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)

(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(meta-information),针对搜索引擎和更新频度的描述和关键词

(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。

(2)Strong标签 强调作用更强调。

(3)i标签 斜体作用。

(4)u标签 下划线标签。

(5)s标签 删除线标签。

(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)

(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

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)

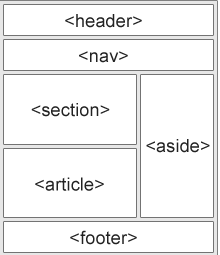
(4)section标签(文档中的节, 具有相似主题的一组内容, 如新闻条目)

(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)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>其他常用标签

(1)hgroup对页面或者区段(section)的标题进行组合

(2)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)

(3)details标签用于描述文档或文档某个部分的细节。显示为产生一个下拉菜单, 只设置这一个标签时为显示一个为详细信息为标题的下拉菜单(配合summary标签)

(4)summary属性规定表格内容的摘要, 显示和普通文本没有任何视觉变化

##details和summary标签配合使用说明

details的第一个子标签summary为下拉菜单的主题

8>处理语义化标签兼容

(1)IE9以上版本也只是选择支持h5

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

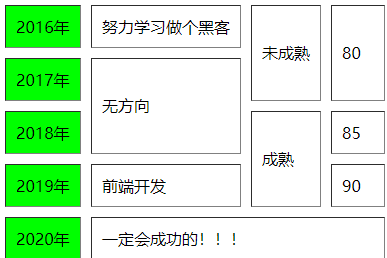
(4)caption(表主题, 只能用语table标签的第一个子元素, 在表格的正上方居中显示)

(5)colgroup标签

##里面的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：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标签的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 -->

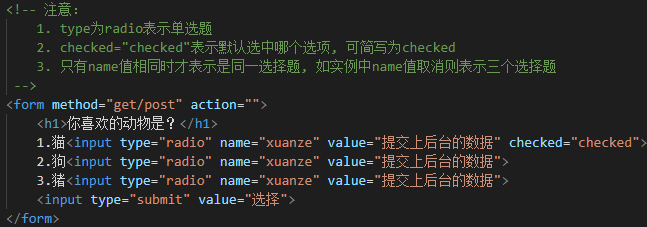


6>label标签(可通过标签的for属性绑定对应的input标签通过id, 绑定后点击label标签的内容光标会自动到指定的input框里面(为增强用户体验) )-->



7>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 -->



8>表单注意点

(1)要有数据值和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

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)

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

(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(浮动后块级元素不能看到浮动元素可能导致布局混乱)

##利用伪元素

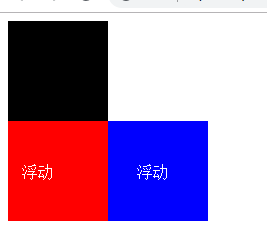
span::before{ content:""；display: block; clear: both }

兼容IE span{zoom:1}

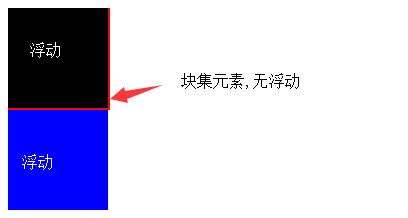
(3)浮动位置

##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. css权重(256进制) 同时选中一个元素权重值大的style覆盖权重值小的style
2. 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. 操作元素扩展

1>H5新增表单事件

(1)oninput 只要内容改变就会触发该事件

(2)oninvalid 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

##data-(标签内设置时), js中获取其属性驼峰式获取

1. H5新增接口

1>网络接口(事件, 与click用法一样)

(1)online 网络连接触发

(2)onoffline 网络断开触发

注意：为了兼容问题一般使用addEventListener绑定

2>全屏接口(方法)

(1)requestFullScreen(); 开启全屏显示

(2)cancelFullScreen(); 退出全屏显示

(3)fullScreenElement; 是否全屏状态

注意：(1)三者都有兼容问题

##谷歌需加上webkit, -->div.webkitRequestFullScreen()

##IE需加上ms,

##火狐须加上moz,

##opera需加上o

(2)requestFullScreen()选中某个dom操作,

cancelFullScreen()和fullScreenElement;需对document操作

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

4>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)添加拖拽事件

##应用于拖拽元素上的事件(绑定在拖拽元素身上)

++ondrag 整个拖拽过程都会调用(持续触发)

++ondragstart 当拖拽开始时调用

++ondragleave 当鼠标离开拖拽元素时调用

++ondragend 当拖拽结束时调用

##应用于目标元素上的事件(绑定在目标元素身上)

++ondragenter 当拖拽元素进入时触发

++ondragover 当拖拽元素停留在目标元素时触发

++ondrop 当拖拽元素放在目标元素松开鼠标时触发

++ondragleave 当鼠标离开目标元素时触发

注意：想要ondrop事件触发不要阻止ondragover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 -->



1. 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

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. 存储
2. 存储
3. 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. 缓存

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. line-height: 行高(文字居中对齐), text-align: center;(文字水平对齐)
2. text-indent: number em 首行缩进number个字符
3. text-decoration: none; 去除下划线(可以定义上中下等划线)
4. verticle-align: middle (垂直方向的对其)文字和图片中线对其 注意和1区分
5. white-space: nowrap; 超出宽度部分强制不换行(默认值是换行normal)
6. text-overflow: ellipsis; 超出部分成...展示 (默认值是clip)

注意:5> 和6> 一般都是配合overflow: hidden使用的

1. text-indent(首行缩进)：数字em；（相对单位）
2. text-decoration:none(无);
3. cursor（光标样式）:help()/move(移动样式);
4. background-image:url(图片地址)；
5. background-size:数字px（设置图片的大小）
6. background-repeat:repeat(多余部分重复出现图片 默认的)/no-repeat（取消重复

出现）

14>background-position:x轴 y轴;(控制图片在容器的位置)

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) 同时选中一个元素权重值大的style覆盖权重值小的style

！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

1. 选择器的四种类型

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 { }

1. 盒模型的计算(box-sizing)

1>普通盒模型

(1)模型

content(内容) + padding(内边距) + border(边框) + margin(外边距)

(2)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

1. IE盒模型

(1)模型

content + padding + border + margin

(2)计算

##border-box值

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

注意：固定宽度后padding撑不开容器

##content-box值(默认值, 就是标准盒模型)

##inherit值(继承父集box-sizing属性)

1. 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

注意：一般使用定位都是父元素relative子元素absolute。

1. 浮动(设置浮动后会产生浮动流)

1>浮动元素的特点： 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

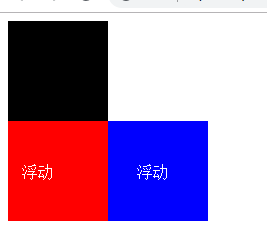
2>清除浮动方式

(1) 利用伪元素--> span::before{ content:""；display: block; clear: both }

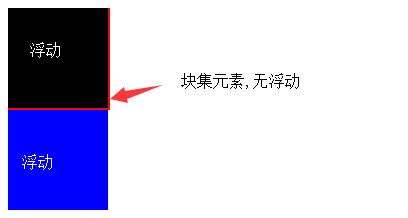
兼容IE span{zoom:1}

3>兄弟集浮动问题

(1)第一个是块集元素浮动设置第二个, 第三个为浮动浮动元素会在第二行显示-->



(2)第一个是浮动元素, 第二个是块集元素, 第三个是浮动元素, 块集元素会被第一个浮动元素挡住在第一行显示-->



1. 触发bfc的四种方式(可解决margin塌陷问题)

1>position:absolute

2>display:inline-block

3>float:left/right

4>overflow:hidden(逸出部分隐藏)

1. 常见css属性问题

1>行高line-height

会继承, 不带单位代表当前盒子的多少倍(一般设在body) -->



表示当前盒子字体大小的1.5倍

2>字体(font-family)

(1)英文

(2)转义编码

(3)中文(不推荐)

3>常见的复合属性

(1)background: 图片地址 图片重复不重复 图片定位 / 图片尺寸 -->

background: url(“./images/bg.png”) no-repeat -60px -109px / 200px 200px;

1. 单位总结

1>绝对单位

(1)像素(px), 计算机屏幕上的一个点, 1px = 1/96in

2>相对单位

(1)百分比(%),

相对于父元素的宽度或者字体大小

(2)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

(3)rem(root em)

相对于html标签的字体大小

(4)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. @规则

1>@charset 设置样式表的编码

2>@import 导入其他样式文件

3>@meida 媒体查询

4>@font-face自定义字体

1. 引入字体图标步骤

1>下载字体包(阿里矢量图, 添加购物车下载)

2>引入字体文件

3>css定义字体, 定义字体名class

4>html写入字体且指定创建好的class