.innerWidth 内容 + padding

.outerWidth 内容 + padding + border

.outerWidth(true) 内容 + padding + border+margin

储存数据

w.sessionStorage session 会议 storage储存

将数据储存到当前页面的内存中，容量为5mb

生命周期为关闭当前页面。自动清除

调用方法 setItem(k,value) 存储数据

GetItem(k) 调用

removeItem(k) 移除

clear() 清除

w.localStorage local 局部

在同一个浏览器不同窗口可以共享数据。

数据存储在硬盘上，永久存在，必须手动清除

播放器常用属性

.currenTime 获取视频播放的当前进度

.duration 获取视频的总时长

.paused 视频播放的状态

地理位置接口

navigator.geolocation getCurrentPosition(success,error,option)

全屏显示（不同浏览器的兼容性）

Div.requestFullScreen() 开启全屏显示

D．cancelFullScreen() 退出全屏

D．fullScreenElement 是否全屏

JSON的转换

Json.parse( ) 把json形式的字符串转换成对象

Json.stringify( ) 把对象转换成字符串

Json\_encode( ) 把数组转换成json形式的字符串

Eval( ) 把字符串解析成JS代码并执行

## DOM获取元素添加元素的方法

.innerHTML 获取添加html元素

.innerText 获取添加文本内容

.disablad 禁用文本框

.onmouseover 鼠标进入

.onmousemove 鼠标经过

.onmouseout 鼠标离开

.onfocus 获取焦点

.onblur 失去焦点

.onkeyup 键盘弹起

.onkeydown 键盘按下

.children 获取子元素

.childNodes 获取子节点

.nextSibling 下一个兄弟节点

.previousSibling 上一个兄弟节点

.nextElementSibling 下一个兄弟元素

.previousElementSibling 上一个兄弟元素

.parentNode 父节点

.parentElement 父元素

.appendChild( ) 追加子节点

.replaceChild(a,b) b替换a

.insertBefore( a , b) 把 a 插到 b的前面

.cloneNode(true) 克隆节点

.removeChilld( ) 移除子节点

.document.createElement( ) 创建元素

## 数组的一些常用方法

arr. indexOf( ) 判断是否存在 存在返回值大于等于0，

截取方法

arr.sort() 将数组按从小到大的顺序 返回一个数组

arr.slice( a,b) 下表重a开始截取到b

arr.splice( a,b,c,d) a 为下表的值 截取b 为重开始的几个 c,d为替换的值 可以是多个

arr.concat(n) 把两个数组合并 返回一个新的数组

arr.join(“”) 将元素用xx连接起来，返回一个string

Arr.pop( )删除数组最后一个元素，并返回

Arr.push( ) 给数组最后添加元素,返回新的数组的长度

Arr.shift ( )删除数组第一个元素，并返回

Arr.unshift( ) 给数组开头添加元素,返回新的数组的长度

Arr.toString( ）将数组转换成一个字符串

## 字符串的一些方法

split(“”) 按照指定字符，对字符串进行分割，返回一个数组

.substr(a,b) 从a开始截取b个

.substring(a,b) 重a开始 截取到b(不包含)

.replace(“替换的值”，“替换值”)只能替换一个

（/替换的值/g，“替换值”）替换所以符合值

.trim ( ) 去除字符串两边的空格

## 解决中文乱码问题

.decodeURI( )码转中 .encodeURI( )中转码

.substr(a,b) 重a 开始 截取b个

## 替换方法

.replace(“要替换的值”，“替换值”)只能替换第一个

.replace(“/要替换的值/g”，“替换值”)替换所以的值

## 计时器转换大小写

.toUpperCase 转换大写

.toLowerCase 转换小写

setTimeout() 延时定时器

setInterval() 间歇定时器

clearTimeout/Interval() 清除定时器

## 字体样式

text-decoration:none 取消下划线删除线

Vertical-align:middle 垂直居中

## 正则

\d 数字 \w 单词字符和 \_ \s 不可见字符 “空格”

\D 非数字 \W非单词字符 \S 可见字符 任意值

| 和 （）可以提高优先级 [ ] 获取范围

“\*”重复零次或多次

“+” 重复多次获一次

“？”重复零次或一次

“{}”自定义重复次数

用. test( ) 判断是否符合规格

*//通过构造函数创建*var regEx = new RegExp(/\d/);*//\d预定义类 表示数字 digit 数字*regEx.test("1");*//可以判断传入的字符串是否符合规则 符合要求返回true不符合返回false***console**.log(regEx.test("1"));*//true***console**.log(regEx.test("a"));*//false 匹配数字 字母不行***console**.log(regEx.test("a1"));*//true 只要有我想要的就可以  
//字面量*var regEx2 = /\d/;  
*//如果这个规则只是使用一次***console**.log(/\d/.test("1"));  
  
  
**console**.log(/\d/.test("1"));  
**console**.log(/\d/.test("a"));  
**console**.log(/\s/.test("a"));  
**console**.log(/\s/.test(" "));  
**console**.log(/\w/.test("a"));  
**console**.log(/\w/.test("1"));  
**console**.log(/\w/.test("\_"));  
**console**.log(/\w/.test("中"));*//单词字符 指的是 英文字母 数字 下划线  
  
  
  
//现在是在学习基础语法 现在的匹配规则是 只要有我想要的就可以***console**.log(/chuan/.test("chuan"));  
**console**.log(/chuan/.test("chuang"));  
**console**.log(/chuan/.test("chua"));*//可以多 不能少  
  
//| 表示或 或的优先级最低 () 可以提升优先级***console**.log(/foot|boot/.test("foot"));  
**console**.log(/foot|boot/.test("boot"));  
**console**.log(/(foot)|(boot)/.test("boot"));  
**console**.log(/f|boot/.test("boot"));  
**console**.log(/(f|b)oot/.test("boot"));  
**console**.log(/a1|b1|c1/.test("boot"));  
**console**.log(/a|b|c1/.test("boot"));  
**console**.log(/(a|b|c)1/.test("boot"));  
**console**.log(/[abc]1/.test("boot"));  
  
  
**console**.log("------简单类------");  
**console**.log(/[abc]/.test("a"));  
**console**.log(/[abc]/.test("b"));  
**console**.log(/[abc]/.test("c"));  
**console**.log(/[abc]/.test("d"));*//false a或者b或者c都可以 其他的不行***console**.log(/[abc]/.test("da"));*//true 只要有我想要的就可以***console**.log("------负向类------");  
**console**.log(/[^abc]/.test("a"));*//不能是a或b或c***console**.log(/[^abc]/.test("b"));*//不能是a或b或c***console**.log(/[^abc]/.test("c"));*//不能是a或b或c***console**.log(/[^abc]/.test("d"));*//true 不能是a或b或c d可以***console**.log(/[^abc]/.test("da"));*//true 只要有我想要的就可以***console**.log("------范围类------");  
**console**.log(/[abcdefghi]/.test("a"));  
**console**.log(/[a-i]/.test("a"));  
**console**.log(/[a-z]/.test("a"));*//所有小写字母***console**.log(/[A-Z]/.test("a"));*//所有大写字母***console**.log(/[0-9]/.test("a"));*//所有数字***console**.log("------组合类------");  
**console**.log(/[a-zA-Z0-9]/.test(""));*//表示所有的大小写字母还有数字***console**.log(/[a-kA-N0-5]/.test(""));*//组合类中的范围随意  
  
  
  
//{} 是量词 用来修饰前面内容出现的次数***console**.log(/^a{2}$/.test("aa"));  
*//[] 表示一个字符的位置***console**.log(/^[abc]$/.test("c"));  
*//() 用来分组 可以提高优先级***console**.log(/^chuan$/.test("chuan"));  
**console**.log(/^chuan{2}$/.test("chuanchuan"));*//false***console**.log(/^chuan{2}$/.test("chuann"));*//true***console**.log(/^(chuan){2}$/.test("chuanchuan"));*//true*