CSS

## CSS样式的引入方式

**内联（行内、行间）样式**：在HTML标签上添加**style属性**来实现

**内部样式：**在**<style>标签**内添加样式

注：内部样式的优点，可以复用代码

区别：内部样式的代码可以复用，符合W3C规范标准，进行让结构和样式分开处理。

**外部样式：**引入一个单独的css文件，name.css（空格的形式不进行限制）

引入形式：

<link>标签，rel属性指定资源和页面的关系，href属性指定资源的地址

@import：有很多问题，不建议使用

区别：<https://www.cnblogs.com/my--sunshine/p/6872224.html>

id前面是要加一个**#**

class前面要加一个**.**

## CSS颜色表示法

1、单词表示法：Blue，white等

2、十六进制表示法：

3、RGB三原色表示法：rgb(255,255,255)取值范围：0-255

## CSS背景样式

**1、Background-color**：背景颜色

**2、Background-image**：背景图片，默认会水平垂直都铺满背景图

**3、Background-repeat**背景图片的平铺方式。

Repeat默认x,y都平铺；repeat-x(y)代表x(y)轴平铺；no-repeat代表不平铺

**4、Background-position**：背景图片的位置。Number（数字或百分比）|英文

**正数：向下和右；负数：向上和左**

X轴：left，center，right

Y轴：top，center，bottom

**5、Background-attachment**：背景图随滚动条的移动方式

背景图片的附着点有三类，分别是浏览器的可视区域、背景容器本身区域和背景容器的内容区域。具体来说，background-attachment的值也就有三种，即：

**scroll：默认值，**背景图片附着在背景容器上，它是相对于背景容器（元素）固定，它会随着背景容器的滚动，而不是随着它的内容滚动（也可以看成是对元素边框固定）。

**fixed：** 背景图片附着在浏览器的可视区域。因为浏览器的可视区域不具备滚动的性格，所以它是不滚动的。（背景位置按照浏览器（若设置为50%，则代表是相对于浏览器的50%）进行偏移）

**local：**这是CSS3新增的属性值。背景图片附着在背景容器的内容区域。它会随着内容的滚动而滚动。

<https://blog.csdn.net/weixin_39256994/article/details/78671202>

<https://www.cnblogs.com/starof/p/4511367.html>

## CSS边框样式

Border-style:边框样式。solid实线，dashed虚线，dotted点线

Border-width:边框大小

Border-color:边框颜色 注：transparent是透明色

注：针对某一条边进行单独设置在border和style/width/color之间加上 -left/right

## CSS文字样式

### font-family

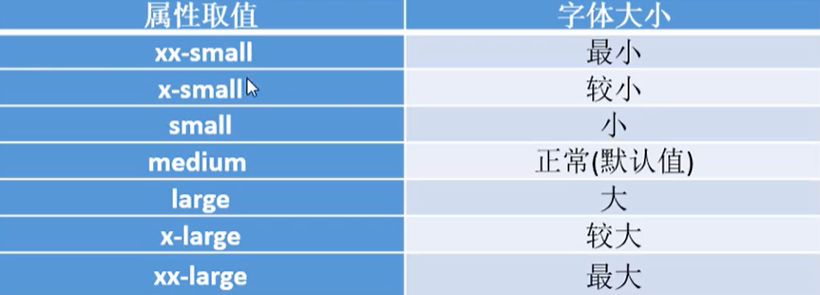
字体类型，字体中出现空格必须加引号

衬线体与非衬线体：<https://www.jianshu.com/p/414ea6c05276>

注意事项：1、设置多字体方式

### font-size

字体大小，默认大小为16px，不推荐设为奇数的单位



**不推荐用上图的表示方法**

### font-weight字体粗细

正常（normal）加粗（bold）|数值表示（100-500正常，600-900加粗）

### font-style字体样式

正常（normal）斜体（italic）注oblique也是斜体，但用得少

区别：italic所有带倾斜字体的可以设置，Oblique没有倾斜属性的字体也可以设为倾斜

Color对字体进行颜色设置

## CSS段落样式

### text-decoration：

文本修饰，下划线（underline），删除线（line-through），上划线（overline），无修饰（none）。可以添加多个修饰，空格隔开

### text-transform：

文本大小写，针对英文段落。小写（lowercase）大写（uppercase）只针对首字母大写（capitalize）

### text-indent：

文本缩进

针对首行缩进，单位px/em

em单位：相对单位（推荐）

### text-align：

文本对齐方式，默认左对齐，对齐方式：left，center，right，justify（两端点对齐）

### line-height：行高

默认行高不是固定值，根据当前字体大小变化

取值：1、number（px）| scale（比例值，和文字大小比较，无单位）

### 强制折行（针对英文）：

**letter-spacing**：字符之间的间距

**word-spacing：**词之间的间距（针对英文段落）

1. Word-break：break-all（非常强烈的折行）
2. Word-wrap：break-Word（不那么墙裂的折行，会产生留白）

## CSS复合样式

复合的写法是通过空格的方式实现的，有的不需要关心顺序，例如background，border；有的是需要顺序的，比如font（最少有两个值，size/line-height和family，顺序固定）

复合样式尽量不要混写，若非要混写，那么一定要先写复合样式再写单一样式，因为若先单一后复合，复合包含了单一样式，会覆盖单一的设置

## CSS选择器

### Id选择器

Css：#elem{}

Html:id=”elem”



注：

1. 在一个页面中是唯一值，只能出现一次
2. 命名规范，字母，\_,-,和数字，且第一个不能是数字
3. 命名方式：驼峰式，下划线式，短线式
4. div#名字+tab快捷创建<div id=”名字”></div>

### class选择器

CSS: .elem{}

Html:class=”elem1 elem2 …”

注：

1. class选择器可以复用
2. 可以添加多个class样式
3. 多个样式时，样式的**优先级根据CSS决定**，而不是class属性中的顺序
4. 标签+类的写法（.elem会对所有class为elem的标签进行css样式修饰，而**div.elem**会对所有class为elem的**div标签**进行css样式修饰）

### 标签（元素）选择器

用法：1、去掉标签的默认样式2、复杂的选择器中如层次选择器中

### 群组选择器

通过逗号的方式给多个不同的选择器添加统一CSS样式

css：如div,#text,.title{background:red}

### 通配选择器

\*{} -> div,p,ul,li…{}尽量避免使用通配选择器，因为会给所有的标签添加样式，慎用

使用场景：

1. 去掉所有标签的默认样式

### 层次选择器

后代：M N{}：只有M下的N才具有大括号中的样式

父子：M > N{}：只关心M的孩子，而M孩子的孩子并不关心

兄弟：M ~ N:当前M**下面**的所有兄弟**N标签**

相邻：M+N{}当前M下面的相邻的**N标签**

### 属性选择器



Div[class]：所有包含class属性的都匹配

Div[**class=box**]：（box不被引号引着）

Div[class=search]：只有完全等于search才匹配

Div[class\*=search]：只要出现了search就匹配，不用完全匹配

^=：起始匹配:起始值为search才匹配

$=：结束匹配，末尾值为search才匹配

多属性匹配：div[class][id][]

### 伪类选择器

CSS伪类用于向某些元素添加特殊效果。一般用于初始样式添加不上的时候，用伪类来添加。

写法： M:伪类{}

**:link** 访问前的样式（只能给a标签）

**:visited** 访问后的样式（只能给a标签）

**:hover** 鼠标移入时的样式（可以给所有标签）

**:active** 鼠标按下时的样式（可以给所有标签）

注：

1. 如果四个伪类都生效，要注意顺序：L V H A
2. 一般网站只这样设置：a{} a:hover{}

**:after :before**  通过伪类的方式给元素添加一个文本内容

例：div:after{content:”添加内容”;color:red;}

:after可以用来清除浮动

**:checked :disabled :focus**都是针对表单元素

用法：:checked/:disabled/:focus{}

结构性伪类选择器：

nth-of-type nth-child

用法举例：li nth-of-type(number | n值【可以为kn用来设置隔行换色】){}第几个（）li标签具有{}中属性

同样以上面的例子进行说明：

first-of-type：第一行具有某样式 nth-child

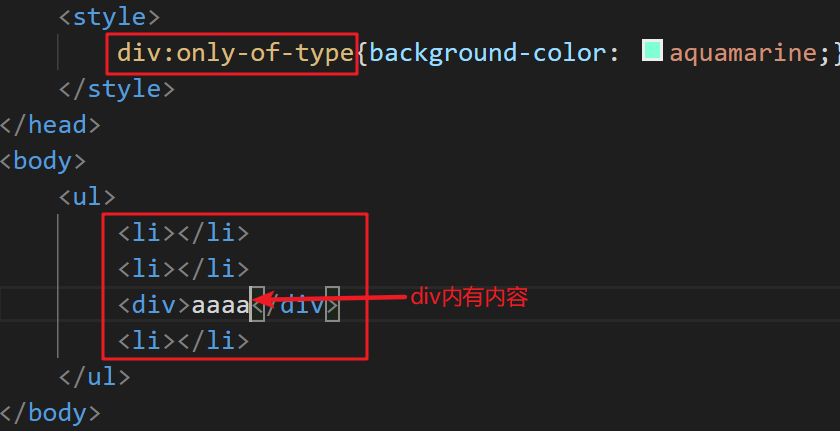
last-of-type：最后一行具有某样式 first-child

only-of-type(只有一个某标签的时候才具有某样式) only-child

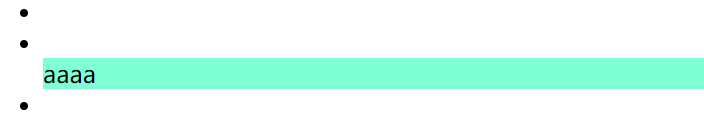
**nth-of-type和nth-child区别（其他同理）**

加li：li:nth-of-type(3)和li:nth-child(3)

前者是第三个li标签会有某属性；后者是第一个li开始后的第三个标签（可以不是li）会有某属性



浏览器效果



## CSS样式继承

文字相关的样式可以被继承，布局相关的样式不可被继承（默认不继承，但是可以设置继承）。示例：p{border:inherit;}

## CSS优先级

**相同样式优先级**：设置**相同样式时，后面的优先级较高**，但不建议出现重复设置样式的情况

**内部样式与外部样式：**优先级相同，若设置了相同样式，那么后写的（在文件后面覆盖了）引入方式优先级高

**单一样式优先级：style行间 > id > class > tag > \* > 继承**

注：

style行间 权重 1000

id 权重 100

class 权重 10

tag 权重 1

**!important：**提升样式优先级，非规范方式，不建议使用，不能针对继承的属性提升优先级

**标签+类与单类：**标签+类的优先级高于单类的优先级

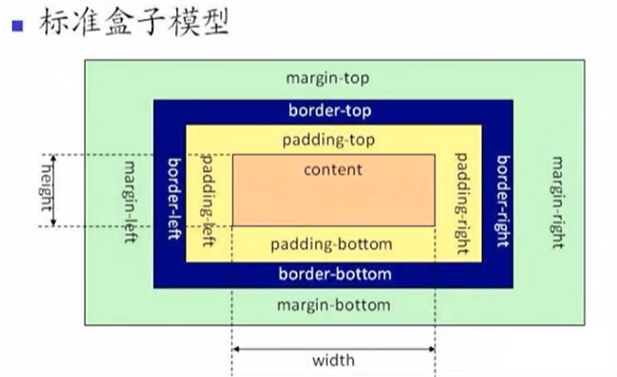
**群组优先级：**群组选择器与单一选择器优先级相同，谁后写谁优先级高

**层次优先级**（建议三层内）：

1. 权重比较：若含多个标签则把所有标签的权重相加进行比较优先级，但是10个tag的权重【1+1+…+1(10个)】< class(10)
2. 约分比较

把等级相同的约掉，比较剩下的权重

## CSS盒子模型



组成：content -> padding -> border -> margin

物品 填充物 包装盒 盒子之间的间距

**content部分：**内容区域，由宽（width）高（height）决定

**padding：**内边距（内填充），内容和边框之间的距离，图中淡黄色

padding-left/right/top/bottom

number：单位px;

写一个值：上下左右；两个值：上下，左右；四个值：上，右，下，左

**margin：**外边距，边框和其他盒子的距离

margin-left/right/top/bottom

number：单位px;

写一个值：上下左右；两个值：上下，左右；四个值：上，右，下，左

注：

1. 背景颜色会填充到margin以内的区域（不包括margin区域）
2. 文字会在content区域
3. padding不能出现负值，margin是可以出现负值

### box-sizing

box-sizing属性允许您以特定的方式定义匹配某个区域的特定元素，取值为content-box（默认值）| border-box。

理解：盒尺寸，可以改变盒子模型的展示形态

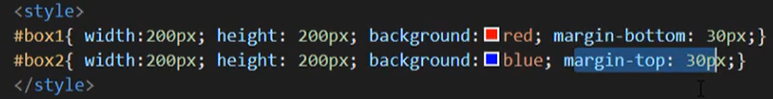
默认值：content-box，宽高对content起作用；而border-box的宽高对content和padding和border都起作用

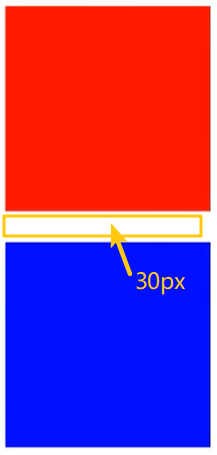
使用场景：

1. 不必计算很多值
2. 解决一些百分比问题（例添加100%的输入框但想加入30px的padding，若直接加会出现滚动条，可以使用border-box）

### 存在问题

**margin叠加：**当给两个盒子同时加上下外边距的时候，就会出现叠加的问题，且这个问题只在上下有，左右不存在这个叠加问题。二者之间的margin值会取较大的作为叠加的值





**Margin问题的解决方案：**

1. BFC规范
2. 想办法只给一个元素添加间距

**Margin传递问题：**

该问题只出现在**嵌套的结构**中，且只有margin-top有该问题，其他方向无。问题描述，对于嵌套的div，若设置里面的div的margin-top会使两个div一起向下移动

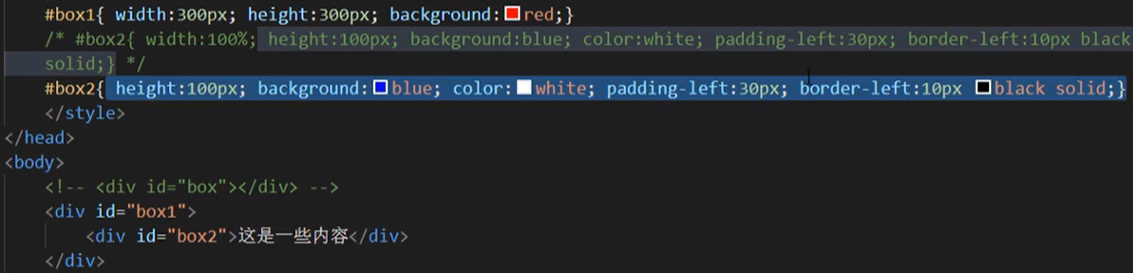
margin传递问题的解决方案：

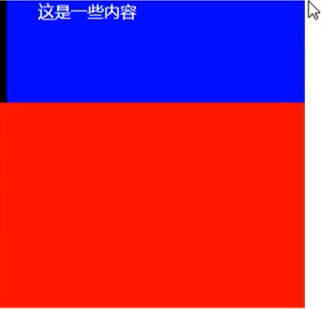
1. BFC规范
2. 给父容器加边框
3. margin换成padding（加给父元素）

扩展：

1、margin-left/right：auto 最左/最右

2、不设置宽高的现象：对盒子模型的影响是会自动计算容器的大小，节省代码





## 标签分类

### 按类型

**block：**块：div,p,ul,li,h1…

特性：

1、独占一行

2、支持所有样式

3、不写宽的时候和父元素的宽相同

4、所占区域是矩形

**lnline：**内联：span,a,em,strong,img…

特性：

1. 挨在一起，左右并排
2. 有些样式不支持，比如宽高，margin，padding（一些方向）
3. 不写宽的时候由内容决定宽度
4. 所占区域不一定是矩形
5. 内联标签之间有空隙，是由换行产生的，若两个内联标签相邻则不会有空隙；若设置body{ font-size:0}，设置内联标签的font-size为n px，也可以消除空隙

注：布局一般用块标签，修饰文本一般用内联 标签

**lnline-block：**内联块：input,select…

特性（具有内联和块的属性）：

1. 挨在一起但是支持宽高
2. 有空隙，所占区域为矩形

### 按内容

Flow：流内容

Metadata：元数据

Sectioning：分区

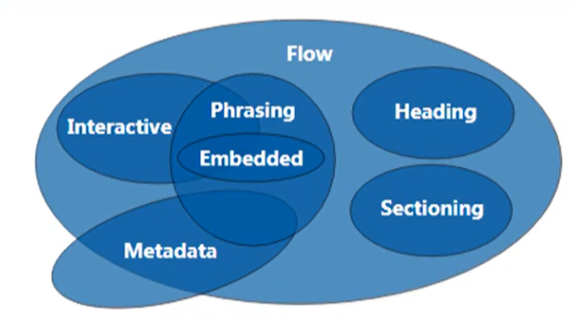
Heading：标题

Phrasing：措辞

Embedded：嵌入的

Interactive：互动的

参考：<https://html.spec.whatwg.org/multipage/dom.html#kinds-of-content>



### 按显示分

替换元素：浏览器根据元素的标签和属性，来决定元素的具体显示内容

示例：img，input…

<img>虽然属于inline但是也属于替换元素，因此支持宽高

非替换元素：将内容直接告诉浏览器，将其显示出来，只写标签没有属性，如h1

示例：div、h1、p…

## 显示框(display)类型

**display：**

display:none;表示此元素将不被显示。

display：block;将元素显示为块元素。（又叫块级元素）

display:inline;将元素显示为内联元素。（又叫行内元素）

display:inline-block;将元素显示为内联块元素。

display:inherit;规定应该从父元素继承 display 属性的值。

注：display:none 与 visibility:hidden 区别

display:none是不占用空间的隐藏，下方的可以占用原来空间

visibility:hidden是占空间的隐藏

## 标签嵌套规范

ul、li；dl、dt、dd；table、tr、td

块标签可以嵌套内联标签，内联不可以嵌套块

特殊：正确的写法

<a href=”#”>

<div></div>

</a>

块标签不一定可以嵌套块标签

错误写法：<p>

<div></div>

</p>

## 溢出隐藏(overflow)

Overflow：

Visible：默认

Hidden：隐藏

Scroll：出现滚轮，多就可滚动，少就不可以

Auto：内容多就出现滚动条，少就不出

x轴、y轴：

overflow-x、overflow-y针对不同轴设置滚轮

## 透明度与手势(opacity,cursor)

opacity：0（透明）-1（不透明）；

注：1、占用空间

2、所有子内容也透明

**3、若想要只对背景设置透明度，使用background:{rgba(num,num,num, [0-1] )}最后一个值为透明度**

**cursor：设置鼠标移入后的手势**

default：默认箭头

实现自定义手势：1、准备图片（.cur/.ico）

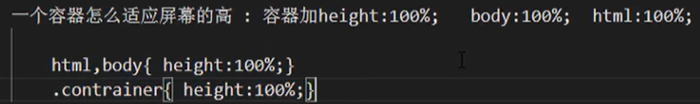
1. cursor:url(图片位置),auto;

## 最大、最小宽高

min-width max-width

min-height max-height div的宽高不定，在某一范围内

注：强化对百分比的理解， 百分比以**父容器**的大小进行换算



## CSS的默认样式

没有默认样式：div、span、…

有默认样式： body、h1…h6、p、ul、…

Body->margin:8px

H1->margin:上下 21.44px,font-weight:bold

p->margin：上下 16px

ul -> margin：上下 16px padding:左 40px

默认点：list-style：disc

a -> text-decoration:underline

## CSS reset

**简单的CSS reset：**

\*{ margin:0; padding:0; }

优点：不用考虑哪些标签会有默认的margin和padding

缺点：稍微的影响性能，改进：body,p,h1,ul{ margin:0; padding:0;}

ul{ list-style:none; }

a{ text-decoration:none; color:#666; }

img{ display:block; }

图片跟容器底部有一些空隙，原因：img是内联标签，内联元素的对其方式是按照文字基线对齐的。改变对齐方式方法1：vertical-align:bottom（底线对齐）方法2：display:block

**重置样式的参考：<https://blog.csdn.net/brain_bo/article/details/81560444>**

写具体页面的时候或者一个布局效果的时候：

1. 写结构
2. Css重置样式
3. 写具体样式

## float浮动

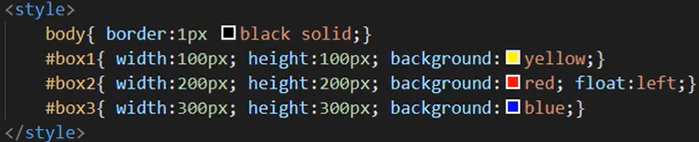
文档流：文档流是文档中可显示对象在排列时所占用的位置。

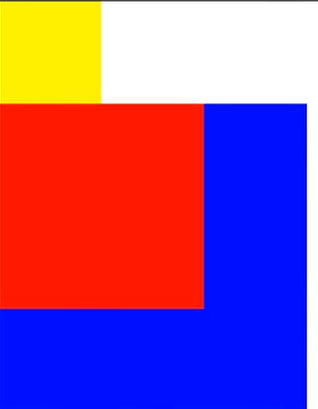
Float浮动特性：加浮动的元素会脱离文档流，会**沿着父容器**靠左或靠右排列，如果之前已经有浮动的元素，会挨着浮动的元素进行排列

取值：left，right，none

**float注意点**

1. 只会影响后面的元素。





2、内容默认提升半层，所以在蓝色div上加内容，内容不会被红色覆盖。

3、默认的width根据内容决定。

4、浮动元素放不下会换行排列。

5、主要给块元素添加，但也可以给内联元素添加。

### 清除浮动（重要）

**上下排列：**使用clear属性清除float浮动clear:left/right/both

**嵌套排列：**

固定宽高 : 不推荐 , 不能把高度固定死，不适合做自适应的效果。

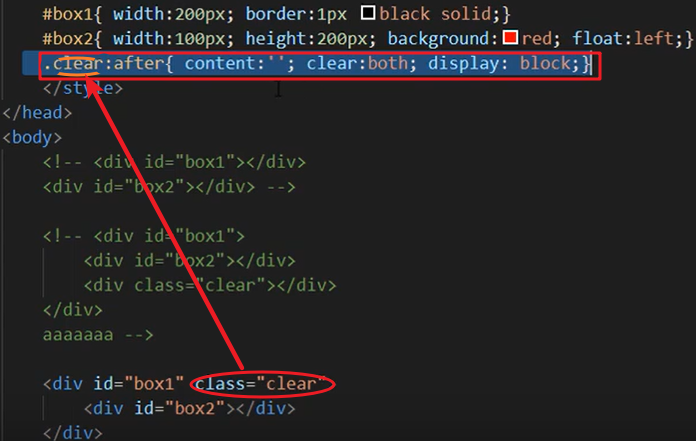
父元素浮动 : 不推荐 , 因为父容器浮动也会影响到后面的元素。

overflow : hidden (BFC规范) , 如果有子元素想溢出，那么会受到影响。

display : inline-block (BFC规范)，不推荐，父容器会影响到后面的元素。

设置空标签 : 不推荐 , 会多添加一个标签。

after伪类 : 推荐，是空标签的加强版，目前各大公司的做法。



注：clear属性只会操作块标签，对内联标签不起作用

## Position定位

### position特性

css position属性用于指定一个元素在文档中的定位方式。top、right、bottom、left 属性则决定了该元素的最终位置。

### position取值

static（默认）

#### relative 相对定位：

1、如果没有定位偏移量，对元素本身没有任何影响

2、不使元素脱离文档流

3、不影响其他元素布局

4、left、top、right、bottom是相对于当前元素自身进行偏移的，正数：向下和右；负数：向上和左

#### absolute 绝对定位

1、使元素完全脱离文档流

2、使内联元素支持宽高 （让内联具备块特性）

3、使块元素默认宽根据内容决定（让块具备内联的特性）

4、如果有定位（**设置了position**）祖先元素，**相对于定位祖先元素发生偏移**，没有定位祖先元素，相对于整个文档发生偏移（绝对、相对、固定）

#### fixed 固定定位

1、使元素完全脱离文档流

2、使内联元素支持宽高 （让内联具备块特性）

3、使块元素默认宽根据内容决定（让块具备内联的特性）

4、**相对于整个浏览器窗口进行偏移**，不受浏览器滚动条的影响

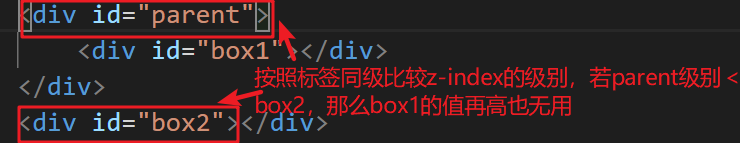
#### sticky 粘性定位

在指定位置进行粘性操作，比如对于设置了position:sticky,top:10px的标签，移动到距离浏览器≤10px的位置时，该标签会保持距离顶部10px而其他标签内容移动

### z-index定位层级

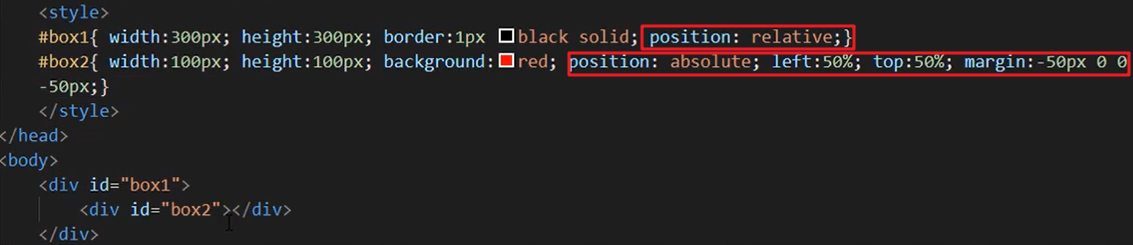
默认层级为0，是为了调整嵌套时候的层级问题（哪个在上面）

1. parent不写position，box1和box2写：后设置样式的优先级高
2. parent不写position，box1和box2写且都写z-index（默认为0），哪个层级高（数大）哪个在前面
3. parent 和box1和box2都写position，z-index，则按照图中说的，按照同级标签比较



### 举例

定位实现元素居中：



## CSS添加省略号

1、width：必须有一个固定的宽

2、white-space : nowrap

不让内容折行

3、overflow : hidden隐藏溢出的内容

4、text-overflow : ellipsis添加省略号

## CSS Sprite

**特性**

CSS雪碧也叫做CSS精灵，是一种网页图片应用处理方式。它允许你将一个页面涉及到的所有零星图片都包含到一张大图中去加载。

**好处**

可以减少图片的质量，网页的图片加载速度快

减少图片的请求的次数，加快网页的打开

## CSS圆角

border-radius：

两个值：左上和右下，右上和左下

四个值：左上，右上，右下，左下

正圆可以用border-radius:50%

border-radius:20(num1)px/40(num2)px 椭圆和矩形相切

## strong与b，em与i

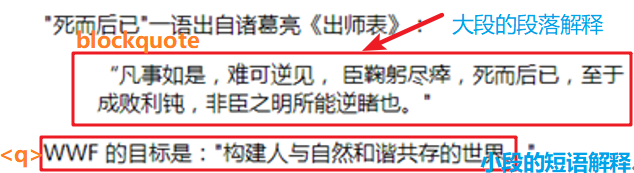
表现形态都是文本加粗和文本斜体，区别在于strong和em是具备语义化的，而b和i不具备语义化。

i，b都是内联标签，可以用来简化选择器操作

## 引用标签

blockquote : 引用大段的段落解释

q : 引用小段的短语解释



abbr : 缩写或首字母缩略词

用法：<abbr title=”全名”>缩写或首字母缩略词</addr>后面的话

address : 引用文档地址信息

cite : 引用著作的标题

语义化标签：便于浏览器找到，

## Iframe嵌套页面

iframe 元素会创建包含另外一个文档的内联框架（即行内框架）。可以引用其他的HTML到当前HTML中显示

|  |  |
| --- | --- |
| **属性** | **含义** |
| **frameborder** | **规定是否显示框架周围的边框** |
| **width** | **定义 iframe 的宽度** |
| **height** | **定义 iframe 的高度** |
| **scrolling** | **规定是否在 iframe 中显示滚动条** |
| **src** | **规定在 iframe 中引入的 URL** |
| **srcdoc(和src同时存在，优先显示)** | **规定在 iframe 中显示的页面内容** |

应用场景：数据传输、共享代码，局部刷新、第三方介入等。

## br与wbr

br 标签表示换行操作，而 wbr 标签表示软换行操作。

提示：如果单词太长，或者您担心浏览器会在错误的位置换行，那么您可以使用 wbr 元素来添加 Word Break Opportunity（单词换行时机）

## pre和code

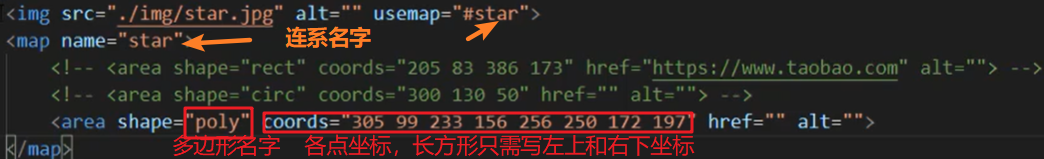
pre 元素可定义预格式化的文本。被包围在 pre 元素中的文本通常会保留空格和换行符。

只应该在表示计算机程序源代码或者其他机器可以阅读的文本内容上使用 code 标签。虽然 code 标签通常只是把文本变成等宽字体，但它暗示着这段文本是源程序代码。

## map与area

定义一个客户端图像映射。图像映射（image-map）指带有可点击区域的一幅图像。area 元素永远嵌套在 map 元素内部。area 元素可定义图像映射中的区域。（给特殊图形添加链接，area能添加的热区的形状，矩形、圆形、多边形）

area 元素的href属性定义区域的URL，shape属性来定义区域的形状，coords属性定义热区的坐标。



## embed与object

embed和object都表示能够嵌入一些多媒体，如flash动画、插件等。基本使用没有太多区别，主要是为了兼容不同的浏览器而已。

object 元素需要配合param 元素一起完成。

用得少，移动端不支持flash

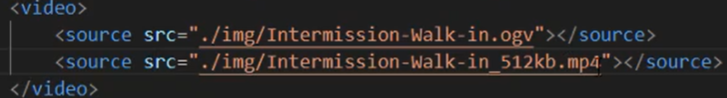
## audio和video

audio标签表示嵌入音频文件，video标签表示嵌入视频文件。默认控件是不显示的，可通过controls属性来显示控件。

用法：<audio src=”音频地址” controls></audio>

loop:循环播放 autplay自动播放

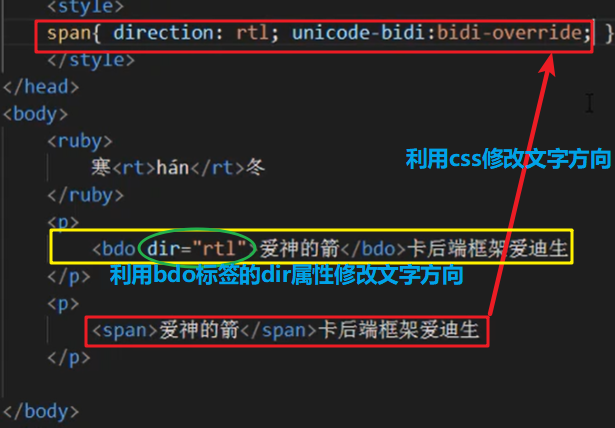
为了能够支持多个备选文件的兼容支持，可以配合source标签做备选功能。



## 文字注解和文字方向

ruby 标签定义 ruby 注释（中文注音或字符），rt 标签定义字符（中文注音或字符）的解释或发音。

bdo 标签可覆盖默认的文本方向。



## 扩展link标签

<link rel="stylesheet" type="text/css" href="theme.css">

<link rel="icon" type="/image/x-icon" href="http://www.mobiletrain.org/favicon.ico">

<link rel="dns-prefetch" href="//static.360buyimg.com">

## meta标签扩展

辅助信息

<https://www.jianshu.com/p/850d2a209ba8>

## HTML5新语义化标签

header : 页眉

footer : 页脚

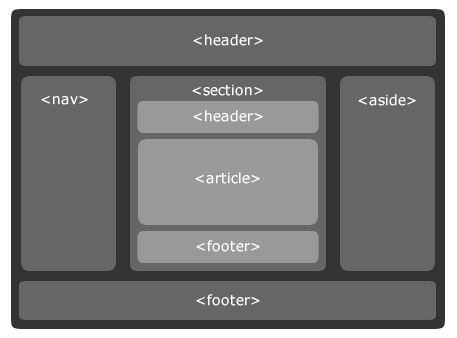
main : 主体

hgroup : 标题组合

nav : 导航

article : 独立的内容

aside : 辅助信息的内容

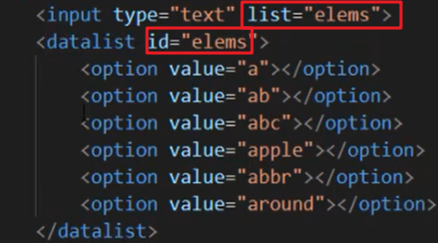


section（在article中嵌套） : 区域

figure : 描述图像或视频

figcaption : 描述图像或视频的标题部分

datalist :选项列表，自动提示（通过id连接input）



details / summary : 文档细节 / 文档标题

progress / meter : 定义进度条 / 定义度量范围





注：meta设置了low和high属性后会根据当前值与两个属性的比较变化颜色

time : 定义日期或时间

mark : 带有记号的文本

参考：<https://blog.csdn.net/qq_41229582/article/details/80637518>

## 表格扩展

添加单线 : border-collapse : collapse

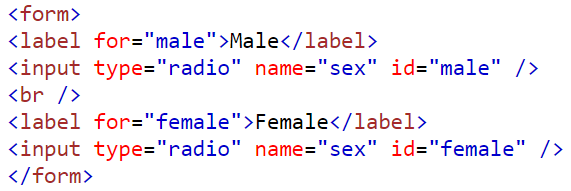
隐藏空单元(style中写): empty-cells : hide

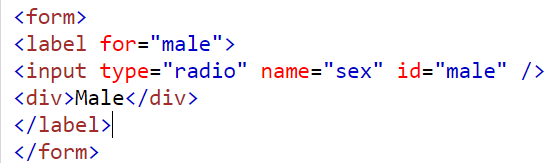
斜线分类 : border 或 rotate

列分组（同时对几列操作） : colgroup 和 col

## 表单扩展

### 美化表单





点击按钮或文本之一，就可以触发相关控件，因此可以利用此原理，将美化效果放在div里

1. label+:checked
2. position+opacity

### 新的input控件

email : 电子邮件地址输入框

url : 网址输入框

number : 数值输入框

range : 滑动条

date / month / week : 日期控件

search : 搜索框

color : 颜色控件

tel : 电话号码输入框（在移动端默认调起数字键盘）

time : 时间控件

### 新的表单属性

autocomplete : 自动完成（默认为on，会保留输入的历史）

autofocus : 获取焦点（打开界面光标自动在含该属性的栏）

required : 不能为空

pattern : 正则验证

method : 数据传输方式（GET/POST POST更安全）

enctype : 数据传输类型

name / value : 数据的键值对

### 扩展标签

fieldset : 表单内元素分组 （和legend组合使用）

legend : 为fieldset元素定义标题

optgroup : 定义选项组

## BFC规范

Formatting context(格式化上下文) 是 W3C CSS2.1 规范中的一个概念。它是页面中的一块渲染区域，并且有一套渲染规则，它决定了其子元素将如何定位，以及和其他元素的关系和相互作用。

BFC 即 Block Formatting Contexts (块级格式化上下文) ，它属于上述中的其中一种规范。具有 BFC 特性的元素可以看作是隔离了的独立容器，容器里面的元素不会在布局上影响到外面的元素，并且 BFC 具有普通容器所没有的一些特性

触发BFC规范的元素可以形成一个独立的容器，**不受外界的影响**，从而解决一些布局问题

### 触发BFC

浮动元素：float 除 none 以外的值

绝对定位元素：position (absolute、fixed)

display 为 inline-block、table-cells、flex

overflow 除了 visible 以外的值 (hidden、auto、scroll)

### BFC的应用

1. **解决margin叠加问题：**

上下两个div都设置距离对方固定距离的margin值会导致二者的距离取margin的最大值而不是和

解决：分别在两个div外再加一个div，使后加的div触动BFC规范，如对其设置overflow或定位等

1. **解决margin传递问题**

举例：两个div嵌套，对里面的盒子设置margin会导致两个盒子一起移动

解决：使里面的盒子触发BFC规范，使里面的盒子成为一个独立的容器，不受外面影响

1. 解决浮动问题

让收到浮动影响的容器触发BFC规范，成为一个独立的容器

1. 解决覆盖问题

## 浏览器前缀（背）

浏览器厂商以前就一直在实施CSS3，但它还未成为真正的标准。为此，当有一些CSS3样式语法还在波动的时候，他们提出了针对浏览器的前缀

|  |  |  |
| --- | --- | --- |
| **浏览器** | **内核** | **前缀** |
| **IE** | Trident | -ms- |
| **Firefox** | Gecko | -moz- |
| **Opera** | Presto | -o- |
| **Chrome** | Webkit | -webkit- |
| **Safari** | Webkit | -webkit- |
| **Opera、Chrome** | Blink |  |

## transtion过渡

transition-property : 规定设置过渡效果的CSS属性的名称。（width/height/all）

transition-duration : 规定完成过渡效果需要多少秒或毫秒。

transition-delay : 定义过渡效果何时开始。（延迟时间可正可负）

transition-timing-function : 规定速度效果的速度曲线。

transition-timing-function属性值

linear

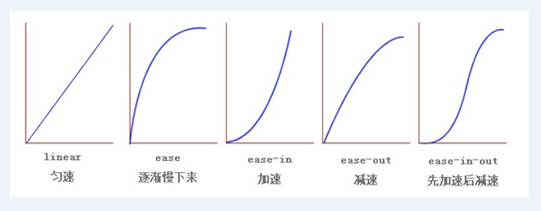
ease（默认值）

ease-in

ease-out

ease-in-out

cubic-bezier：贝赛尔曲线，各数值可以在下面的链接中获取



（ [http://cubic-bezier.com](%20http:/cubic-bezier.com%20) ）

复合写法

Transition：property（默认all） 共需多少时间duration 延迟多少时间delay 曲线类型

如：transition:1s 2s linear

**注：不要加到hover上**

## C

### R