1. 语义化标签

1>article用来规定独立的自包含内容, 通常应有自身的意义, 论坛的帖子,报纸文章。

2>section定义文档中的节一个区域，如章节，页眉，页脚或者文档中的其他部分。

3>nav用来定义导航链接部分。

4>aside定义其所处内容之外的内容，如文章的注解。

5>header定义文档的页眉的如，介绍信息

6>footer定义页脚的如，文档作责，版权信息，使用条款链接，联系信息

7>hgroup对页面或者区段(section)的标题进行组合

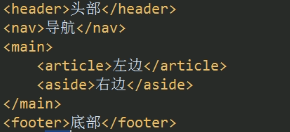
8>address定义文档或者文章作者或拥有者的联系信息(一般成斜体展示)

9>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

10>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。

注意：details一般配合summary使用 details的第一个子标签summary为下拉菜单的主题

1. 基本使用案例 -->



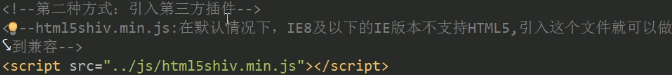
1. IE对H5的兼容问题
2. IE9以上版本也只是选择支持h5(会把语义化标签解析成inline元素, 兼容需要设置语

义标签设置为display: block)

1. IE8以及以下版本完全不支持h5(完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;)

1. 方便IE兼容方案(每次都要手动声明)



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. 新增其他标签
2. progress标签 (进度条) 其有max属性(最大进度)和value属性(当前进度)
3. 新增表单type属性

1>email 会自动验证 输入的内容必须包含@(合法邮箱)

2>tel不会自动验证(面向全球服务) 移动端自动弹出数字键盘

3>url 会自动验证 输入必须包含http://(合法网站)

4>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

5>search 可以提供更好的体验(input框右边有删除全部内容按钮)

6>range范围 -->  其有max属性和min属性

7>color 拾色器-->

8>time时间 时分秒-->

9>datatime日期时分秒 -->

注意：大部分不支持只有苹果的safari

10>file文件

11>month月week星期

1. Input表单新增的属性

1>placeholder=”要提示的内容” -->

2>autofocus=”on”(打开自动聚焦) off关闭

3>autocpmplate=”on”(打开历史记录) off关闭 注意:需要按提交按钮才会记录

4>required要求输入

5>pattern=”正则验证”(一般配合4>使用且用于验证手机号) -->



6>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”)

注意：一个input框输入多个邮箱时, 号分开

7>from=”form标签的id” 指定控件提交到哪个表单

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

1. 新增事件

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 -->



1. 音频和视频

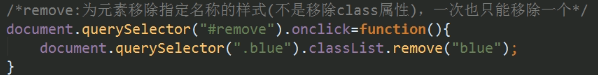
自动播放等属性查看文档

1. 新增操作元素样式

1>.classList.add(“添加的style(只能一个(需要多个可重复创建))”) 添加样式-->



2>.classList.romove(“删除的style(只能一个(需要多个可重复创建))”) 删除样式-->



3>.classList.toggle(“替换的style(只能一个(需要多个可重复创建))”)

4>.classList.contains(“”)

5>.classList.item(index)找属性

1. 自定义属性
2. 规范

(1)data-开头

(2)data-后面必须最少一个字符

2>获取其属性

(1)选中定义属性的元素

(2)dom.dataset[“自定义属性名(data不需要)”] 符合单词大写

1. 新增的接口
2. 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

1. 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

1. requestFullScreen()选中某个dom操作,

cancelFullScreen()和fullScreenElement;需对document操作

1. 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

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));

实例-->



5>地图接口 --详细查看第二天笔记

(1)基本形式 -->

navigator.getCurrentPosition(successCallback, errorCallback, options)

获取当前地理信息

(2)调用第三方接口

1. css3的常用属性

1>css3的边框属性

(1)border-radius把边框变成圆角----->border-top-left-radius:20px;

(2)box-shadow 向边框添加一个或 个阴影

(3border-image 用图像来绘制边框 :url(图片地址)

2>2D变形

transform中的方法---------->transform:translate(10px, 20px);

(1)translate(x轴偏移，y轴偏移 单位为px或者left这样的具体英文) 偏移

(2)rotate(旋转多少度单位为deg) 旋转

可以设置中心点为transform-orige:x轴上的点，y轴上的点，默认为0,0(方块中心)

(3)scale(x轴上缩放, y轴上的缩放 无单位, 一个值时表示x轴和y轴的缩放) 缩放

(4)skew(x轴上的倾斜，y轴上的倾斜 单位为deg) 倾斜

(4)\*\*\*Matrix(a,b,c,d,e,f);

3>3D变形

(1)ranslate3d(里面的值不可以是百分比),translateZ()3D位移

(2)rotate3d(x轴, y轴, z轴, 度数)x,y,z为矢量值,rotateX(),rotateY(),rotateZ()3D旋转

(3)Scale3d(),scaleZ()3D缩放

(4)Matrix3d()4\*4矩阵的3D转变

4>3D变形的转化属性

(1)tranceform-style:flat(指定元素呈2D平面内且是默认属性)/preserve-3d(指定元素定位在三维空间内)

注意:需要设置在需要style属性的父元素上

(2)Perspective:none/数值 定义距离值单位为px一个数值，如800距离800

(3)perspective-orige:百分比/数值/left等具体数值 定义观看点的位置 两值x轴y轴

(4)Backface-visibility:visible(可见，默认值)/hidden(不可见) 定义元素不面对屏幕是否可见

(5)Transform-origin-z:控制三维空间的中心点。

5>过度

transition中的方法-------->transition-property:width;定义对宽度进行过度

(1)-property: 检索或设置对象中的参与过度的属性如width对宽度进行过度

(2)-duration: 持续时间如2s 两秒运行完成

(3)-timing-function: 运动曲线速度如linear 规定以相同速度开始至结束的过渡效果。

(4)-delay: 至几秒开始运行如2s两秒后开始运行

可以把这些简化写 -->transition:width 2s linear 2s,height 2s linear 2s;

6>动画

animation中的方法------->animation-name:@keyframes定义的属名称,时间。

(1)-name:keyframes 名称{from从{} to{}什么}/{0%{}25%{}}

(2)-duration:时间 定义一个周期几秒完成

注意：上面这两个属性定义动画时一定要写。

可简称成 animation:nkeyframes名, duration的值,其他的属性值;

(3)-timing-function:规定速度曲线，默认值为ease

(4)-delay:规定动画何时开始，默认值为0

(5)-Iteration-count:规定动画播放次数，默认值为1

(6)-direction:规定下个周期逆向的播放，默认值为”normal”。

总结：

1. position:absolute;和float:left/right;会把元素转化成行级块元素inline-block；
2. 定位居中

position:absolute; top=50%; left=50%; margin-top=-height/2 margin-left=-width/2

1. body默认的margin为8px。