**CSS层叠样式表（cascading style sheets）**

用来定义html内容在浏览器内的显示样式。

**优点：**

1、简化html相关标签，网页体积小，下载快

2、解决内容与表现分离的问题

3、更好的维护网页，提高工作效率

Html是网页内容的载体，用来搭建网页的结构

Css是样式表现，用来控制外观

Javascript是行为，用来实现网页特效

CSS样式规则：选择器（相应要添加样式的标签内容）{声明（属性：值；）}

<!doctype html>

<html>

<head>

<meta charset=”utf8”>

<title>1234<title>

<style type=”text/css”>

/\*CSS注释\*/

P,h1,h3{font-size:30px;

Color:blue;

Front-family:”隶书”;}

H2{color:red;}

</style>

</head>

CSS样式使用方式

1、行内样式：直接在内容标签内添加style属性。

2、内部样式（嵌入样式）：在head标签内添加style标签即<style type=”text/css”>，添加type属性，然后再添加选择器和申明

\*为了适应低版本的浏览器，选择器和申明需要使用注释标签包裹，这样就算低版本浏览器不识别style标签也不会把选择器和申明显示出来，而高版本会直接忽略注释标签。

3、外部样式 ：直接把样式表里的内容写到外部文件中就可以了，样式文件扩展名为“.css”；然后在html文件中引入外部文件——<link href=”XX.css” rel=”stylesheet” style=”text/css”> href外部样式文件地址，rel说明外部文件与HTML文件的关系，type说明文件类型。 \*link标签写在head标签中。

4、导入样式：加载完页面之后再加载样式。

<head>

<style type=”text/css”>

@import url(XXX.css); (括号内写入样式文件地址)

</style>

</head>



**样式使用方式优先级**

行内样式>内部样式>外部样式；

外部样式的优先级取决于所处的位置，最后定义的优先级最高。

类选择器

对应内容中添加class属性，例如<p class=”specia”>，然后在style样式标签中引用相应样式——.specia{XX:XX;}

不同类元素的同一个名称的类选择器设置不同样式规则:标签加类名，例p.specia{XX:XX;} \*\*\*class值不能是数字！！！！

Class属性可以添加多个值，值与值之间用空格隔开。

ID选择器：对应内容标签中添加id属性，<p id=”name”>然后在style标签内设置id样式——#name{XX:XX;} \*id是唯一的。

全局选择器：选择所有内容添加样式 \*{：；}

后代选择器：某标签下的某标签下的某标签，选择器之间用空格隔开：

div.abcd h1 em{ : ;}含义是：给class定义为abcd的div标签下的h1标签下的所有em标签设置style属性

伪类选择器



a:link{ } a:visited{ } a:hover{ } a:active{ }

\*其他元素也支持:hover、:active状态，比如：p:hover{ }

\*IE6及更早的版本都是支持<a>标签的四种状态，但不支持其他元素的hover跟active

\*注意书写顺序：link-visit-hover-avtive使用方法：

Id选择器 优先于 class选择器 优先于 标签选择器

!imortant声明优先级是最高的

直接在标签属性语句后面添加！important（注意要写在；号前面）

常用的CSS样式命名:header、main、footer、content、container、navigator、sidebar、column、页面外围控制wrapper、menu、submenu、summary

Css文件不能以数字和 - \_ 开头来命名

**文字样式属性**

字体、字体集font-family font-family：”字体一”，”字体二”

\*字体集：serif、sans-serif、monospace、cursive、fantasy（具有不同的装饰效果）

\*含有空格的字体、中文字体，用英文引号””括起。

\*设置多个字体，用英文逗号隔开。（浏览器首选第一个字体，逐一往下最后是默认字体）

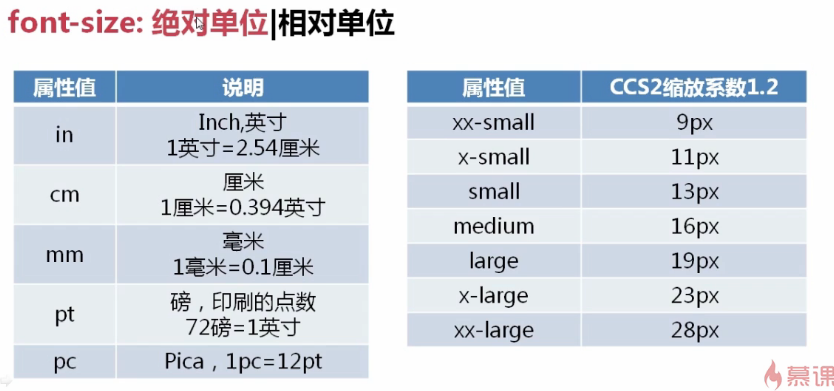
\*引号嵌套，外层括号用双引号，内层用单引号。

大小font-size

\*不设置字体大小时，是浏览器默认字体大小（一般是16px）

\*px单位下文字大小手屏幕分辨率影响较大

\*smaller、larger、em、percent单位值都是相对父元素的设置，em代表的是父元素的几倍。Percent也是父元素的百分之几



颜色color：值为名称、RGB值、十六进制

\*color：reb(X,X,X) 数字0~255 百分比0%~100%

\*十六进制：#开头，六位，0~F(0~9+a~f)

粗细font-weight

值:normal、bold、bolder、lighter、100~900

\*默认值为normal，400等同于normal，700等同于bold

样式font-style:值：normal(默认样式)、italic(斜体)、oblique(倾斜)

样式font-variant：设置文本为小型大写字母。值：normal、small-caps

Font属性简写

Font: style值 variant值 weight值 size值/line-height(行高值) family值

\*值之间空格隔开

\*值之间是有书写顺序的

文本对对齐方式

Text-align：left（right、center、justify）

\*该属性只对块级元素设置有效

Vertical-align：baseline、sub、super、top、tex-top、middle、bottom、text-bottom、长度、百分比

\*该属性只对行内元素设置有效

\*可以在其父元素里面添加“display：table”，以及自身添加“display：table-cell”

Line-height:值可以是px、em、%，em与%都是相对于字体的。

Word-spacing:单词之间的间距

Letter-spacing：字母之间的间距

\*可以是正值也可以是负值

Text-transform：capitalize/uppercase/lowercase/none

Text-decoration: underline/overline/line-throgh/blink/none

盒子模型

Max-width、min-width最大宽度和最小宽度，有兼容问题，IE不支持

块级元素和替换元素可以设置高和宽

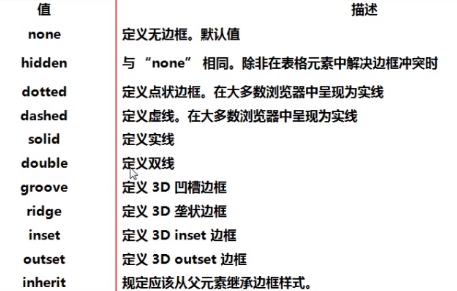
高宽属性都是内容的高和宽

边框

Border-width：thin、medium、think、值

Border-color

Border-style：



4个方向来具体控制上下左右边框：Border-left-width

简写：border：宽度、样式、颜色

内边距padding：内容与边框的距离

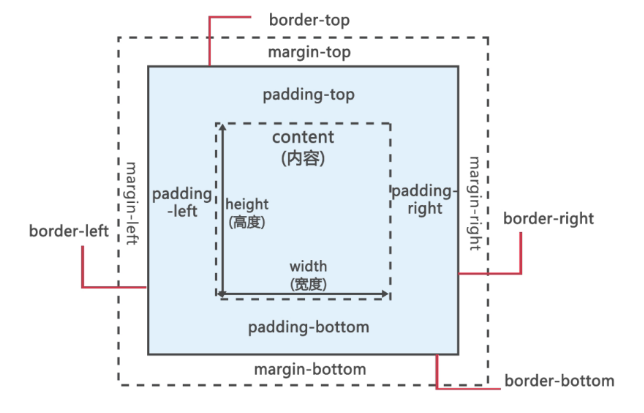
同样可以用上下左右来控制具体方向的距离

Padding-top、padding-right、padding-bottom、padding-left

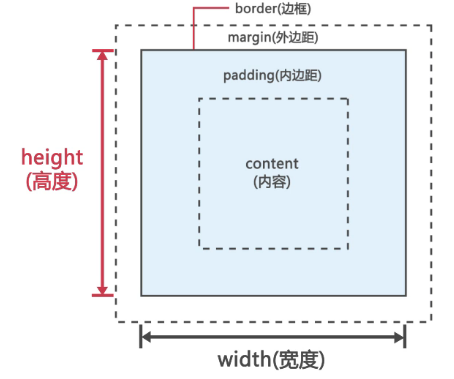
外边距margin，margin值设置为auto时，元素会针对其父元素居中显示

垂直方向两个相邻元素都设置外边距，外边距会发生合并，高度为两个发生合并的边距中的最大值。

标准盒子模型

文档开头的doctype html声明就决定了所有浏览器都会使用标准盒子模型来解析这个网页。

IE盒子模型



Display：inline(元素显示为内联元素)、block(元素显示为块级元素)、inline-block(行内块元素，元素呈现为行内元素，具有块级元素特征)、none(元素不显示)

Background-color：

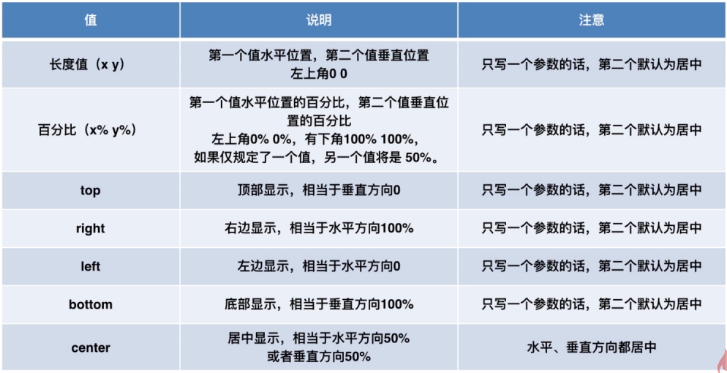
Background-image：

Background-repeat：

Background-attachment：fixed、scroll

Background-position：图片定位，相对于图片的容器来说的。

值：长度值(X Y)、百分比(X% Y%)、距左边边框的值与距顶部边框的值，只写一个值的话第二个默认为居中 top、right、left、bottom、center



Background：简写形式，不分先后

列表项标记

List-style-type：none、disc、square、circle、decimal、lower-roman、upper-roman、lower-alpha、upper-alpha

List-style-image：url(xxxx.jpg)

List-style-position:inside(作为文本，占文本位置)、outside(默认项，在文本外，文本不环绕图标)

List-style：值之间用空格分割，不固定位置，list-style-image会覆盖list-style-type

浮动——float：right、left、none

浮动产生的问题：高度塌陷

清除浮动（闭合浮动）的语法——clear：none、left、right、both

清除浮动的常用方法：

1、添加空盒子浮动，然后清除这个盒子的浮动

2、给添加了浮动的容器添加overflow：hidden属性

3、给添加了浮动的容器添加clearfix这个class，然后在样式中添加——

.clearfix:after{

Content:”.”;

Height:0;

Display:block;

Visibility:hidden;

Clear:both }

4、给父元素添加固定高度

5、给父元素也添加浮动，不过这种方法会引起新的浮动问题

\*添加在方法1和2之后，解决浏览器版本不兼容问题：Zoom:1

z-index：设置元素的堆叠顺序，拥有更高堆叠值的元素会处于堆叠值较低的元素的前面。（更靠近视窗）它的值可以是数字（可正可负）也可以是从父元素处继承得到的。

nth-child(n)选择器：选择属于其父元素的第n个子元素。N可以是数字、公式、关键词。示例——p nth-child(8) ：选取标签为p的第8个元素。

定位position

Static：静态定位/常规定位/自然定位

Relative：可定位祖先元素。

1、可以使用top/right/bottom/left/z-index进行定位（相对常规流中的自己偏移）

2、相对定位的元素不会离开常规流（常规流中依然保留元素位置）

3、任何元素都可以设置为relative，它的绝对定位后代都可以相对于它进行定位。

4、可以使浮动元素发生偏移，并控制它们的堆叠顺序

（参照物：常规流中的自身）

Absolute：绝对定位

1、脱离常规流

2、top/right/bottom/left值可以为百分比（比的是最近的定位祖先元素，没有relative定位祖先元素时，比的是body标签）

3、top/right/bottom/left全部设置为0时，将填充满容器。单个为0则对齐边。全部设置为auto时，将会返回常规流中的位置

（参照物：最近的定位祖先元素，没有则为body）

Fixed：固定定位

相对于视窗固定，其余性质与relative一样。

（参照物：视窗）

Sticky：磁铁定位/粘性定位/吸附定位

1、如果产生偏移，原位置仍在常规流中

2、如果最近祖先元素有滚动，以最近的祖先元素作为滚动标尺

3、如果最近祖先元素没有滚动，以视窗作为偏移标尺

\* 该属性在IE中并不支持，相同效果可以使用JS实现

（参照物：视窗、可滚动的祖先元素）

Cursor:定义光标形状（有很多属性，其中pointer就是将光标变成手）

Display：none与visible：hidden的区别：none完全在页面中消失，不再具有宽高等属性，而hidden相当于透明，看不见内容，但是其宽高等空间属性依然保留。

行布局

水平居中：设置绝对定位与margin

Opacity透明度值为0~1

Border-radius边框圆角

Background-color：transparent透明

浮动状态下，负边界值会导致元素上移。