1. 标签

1>常用的标签

p, h1-h6, div, img, a, span等标签

2>列表标签

(1)ol li(有序列表, 默认自带排序, 在ol上设置type属性可以控制用什么排序(I, 1, a))

(2)ul li(无序列表, 默认自带小黑点, 可以使用list-style: none清除小圆点, 常用)

(3)dl(定义列表含义) dt(定义列表项目) dd(定义列表内容) (自定义列表)

3>单标签(只需要一个标签)

(1)<meta>标签

提供有关页面的元信息, 针对搜索引擎和更新频度的描述和关键词。

##name属性表述文档信息, 对搜索引擎友好。

##content属性定义http-equiv或者name属性相关的元信息

(2)<br>(换行符)

(3)<hr>(水平线)  
5>其他常见标签

(1)em标签 强调作用(效果为倾斜)。

(2)strong标签 强调作用更强调(效果为加粗)。

(3)i标签 斜体作用(效果为倾斜)。

(4)u标签 下划线标签(效果为下划线)。

(5)s标签 删除线标签。(效果为字体中间删除线)

(6)small标签 小号字体标签。

(7)code标签 代码原样输出(按文本格式输出, 一行写完再换行)

(8)pre标签 代码原样输出(保留在html写的空格, 与代码里面内容完全一致)

(9)del标签 定义文档中已经删除的文本 显示效果等同于删除线s标签

6>常见语义化标签

(1)header标签(文档的页眉部分, 介绍信息)

(2)nav标签(导航链接部分, 导航栏)

(3)main标签(文档主要内容部分, 一个页面只使用一次, main标签ie11不支持)少用

(4)section标签(文档中的节, 具有相似主题的一组内容, 如新闻条目)

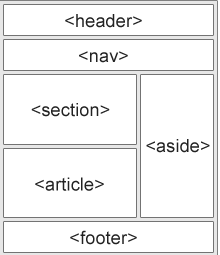
(5)article标签(独立的自包含内容, 如论坛帖子, 包装文章等)

(6)aside标签(附注栏, 如引述, 侧栏, 指向文章的一组链接, 广告, 相关产品列表等)

(7)footer标签(文档或节的页脚, 如文档作者, 版权信息, 使用条款链接等)

(8)address定义文档或者文章作者或拥有者的联系信息(一般成斜体展示)

(9)以上标签使用基本案例 -->

 可以使用main标签把例子中的section article aside标签包裹

7>处理语义化标签兼容

(1)IE9以上版本也只是选择支持h5(main标签ie11不支持)

会把语义化标签解析成inline元素, 兼容需要设置语义标签设置为display: block;

(2)IE8以及以下版本完全不支持h5

全不能解析语义化标签, 兼容需要用js手动创建语义化标签, 在将需要为块级元素的标签的style中设为display: block;

(3)IE兼容方案, 引入第三方库(解决繁琐操作, 手动JS声明标签)

<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

注意：位置需要在使用标签之前(常用在IE8及以下版本兼容方案)

1. 表格
2. 表格一般都是一个table标签包裹其里面的标签有

(1)tr(行)

(2)td(单元格)

(3)th(表头单元格), 效果为字体加粗

(4)thead(表头标签), tbody(表主体标签),tfoot(表尾标签)

(5)caption(表主题)只能用语table标签的第一个子元素, 在表格的正上方居中显示

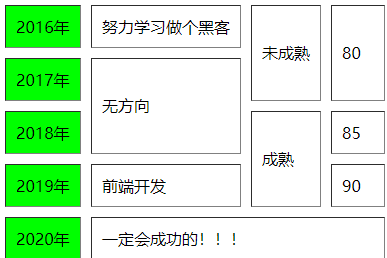
(6)colgroup标签

##里面col标签可以设置width属性可以规定表格可以规定每列多宽,

几个width表示分成几列

##里面的span=”1”属性规定表格几列什么样式 -->





2>表格上常用的属性

(1)table标签上有

border=”0”(设置变宽大小, 默认是无变宽的0),

align=”center”(整个表格的位置) ,

cellspacing=”0”(单元格之间的margin),

cellpadding=”0”(每个单元格里面文字的padding)

(2)td和th标签上有

rowspan(跨行合并), rowspan=”合并几行”

colspan(跨列合并) colspan=”合并几列”

注意：一般colspan在thead不生效, 因为里面只有一个tr。

注意：除去表格默认双边框方法有

(1)cellspacing=”0”

(2)在其style上设置为border-collapse: collapse(其默认值为separate(双边框))

1. 表单(可配合表单制作文件例子看)

1>使用场景

一般表单为一个form标签包裹很多个表单控件(input标签)

2>form标签上的属性有

(1)method=”方法”属性 表单提交的方法默认值为get, 还有个为post方法

(2)action=”url”属性 添加表单提交到的地址

(3)target=”\_blank” 提交表单时打开新的页面

(4)name=”名字” 规定表单名称

(5)autocomplate=”on” 开启自动完成功能(历史记录), 默认值为off

3>input标签上的属性有

(1)checked默认选中

(2)placeholder=”提示信息” 规定提示信息 -->

(3)required 规定字段必须提交

(4)autocomplate=”on” 开启自动完成功能(历史记录), 默认值为off

注意: 需要按提交按钮后才会记录

(5)multiple可以选择多个东西(文件 type=”file”加上这个属性之后可选择多个文件, 也可用 于写入多个邮箱type=”emil”)

(6)form=”form标签id” 把该字段提交到哪个表单

(7)autofocus=”on”(打开自动聚焦) off关闭

(8)novalidate取消验证

(9)pattern=”正则验证” 规定提交格式( 一般配合required属性 ) -->



(10)list=”datalist的id”引用输入字段额预定选项 datalist>option

(11)max=”数字”输入字段的最大值, min=”数字”输入字段的最小值 (一般配合使用规定范围)

(12)maxlength=”数字”规定输入字段中的字符额最大长度

(13)step=”数字”字符间隔

(14)size=”数字” 输入字段宽度

4>input标签上type属性值有

(1)radio单选

(2)checkbox复选

(3)from=”form标签的id” 指定控件提交到哪个表单

(4)file选择文件

配合multiple属性可以选择多个文件

(5)email 会自动验证 输入的内容必须包含@(合法邮箱)

(6)tel不会自动验证(面向全球服务) 移动端自动弹出数字键盘

(7)url 会自动验证 输入必须包含http://(合法网站)

(8)number只能输入数字(包含小数点) input框右边有+1, -1按钮

其有max属性和min属性

(9)search 可以提供更好的体验(input框右边有删除全部内容按钮)

(10)range范围 -->  其有max属性和min属性

(11)color 拾色器-->

(12)time时间 时分秒-->

(13)datatime日期时分秒 -->

注意：大部分不支持只有苹果的safari

(14)month月week星期

(15)list=”datalist标签的id”(不常用datalist标签的兼容性不好)

5>textarea文本域标签 想要input多行显示使用如内容 -->



6>label标签(可通过标签的for属性绑定对应的input标签通过id, 绑定后点击label标签的内容光标会自动到指定的input框里面(为增强用户体验) )-->



注意：input写在label标签里面也无法实现点击聚焦到input必须通过for属性绑定

7>表单注意点

(1)要有value数据值和name数据名表单才能发送成功。

(2)name值相同才知道是同一题目

(3)一个input框输入多个邮箱时, 号分开

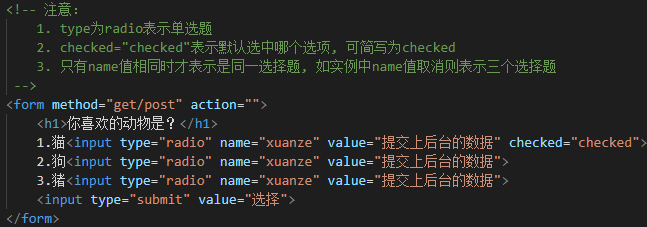
(4)name属性

##name 属性用于对提交到服务器后的表单数据进行标识，或者在客户端通过 JavaScript 引用表单数据

##只有设置了 name 属性的表单元素才能在提交表单时传递它们的值，因为服务端获取表单提交的数据是通过表单元素的 name 属性的值而得到的，没有 name 属性就无法得到表单元素提交给服务端的值

8>表单实例

(1)表单实例制作单选题 -->



(2)表单实例制作复选题 -->



(3)表单实例制作下拉菜单 -->



1. 多媒体标签

1>音频播放：audio标签的使用:

(1)属性：

|  |  |  |
| --- | --- | --- |
| [autoplay](http://www.w3school.com.cn/html5/att_audio_autoplay.asp" \o "HTML5 <audio> autoplay 属性) | autoplay | 如果出现该属性，则音频在就绪后马上播放。 |
| [controls](http://www.w3school.com.cn/html5/att_audio_controls.asp" \o "HTML5 <audio> controls 属性) | controls | 如果出现该属性，则向用户显示控件，比如播放按钮。 |
| [loop](http://www.w3school.com.cn/html5/att_audio_loop.asp" \o "HTML5 <audio> loop 属性) | loop | 如果出现该属性，则每当音频结束时重新开始播放。 |
| [preload](http://www.w3school.com.cn/html5/att_audio_preload.asp" \o "HTML5 <audio> preload 属性) | preload | 如果出现该属性，则音频在页面加载时进行加载，并预备播放。  如果使用 "autoplay"，则忽略该属性。 |
| [src](http://www.w3school.com.cn/html5/att_audio_src.asp" \o "HTML5 <audio> src 属性) | *url* | 要播放的音频的 URL。 |

(2)示例:播放音频

(2)示例:播放音频

<audio src="../mp3/See.mp3" controls autoplay></audio>

2>视频播放：video标签的使用

(1)属性：

|  |  |  |
| --- | --- | --- |
| [autoplay](http://www.w3school.com.cn/html5/att_video_autoplay.asp" \o "HTML5 <video> autoplay 属性) | autoplay | 如果出现该属性，则视频在就绪后马上播放。 |
| [controls](http://www.w3school.com.cn/html5/att_video_controls.asp" \o "HTML5 <video> controls 属性) | controls | 如果出现该属性，则向用户显示控件，比如播放按钮。 |
| [height](http://www.w3school.com.cn/html5/att_video_height.asp" \o "HTML5 <video> height 属性) | *pixels* | 设置视频播放器的高度。 |
| [loop](http://www.w3school.com.cn/html5/att_video_loop.asp" \o "HTML5 <video> loop 属性) | loop | 如果出现该属性，则当媒介文件完成播放后再次开始播放。 |
| [preload](http://www.w3school.com.cn/html5/att_video_preload.asp" \o "HTML5 <video> preload 属性) | preload | 如果出现该属性，则视频在页面加载时进行加载，并预备播放。  如果使用 "autoplay"，则忽略该属性。 |
| [src](http://www.w3school.com.cn/html5/att_video_src.asp" \o "HTML5 <video> src 属性) | *url* | 要播放的视频的 URL。 |
| [width](http://www.w3school.com.cn/html5/att_video_width.asp" \o "HTML5 <video> width 属性) | *pixels* | 设置视频播放器的宽度。 |

(2)视频播放

<video src="../mp3/561902ae6ac6e6649.mp4" controls></video>

(3)说明：由于版权等原因，不同的浏览器可支持播放的格式是不一样的



3>常用方法：

load() 加载、 play() 播放、 pause() 暂停

4>常用属性：

(1)currentTime 视频播放的当前进度、

(2)duration:视频的总时间 100000/60

(3)paused:视频播放的状态.

5>常用事件：

(1)oncanplay: 事件在用户可以开始播放视频/音频（audio/video）时触发。

(2)ontimeupdate:通过该事件来报告当前的播放进度.

(3)onended:播放完时触发—重置

注意：Jq没有提供对视频播放控件的方式，也就意味着如果要操作视频播放，必须使用原生的js方法—dom元素

1. css基础(必看)
2. 元素分类

(1)行级元素又叫内联元素 display:inline

内容决定元素所占位置, 不可以改变其宽高 -->sapn, strong, em a, del等标签

(2)块级元素 display:block

独占一行, 可以通过css改变其宽高 --> div, p, ul, li, ol, form, address等标签

(3)行级块元素 display:inline-block

内容决定大小, 可以改宽高 -->img等标签

注意：三种类型可以随意转化通过display属性, 带有inline的都属于文字属性(无论多少个空格只显示一个空格)

1. css权重(256进制)

(1)权重表

！important Infinity(无穷大)

行间样式 1000

id 100

class|属性|伪类 10

标签|伪元素 1

通配符 0

(2)权重应用

##同时选中一个元素权重值大的style覆盖权重值小的style

##同时选中一个元素相同权重写在下面的覆盖上面的style

1. 四种基本选择器(更多选择器查看四五六天笔记)

(1)父子选择器/派生选择器 -->div p{ }

(2)直接子元素选择器 -->div>p{ }

(3)并列选择器 -->div.class{ } (选择同一个元素上的多个属性)

(4)分组选择器 -->demo1, demo2 { }

4>盒模型

(1)普通盒模型计算

元素的width+padding左右边距+border左右大小+margin左右大小高度同理

(2)IE盒模型计算

元素的width(border+padding+content) + margin

注意：设置成IE盒模型后后padding, border撑不开容器, 容器大小又width决定

(3)设置盒模型(box-sizing: 以下值)

##border-box值

##content-box值(默认值, 就是标准盒模型)

##inherit值(继承父集box-sizing属性)

(4)盒模型兼容

IE8及以上版本支持该属性，Firefox 需要加上浏览器厂商前缀-moz-，对于低版本的IOS和Android浏览器也需要加上-webkit-

5>定位(配合left, top, bottom, right使用一般两两配合使用, z-index属性(第几层))

(1)absolute绝对定位 相对于最近的定位元素进行定位

(2)relative相对定位 相对于自己原来的位置进行定位

(3)fixd固定定位 相对于浏览器进行定位

(4)定位应用以及注意点

##一般使用定位都是父元素relative子元素absolute。

##子元素设置定位后撑不开父容器

6>浮动(产生浮动, 脱离文档流造成margin高度塌陷, 在父子元素中)

(1)特点

##块级元素不能看到浮动元素,

##触发了bfc的元素, 文本类属性元素(带有inline属性的元素)能看到浮动元素

(2)清除浮动方式

##第一个子元素和最后一个子元素设置为块级元素且设置clear：both属性(与浮动元素相同层级), 推荐在父元素使用伪元素 -->

span::before{ content:""；display: block; clear: both }

兼容IE span{zoom:1}

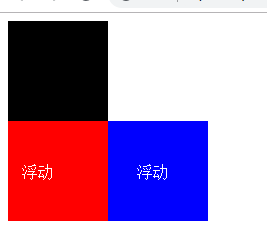
##父元素触发BFC。使父元素强行包裹浮动元素，并隔绝内外环境。外面的元素不会受里面浮动元素影响。

##为父元素设置高度。(不推荐)

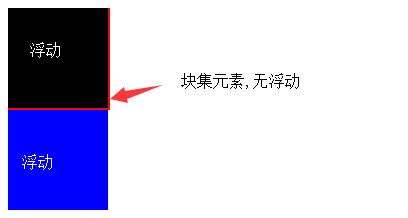
(3)浮动位置

##实例

第一个为块级元素, 第二个为浮动元素, 第三个为浮动元素, 那么两个块级元素会被挤到第二行, 原因是第一行无位置了 -->



##无位置详解 -->



7>触发bfc条件

(1)根元素 float属性不为none

(2)position为absolute或fixed

(3)display为inline-block, table-cell, table-caption, flex, inline-flex

(4)overflow不为visible

8>单位总结

(1)绝对单位

##像素(px), 计算机屏幕上的一个点, 1px = 1/96in

(2)相对单位

##百分比(%),

相对于父元素的宽度或者字体大小

##em,

相对于当前元素文本字体的大小, 1em=当前字体的大小, 2em=当前字体大小\*2

##rem(root em)

相对于html标签的字体大小

##vw, vh

1vw = 1%视口(浏览器可视区域)的宽度

1vh = 1%视口(浏览器可视区域)的高度

1. 操作元素扩展

1>H5新增表单事件

(1)input 只要内容改变就会触发该事件

(2)invalid 当前验证不通过时候触发一般用于改变input验证默认信

2>H5新增操作元素类

(1)dom.classList.add(“class”) 添加class

(2)dom.classList.romove('class') 移除class

(3)dom.classList.toggle('class') 切换class，有则移除，无则添加

注意：()只能操作一个class需要操作多个需重新创建

(4)dom.classList.contains('class') 检测是否存在class, 有则true, 无则false

(5)dom.classList.item(index) 找dom的第索引位class, 返回其类名

3>H5自定义属性

(1)设置自定义属性

##标签内设置(和标签添加class一样)

data-自定义属性名="设置的属性值"

##在js中设置

dom.dataset.自定义属性名 = “设置的属性值”

(2)获取其属性(js中获取)

##dom.dataset.自定义的属性名

(3)注意设置属性名时候的规范

##dom.dataset(data-(标签内设置的属性)), js中获取其属性驼峰式获取

1. H5新增常用接口

1>拖拽接口

(1)将需要拖拽的元素设置其属性draggable=”true”(图片和超链接默认可以拖拽)

(2)拖拽事件

++drag 拖拽整个过程都会调用(持续触发)

++dragstart 拖拽开始时调用

++dragleave 拖拽元素离开绑定元素时调用

++dragend 当拖拽元素松开鼠标时触发

++dragenter 当拖拽元素进入绑定事件元素时触发

++dragover 当拖拽元素停留在绑定事件元素时触发

++drop 当拖拽元素在绑定元素内松开鼠标调用。

注意：想要drop事件触发不要阻止dragover的默认行为(实现拖拽主要靠这两个事件, 其他事件主要为了给拖拽过程实现效果)

(3)拖拽实例总结

注意：在拖动元素时，每隔 350 毫秒会触发 ondrag 事件

实现要点: ##使用一个对象接收被拖拽的元素用于插入目标元素

##利用ondragstart和ondragend事件实现拖拽时的效果

##利用ondragover和ondrop事件实现拖拽功能

##优化(实例演示无说明)

++为了防止定义的对象影响全局变量(优化后可不定义对象)

++使用dataTransfer方法来存取拖拽元素

--setData存(ondragstart事件中存) -->

// 通过dataTransfer来存储数据和获取

// setData(format, data)

// fomrmat: 数据类型(text/html text/url-list)

// data: 数据(一般时字符串值)

e.dataTransfer.setData("text/html", e.target.id);

--getData取(ondrop事件中取)

/\*通过e.dataTransfer.setData存储的数据，只能在drop事件中获取\*/

var id = e.dataTransfer.getData("text/html");

e.target.appendChild(document.getElementById(id));

(4)实例 -->



2>地图接口

(1)概念

在HTML规范中，增加了获取用户地理信息的API，这样使得我们可以基于用户位置开发互联网应用，即基于位置服务 (Location Base Service), 浏览器会自动以最优方式去获取用户地理信息。

(2)定位方式优缺点



(3)隐私: 推送通知

HTML5 Geolocation(地理位置定位) 规范提供了一套保护用户隐私的机制。必须先得到用户明确许可，才能获取用户的位置信息

(4)API说明

## navigator.getCurrentPosition(successCallback, errorCallback, options)

获取当前地理信息

##navigator.watchPosition(successCallback, errorCallback, options)

重复获取当前地理信息

##当成功获取地理信息后，会调用succssCallback，并返回一个包含位置信息的对象position。Coords(坐标)

##position.coords.latitude纬度

##position.coords.longitude经度

##当获取地理信息失败后，会调用errorCallback，并返回错误信息error

##可选参数 options 对象可以调整位置信息数据收集方式

3>FileReader接口

(1)readAsText()

##读取文本文件(需可以用Txt打开的文件),

##返回文本字符串

##默认编码为utf-8

(2)readAsBinaryString()

##读取任意类型的文件

##返回二进制字符串,

##一般不是读取文件给用户看, 而是存储文件(传递给后台, 后台接收后存储)

(3)readAsDataURL()

##读取文件获取一段以data开头的字符串, 这段字符串的本质是DataURL,

##DataURL是一种将文件(这个文件一般指的是图像或者能够嵌入到文档的文件

格式)嵌入到文档的方案, 能将服务器资源转化为base64编码 的字符形式, 且将这些内容直接存储到url中》》优化网站加载速度和执行效率

##无返回值, 需要传递一个参数(二进制的大文件, 如图文或者能嵌入文档的文件)

##读取完文件之后会将读取的结果存储在读取文件对象的result中

##可以使用onload事件判断是够读取完成文件

##使用方式 -->img(获取的imgDOM).src = reader(读取文件对象).result

##创建文件对象方式 -->var reader = new FileReader();

(4)abort() 中断文件读取

1. 移动端的touch触摸和轻击(faskclick)
2. 触摸

(1)事件

##touchstart 当手指触摸屏幕时候触发

##touchmove 当手指在屏幕来回滑动时候触发

##touchend 当手指离开屏幕时候触发

##touchcancel 当被迫中止滑动时候触发, 如来电, 弹消息

(2)事件对象

##changedTouchs 改变后的触摸点集合

##targetTouches 当前元素的触摸点集合

##touches 页面上所有触发点的集合

(3)事件对象三者关系

##名字都是touchList(触摸点的集合, 一个手指一个触摸点)

##changedTouchs在接触屏幕和离开屏幕都会有记录而targetTouches 和targetTouches不会有记录( length: 0 )

(4)实战说明

##通过绑定touchstart和touchmove计算其e.touches[0].clientX(鼠标坐标)值

计算其值, 判断左右滑动手势。

2>tap轻击(faskclick)

(1)前言

tap事件是touch事件衍生出来的(其封装), 为了解决移动端click事件延迟300ms(移动端click事件延迟300ms为了区分用户是点击还是滑动) , 谷歌浏览器看不到延迟效果, 在真机才有300ms延迟效果

(2)可以使用第三方插件引入就可以直接使用了

fastclick.js插件, 提供移动端响应速度, 引入之后在需要轻击事件的页面执行 -->

document.addEventListener(“DOMContentLoaded”, function () {

FastClick.attach(document.body);

})

既可以使用, 页面使用click绑定事件即使轻击事件

zepto.js库(基于高版本开发)里面也有tap轻击事件的封装

(3)原生js封装tap事件



1. 适配方案(通过视口, 配合流式布局)
2. 流式布局

宽度使用百分比(百分比布局), 高度用px来固定住

1. viewport(视觉窗口, 即视口)

(1)简介

viewport是移动端特有的, 是一个虚拟的区域, 承载网页, 浏览器承载视口, 视口承载网页(使得PC端页面放到移动端不出现滚动条, 原理是缩放)

缺点：内容也会缩放(采用viewport默认值会)

(2)标准适配方案(改变viewport默认设置以达到内容正常显示, 且无滚动条)

##注意点：

++设置viewport需紧跟在编码(<meta charset=”UTF-8”>)后面

++s通过meta标签的content属性来传递参数来改变其默认值 -->

<meta content=”里面设置参数”>

++开启默认缩放(user-scalable=yes)之后maximum和maximum才有用

##viewport的主要功能(参数)

++width 设置宽度 (device-width当前设备的宽度)

++height 设置高度

++initial-scale 缩放比例

++user-scalable 是否允许用户缩放 (no, 0用户不能缩放, yes, 1用户能缩放)

++maximum 最大缩放比

++maximum 最小缩放比

(3)国际标准适配方案



(4)生成标准适配方案快捷键meta:vp+tab

1. H5C3拓展笔记内容
2. 兼容前缀
3. 新增属性选择器, 伪类选择器(n-th-child)
4. 颜色取值(rgba, hsla)
5. 文本阴影, 盒阴影, 圆角
6. 渐变, 背景属性详解, 过度
7. 2d旋转, 3d旋转, 动画
8. 列布局(操作文本内容), 伸缩布局(弹性盒子)

1. 本地缓存
2. window.sessionStorage存储

(1)sessionStorage约5M

(2)生命周期为关闭浏览器窗口：相当于存储在当前页面的内内存中(在同一个窗口下数据可以共享(在当前页面下可以获取到，换另外一个页面下不能获取到))

2>window.localStorage存储

(1)localStorage约20M

(2)永久生效，除非手动删除：存储在硬盘上, 可以多窗口共享。但是不能跨浏览器

3>两者共同点

(1)只能存储字符串，可以将对象JSON.stringify() 编码后存储

(2)方法

## setItem(key,value):设置数据，以键值对的方式

## getItem(key):通过指定的键获取对应的值内容

## removeItem(key):删除指定的key及对应的值内容

## clear():清空所有存储内容

1. 使用案例

<script>  
 **var** userData=document.getElementById("userName");  
 //存储数据  
 document.getElementById("setData").onclick=**function**(){  
 window.sessionStorage.setItem("userName",userData.value);  
 }  
 //获取数据  
 document.getElementById("getData").onclick=**function**(){  
 **var** value=window.sessionStorage.getItem("userName");  
 alert(value);  
 }

</script>

1. 离线存储

1>概念

通过创建 cache manifest 文件, 可以轻松地创建 web 应用的离线版本

2>基本使用

(1)如需启用应用程序缓存，请在文档的 <html> 标签中包含 manifest 属性 -->

<!DOCTYPE HTML>

<html manifest="demo.appcache">

...

</html>

(2)创建manifest 文件的建议的文件扩展名是：".appcache"

注意：manifest 文件需要配置正确的 MIME-type，即 "text/cache-manifest"。必须在 web 服务器上进行配置

(3)每个指定了 manifest 的页面在用户对其访问时都会被缓存。如果未指定 manifest 属性，则页面不会被缓存（除非在 manifest 文件中直接指定了该页面）

3>Manifest 文件

(1)概念：

manifest文件是简单的文本文件, 它告知浏览器被缓存的内容(以及不缓存的内容)

(2)manifest 文件可分为三个部分

##CACHE MANIFEST – 开始

CACHE在此标题下列出的文件将在首次下载后进行缓存

##NETWORK - 在此标题下列出的文件需要与服务器的连接，且不会被缓存

##FALLBACK - 在此标题下列出的文件规定当页面无法访问时的回退页面(比如 404 页面)

(3)三部分说明

##CACHE MANIFEST说明：

CACHE MANIFEST，放置在第一行，是必需的 -->

CACHE ：

/theme.css

/logo.gif

/main.js

上面的 manifest 文件列出了三个资源：一个 CSS 文件，一个 GIF 图像，以及一个 JavaScript 文件。当 manifest 文件加载后，浏览器会从网站的根目录下载这三个文件。然后，无论用户何时与因特网断开连接，这些资源依然是可用的

##NETWORK说明：

NETWORK小节规定文件"login.asp"永远不会被缓存, 且离线时是不可用的 -->

NETWORK:

login.asp

##FALLBACK说明：

FALLBACK小节规定如果无法建立因特网连接, 就使用指定的资源代替所请求的url的资源, -->

FALLBACK:

/html5/ /404.html

注释：当html5资源在离线状态下无法请求的时候，就使用404.html代替

(4)总结说明

##CACHE: 可以省略，这种情况下将需要缓存的资源写在CACHE MANIFEST

##可以指定多个CACHE: NETWORK: FALLBACK:，无顺序限制

## #表示注释, 只有当demo.appcache文件内容发生改变时或者手动清除缓存后，才会重新缓存。

##chrome 可以通过chrome://appcache-internals/工具和离线（offline）模式来调试管理应用缓存

(5)更新缓存

一旦文件被缓存，则浏览器会继续展示已缓存的版本，即使修改了服务器上的文件。为了确保浏览器更新缓存，也需要更新 manifest 文件，也就意味着一旦应用被缓存，它就会保持缓存直到发生下列情况：

1. 用户清空浏览器缓存
2. manifest 文件被修改（参阅下面的提示）
3. 由程序来更新应用缓存

说明：更新注释行中的日期和版本号是一种使浏览器重新缓存文件的办法

1. 移动端
2. 屏幕像素和px区别

屏幕像素：物理像素, 即像素点, 设备屏幕的最小可视颗粒

px:页面的尺寸单位

1. 移动开发可以使用jquery但不推荐使用

原因：jquery做了很多PC端浏览器兼容, 特别是ie, 移动端没有ie

1. 移动端内核都是webkit或者blink 兼容都是加-webkit-前缀
2. 建议使用h5的pai或者使用 zepto.js的库(基于高版本浏览器开发)
3. 常用box-sizing原因

(1)移动开发常用流式布局(宽度百分比, 高度固定)

(2)普通盒模型无法准确计算盒子大小 -->

.box{ wdith: 100%; height: 200px; border: 20px solid red; }

普通盒子无法计算盒子大小会导致出现横向滚动条

ie盒模型其大小为100%, 200px无横向滚动条

(3)设置IE盒模型之后都是从边框开始计算盒子尺寸的

6>使用二倍图的原因

物理像素比 非矢量网站可能会失真, (标准视口使用二倍图可以解决失真问题)

7>屏幕划分(宽度)

(1)<768px 超小屏设备(手机)

(2)[769px, 992px) 小屏设备(平板电脑)

(3)[992,px 1200px) 中屏设备(台式电脑)

(4)>=1200px 大型设备(台式电脑)

1. 开发常见库及其使用方式
2. iScroll插件区域滚动插件

(1)使用插件new(“父容器(小容器)”, { 配置参数} )

(2)需要滚动的子容器必须要大于父容器

(3)父容器和子容器之前需要夹一层容器 -->

div.wrap(父容器)>ul>li(需要滚动的容器)

2>zepto.js库(基于高版本开发)

script标签引入使用, jq用法差不多

3>swipter..js轮播图插件

引入用其html

1. 项目初始化常见设置

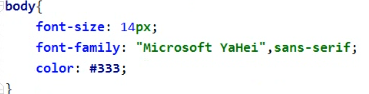
1>设置成IE盒模型

\*, \*::before, \*::after{ box-sizing: border-box; -webkit-box-sizing: border-box}

2>清除高亮属性(移动端特有的默认样式, 注意和伪类:active区分)

-webkit-tap-highlight-color: transparent;清除高亮

3>设置字体

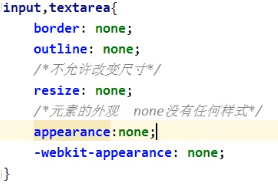


没有Microsoft YaHei就用系统自带的字体

4>去列表原点



5>输入框(resize:none属性设置使文本域大小不可以使用鼠标拖动改变其大小)



1. @规则

1>@media 媒体查询

(1)常见的@media

##@media screen and ( min-width: min ) and ( max-width: max )

{ .layout{ color: red}(设置样式) }

设置屏幕宽度在(min, max)区间时指定元素设置其样式

注意：小值要在前面, 大值要在后面

##@media screen and(min-width: max)

{ .layout{ color: red}(设置样式) }

设置屏幕宽度在max下区间生效, 适当时可以替换上面的设置max, min区间,

需要注意的是设置一个值min-width: max(最大宽度)是全部生效需要把max值从小到大依次放置(按顺序放置), 原因是所有屏幕对同一个元素处理代码都会生效, 为了小屏设备元素样式覆盖大屏设备样式

##@media screen and ( width: min ) and ( width: max )

{ .layout{ color: red}(设置样式) }

设置屏幕宽度在min, max值时指定元素设置其样式

##@media screen and (orientation: landscape)横屏

##@media screen and (orientation: portrait)竖屏

1. @import 导入其他样式文件

(1)基本使用

在css中使用@import “导入文件的地址”;

注意：一定要加;

3>@font-face自定义字体(示例查看拓展笔记)

(1)使用步骤

##选择自己需要的字体图标添加至项目下载

##解压下载的文件将其.woff2, .woff, ttf, svg, eot放至项目fonts文件下

##在css中定义(@font-face), 通过(font-family定义总名)

##在css中创建类引用字体图标(通过font-family引用), 且定义其样式

##在html中添加第一个类(创建类), 第二个类(指定具体某个字体图标)

(2)注意点

##@font-face可以单独创建一个文件(iconfont.css)

##在@font-face后一个创建类, 所有字体图标样式

##在html第二个类可以是一个类(icon-sousou), 可以是一个编码(&#xe633;)

1. @charset 设置样式表的编码

(1)常见设置

在less文件中第一行写 @charset “utf-8”

1. 处理文字css

1>三件套

overflow: hidden; 查出部分隐藏

white-space: nowrap; 强制不换行

text-overflow: ellipsis; 查出的部分成…展示

2>文字换行css

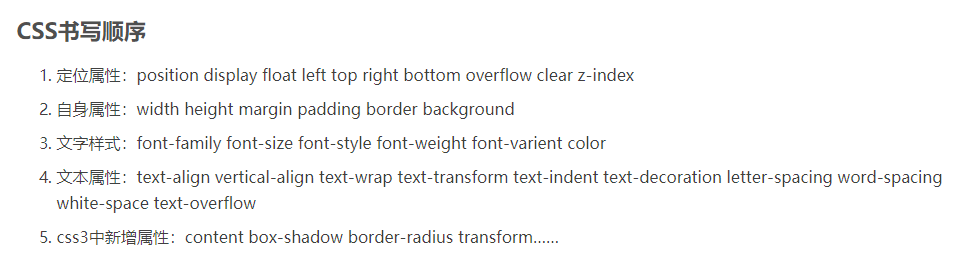
word-break:break-all;自动换行(会折断长的英文单词)

word-break:break-word;自动换行(不会会折断长的英文单词)

注意: 元素拥有默认的white-space:normal 可能设置了nowrap属性导致word-break失效，可以先设置white-space:normal在设置word-break可以生效 🡪

white-space:normal;  
word-break:break-all;

1. css书写规范



1. 小知识总结
2. line-height值不设置单位代表当前元素盒子所继承的字体大小的多少倍
3. line-height值设置在block元素才会使文字居中
4. :nth-of-type和:first-child的区别

(1)p:nth-of-type意思是找到P元素, 通过p找到父元素, 通过父元素找子元素当中类型为p的元素, 然后在去找第几个

(2)p:nth-child 意思是找到P元素, 通过p找到父元素, 通过父元素找到所有的子元素, 找到第一个元素, 匹判断类型(如果不符合类型则此选择器是无效选择器)

4>在html元素设置字体大小会影响rem,

body元素设置字体大小会影响继承其字体大小属性的em

5>p标签嵌套块级元素, a标签嵌套a标签都会解析成兄弟结构(不符合w3c规范)

1. 响应式(bootstrap)
2. 核心模块使用@规则实现
3. 使用less方便维护
4. bootstrap常见类

(1)栅格系统

##-xs超小屏设备

##-sm小屏设备

##-md中屏设备

##-lg大屏设备

(2)容器

##.container包裹栅格系统容器

##.container-fluid宽度100%包裹栅格系统容器

(3)操作栅格(都是相对左操作)

##-offset-2在左侧补2列

##-push-2从左侧往右侧推2列

##-pull-2往左拉2列

(4)排版

##small内联子标题

##lead引导主体副本

##text-left文本左对齐

(5)辅助类

##pull-left/right 元素浮动到左边/右边

##center-block 设置元素为 display:block 并居中显示

##clearfix 清除浮动

##show/hidden 强制显示/隐藏

##close 显示关闭按钮

##caret 显示下拉式功能

##divider 分隔线

1. less
2. 变量
3. 函数
4. 内置方法
5. 其他

(1)变量拼接@{变量}

(2)&暗特符 去空格 -->

.container{

:hover{

color: red;

}

} 生成的css -->

.container :hover{color: red} 需要在:hover前加上&此hover效果生效(去空格)