操作符时，返回true  
alert('NaN'==NaN);//false  
alert(NaN==NaN);//false  
alert(8==NaN);//false  
alert(NaN!=NaN);//true  
  
//如果是布尔型，自动转成数值型  
alert(false==0);//true  
alert(true==1);//true  
alert(true==2);//false  
  
//undefined，null 在比较操作之前，不做任何类型转换  
alert(undefined==0);//false  
alert(null==0);//false  
  
//字符型自动转成数值型进行比较  
alert('8'==8);//true  
  
//===恒等于（全等于），比较类型  
alert('23'==23);//true  
alert('23'===23);//false  
alert(null===undefined);//false  
  
var a=!"" ; //t  
var a=!"ye" //f  
var a=!0 //t  
var a=!23 //f  
var a=!NaN //t  
var a=!null //t  
var a=!undefined //t  
alert(Boolean(a));

1. 逻辑运算符  
    && || ！

var a=34;  
var b=5;  
if(b>5&&++a>30){  
 alert("yes");  
}else{  
 alert("no");//输出  
}  
alert(a);//34  
if(b>5&++a>30){  
 alert("yes");  
}else{  
 alert("no");//输出  
}  
alert(a);//35

//面试题：

&&：短路运算符 如果一个操作数能绝定结果，就不会再对第二操作符运算  
 &：所有操作符全判断

//特例  
alert(null&&45);//null  
alert(NaN&&45);//NaN  
alert(undefined&&45);//undefined

//&&:js会依次获取每个操作数，将他们转换成布尔值，如为“false”，则返回操作数的值,否则继续处理下一个操作数，返回最后操作数的值

var a=0 &&true;  
alert(a);//0  
var a="hello" && true && "abc";  
alert(a);//abc  
var a='23' && 'true' && 0 && false && '123';  
alert(a);//0  
var a='23' && 'true' && false && '123';  
alert(a);//false  
var a='23' && 'true' && '123';  
alert(a);//123  
  
var a=34;  
var b=5;  
if(b>3 || ++a>30){  
 alert("yes");//输出  
}else{  
 alert("no");  
}  
alert(a);//34  
if(b>3 | ++a>30){  
 alert("yes");//输出  
}else{  
 alert("no");  
}  
alert(a);//35  
  
var a='abc'||'123'; //abc  
var a=false || "" || 0; //0  
alert(a);

1. 三目（元）运算符  
   单目（一元）运算符：! ++ -- ~ ....  
   双目（二元）运算符：+ - \* / % ....  
   三目（三元）运算符：?:

var a=5;  
if(a>5){  
 alert("y");  
}else{  
 alert("n");  
}  
等价于---三目  
var c=(a>5)?alert("y"):alert("n");

六、位运算  
 & 与  
 | 或  
 ~ 非 取反~：-N-1  
 ^ 异或

>>右移  
 <<左移  
 >>>无符号右移

18 00000000 00000000 00000000 00010010  
-18 取反：11111111 11111111 11111111 11101101  
 加一：11111111 11111111 11111111 11101110  
var a=-18;  
alert(a.toString(2));//-10010  
alert(~18);//-19  
~ 00000000 00000000 00000000 00010010 //18  
----------------------------------------------  
 11111111 11111111 11111111 11101101 //取反 -19  
----------------------------------------------  
 11111111 11111111 11111111 11101100 //减一  
----------------------------------------------  
 00000000 00000000 00000000 00010011 //取反 19 加负号 为-19  
取反~：-N-1

&:都为1，结果为1  
alert(25&3);//1  
 00011001  
& 00000011  
-------------  
 00000001  
alert(-25&3);//3  
 11100111  
& 00000011  
---------------  
 00000011  
alert(-25&-3);//-27  
 11100111  
& 11111101  
---------------  
 11100101 11100100 00011011 27 加负号 为-27  
  
|:只要有一，结果为一  
alert(23|3);//23  
 00010111  
| 00000011  
---------------  
 00010111  
  
**^** :相同为0，不同为一  
alert(23^3);//23  
 00010111  
^ 00000011  
---------------  
 00010100  
  
<<:  
alert(2<<5);//64 2左移5位，空位0填充  
00000000 00000010-->00000000 01000000  
alert(-2<<5);//-64  
11111111 11111110-->11111111 11000000 --- -64  
alert(9<<2);//36  
00000000 00001001-->00000000 00100100  
alert(-9<<2);//-36  
11111111 11110111-->11111111 11011100 --- -36  
  
>>:有符号右移>> 用符号位（0、1）来填充所有空位  
 无符号右移>>> 用0来填充所有空位 必须用32位算  
alert(9>>3);//1  
00001001-->00000001  
alert(-9>>3);//-2  
11110111-->11111110  
alert(9>>>4);//0  
00000000 00000000 00000000 00001001  
00000000 00000000 00000000 00000000  
alert(-9>>>4);//268435455  
11111111 11111111 11111111 11110111 --- -9  
000011111111 11111111 11111111 1111 --- 268435455

//面试题 -13>>2 13>>>2  
alert(-13>>2);//-4  
11110011-->11111100 11111011 00000100  
alert(-13>>>2);//1073741820  
11111111 11111111 11111111 11110011  
0011111111 11111111 11111111 111100

23.

var a="A";  
alert(a.charCodeAt()); //65  
alert(String.fromCharCode(65)); //A //解码

1. 加密与解密 （对中文有效）  
   var v1='a';  
   var key='8';  
   //加密 （用 异或 方法）  
   var v2=v1.charCodeAt()^key.charCodeAt();  
   var v3=String.fromCharCode(v2);  
   alert(v3);//Y  
   //解密  
   var v4=v3.charCodeAt()^key.charCodeAt();  
   var v5=String.fromCharCode(v4);  
   alert(v5);//a

//练习：8个交通灯对应8个车道，要求测试1、3、5、7是否开通（0-红灯 1-绿灯）  
var a=255; //11111111  
if(a&85==85){ //1010101  
 alert("1、3、5、7开通");  
}

25.数组

（1）数组定义  
一、构造函数实现方式

var arr1=new Array();//空数组  
arr1[0]='a';//数组赋值  
arr1[9]='b';  
alert(arr1[0]);  
alert(arr1.length); //10  
for(var i=0;i<arr1.length;i++){  
 alert(arr1[i]);//未赋值的数组，输出undefined   
}

var arr1=new Array(5);//指定数组长度  
arr1[0]='a';  
arr1[9]='b';  
alert(arr1.length);//10 与内容有关 与指定长度无关  
for(var i=0;i<arr1.length;i++){  
 document.write(arr1[i]);  
}

//数组循环 for...in... 只输出实际定义的内容

for(var v1 in arr1){  
 document.write(arr1[v1]);  
}

二、直接量实现

var arr2=[12,12,34,'aa',true];//数组中可以为任意类型  
document.write(arr2);  
  
var arr2=[];  
arr2[2]='a'//空数组赋值  
  
var arr2=['a','b',];//ie8及以下a\b\undefined arr2.length--3  
var arr3=[,,,,,,];//ie8及以下七个undefined ie8以上六个undefined  
for(var i=0;i<arr2.length;i++){  
 document.write(arr2[i]);  
}

（2）数组的添加与删除  
//push:在数组尾部增加数据，返回数组长度  
//pop:删除数组尾部一个数据，返回被删除的元素,如删除数组为空，返回undefined  
//shift:删除数组头部一个数据，返回被删除的元素,如删除数组为空，返回undefined  
//unshift:在数组头部增加数据，返回数组长度

var arr1=['a','b'];  
var v1=arr1.push("d");  
alert(arr1);  
alert(v1);//3  
  
var v1=arr1.pop();  
alert(arr1);  
alert(v1);//b  
  
var v1=arr1.shift();  
alert(arr1);  
alert(v1);//a  
  
var v1=arr1.unshift("d");  
alert(arr1);  
alert(v1);//3

（3）例：创建ul列表，添加数组值到li中，并添加ul li样式

<style>  
 .my\_ul{  
 list-style: none;  
 }  
 .my\_ul li{  
 color:green;  
 font-size: 30px;  
 }  
</style>

<script>  
 var arr1=['aaa','bbb'];  
 var v\_ul=document.createElement("ul");  
 for(var i=0;i<arr1.length;i++){  
 var v\_li=document.createElement("li");  
 v\_li.innerHTML="<span style='color:red;'>"+arr1[i]+"</span>";

//<li>aaa</li>  
 v\_ul.appendChild(v\_li);  
 }  
 v\_ul.setAttribute("class","my\_ul");  
 document.body.appendChild(v\_ul);  
</script>

26.<!--多维数组-->

var arr1=[1,2,3];//一维数组  
var arr2=[[1,2],[3,4],[23,56]];//多维数组  
var arr3=[[2,3],'a'['b',[3,4,[7,8]]]];//语法支持

27.数组属性 sort()

//sort--对数组元素进行排序  
//对字母排序是根据字母在编码表中的顺序进行的  
//sort方法中可以放“比较函数”做参数  
//比较函数,两个参数，a，b  
 该函数只判断返回值：如返回值>0 a在b前 ab  
 返回值<0 a在b后 ba  
 返回值=0

var arr1=[23,45,12,2,8,56];  
alert(arr1.sort());  
var arr2=['m2','peter',34,'Mart','cab','Ass',13,12,7];  
alert(arr2.sort());

//特例：

//对数值排序

var arr1=[23,45,12,2,8,56];  
function f1(a,b){  
 var c=a-b;  
 return c;  
}  
arr1.sort(f1);  
alert(arr1);

//对数值排序--奇偶排序

var arr1=[23,2,12,45,8,56];  
function f1(a,b){  
 var c=b%2-a%2;  
 return c;  
}  
arr1.sort(f1);  
alert(arr1);

//对数值排序--质数在前

var arr1=[23,2,12,3,5,45,8,56];  
function f1(a,b){  
 for(var i=2;i<a;i++){  
 if(a%i==0){  
 return 1;  
 }  
 }  
 return -1;  
}  
arr1.sort(f1);  
alert(arr1);

//不区分大小写排序---降序

var arr1=['m2','peter','Mart','cab','Ass'];  
function f1(a,b){  
 var a1= a.toLocaleLowerCase();  
 var b1=b.toLocaleLowerCase();  
 if(a1>b1){  
 return -1;  
 }else{  
 return 1;  
 }  
}  
arr1.sort(f1);//sort修改原数组  
alert(arr1);

//中文排序

var v1=["达到","啊啊","方法","版本"];  
function fun1(a,b){  
 return a.localeCompare(b);//按拼音  
}  
alert(v1.sort(fun1));

1）reverse() --反转 直接修改原数组

var arr1=[23,2,12,3,5,45,8,56];  
arr1.reverse();  
alert(arr1.toString());

2）join() --将所有数组元素的字符串转换连接成一个字符串  
var arr1=["aa","bb","cc","dd"];  
document.write(arr1.join() +"<br>");  
document.write(arr1.join("+"));

3）concat() --连接两个数组并返回一个新的数组。  
var arr1=["aa","bb","cc"];  
var arr2=[11,22,33];  
document.write(arr1.concat(arr2));

4）slice() 提取一段数组并返回一个新的数组。  
var arr = ["George","John","Thomas"];  
document.write(arr + "<br>");  
document.write(arr.slice(0,1) + "<br>");  
document.write(arr.slice(1));

5）splice() 改变数组的内容,添加新的元素,删除旧的元素。  
var arr = ["George","John","Thomas","James","Adrew","Martin"];  
document.write(arr + "<br>");  
arr.splice(2,0,"William");//在位置2上添加1个元素  
document.write(arr + "<br>");  
arr.splice(2,1,"Jane");//在位置2上删除1个元素  
document.write(arr + "<br>");

28.

//页面加载 <body onload="fun1()">

var arr1=['a','b','c'];  
var arr2=[12,23,45];  
function fun1(){  
 var v\_sel=document.getElementById("sel01");  
 for(var i=0;i<arr1.length;i++){  
 var v\_o=document.createElement("option");//<option></option>  
 v\_o.innerText=arr1[i];  
 v\_sel.appendChild(v\_o);  
 }  
}  
//下拉列表切换函数  
function fun2(){  
 var v1=document.getElementById("sel01").value;  
 //根据数组值获取数组中的下标值（索引值）  
 var v\_in=arr1.indexOf(v1);  
 document.getElementById("show").innerText=arr2[v\_in];  
}

<body onload="fun1()">  
<select id="sel01" onchange="fun2()">  
 <option>请选择</option>  
</select>  
<div id="show"></div>  
</body>

29.//省市级联

<script>  
var pro\_arr=["黑龙江","河北"];  
var city\_arr=[['哈尔滨','大庆'],['石家庄','张家口']];  
function fun1(){  
 var v\_sel=document.getElementById("province");  
 for(var i=0;i<pro\_arr.length;i++){  
 var v\_o=document.createElement("option");  
 v\_o.innerText=pro\_arr[i];  
 v\_sel.appendChild(v\_o);  
 }  
}  
function fun2(){  
 var v\_sel=document.getElementById("city");  
 v\_sel.innerText="";  
 for(var i=0;i<city\_arr.length;i++){  
 var v\_o=document.createElement("option");  
 var v1=document.getElementById("province").value;  
 var v\_in=pro\_arr.indexOf(v1);  
 v\_o.innerText=city\_arr[v\_in][i];  
 v\_sel.appendChild(v\_o);  
 }  
}

</script>  
<body onload="fun1()">  
<select id="province" onchange="fun2()">  
 <option>请选择</option>  
</select>  
<select id="city">  
 <option>请选择</option>  
</select>  
</body>

30.//原型封装--按索引值删除数组元素--数组原型 prototype  
Array.prototype.mydel=function(arg1){  
 //this---当前对象  
 if(arg1>=this.length||arg1<0){  
 alert("输入错误!");  
 }else{  
 this.splice(arg1,1);//删除操作  
 }  
};  
var arr1=['a','b','c','d'];  
arr1.mydel(2);  
arr1.mydel(-1);  
alert(arr1);

//原型封装--删除数组中的重复元素  
Array.prototype.delcf=function(){  
 for(var i=0;i<this.length-1;i++){  
 for(var j=i+1;j<this.length;j++){  
 if(this[i]==this[j]){  
 this.splice(j,1);  
 j--;  
 }  
 }  
 }  
};

var arr1=['a','b','c','d','a','e'];  
arr1.delcf();  
alert(arr1);

31.//数组检测  
var arr2=['a','b'];  
var v1='ssss';  
//方法一：  
if(arr2 instanceof Array){  
 alert("是数组");  
}else{  
 alert("不是数组");  
}  
//方法二：ECMAScript 5 新增  
if(Array.isArray(v1)){  
 alert("是数组");  
}else{  
 alert("不是数组");  
}

32.迭代方法： ECMAScript 5 新增  
1)every():对数组中的每一项运行给定的函数,如果该函数对每一项都返回true，才返回true  
(item:数组项的值 index:该数组项在数组中的位置（索引值） array:数组对象本身)  
var arr2=[30,50,23,12];  
var v=arr2.every(function(item,index,array){  
 return item>10;//如果数组中的每一项都大于时，返回true  
});  
alert(v);

2)some():对数组中的每一项执行测试函数，直到获得返回 true 的项。 使用此方法确定数组中的所有项是否满足条件.类似于||的含义  
var arr2=[3,5,2,12];  
var v=arr2.some(function(item,index,array){  
 return item>10;//如果数组中只要有一项都大于时，返回true  
});  
alert(v);

3)filter():对数组中的每一项执行测试函数，并构造一个新数组，返回 true的项被添加进新数组。 如果某项返回 false，则新数组中将不包含此项  
var arr2=[3,5,23,12];  
var v=arr2.filter(function(item,index,array){  
 return item>10;//输出符合要求的项  
});  
alert(v);

4)map():对数组中的每一项执行函数并构造一个新数组，并将原始数组中的每一项的函数结添加进新数组。  
var arr2=['a','b','c','d'];  
var v=arr2.map(function(item,index,array){  
 return item.toUpperCase();  
});  
alert(v);

5)forEach():对数组中的每一项执行函数,不返回值  
var arr = [3,4,5,6,7,"a"];  
var res = arr.forEach(function(elem,index){  
 console.log(index+"."+elem);  
});  
console.log(res);

1. 归并方法 reduce() reduceRight()  
   1）reduce():从第一项开始逐个遍历到最后。  
   2）reduceRight():从数组的最后一项开始，遍历到数组的第一项。  
   //这个函数返回的任何值斗殴会作为第一个参数自动传给下一项。第一次迭代发生在数组的第二项上，因此第一个参数是数组的第一项，第二个参数是数组的第二项,和 作为归并基础的初始值。  
   var values = [1, 2, 3, 4, 5];  
   var sum = values.reduce(function (prev, cur, index, array) {  
    return prev + cur;  
   });  
   alert(sum);//15  
   //结果一样，只是方向相反而已  
   var sum2=values.reduceRight(function (prev,cur,index,array) {  
    return prev+cur;  
   });  
   alert(sum2);//15

34.