目录

[一、 HTML5简介 4](#_Toc458280013)

[1．HTML5特性 4](#_Toc458280014)

[2．Html5 文档类型 <!DOCTYPE html5>的必要性。 4](#_Toc458280015)

[3．lang与charset区别 4](#_Toc458280016)

[4．常见浏览器引擎及厂商前缀 4](#_Toc458280017)

[5．路径问题 5](#_Toc458280018)

[6．元素属性href 和 src 路径问题及区别 5](#_Toc458280019)

[二、HTML5 常用技术点 5](#_Toc458280020)

[1．DOM对象的classList属性 5](#_Toc458280021)

[2．自定义属性：data- 6](#_Toc458280022)

[3．getBoundingClientRect()方法 6](#_Toc458280023)

[4．懒加载（lazy loading） 6](#_Toc458280024)

[5．CSS3函数attr() 6](#_Toc458280025)

[6．伪元素与元素及伪类的区别 6](#_Toc458280026)

[7．CSS3 calc() 计算函数 7](#_Toc458280027)

[8．CSS3颜色函数rgba() 7](#_Toc458280028)

[9．CSS3计数函数counter() 7](#_Toc458280029)

[10．CSS3选择器 7](#_Toc458280030)

[11．其他知识点 8](#_Toc458280031)

[三、CSS3文本 8](#_Toc458280032)

[1．文本换行 8](#_Toc458280033)

[2．文本阴影 9](#_Toc458280034)

[3．文本溢出 9](#_Toc458280035)

[四、CSS3边框 10](#_Toc458280036)

[1．边框颜色属性 10](#_Toc458280037)

[2．图片边框属性 10](#_Toc458280038)

[3．圆角边框属性 11](#_Toc458280039)

[4．盒子阴影 11](#_Toc458280040)

[5．元素外轮廓 12](#_Toc458280041)

[6．调整元素盒子大小 12](#_Toc458280042)

[五、CSS3渐变 12](#_Toc458280043)

[1．线性渐变 12](#_Toc458280044)

[2．径向渐变 13](#_Toc458280045)

[六、CSS居中 13](#_Toc458280046)

[1．水平居中 13](#_Toc458280047)

[2．垂直居中 13](#_Toc458280048)

[七、CSS3背景 14](#_Toc458280049)

[1．背景原点属性 14](#_Toc458280050)

[2．背景裁剪属性 14](#_Toc458280051)

[3．背景尺寸属性 14](#_Toc458280052)

[4．多背景 15](#_Toc458280053)

[八、CSS3动画 15](#_Toc458280054)

[1．过渡（transition） 15](#_Toc458280055)

[2．变换（transform） 15](#_Toc458280056)

[3．动画（animation） 16](#_Toc458280057)

[九、3D 18](#_Toc458280058)

[1．transform-style:preserve-3d 18](#_Toc458280059)

[2．transform 3D相关函数 18](#_Toc458280060)

[十、HTML5语义元素 18](#_Toc458280061)

[1．语义的意义 18](#_Toc458280062)

[2．常用布局语义元素 18](#_Toc458280063)

[3．IE兼容 19](#_Toc458280064)

[十一、HTML5语义表单 19](#_Toc458280065)

[1．表单method分类 19](#_Toc458280066)

[2．所有input元素均是inline-block元素 20](#_Toc458280067)

[3．<input>输入类 20](#_Toc458280068)

[4．<input>选择类 20](#_Toc458280069)

[5．新属性 20](#_Toc458280070)

[6．新的元素 21](#_Toc458280071)

[7、验证 21](#_Toc458280072)

[8．相关知识点 21](#_Toc458280073)

[十二、离线存储 21](#_Toc458280074)

[1．Cookie 21](#_Toc458280075)

[2．Httpsession 22](#_Toc458280076)

[3．sessionStorage 22](#_Toc458280077)

[4．localStorage 22](#_Toc458280078)

[5．同源（origin） 22](#_Toc458280079)

[十三、伸缩布局盒模型 23](#_Toc458280080)

[1．特点 23](#_Toc458280081)

[2．专有名词 23](#_Toc458280082)

[3．属性 23](#_Toc458280083)

[十四、布局 25](#_Toc458280084)

[1．table-cell三列布局 25](#_Toc458280085)

[2．Calc函数三列布局 25](#_Toc458280086)

[3．右侧固定布局 26](#_Toc458280087)

[4．侧边栏 27](#_Toc458280088)

[5．两列 27](#_Toc458280089)

[6．Column 27](#_Toc458280090)

[十五、多媒体特性与响应式布局 28](#_Toc458280091)

[1．媒体 28](#_Toc458280092)

[2．响应式布局与自适应布局 28](#_Toc458280093)

[3．视口(viewport) 28](#_Toc458280094)

[4．@media选择器 29](#_Toc458280095)

[5．单位rem 29](#_Toc458280096)

[6．渐进增强和优雅降级 30](#_Toc458280097)

[十六、Bootstrap 30](#_Toc458280098)

[1．布局容器class 30](#_Toc458280099)

[2．行(.row) 30](#_Toc458280100)

[3．列(.col) 30](#_Toc458280101)

[4．字体图标 31](#_Toc458280102)

[十七、Less 31](#_Toc458280103)

[1．简介 31](#_Toc458280104)

[2．变量 31](#_Toc458280105)

[3．函数 31](#_Toc458280106)

[4．选择器 31](#_Toc458280107)

[5．注释 32](#_Toc458280108)

# HTML5简介

## 1．HTML5特性

1.1．语义化（ Semantics ）

可以更精确的描述HTML文档内容，清晰的呈现HTML的文档结构。

1.2．连通性（ Connectivity ）

使用新的方式与服务器后台通信。

1.3．离线存储（ Offline and storage ）

网页可以把数据缓存到本地，高效的操作离线数据。

1.4．多媒体（ Multimedia ）

使用<video>和<audio>实现音频视频播放。

1.5．2D和3D图形渲染（ 2D/3D graphics and effects ）

提供更多的呈现方式。

1.6．性能与集成（ Performance and integration ）

提供更快的速度优化和更好的使用计算机硬件的方式。

1.7．设备访问（ Device access）

允许使用各种输入和输出设备。

1.8．样式（ Styling ）

让开发者可以开发更复杂的页面。也就是CSS3.

## 2．Html5 文档类型 <!DOCTYPE html5>的必要性。

2.1．告诉浏览器基于哪种规范解析HTML。

2.2．如果不添加文档类型，会触发IE7及以下进入怪异模式。

正常模式：width = content

怪异模式：width = content + padding + border

## 3．lang与charset区别

3.1．lang

设置语言，使用什么人的语言编写的（中国人等）

3.2．charset

字符编码，改变字符编码可能会导致字符乱码

## 4．常见浏览器引擎及厂商前缀

|  |  |  |
| --- | --- | --- |
| **浏览器** | **内核** | **厂商前缀** |
| **IE** | **trident** | **-ms-** |
| **Chrome** | **blink** | **-webkit-** |
| **Opera** | **blink** | **-o-** |
| **Safari** | **webkit** | **-webkit-** |
| **Fierfox** | **gecko** | **-moz-** |

判断浏览器内核

navigator.userAgent

str.indexOf("keyword")>-1

注意：最好根据需要判断某一特性是否支持该浏览器

## 5．路径问题

5.1．网址

5.2．相对URL，以/开头

从网址下项目的根目录开始寻找

5.3．相对URL，不以/开头

从当前文件所在目录开始寻找

## 6．元素属性href 和 src 路径问题及区别

6.1．Web资源： html, css, js, img(png ,jpg)…..

示例

<link href=”“>

<img src=””>

<script src=””>

6.2．Href

异步加载web资源。如果网速慢，会造成网页布局混乱（CSS未加载出来）。

6.3．Src

同步加载web资源，加载完src才继续加载下面代码。若网速慢，会造成网页白屏。

# 二、HTML5 常用技术点

## 1．DOM对象的classList属性

1.1．作用：获取元素所有的class名

1.2．属性：length：class的长度

1.3．方法

--add(value)：添加class方法

--contains(value)

--remove(value)：删除class方法

--toggle(value)：切换class方法

## 2．自定义属性：data-

2.1．介绍：以data-开头的属性。

是将自定义的数据存储到页面或应用程序中，这些数据不是用来呈现给用户的。

2.2．属性：dataset

2.3．示例

<div id=" attr "data-md-width="140px"></div>

读取值：attr . dataset . mdWidth

设置新值：attr. dataset. mdWidth = “26px”;

## 3．getBoundingClientRect()方法

3.1．作用：获取元素距视窗顶部和左边的位置

3.2．返回值：元素左上角的位置及元素本身的height、width

top、bottom、left、right、height、width

## 4．懒加载（lazy loading）

当图片出现在可视区域中才加载图片

## 5．CSS3函数attr()

5.1．需要与伪元素的：content一起使用

示例

a:after{

content: " [" attr(href) "] " ;

}

5.2．试验阶段语法

/\* 指定属性类型 \*/

attr(src url);

attr(data-count number);

attr(data-width px);

/\* 设置回退方案 \*/

attr(data-count number, 0);

attr(src url, '');

attr(data-width px, inherit);

attr(data-something, 'default');

## 6．伪元素与元素及伪类的区别

6.1．伪元素VS元素：

不能被选中，不能通过JS获取

在页面显示：::before ::after

伪元素可以被元素替换。

6.2．伪元素VS伪类：

伪类是控制元素的显示状态的。

不能被其他元素替换。

## 7．CSS3 calc() 计算函数

动态计算尺寸，可以使用+-\*/混合运算，IE9以下不支持，常用于布局

calc()内部嵌套另一个calc()，但仅欧朋浏览器支持

必须在运算符左右加空格。

在低版本的浏览器使用，需要增加浏览器厂商前缀。

## 8．CSS3颜色函数rgba()

8.1．rgba(颜色值，颜色值，颜色值，透明度)

8.2．opacity VS rgba() VS transparent

--rgba()

颜色透明度

--opacity

元素及其内容，子元素透明度。

--transparent = rgba( 255, 0, 0, 0)

## 9．CSS3计数函数counter()

9.1．counter-reset: counter-name ;

counter()元素的父元素中申明了一个变量，且初始值为0

9.2．counter-increment: counter-name ;

使用前自增1，在子元素中写

9.3．content: counter( counter-name ); //用在无序列表里

把自增后的数据给content属性

9.4．content: counters( counter-name, “.”); //用在有序列表里

二级自增

## 10．CSS3选择器

10.1．:empty

选择无文本及子元素的元素

10.2．:target

选择被其他元素作为目标的元素

示例：

<a href="#id"></a>

#id 对应的元素

## 11．其他知识点

11.1．box-sizing

属性值：

border-box

等同于开启IE7以下的怪异模式

即width = 内容区+padding+border

content-box

正常模式

即width = 内容区

11.2．解析选择器：从右往左，因此尽量只写一层选择器

# 三、CSS3文本

## 1．文本换行

1.1．word-wrap

1.1.1．介绍：浏览器为了防止文本溢出，是否可以在单词内换行。

1.1.2．属性值：

--normal

默认情况，浏览器只在“空格”或者“半角”的位置换行。

--break-word

浏览器自行决定在何处“截断”单词。默认情况是禁止浏览器截断单词。

IE6及以上浏览器对该属性支持。

1.1.3．在W3C中已被overflow-wrap代替。目前只有Chrome实现了该属性

1.2．word-break(忽略换行，将换行当作空格处理)

1.2.1．介绍：让文本在任何位置换行

1.2.2．属性值：

--normal

默认，根据语言自己的规则确定换行方式，中文到边界上的汉字换行，英文整个单词换行。

--break-all

为了保证每一行的空间不浪费，强行“截断”英文单词。

--keep-all

不准许“截断”。如果是中文把前后标点符号（全角）后开始换行，英文单词整个换行。

1.2.3．IE6及以上浏览器对该属性支持。

1.3．white-space

1.3.1．介绍：处理元素中的空白符（CSS3新增）

1.3.2．属性值：

--normal

默认值。空白处会被浏览器忽略（即多余空格被浏览器删除）。只保留正常的空格。

--pre：

文本空白会被浏览器保留。类似<pre>效果。

--nowrap:

文本不换行，文本在同一行显示，直到遇到<br/>位置。空白处会被浏览器忽略

--pre-line:

与normal类似，空白处会被浏览器忽略。不同点是保留换行符。

IE7及以下浏览器不支持此属性。

--pre-wrap:

保留空白符序列，换行单独一行显示。

IE7及以下浏览器不支持此属性。

--Inherit:

继承父元素的值。

IE所有浏览器均不支持此属性。

## 2．文本阴影

2.1．text-shadow

2.1.1．参数：offset-x | offset-y | blur-radius | color

--offset-x：X轴偏移量

--offset-y：Y轴偏移量

--blur-radius：模糊半径

--color：阴影颜色

2.1.2．可在text-shadow中设置多个值（x|y|blur|color,x|y|blur|color ……）来设定多个阴影

2.2．IE8及以下浏览器兼容方案（一般不用）

fliter: shadow([Color], [Direction], [Strength]);

--Color：阴影颜色

--Direction：设定投影的方向。0 阴影在文本上；45度阴影在文本右上角。

--Strength

## 3．文本溢出

text-overflow

3.1．属性值：

--ellipsis

当文本溢出时，省略号显示。

告诉用户文本未显示完，提示用户点进去观看，提升了用户的友好度

与title搭配使用，title中写完整内容

--clip

仅仅简单的剪裁，不使用省略号。

3.2．上述两个属性需要配合以下一起使用，且容器需要定义宽度。只在块元素内生效，即可设宽度元素才有效果。

white-space: nowrap;

overflow: hidden;

# 四、CSS3边框

## 1．边框颜色属性

1.1．border-(top|right|bottom|left)-color

分别设置元素四个边框的颜色。由于浏览器兼容性问题，该属性极少在商业网站中使用。处于实验室阶段。

简写方式，border-color:颜色

1.2．为边框设置多个颜色（仅Firefox浏览器支持）

1.2.1．写法：

-moz-border-top-colors：[color]{1,n}

-moz-border-right-colors：[color]{1,n}

-moz-border-bottom-colors：[color]{1,n}

-moz-border-left-colors：[color]{1,n}

需要配合border: 1px solid使用

1.2.2．注意事项：

上述属性仅在Firefox3.0以上有效，它不是标准语法。而且目前仍在实验阶段，实际使用一定要慎重。

上述样式的优先级比border-[top|right|bottom|left]-color优先级高。

不支持动画效果

## 2．图片边框属性

2.1．介绍：浏览器支持较好，但商业网站应用很少，主要用在个人博客中。

2.2．border-image-source:url()

设置图片来源

2.3．border-image-slice:[<percentage>|<number>]｛1,4｝&&fill

百分比值或纯数字（不需要写单位，默认为px）。

表示四方向切割图片的位置。

高版本chrome需要加border-style:solid才可显示

2.4．border-image-width:[<length>]｛1,4｝

用来表示背景图片显示的大小，与border-width设置效果相同;

2.5．border-image-outset:[<length>]｛1,4｝

图片外边框的位置，只能为整数，其大小的改变不影响盒子的大小。

2.6．border-image-repeat

2.6.1．平铺边框

2.6.2．属性值：

--stretch:默认值，拉伸

--round:填满，常用此值

--repeat:重复

2.7．简写方式

border-image:<source>||<slice>[/<width>||<repeat>]

## 3．圆角边框属性

3.1．介绍：圆角边框属性，浏览器支持较好，应用非常广泛。

3.2．写法：

border-radius：top-left/bottom-right/top-right/bottom-left

border-radius：top-left/bottom-right , top-right/bottom-left

border-radius：top-left , top-right/bottom-left , bottom-right

border-radius：top-left/top-right/bottom-right/bottom-left

border-radius：value1**/**value2

value1水平方向半径，value2垂直方向半径

3.3．IE兼容问题

3.3.1．使用插件PIE.js文件

PIE.js文件支持：border-radius：圆，box-shadow：阴，border-image：图片边框

3.3.2．通过<!-[if ]><![endif]>导入

3.3.3．编写样式的同时写入

behavior：url(../PIE/PIE.htc);

## 4．盒子阴影

4.1．介绍：

CSS3盒子阴影属性，应用非常广泛，主要用来呈现元素立体效果

4.2．写法：

box-shadow：none或inset x-offset y-offset blur-radius spread-radius color

可写多个阴影即多个inset x-offset y-offset blur-radius spread-radius color

4.2．属性值：

None：默认值，不设置阴影。

inset：阴影类型，可选。设置inset为内阴影，否则为外阴影

x-offset：阴影水平偏移。正值，阴影在右侧，负值在左侧。

y-offset：阴影垂直偏移。正式，阴影在底部，负值在顶部。

blur-radius：阴影模糊半径，可选。只能为正值，如果设为0，不具模糊效果。

spread-radius：阴影扩展半径。可选。正值，阴影扩大；负值，阴影缩小。

color：阴影颜色。可选

## 5．元素外轮廓

5.1．元素外轮廓

outline-color外轮廓颜色

outline-style外轮廓样式

outline-width 外轮廓宽度

简写：outline：color,style, width

5.2．outline-offset

正值：外偏移；负值：内偏移

5.3．外轮廓不会改变元素盒子大小

## 6．调整元素盒子大小

resize:

none：用户不能调整元素宽高。

both：用户可以调整元素宽高。

horizontal：用户只能调整元素宽度

vertical：用户只能调整元素高度

inherit：默认继承

注意事项：

resize必须搭配overflow：auto;一起使用

# 五、CSS3渐变

## 1．线性渐变

1.1．linear-gradient(角度/to方向,colo,color,……..)

用法：将此函数设置给background-img，默认从上到下过渡

角度：单位deg，按顺时针旋转，默认角度180deg

颜色：后接数值指该颜色过渡开始的起始位置

如果指定过渡位置，那么它之前的空间全部由该颜色填充

1.2．repeating-linear-gradient ()

重复渐变

参数：方向，颜色1开始位置，颜色1结束位置，……..

开始位置--结束位置=宽度，颜色2开始位置=颜色1结束位置

示例：

background:repeating-linear-gradient(to right,white,white 25px, black 25px,black 50px);

1.3．渐变本质：创建了一张图片，均匀过渡，颜色不失真

## 2．径向渐变

2.1．radial-gradient(color, color……)

参数：

circle/ellipse 半径 at圆心X值 圆心Y值,颜色 渐变开始位置,……..

circle：圆形

ellipse：椭圆，默认按椭圆渐变，因此可不写，半径同时写X轴Y轴

关键字：设置在circle/ellipse后

closest-side：渐变的圆心距离元素最近（两）边的距离作为径向渐变半径。

closest-corner：渐变的圆心距离元素最近角（两）边的距离作为径向渐变半径。

farthest-side：渐变的圆心距离元素最近（两）边的距离作为径向渐变半径。

farthest-corner：渐变的圆心距离元素最近角（两）边的距离作为径向渐变半径

2.2．repeating- radial -gradient ()

用法基本同线性渐变

# 六、CSS居中

DIV容器元素（布局元素）

## 1．水平居中

Inline/Inline-block

Text-align: center

Block

margin: 0 auto;

## 2．垂直居中

Inline

字体大小会影响inline元素（文本）的位置。

多个inline元素，以字体最大元素基线作为该行的基线。

Inline-block

Inline-block元素与同一行的元素基线相同

Vertical-align: middle指inline-block基于基线所在的位置。

没有内容的Inline-block元素会影响基线的位置。

Img就是一种无内容放入Inline-block元素

Block

使用定位居中：使用top、bottom值设定居中

在margin中通过计算函数calc()计算出居中位置

增加一个父元素设置其display:table，并将需要居中的元素设置为display:table-cell

通常子元素不设置高度，直接设置父元素margin来使子元素垂直居中

# 七、CSS3背景

## 1．背景原点属性

1.1．background-origin: 背景图片左上角位置 背景图片绘制原点

1.2．属性值：

padding-box;（默认值）

背景图左上角与元素padding最外侧左上角保持一致。

border-box;

背景图左上角与元素外边框左上角保持一致。

content-box;

背景图左上角与元素内容区content左上角保持一致。

1.3．background-position也会影响左上角位置

## 2．背景裁剪属性

2.1．background-clip:定义背景图片的切割范围

2.2．属性值：

border-box;（默认值）

元素外边框外的背景图片都将被裁裁剪掉。

padding-box;

元素padding区域外的背景图片都将被裁裁剪掉。

content-box;

元素内容区域外背景图片都将被裁裁剪掉。

2.3．-webkit-background-clip: text;

-webkit-text-fill-color: transparent

使用图片填充文本

与background-image搭配使用，里面可以使用渐变函数

只有webkit内核浏览器可以使用

## 3．背景尺寸属性

3.1．background-size:设置背景图片尺寸

3.2．属性值：

auto

默认值，背景图保持原样。

整数值或百分百

如果只设置一个值，代表背景的宽度，第二个值为auto（根据图片宽度等比计算高度）。百分百相对于背景图所在元素的宽度计算。

cover

背景图片填满整个元素，但不居中显示

如果需要居中显示需要设置background-position: center;。

contain

保持背景图片自身的宽高比例。

## 4．多背景

将每个属性单独写，使用“，”分割控制多张背景图

# 八、CSS3动画

## 1．过渡（transition）

1.1．介绍：让CSS样式从初始状态平滑地过渡到结束状态。不是所有的CSS样式都具备过渡特性。用数值控制的属性都可以进行过渡。

1.2．效果生效：鼠标单击、获得焦点、被点击或对元素任何改变

1.3．transition-property

设置过渡的属性，可写多个属性，默认all

参数：属性名

1.4．transition-duration

设置过渡的持续时间

参数：浮点型+s

1.5．transition-timing-function

过渡速度控制

liner：线性

ease：先慢后快

ease-in：以低速开始

ease-out：以低速结束

ease-in-out：以低速开始、介绍

cubic-bezier(n,n,n,n)：贝塞尔曲线，控制过渡变化速度、控制过渡回弹

1.6．transition-delay

控制动画延迟执行 单位s

可同时写多个控制多个property中的属性，能让样式过渡分开执行

1.7．简写：transition

最好固定顺序：property，duration，timing-function，delay

1.8．维持过渡结束状态

通过JS动态添加结束状态样式（使用classlist方法）

## 2．变换（transform）

2.1．介绍：

让该元素产生旋转、缩放、移动或倾斜倾斜。由transform导致的形状改变及位置变化

2.2．斜切

skew(值)：默认，左右两边逆时针斜切

skewX(值)：水平斜切，左右两边逆时针斜切

skewY(值)：垂直斜切，上下两边顺时针斜切

skew(X度,Y度)

会导致所有子元素包括文本等均倾斜

2.3．移动

translate(值)：默认，沿X轴移动

translateX(值)：沿X轴移动，正值向右移

translateY(值)：沿Y轴移动，正值向下移

translate (水平位移,垂直位移)

2.4．旋转

rotate()：默认Z轴旋转，中心点（50%，50%）为旋转圆心，正值为顺时针

rotateX()：X轴旋转

rotateY()：Y轴旋转

transform-origin: par1 par2控制旋转圆心，默认为元素中心点

par1: 相对于元素左上角的水平方向距离

par2：相对于元素左上角的垂直方向距离

2.5．等比缩放

scale(n)：X、Y轴等比缩放n倍

scale(n1, n2)：X轴等比缩放n1倍，Y轴等比缩放n2倍

运用：chrome不支持font-size=12px以下，可使用scale(.8)来处理，但必须将元素设置为非inline元素

2.6．简写

transform：斜切、移动、旋转、等比缩放函数

## 3．动画（animation）

3.1．介绍：Animation 是类似flash 逐帧播放动画。

3.2．@keyframes animName{}

定义关键帧，指定动画的名字

{}中使用n%{书写样式}

n%：播放时长的百分之n开始播放{}中的样式，可合并书写n1%,n2%{}

播放后样式回归播放前设定的样式

3.3．动画属性

3.2.1．animation-name: animName

指定动画的关键帧名称

3.2.2．animation-duration: n s

设定播放时长，默认0s

3.2.3．animation-timing-function：规定动画的速度曲线，默认为ease

3.2.4．animation-delay: n s

设置延迟执行，默认0s

3.2.5．animation-iteration-count: 1 / infinite

设定播放次数，默认1次，infinite：无限循环

3.2.6．animation-direction：控制轮播方向

属性值：

normal：逐帧顺序播放

reverse：逐帧反向播放

alternate：顺讯循环播放

alternate-reverse：反向循环播放

3.2.7．animation-fill-mode：

属性值：

none：默认不改变行为

forwards：动画结束后保留最后一个属性值

backwards：动画开始前即delay所指定的那段时间内应用第一个属性值

both：forwards、backwards均被使用

3.2.8．animation-play-state

属性值：

running设置动画播放

paused设置动画暂定

3.4．JS动画与CSS3动画区别

3.4.1．所有transform变换不会导致页面布局发生改变

Reflow回流：CSS3动画能避免触发回流，不会导致浏览器重新计算页面元素位置及宽高

3.4.2．CSS3动画会开启GPU加速，防止出现JS动画会出现的闪屏

问题：会消耗手机电量，因此在移动端避免过度使用动画效果

# 九、3D

## 1．transform-style:preserve-3d

建立3D空间

## 2．transform 3D相关函数

rotateX()：沿X轴旋转

rotateY()：沿Y轴旋转

rotateZ()：沿Z轴旋转

transform-origin:参数1,参数2,参数3

设置圆心，一般设置为center

translateZ()：沿Z移动元素（父元素必须开启 preserve-3d）

scaleZ()：沿Z轴等比缩放元素

# 十、HTML5语义元素

## 1．语义的意义

程序利益相关方（程序员、浏览器、搜索引擎）能很好的协同工作

## 2．常用布局语义元素

2.1．<header>

网页或section的页眉，其中放h1-h6、logo、hgroup、用户相关链接

没有个数限制。如果hgroup或h1-h6自己就能工作的很好，就不用header

2.2．<footer>

网页或section的页脚，其中放联系信息、常用链接等。没有个数限制

2.3．<hgroup>

网页或“section的标题

如果只需要一个h1-h6标签就不用hgroup，如果有连续多个h1-h6标签就用hgroup

如果有连续多个标题和其他文章数据，h1-h6标签就用hgroup包住，和其他文章元数据一起放入header标签。

2.4．<nav>：导航条

2.5．<aside>

配合article使用，是article的附属信息部分，例：侧边栏

2.6．<section>

文档中的节、段

一张页面可以用section划分为简介、文章条目和联系信息。但在文章内页，最好用article。section不是一般意义上的容器元素，如果想作为样式展示和脚本的便利，可以用div。

article、nav、aside可以理解为特殊的section，若可以用article、nav、aside就不要用section，没实际意义的就用div

2.7．<article>

表示文档中的节或者段，代表一个在文档，页面或者网站中自成一体的内容，其目的是为了让开发者独立开发或重用

2.8．<progress>

显示一项工作做完成的进度

属性：

max

大于0的浮点型数字，表示总工作量

value

0-max的任意数字，表进度值

Position（只读属性）

返回double值，即value/max

若进度条未设置max和value返回-1

2.9．<address>

表联系信息

独占一行，自动斜体，子元素不会自动换行

## 3．IE兼容

使用插件兼容IE8及以下

<script src="js/html5shiv.js" type="text/javascript"></script>

# 十一、HTML5语义表单

## 1．表单method分类

1.1．get

输入的表单数据会显示在地址栏

1.2．post

输入的表单数据不会显示在地址栏，隐藏提交，相对较安全

## 2．所有input元素均是inline-block元素

## 3．<input>输入类

3.1．email：电子邮件，只验证是否存在@及@前后是否有内容

3.2．tel：电话号码，不做其他验证

3.3．url：网址，只验证http://

3.4．search：搜索框，比输入框多一个清除的X按钮

3.5．number：数字框，比输入框多上下按钮键

属性：

step = “n”：按上下按钮是每次加减n

min/max = “n”：限定上下按钮数字的范围

## 4．<input>选择类

4.1．range

特定范围内数值选择器

属性：

step = “n”：按按钮移动数值每次加减n

min/max = “n”：限定上下按钮数字的范围

value：默认50，范围为1-100

4.2．color：颜色选择器，value值为十六进制颜色

4.3．datetime-local：日期选择器，年月日及时间

IE、火狐均不兼容，只有chrome可用

4.4．time：时间选择器，只有chrome可用

4.5．date：日期选择器，只有chrome可用

4.6．week：星期选择器，一年的第几周，只有chrome可用

4.7．month：月份选择器，只有chrome可用

## 5．新属性

5.1．autocomplete

是否保存用户输入值，显示之前输入的内容。

默认为on显示，off不显示

5.2．autofocus

自动获取输入焦点

布尔型属性，属性值autofocus

5.3．required

必须输入值，不能为空值

布尔型属性，属性值required

5.4．pattern

验证正则表达式，相对安全

5.5．placeholder：输入框提示信息

5.6．formaction：在submit里定义提交地址。常见应用场景：保存草稿

Formnovalidate：不做表单验证。<input type="submit" formnovalidate=”formnovalidate”>

## 6．新的元素

<datalist id=”ID”> <option></option></datalist>

仅定义了一个数据源。必须配合input使用,<input list=”ID” />

Input中输入值时有过滤功能

## 7、验证

Form表单中的验证在浏览器开发者项目（F12）中修改，再进行提交是可以提交到后台的，因此在后台必须验证传入数据的正确性

## 8．相关知识点

8.1．input:required{}选择器

表必填项，当某元素为必填项时，显示特定样式

8.2．input:invalid{}选择器

当用户输错是显示的样式

8.3．onchange事件

输入值发生改变时触发

8.4．invalid事件：验证失败时触发

使用方法：inputElement.addEventListener(“invalid”，function(){}，false)

方法中this.validity：查看验证失败原因，失败的原因值为true，其他为false

8.5．自定义验证失败提示

oninput事件：监控输入信息，每输入一个字触发一次

事件函数中this.setCustomValidity(“自定义提示的信息”)，this指此输入元素

8.6．表单作用域

元素必须在form中才能进行提交

但给form设置id值，再给form外的input元素设置属性form=”设置的id”也可提交

# 十二、离线存储

## 1．Cookie

1.1．使用场景：记录（跟踪）用户信息，确保服务器能识别请求是不是来自一个会话

1.2．生成位置：服务器

1.3．存储位置：浏览器

1.4．生命周期：服务器自定义

1.5．大小：4KB

1.6．问题：需要在客户端和服务器端来回地传送，繁琐且消耗带宽

## 2．Httpsession

2.1．使用场景：服务器端用作数据存储

2.2．生成位置：服务器

2.3．存储位置：服务器

2.4．生命周期：页签关闭前

## 3．sessionStorage

3.1．生成位置：浏览器

3.2．存储位置：浏览器

3.3．生命周期：页签关闭前

3.4．大小：5MB

3.5．存储：sessionStorage.setItem(“name”,”value”)，name不能重复，否则会覆盖前一个值

3.6．获取值：sessionStorage.getItem(“name”)

3.7．删除值：sessionStorage.removeItem(“name”)

3.8．取第i个name：sessionStorage.key(i)

3.9．一共几个值：sessionStorage.length

3.10．清除所有值：sessionStorage.clear()

## 4．localStorage

4.1．生成位置：浏览器

4.2．存储位置：浏览器

4.3．生命周期：永久，直到用户手动清除

4.4．大小：5MB

4.5．语法同sessionStorage

## 5．同源（origin）

5.1．含义：指协议、域名、端口必须完全一致。

5.2．如果一个网址与另一个网址不同源，那么他们的sessionStroage和localStorage的数据是不能共享的。

5.3．Localhost = 127.0.0.1

# 十三、伸缩布局盒模型

## 1．特点

Flex容器元素可以改变子元素的排序。

Flex容器元素会使子元素伸缩来填满可用空间，或者缩小防止溢出容器

## 2．专有名词

2.1．伸缩容器：display: flex

2.2．伸缩项目: 不指定的固定width和height的子元素

2.3．主轴/主轴对齐：元素在水平方向排布

2.4．侧轴/测轴对齐：元素在垂直方向排布

## 3．属性

3.1．display

属性值：

flex：声明元素为伸缩容器

inline-flex：声明元素为行内伸缩容器

3.2．flex-direction：方向

属性值：

row：伸缩容器中伸缩项目水平方向从左到右排列，简称ltr

row-reverse：伸缩容器中伸缩项目水平方向从右到左排列，简称rtl

column：伸缩容器中伸缩项目垂直方向从上到下排列

column-reverse：声明伸缩容器中的伸缩项目垂直方向从下到上排列

3.3．flex-wrap：是否包裹伸缩项目

属性值：

nowrap：伸缩容器不包裹伸缩项目，伸缩项目会“溢出”

wrap：伸缩容器包裹伸缩项目。水平空间不够时，自动换行

wrap-reverse：伸缩容器反向包裹伸缩项目，与direction相反

3.4．flex-flow简写flex-direction和flex-wrap

3.5．justify-content：伸缩项目沿伸缩容器水平对齐

伸缩容器高度必须比伸缩项目高，否则没效果

属性值：

flex-start：默认，伸缩项目左侧对齐

flex-end：伸缩项目右侧对齐

center：伸缩项目居中对齐

space-between：均匀分布伸缩项目，间隙在伸缩项目中间

space-round：均匀分布伸缩项目，间隙围绕伸缩项目，左右侧均有间隙

3.6．align-items：伸缩项目沿伸缩容器垂直对齐方式

属性值：

flex-start：伸缩项目顶部对齐

flex-end：伸缩项目底部对齐

center：伸缩项目居中对齐

baseline：伸缩项目基线对齐

stretch：不指定高度时，伸缩项目垂直方向充满整个高度，指定了高度，则贴着此行顶部边，不会拉伸

3.7．align-content：内容垂直对齐

样式生效前提

伸缩容器固定高度，伸缩元素有两行及以上，伸缩元素flex-wrap:wrap

属性值：

flex-start：所有伸缩元素在伸缩容器中顶部对齐，两行间无间隙

flex-end：所有伸缩元素在伸缩容器中底部对齐

center：所有伸缩元素在伸缩容器居中对齐

baseline：所有伸缩元素在伸缩容器中基线对齐（仅firfox支持）

stretch：未指定高度的所有伸缩元素在伸缩容器占满整个屏幕，指定高度的伸缩元素贴着每行的边顶部，不会拉伸

3.8．align-self ：配合align-items使用，自定义伸缩元素垂直对齐方式

属性值：

flex-start：伸缩项目垂直顶部对齐

flex-end：伸缩项目垂直底部对齐

center：伸缩项目垂直方居中对齐

baseline：伸缩项目垂直基线对齐

stretch：未指定高度的所有伸缩元素在伸缩容器占满整个屏幕，指定高度的伸缩元素贴着每行的边顶部，不会拉伸

3.9．order：number

伸缩项目排序

所有指定order属性的伸缩元素重新按number排列

未设定order属性的伸缩元素使用默认排序，且排列在指定order属性的元素之前

3.10．flex-grow:number

伸缩项目占用的剩余空间比例

宽度=原宽度+除去伸缩项目宽度的伸缩容器宽度X number/number总和

原宽度不设置时为0

3.11．flex-shrink:number

伸缩空间不够时，伸缩项目收缩比例，默认1

压缩后宽度=原宽度--需要压缩的宽度X number/number总和

# 十四、布局

## 1．table-cell三列布局

.wrapper {

margin: 0 auto;

background-color: #f0f0f0;

display: table;}

.box {

background-color: #00CCFF;

display: table-cell;}

.box1{

background-color: red;}

.box2{

width: 100%;}

.box3{

background-color: green;}

<div class="wrapper">

<div class="box box1">11</div>

<div class="box box2"></div>

<div class="box box3">22</div>

</div>

## 2．Calc函数三列布局

.wrapper {

width: 100%;

margin: 0 auto;

background-color: #f0f0f0;

display: table;}

.box {

background-color: #00CCFF;

display: table-cell;}

.box1{

width: 200px;

background-color: red;}

.box2{

width: calc(100% - 400px);}

.box3{

width: 200px;

background-color: green;}

<div class="wrapper">

<div class="box box1">

<img src="1.png">

</div>

<div class="box box2"></div>

<div class="box box3">

<img src="1.png">

</div>

</div>

## 3．右侧固定布局

.col-3{

overflow: hidden;}

.col-2{

float: left;

width: 100%;

/\*background-color: red;\*/

margin-right: -200px;}

aside{

float: right;

height: 50px;

width: 200px;

background-color: blue;}

nav{

float: left;

height: 50px;

width: 200px;

background-color: green;}

article{

height: 55px;

width: auto;

background-color: yellow;

margin: 0 200px;}

<div class="col-3">

<div class="col-2">

<nav></nav>

<article></article>

</div>

<aside></aside>

</div>

## 4．侧边栏

html, body {

height: 100%;

margin: 0;}

.box {

width: 200px;

background-color: #00CCFF;

float: left;

height: 100%;

/\*position: fixed;

top: 0;

left: 0;

bottom: 0;\*/}

<div class="box"></div>

## 5．两列

.col-left {

width: 200px;

height: 200px;

background-color: #00CCFF;

float: left;}

.col-right{

height: 200px;

background-color: red;

overflow: hidden;}

<div class="col-left"></div>

<div class="col-right"></div>

## 6．Column

6.1．column-count：number

将内容分为number列，只用于排版，不可用于布局

6.2．column-width：number px

指定每列的宽度

6.3．column-span：all

规定元素应该横跨的列数，all：横跨所有列

6.4．column-rule：style，color，width

指定列与列之间的边框，用法同border

6.5．column-gap：number px

指定列与列之间的间隙

# 十五、多媒体特性与响应式布局

## 1．媒体

1.1．概念

即媒体特性，指手机、pad、打印机、PC电脑、盲人触屏设备等用来呈现网页的显示器

1.2．meta标签

告诉浏览器如何解析或设置网页

## 2．响应式布局与自适应布局

2.1．响应式布局

一套HTML代码，适应不同的设备屏幕（屏幕大小）。不同的宽度的屏幕显示的结构不同

2.2．自适应布局

一套HTML代码，网页随屏幕稍微变化而变化，不会存在网页结构的调整

2.3．设置图片自适应

img{

display:block;//设置独占一行

width:100%

max-width:100%;}//保证手机一定能用

2.4．响应式设计技巧

2.2.1．尽量少用无关紧要的div

2.2.2．不适用内联元素（不能设置宽度）

2.2.3．尽量少用JS或flash（加载太慢，可能会出现卡顿）

2.2.4．丢弃无用的相对/绝对定位

2.2.5．舍弃任何冗余结构

## 3．视口(viewport)

3.1．概念

在移动端，视口的宽度是网页的宽度

3.2．用法

<meta name=”viewport” content=””>

3.3．属性值

width：可视区域宽度，number或device-width。

height：可视区域高度，number或device-height。

inital-scale：页面首次显示时，可视区域的缩放比例，取值1.0页面按实际比列显示

minimun-scale：可视区域最小缩放级别，取值1.0禁止用户缩放至实际尺寸以下。

maximun-sacle：可视区域最大缩放级别，取值1.0禁止用户放大至实际尺寸以上。

user-scalable：用户是否可以对页面进行缩放。yes可以缩放，no禁止缩放。

3.4．移动端网页设置的meta

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">

## 4．@media选择器

4.1．介绍

监控设备尺寸，通过尺寸判断设备类型（PC、pad、mobel）来判断设备类型，实现3端不同样式显示

4.2．属性值

all：默认全部

screen：屏幕

print：打印机

speech：发声设备

4.3．属性值使用方法

@media all/screen/print/ speech{}

4.4．媒体属性

4.4.1．width：监控手机的宽度

max-width：最大宽度

min-width：最小宽度

4.4.2．orientation：监控手机是否横屏

landscape：横屏

portrait：竖屏

4.5．总使用方法

4.5.1．@media screen and (max/min-width：number px)

and (orientation：landscape/ portrait){样式}

number：断点

max–width：若超过断点，显示默认样式

min-width：若小于断点，显示默认样式

and与()间必须以空格隔开，可以写多个and()

4.5.2．分离写法

将{ }中的内容写在一个css文件中，导入时

<link rel="stylesheet" href="路径" media="@media与{}间的全部内容 ">

示例：

<link rel="stylesheet" href="css/mobel.css" media="screen and (max-width:n px)">

4.6．IE兼容方法

使用respond.js插件兼容IE8及以下

## 5．单位rem

5.1．介绍

rem (root em)，与px、em一样也是一个单位。不同的是它是基于html元素计算。

常与媒体查询一起使用

5.2．优点

只要改变html的font-size，所有基于该元素计算的元素相关尺寸随之等比改变。

5.3．缺点

1 rem = 16px 需要进行换算，非常麻烦。

解决办法：html{ font-size: 625%}（625%X16px=100px），此时可以使用.12rem表示12px，计算比较方便。但是Chrome浏览器最小字体是12px，如果设置了11px及以下，仍显示12px。

## 6．渐进增强和优雅降级

6.1．渐进增强

针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

6.2．优雅降级

一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

# 十六、Bootstrap

## 1．布局容器class

1.1．.container：两侧留白

1.2．.container-fluid：两侧不留白

## 2．行(.row)

必须包含在.container或者.container-fluid中，行中可放置列(.col)

## 3．列(.col)

3.1．介绍：把一个横行平均分成了12份

3.2．写法：.col-xs/sm/md/lg-number

3.3．相关值

xs：超小屏幕，手机，<768px

sm：小屏幕，平板，>=768px

md：中等屏幕，桌面显示器，>=992px

lg：大屏幕，大桌面显示器，>=1200px

number：表示在12份中占几份

3.4．xs/sm/md/lg只控制一种类型的屏幕，可以为一列设置多种类型的col，以控制多种屏幕显示的不同

3.5．希望在某种类型的屏幕上隐藏：hidden- xs/sm/md/lg

3.6．希望在某种类型的屏幕上隐藏：visible- xs/sm/md/lg

## 4．字体图标

可当做字体使用，字体的所有样式均可使用

@font-face {

font-family: 'northwood\_highregular';

src: url('../fonts/northwood\_high-webfont.woff2') format('woff2'),

url('../fonts/northwood\_high-webfont.woff') format('woff');

font-weight: normal;

font-style: normal;

}

# 十七、Less

## 1．简介

Less是一种动态样式语言，属于CSS预处理语言的一种.

它使用类似CSS的语法，为CSS的赋予了动态语言的特性:变量、继承、运算、函数等，更方便CSS的编写和维护。

css预处理就是通过编写less通过处理转化成css

## 2．变量

2.1．定义变量：@name：value

2.2．使用变量：属性值：@name

2.3．同一个变量赋值多次，只认最后一次

## 3．函数

3.1．自定义参数函数

.functionName(@name){属性：@name }

3.2．自定义默认参数函数

.functionName (@name：value){属性：@name }

3.3．函数的使用

选择器{.函数名(参数)}

3.4．函数中可以传多个参数

## 4．选择器

4.1．后代选择器

选择器1{选择器2{}}

示例：

Less：

#header {

color: black;

.navigation {

font-size: 12px;}}

生成CSS：

#header .navigation {

font-size: 12px;}

4.2．串联选择器

选择器1{&选择器2{}}

常用于设置伪类

示例：

Less：

#header {

.logo {

width: 300px;

&:hover { text-decoration: none }}}

生成CSS：

#header .logo:hover {

text-decoration: none;}

## 5．注释

同CSS基本相同，但/\*\*/中的内容会在自动生成的css文件中显示，但//后的内容不会显示在css文件中