js笔记

一、js简介

1、js是什么

2、js作用

3、组成

4、引入方式

二、基本语法

1、变量

2、原始数据类型

3、引入数据类型

4、运算符

5、逻辑语句

三、js内建对象

(1)Number

创建方式：

```
var myNum=new Number(value);
var myNum=Number(value);
```

属性和方法：

toString():转成字符串

valueOf()：返回一个 Number 对象的基本数字值

(2)Boolean

创建方式：

```
var bool = new Boolean(value);
var bool = Boolean(value);
```

属性和方法：

toString():转成字符串

valueOf()：返回一个 Boolean 对象的基本值(boolean)

(3)String

创建方式：

```
var str = new String(s);
```

var str = String(s);

属性和方法：

length:字符串的长度

charAt():返回索引字符

charCodeAt:返回索引字符unicode

indexOf():返回字符的索引

lastIndexOf();逆向返回字符的索引

split();将字符串按照特殊字符切割成数组

substr():从起始索引号提取字符串中指定数目的字符

substring():提取字符串中两个指定的索引号之间的字符

toUpperCase();转大写

示例：

(4)Array

创建方式：

var arr = new Array();//空数组

var arr = new Array(size);//创建一个指定长度的数据

var arr = new Array(element0, element1, ..., elementn);//创建数组直接实例化元素

var arr = [];//空数组

var arr = [1,2,5,"java"];//创建数组直接实例化元素

属性和方法：

length:数组长度

join()：把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔一个

pop():删除并返回最后元素

push()：向数组的末尾添加一个或更多元素，并返回新的长度

reverse();反转数组

sort();排序

(5)Date

创建方式：

var myDate = new Date();

var myDate = new Date(毫秒值);//代表从1970-1-1到现在的一个毫秒值

属性和方法

getFullYear():年

getMonth():月 0-11

getDate():日 1-31

getDay()：星期 0-6

getTime():返回1970年1月1日午夜到指定日期（字符串）的毫秒数

toLocalString();获得本地时间格式的字符串

(6)Math

创建方式：

Math 对象并不像 Date 和 String 那样是对象的类，因此没有构造函数 Math()，像 Math.sin() 这样的函数只是函数，

不是某个对象的方法。您无需创建它，通过把 Math 作为对象使用就可以调用其所有属性和方法。

属性和方法

PI：圆周率

abs():绝对值

ceil():对数进行上舍入

floor():对数进行下舍入

pow(x,y)：返回 x 的 y 次幂

random():0-1之间的随机数

round():四舍五入

(7)RegExp

创建方式：

```
var reg = new RegExp(pattern);
var reg = /^正则规则$/;
```

规则的写法：

[0-9]

[A-Z]

[a-z]

[A-z]

\d 代表数据

\D　　非数字

\w　　查找单词字符

\W　　查找非单词字符

\s　　查找空白字符

\S　　查找非空白字符

n+　　出现至少一次

n*　　出现0次或多次

n?　　出现0次或1次

{5} 出现5

{2,8} 2到8次

方法：

test(str):检索字符串中指定的值。返回 true 或 false

需求：

校验邮箱：

```
var email = haohao_827@163.com
var reg = /^[A-z]+[A-z0-9_-]*\@[A-z0-9]+\.[A-z]+$/;
reg.test(email);
```

四、js的函数

1、js函数定义的方式

(1)普通方式

语法：function 函数名(参数列表){函数体}

示例：

```
function method(){
        alert("xxx");
}
method();
```

(2)匿名函数

语法：function(参数列表){函数体}

示例：

```
var method = function(){
        alert("yyy");
};
method();
```

(3)对象函数

语法：new Function(参数1,参数2,...,函数体);

注意：参数名称必须使用字符串形式、最后一个默认是函数体且函数体需要字符串形式

示例：

```
var fn = new Function("a","b","alert(a+b)");
fn(2,5);
```

2、函数的参数

(1)形参没有var去修饰

(2)形参和实参个数不一定相等

(3)arguments对象 是个数组 会将传递的实参进行封装

```
function fn(a,b,c){
        //var sum = a+b+c;
        //alert(sum);
        //arguments是个数组 会将传递的实参进行封装
        for(var i=0;i<arguments.length;i++){
                alert(arguments[i]);
        }
}
fn(1,2,4,8);
```

3、返回值

(1)在定义函数的时候不必表明是否具有返回值

(2)返回值仅仅通过return关键字就可以了 return后的代码不执行

```
function fn(a,b){
        return a+b;
        //alert("xxxx");
}
alert(fn(2,3));
```

4、js的全局函数

(1)编码和解码

encodeURI() decodeURI()

encodeURIComponet() decodeURIComponent()

escape() unescape()

三者区别：

进行编码的符号范围不同吧，实际开发中常使用第一种

(2)强制转换

Number()

String()

Boolean()

(3)转成数字

parseInt()

parseFloat()

(4)eval()方法

将字符串当作脚本进行解析运行

```
//var str = "var a=2;var b=3;alert(a+b)";
//eval(str);
function print(str){
        eval(str);
}
print("自定义逻辑");
```

五、js的事件

事件

事件源

响应行为

1、js的常用事件

onclick:点击事件

onchange:域内容被改变的事件

需求：实现二级联动

```
<select id="city">
        <option value="bj">北京</option>
        <option value="tj">天津</option>
        <option value="sh">上海</option>
</select>
<select id="area">
        <option>海淀</option>
        <option>朝阳</option>
        <option>东城</option>
</select>
<script type="text/javascript">
        var select = document.getElementById("city");
        select.onchange = function(){
```

```
                                        var optionVal = select.value;
                                        switch(optionVal){
                                                case 'bj':
                                                        var area =
document.getElementById("area");
                                                        area.innerHTML = "
<option>海淀</option><option>朝阳</option><option>东城</option>";
                                                        break;
                                                case 'tj':
                                                        var area =
document.getElementById("area");
                                                        area.innerHTML = "
<option>南开</option><option>西青</option><option>河西</option>";
                                                        break;
                                                case 'sh':
                                                        var area =
document.getElementById("area");
                                                        area.innerHTML = "
<option>浦东</option><option>杨浦</option>";
                                                        break;
                                                default:
                                                        alert("error");
                                        }
                                };

                        </script>
```

onfoucus:获得焦点的事件

onblur:失去焦点的事件

需求：　当输入框获得焦点的时候，提示输入的内容格式

当输入框失去焦点的时候，提示输入有误

```
                        <label for="txt">name</label>
                        <input id="txt" type="text" /><span id="action"></span>
                        <script type="text/javascript">
                                var txt = document.getElementById("txt");
                                txt.onfocus = function(){
```

```
                                        //友好提示
                                        var span =
document.getElementById("action");
                                        span.innerHTML = "用户名格式最小8位";
                                        span.style.color = "green";
                                };
                                txt.onblur = function(){
                                        //错误提示
                                        var span =
document.getElementById("action");
                                        span.innerHTML = "对不起 格式不正确";
                                        span.style.color = "red";
                                };
                        </script>
```

onmouseover:鼠标悬浮的事件

onmouseout:鼠标离开的事件

需求：div元素 鼠标移入变为绿色 移出恢复原色

```
#d1{background-color: red;width:200px;height: 200px;}
<div id="d1"></div>
<script type="text/javascript">
        var div = document.getElementById("d1");
        div.onmouseover = function(){
                this.style.backgroundColor = "green";
        };
        div.onmouseout = function(){
                this.style.backgroundColor = "red";
        };
</script>
```

onload:加载完毕的事件

等到页面加载完毕在执行onload事件所指向的函数

```
<span id="span"></span>
<script type="text/javascript">
```

```
window.onload = function(){
        var span =
document.getElementById("span");
        alert(span);
        span.innerHTML = "hello js";
};
</script>
```

2、事件的绑定方式

(1)将事件和响应行为都内嵌到html标签中

```
<input type="button" value="button"
onclick="alert('xxx')"/>
```

(2)将事件内嵌到html中而响应行为用函数进行封装

```
<input type="button" value="button" onclick="fn()" />
<script type="text/javascript">
        function fn(){
                alert("yyy");
        }
</script>
```

(3)将事件和响应行为 与html标签完全分离

```
<input id="btn" type="button" value="button"/>
<script type="text/javascript">
        var btn = document.getElementById("btn");
        btn.onclick = function(){
                alert("zzz");
        };
</script>
```

****this关键字

this经过事件的函数进行传递的是html标签对象

```
<input id="btn" name="mybtn" type="button"
value="button123" onclick="fn(this)"/>
<script type="text/javascript">
        function fn(obj){
                alert(obj.name);
```

```
                }
            </script>
```

3、阻止事件的默认行为

```
IE：window.event.returnValue = false;
W3c：传递过来的事件对象.preventDefault();
//ie：window.event.returnValue = false;
//W3c：传递过来的事件对象.preventDefault();
//W3c标准
if(e&&e.preventDefault){
        alert("w3c");
        e.preventDefault();
//IE标签
}else{
        alert("ie");
        window.event.returnValue = false;
}


//通过事件返回false也可以阻止事件的默认行为
<a href="demo11.html" onclick="return false">点击我吧</a>
```

4、阻止事件的传播

```
IE：window.event.cancelBubble = true;
W3c：传递过来的事件对象.stopPropagation();
if(e&&e.stopPropagation){
        alert("w3c");
        e.stopPropagation();
//IE标签
}else{
        alert("ie");
        window.event.cancelBubble = true;
}
```

六、js的bom

(1)window对象

弹框的方法：

提示框：alert("提示信息");

确认框：confirm("确认信息");

有返回值：如果点击确认返回true 如果点击取消 返回false

```
var res = confirm("您确认要删除吗？");
alert(res);
```

输入框：prompt("提示信息");

有返回值：如果点击确认返回输入框的文本 点击取消返回null

```
var res =  prompt("请输入密码？");
alert(res);
```

open方法：

window.open("url地址");

open("../jsCore/demo10.html");

定时器：

setTimeout(函数,毫秒值);

```
setTimeout(
        function(){
                alert("xx");
        },
        3000
);
```

clearTimeout(定时器的名称);

```
var timer;
var fn = function(){
        alert("x");
        timer = setTimeout(fn,2000);
};
var closer = function(){
        clearTimeout(timer);
};
fn();
```

```
setInterval(函数,毫秒值);
clearInterval(定时器的名称)
        var timer = setInterval(
        function(){
                alert("nihao");
        },
        2000
);
var closer = function(){
        clearInterval(timer);
};
```

需求：注册后5秒钟跳转首页

恭喜您注册成功，5秒后跳转到首页，如果不跳转请点击这里

```
<script type="text/javascript">
        var time = 5;
        var timer;
        timer = setInterval(
                function(){
                        var second = document.getElementById("second");
                        if(time>=1){
                                second.innerHTML = time;
                                time--;
                        }else{
                                clearInterval(timer);

location.href="../jsCore/demo10.html";
                        }
                },
                1000
        );
</script>
```

(2)location

location.href="url地址";

(3)history

back();

forward();

go();

后一页

<input type="button" value="上一页" onclick="history.back()">

<input type="button" value="下一页" onclick="history.forward()">

<input type="button" value="上一页" onclick="history.go(-1)">

<input type="button" value="下一页" onclick="history.go(1)">

七、js的dom

1、理解一下文档对象模型

html文件加载到内存之后会形成一颗dom树，根据这些节点对象可以进行脚本代码的动态修改

在dom树当中 一切皆为节点对象

2、dom方法和属性

笔记见代码