**六、javascript字符串日期对象**

**一、Math对象**

1.Math.round(3.6) //四舍五入

2.Math.random() //返回0-1之间的随机数

3.Math.max(num1, num2) //返回较大的数

4.Math.min(num1, num2) //返回较小的数

5.Math.abs(num) //绝对值

6.Math.ceil(19.3) //向上取整

7.Math.floor(11.8) //向下取整

8.Math.pow(x,y) //x的y次方

9.Math.sqrt(num) //开平方

**二、对象**

**1.认识**

(1)对象是一种数据类型。

(2)是引用数据类型/复合数据类型。

(3)将数据和功能组织在一起

**2.数据结构学习历程：**

基本数据类型(存储单数数据) -> 数组(存储批量数据) -> 对象(既存储了数据，又存储了函数)

【好处】我们可以使用生活逻辑编写代码，可以通过代码解决生活中的一些问题。

//声明一个人

var person = {

name: "tianyufei",

age: 18,

showName: function(){

alert("我的名字叫做:" + person.name);

}

}

**3.对象的声明方式**

三种

//1、通过运算符创建对象

var obj = new Object();

(1)//添加属性 （属性/变量）

obj.name = "tianyufei";

obj.age = 18;

(2)//添加方法

obj.showName = function(){

alert(obj.name);

}

(3)//访问属性

// alert(obj.name);

(4)//调用函数

//obj.showName();

//2、new运算符可以省略

var obj = Object();

//3、使用常量/json声明对象（最常用的）

var obj = {

name: "tianyufei",

age: 18,

showName: function(){

alert(obj.name);

}

}

//【注】删除对象的属性delete

alert(obj.name);

delete obj.name;

alert(obj.name); //undefined

属性/方法 进行 赋值/取值

对象名.属性/方法

对象名["属性"/"方法"]

//obj.sex 等价

obj["sex"] = "男";

alert(obj["sex"]);

//【注】所有的复合数据类型都是对象，数组也是对象，数组有很多属性/方法。

**4.日期对象**

(1)获取时间（new不可省略）

var d = new Date();

alert(d);

//Fri Aug 18 2017 16:12:10 GMT+0800 (CST)

//格林尼治时间 格林尼治天文台 子午线

//创建日期对象

//1、2015/08/22

/\*var d = new Date("2015/08/22");

alert(d);\*///Sat Aug 22 2015 00:00:00 GMT+0800

//2、2015-08-22

/\*var d = new Date("2015-08-22");

alert(d);\*/

//3、2016,04,13,14,34

/\*var d = new Date(2016,04,13,14,34);

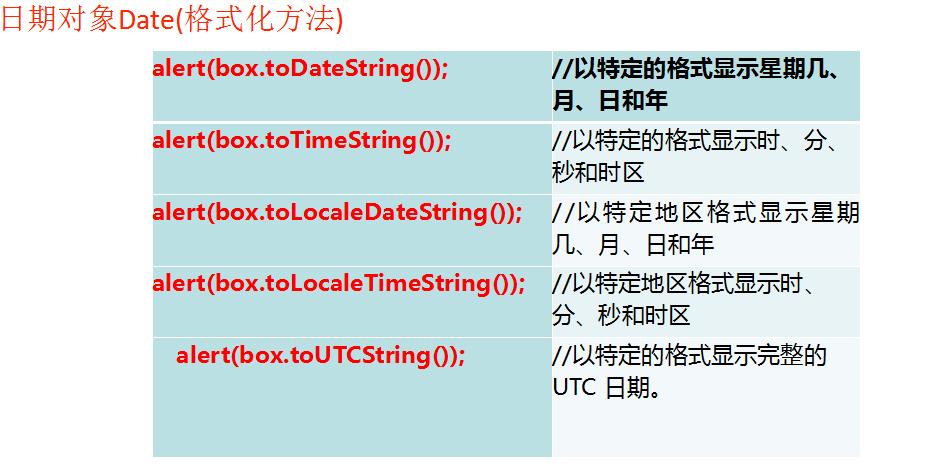
alert(d);\*///Fri May 13 2016 14:34:00 GMT+0800

//4、传毫秒数 会根据毫秒数，以1970年1月1日0时为参照物，计算时间

var d = new Date(1000);

alert(d);

**5.日期对象方法**



【例】(1)Date.parse(“2015-08-24”);

//转换格式默认支持2015-08-24或2015/08/24

//返回距离1970年1月1日0时的毫秒数

(2)d.getTime()/d.setTime()

getTime获取某个日期自1970年以来的毫秒数

setTime修改日期的毫秒数，对应的日期时间也会修改

**日期对象：**

set/getDate() 从Date对象中返回一个月中的某一天(1~31)

getDay() 从Date对象返回一周中的某一天(0~6)

set/getMonth() 从Date对象中返回月份(0~11)

set/getFullYear() 从Date对象以四位数返回年份

set/getHours() 返回Date对象的小时(0~23)

set/getMinutes() 返回Date对象的分钟(0~59)

set/getSeconds() 返回Date对象的秒数(0~59)

set/getMilliseconds()返回Date对象的毫秒

set/getTime() 返回1970年1月1日至今的毫秒数

getTimezoneOffset() 返回本地时间与格林威治标准时间(GMT)的分钟差

set/getUTCDate() 根据世界时从Date对象返回月中的一天(1~31)

getUTCDay() 根据世界时从Date对象返回周中的一天(0~6)

set/getUTCMonth() 根据世界时从Date对象返回月份(0~11)

set/getUTCFullYear()根据世界时从Date对象返回四位年份

set/getUTCHours() 根据世界时返回Date对象的小时(0~23)

set/getUTCMinutes()根据世界时返回Date对象的分钟(0~59)

set/getUTCSeconds()根据世界时返回Date对象的秒数(0~59)

set/getUTCMilliseconds() 根据世界时返回Date对象的毫秒(0~999)

oString() 把Date对象转换为字符串

toTimeString() 把Date对象的时间部分转换为字符串

toDateString() 把Date对象的日期部分转换成字符串

toUTCString() 根据世界时,把Date对象转换为字符串

toLocaleString() 根据本地时间格式,把Date转换为字符串

toLocaleTimeString() 根据本地时间格式，把Date时间部分转换为字符串

toLocaleDateString() 根据本地时间格式,把Date对象的日期部分转换为字符串

UTC() 根据世界时返回1970年1月1日到指定日期的毫秒数

valueOf() 返回Date对象的原始值

**【注意】：**

**1、月份是从0-11，所以取得的值要加1才是当前月份**

**2、星期天是从0-6,0代表星期天**

**6.innerHTML**

<style>

#div1{width: 100px; height: 100px; background-color: red}

</style>

<script>

window.onload = function(){

/\*

innerHTML

【注】标签间的所有内容。

【注】如果设置的文本中含有标签，那么会将标签进行解析。

\*/

var oDiv = document.getElementById("div1")

oDiv.onclick = function(){

alert(oDiv.innerHTML);

oDiv.innerHTML = "<h1>修改后的文本</h1>";

}

}

</script>

</head>

<body>

<div id = "div1"><em>原来的文本</em></div>

</body>

**7.定时器**

(1)setInterval(函数, 毫秒数);

【功能】每隔对应毫秒数去执行一次函数。

【返回值】当前定时器在页面上的唯一标识，当前定时器的id。

(2)取消定时器

clearInterval(timer);

var i = 0;

var timer = setInterval(function(){

document.title = i++;

}, 1000);

window.onload= function(){

var oBtn = document.getElementById("btn");

oBtn.onclick = function(){

clearInterval(timer);

}

}

</script>

</head>

<body>

<input type = "button" value = "按钮" id = "btn"/>

</body>

**8.秒表**

(1)开始、暂停、停止三键秒表

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<style>

.watch{

width:400px;

height:500px;

margin:50px auto 0;

background:gray;

text-align:center;

}

#show{

width:200px;

height:150px;

text-align:center;

font-size:35px;

margin:30px 0;

}

.btn{

width:200px;

height:50px;

background:red;

font-size:20px;

font-weight:900;

margin:10px 0;

text-align:center;

}

</style>

<script>

var int;

var i = 0;

window.onload = function(){

function time(){

i++;

var hour = parseInt(i / 36000) % 24;

var minute = parseInt(i / 600) % 60;

var second = parseInt(i / 10) % 60;

var millisecond = parseInt(i % 10);

$("show").value = show(hour) + ":" + show(minute) + ":" + show(second) + ":" + millisecond;

}

function show(i){

if(i < 10){

return "0" + i;

}else{

return i;

}

}

begin.onclick = function(){

int = setInterval(time, 100);

}

reset.onclick = function(){

clearInterval(int);

i = 0;

$("show").value = "00:00:00:0";

}

pause.onclick = function(){

clearInterval(int);

}

}

function $(id){

return document.getElementById(id);

}

</script>

</head>

<body>

<div class = "watch">

<input id = "show" type="text" value = "00:00:00:0" />

<input class = "btn" type="button" id = "begin" value = "开始" />

<input class = "btn" type="button" id = "pause" value = "暂停" />

<input class = "btn" type="button" id = "reset" value = "停止" />

</div>

</body>

</html>

(2)开始/暂停、停止二键秒表

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<style>

.watch{

width:400px;

height:500px;

margin:50px auto 0;

background:gray;

text-align:center;

}

#show{

width:200px;

height:150px;

text-align:center;

font-size:35px;

margin:30px 0;

}

.btn{

width:200px;

height:50px;

background:red;

font-size:20px;

font-weight:900;

margin:10px 0;

text-align:center;

}

</style>

<script>

var int;

var i = 0;

window.onload = function(){

function time(){

i++;

var hour = parseInt(i / 36000) % 24;

var minute = parseInt(i / 600) % 60;

var second = parseInt(i / 10) % 60;

var millisecond = parseInt(i % 10);

$("show").value = show(hour) + ":" + show(minute) + ":" + show(second) + ":" +

millisecond;

}

function show(i){

if(i < 10){

return "0" + i;

}else{

return i;

}

}

var isStart = false;

begin.onclick = function(){

if(isStart){

$("begin").value = "开始";

clearInterval(int);

}else{

$("begin").value = "暂停";

int = setInterval(time, 100);

}

isStart = !isStart;

}

reset.onclick = function(){

clearInterval(int);

i = 0;

$("show").value = "00:00:00:0";

}

}

function $(id){

return document.getElementById(id);

}

</script>

</head>

<body>

<div class = "watch">

<input id = "show" type="text" value = "00:00:00:0" />

<input class = "btn" type="button" id = "begin" value = "开始" />

<input class = "btn" type="button" id = "reset" value = "停止" />

</div>

</body>

</html>

**三、字符串**

**1.字符串声明**

(1)①通过new运算符创建

var str1 = new String(10);

//alert(typeof str);

②省略new运算符创建字符串

var str2 = String(100);

③直接通过常量证明字符串

var str3 = "hello world";

//alert(typeof str1 + ", " + typeof str2 + ", " + typeof str3);

(2)复合数据类型 => 对象 => 属性和方法

①字符串的属性:

length

【注】代表字符串的长度，字符串中字符的个数。

②【特点】字符串一旦被创建就没有办法被修改。

如果要创建新的字符串，那么必须销毁原来的字符串，生成新的字符串。

/\*var str = "hello world";

str = "hxllo world"

alert(str);\*/

③字符串的方法

【注】必须在document.write()中使用

【格式】字符串.函数()

document.write("hello".sub() + "world".sup());

**2.字符串函数**

(1)charAt(index)

【格式】字符串.charAt(index);

【作用】访问对应下标的字符

【注】字符可以通过下标去访问。

/\*var str = "hello world";

alert(str.charAt(1));

alert(str[1]);\*/

(2)charCodeAt()

【格式】字符串.charCodeAt(index);

【作用】访问对应下标的字符的ASCII码值。

/\*var str = "hello world";

alert(str.charCodeAt(1));\*/

(3)concat()

【格式】字符串1.concat(字符串2)

【作用】链接字符串

/\*var str1 = "hello";

var str2 = " world";

// var str = str1.concat(str2);

var str = str1 + str2;

alert(str);\*/

(4)String.fromCharCode();

【格式】String.fromCharCode(ASCII码值...);

【作用】将对应的ASIIC码值转成字符串。

【返回值】组成的字符串

/\*var str = String.fromCharCode(97, 98, 99);

alert(str); //abc\*/

**【例】随机输出字母与数字混合的四位验证码**

<script>

/\*

A-Z 65-90

a-z 97-122

0-9 0-9

10-35 + 87

随机0~100的数

i j k l

\*/

function authCode(n){

var arr = [];

for(var i = 0; i < n; i++){

var num = parseInt(Math.random() \* 100);

if(num >= 0 && num < 10){

arr.push(num);

}else if(num >= 10 && num <= 35){

arr.push(String.fromCharCode(num + 87));

}else if(num >= 65 && num <= 90){

arr.push(String.fromCharCode(num));

}else{

//无用数

i--;

continue;

}

}

return arr.join("");

}

window.onload = function(){

var oBtn = document.getElementById("btn\_id");

oBtn.value = authCode(4);

oBtn.onclick = function(){

oBtn.value = authCode(4);

}

}

</script>

</head>

<body>

<input type = "button" id = "btn\_id" />

</body>

(5)toLowerCase()方法用于把字符串转换成小写

(6)toUpperCase()方法用于把字符串转换成大写

var str = "hello WORLD";

alert(str.toLowerCase());

alert(str.toUpperCase());

(7)indexOf()

【格式】父字符串.indexOf(子字符串, 开始查找的下标位置);

【功能】在父字符串中查找子字符串第一次出现的位置。

【返回值】如果查找到，将子字符串出现的下标位置返回，否则返回-1。

/\*var supStr = "abcabcabc";

var subStr = "abcd";

var index = supStr.indexOf(subStr, 1); //默认从0开始查找

alert(index);\*/

(8)lastIndexOf() 查找字符串最后一次出现的位置。

【格式】父字符串.lastIndexOf(子字符串)

【返回值】如果查找到，将子字符串出现的下标位置返回，否则返回-1。

/\*var supStr = "abcabcabc";

var subStr = "abc";

alert(supStr.lastIndexOf(subStr));\*/

(9)search()

【格式】父字符串.search(字符串/正则表达式);

【返回值】如果查找到，将子字符串出现的下标位置返回，否则返回-1。

\*/

var supStr = "Abcabcabc";

var subStr = "abc"

// alert(supStr.search(subStr)); //3

alert(supStr.search(/abc/i)); //0

/\*

"abc"

【正则表达式】/abc/ i 忽略大小写 g全局匹配

**3.replace**

replace

【格式】原字符串.replace(替换的子串, 替换成的字符串);

【功能】字符串替换

【返回值】生成一个替换成功后的新字符串。原字符串不受影响。

var str = "How are Are you!";

/\*var newStr = str.replace("are", "old are");

alert(newStr);

alert(str);

\*/

var newStr = str.replace(/are/ig, "old are");

alert(newStr);

**4.substring**

substring

【格式】字符串.substring(start, end);

【功能】在指定范围内提取子串，并生成新的字符串并返回。

【返回值】新的字符串

/\*var str = "helloworld";

var subStr = str.substring(2, 5);

alert(subStr);\*/

**5.split**

split();

【格式】字符串.split(分割符);

【功能】字符串分割

【返回值】字符串使用分割符分割成的子串所组成的数组。

字符串到数组：split

数组到字符串：join 经常配合使用

/\*var str = "how \*are you";

var arr = str.split(" \*");

alert(arr);\*/

//【注】如果，分隔符是空字符串，会将所有的字符单个分离，存储在数组中。

var str = "helloworld";

var arr = str.split("");

alert(arr); //h,e,l,l,o,w,o,r,l,d

**6.敏感词过滤**

敏感词过滤

(1)将所有的敏感词放在数组里

var arr = [/TMD/ig, /XXX/ig];

var str = "我TMDsdadaddada，XXX大大大啊XXX";

(2)通过循环去完成

for(var i = 0; i < arr.length; i++){

str = str.replace(arr[i], "\*");

}

alert(str);