快捷键：

注释： Ctrl+/

自动补全标签： 标签名+tap

补全多个标签： 标签名\*数字 回车

有时候，我们不能直接书写部分符号，比如< > 和连续的空格，

< >可以在中间加空格解决，其他使用html中的实体（转义字符）

&nbsp; 空格

&lt;小于

！+tap直接生成网页模板

Ctrl+alt +o截图识别文字（腾讯OCR）

Ul>li \* 5 + tab可以生成ul里面包含五个li

搜狗输入法中文v1可以输出符号

多行编辑(列编辑): Alt+Shift+鼠标左键 ，ctrl+Alt+Down/Up

连点两下 选中这一行

第一章：introduce

1.网页的结构

结构(HTML )

表现(CSS)

行为(JavaScript)

CSS：层叠样式表

2.meta标签

主要用于设置网页中的一些元数据，元数据不是给用户看

charset指定网页的字符集

name指定的数据的名称

content指定的数据的内容

keywords表示网站的关键字，可以同时指定多个关键字，关键字间使用,隔开

title标签的内容会作为搜索结果的超链接上的文字显示

例如：<meta name="keywords" content="HTML5,前端,CSS3">

<meta http-equiv=" refresh" confent= "3 ;url=https:/ /www . mozilla.org">将页面重定向到另一个网站

3.语义化标签

在网页中HTML专门用来负责网页的结构

所以在使用html标签时，应该关注的是标签的语义，而不是它的样式，比如不要管h1是不是比h2大。

标题标签:

h1 ~ h6一共有六级标题

从h1~h6重要性递减，h1最重要，h6最不重要

h1在网页中的重要性仅次于title标签，一般情况下一个页 面中只会有一个h1

一般情况下标题标签 只会使用到h1~h3, h4~h6很少用

4.块元素

在页面中独占一行的元素称为块元素(block element)

在网页中-般通过块元素来对页面进行布局

hgroup标签用来为标题分组，可以将一组相关的标题同时放入到hgroup

例如：

<hgroup>

<h1>回乡偶书二首</h1>

<h2>其-</h2>

</hgroup>

4.行内元素

在页面中不会独占一行的元素称为行内元素(inline element)。

行内元素主要用来包裹文字 如：

Em标签用于表示语音语调的一个加重，不要关注他是否变斜了

strong表示强调，重要内容!   
blockquote 表示一个长引用

q表示一个短引用

一般情况下会在块元素中放行内元素，而不会在行内元素中放块元素

块元素中基本上什么都能放

p元素中不能放任何的块元素

浏览器在解析网页时，会自动对网页中不符合规范的内容进行修正

比如:

标签写在了根元素的外部

p元素中嵌套了块元素

根元素中出现了除head和body以外的子元素

5.布局标签(结构化语义标签)

header表示网页的头部

main表示网页的主体部分(一个页面中只会有一个。main)

footer表示网页的底部

nav表示网页中的导航

aside和主体相关的其他内容(侧边栏)

article表示一个独立的文章

section 表示一个独立的区块，上边的标签都不能表示时使用secctiion

div 没有语义， 就用来表示一个区块，目前来讲div还是我们主要的布局元素

span行内元素，没有任何的语义，一般用于在网页中选中文字

6.列表

1、有序列表

2、无序列表

3、定义列表

无序列表，使用ol标签来创建无序列表

使用li表示列表项

无序列表，使用ul标签来创建无序列表

使用li表示列表项

定义列表，使用dl标签来创建一个 定义列表

使用dt来表示定义的内容

使用dd来对内容进行解释说明

列表之间可以互相嵌套

7.超链接：

可以让我们从一个页面跳转到其他页面，

或者是当前页面的其他的位置

使用a标签来定义超链接

属性:

href指定跳转的目标路径，当我们需要跳转一个服务器内部的页面时，一般我们都会使用相对路径

超链接是也是一个行内元素，在a标签中可以嵌套除它自身外的任何元素

相对路径都会使用./或. ./开头，且./可以省略

值可以是一个外部网站的地址

也可以写一个内部页面的地址

target属性，用来指定超链接打开的位置

可选值:

self默认值在当前页面中打开超链接

blank在个新的要么中打开超链接

可以直接将超链接的href属性设置为#，这样点击超链接以后

页面不会发生跳转，而是转到当前页面的顶部的位置

可以跳转到页面的指定位置，只需将href属性设置#目标元素的id属性值

id属性(唯一不重复的)：每一个标签都可以添加一个d属性

id属性就是元素的唯一标识，同一个页面中不能出现重复的id属性。

可以使用javascript:;来作为href的属性，此时点击这个超链接什么也不会发生

8.图片：

图片标签用于向当前页面中引入一个外部图片

使用img标签来引入外部图片，img标签是 一个自结束 标签

img这种元素属于替换元素(块和行内元素之间，具有两种元素的特点)

属性:

src. 属性指定的是外部图片的路径(路径规则和超链接是一 样的)

alt图片的描述，这个描述默认情况下不会显示，有些浏览器会图片无法加载时显示

搜索引擎会根据alt中的内容来识别图片，如果不写alt属性则图片不会被搜索引擎所收录

width图片的宽度( 单位是像素)

height图片的高度

宽度和高度中如果只修改了一个，则另一个会等比例缩放

注意:

一般情况在pc端，不建议修改图片的大小，需要多大的图片就裁多大

但是在移动端，经常需要对图片进行缩放(大图缩小)

9.图片的格式:

jpeg(jpg)

一支持的颜色比较丰富，不支持透明效果，不支持动图

一般用来显示照片

gif

支持的颜色比较少，支持简单透明， 支 持动图

颜色单一的图片，动图

png

支持的颜色丰富，支持复杂透明，不支持动图

颜色丰富，复杂透明图片(专为网页而生)

webp

这种格式是谷歌新推出的专门用来表示网页中的图片的一种格式

它具备其他图片格式的所有优点，而且文件还特别的小

缺点:兼容性不好

base64

将图片使用base64编码，这样可以将图片转换为字符，通过字符的形式来

引入图片

一般都是一些需要和网页-起加载的图片才会使用base64

效果一样，用小的

效果不一样，用效果好的

10.内联框架

内联框架，用于向当前页面中引入一个其他页面

src 指定要引入的网页的路径

frameborder 指定内联框架的边框

11.音频播放

用来向页面中引入一个外部的音频文件的

音视频文件引入时，默认情况下不允许用户自已控制播放停止

属性:

controls是否允许用户控制播放

autoplay音频文件是否自动播放，有的浏览器不支持

一如果设置了autoplay 则音乐在打开页面时会自动播放

但是目前来讲大部分浏览器都不会自动对音乐进行播放

Loop 音乐是否循环播放

除了通过src来指定外部文件的路径以外，还可以通过source来指定文件的路径

<audio controls>

对不起，您的浏览器不支持播放音频!请升级浏览器!

<source src=" ./source/audio . mp3">

<source src=" ./ source/ audio. ogg">这两个文件优先播放上面那个

<embed src=" ./ source/ audio . mp3" type=" audio/ mp3" width="300" height=" 100'>

</ audio>

用source引用可以让不支持的浏览器显示文字

Embed 是一个老式的标签，可以适用所有浏览器

第二章CSS

1.css简介

第一种方式：(内联样式，行内样式):

在标签内部通过style属性来设置元素的样式

问题:

使用内联样式，样式只能对一个标签生效，

如果希望影响到多个元素必须在每一个元素中都复制一遍

并且当样式发生变化时，我们必须要一个一个的修改，非常的不方便

注意:开发时绝对不要使用内联样式

第二种方式：(内部样式表)

将样式编写到head中的style标签里

然后通过css的选择器来选中元素并为其设置各种样式

可以同时为多个标签设置样式，并且修改时只需要修改-处即可全部应用

内部样式表更加方便对样式进行复用

问题:

我们的内部祥式表只能对一个网页起作用，

它里边的样式不能跨页面进行复用

第三种方式：(外部样式表) 最佳实践

可以将CSS样式编写到一个外部的CSS文件中，

然后通过link 标签来引入外部的cSS.文件

外部样式表需要通过link标签进行引入，

意味着只要想使用这些样式的网页都可以对其进行引用

使样式可以在不同页面之间进行复用

将样式编写到外部的CSS文件中，可以使用到浏览器的缓存机制，

从而加快网页的加载速度，提高用户的体验。

2.css的基本语法:

选择器 + 声明块

选择器，通过选择器可以选中页面中的指定元素

声明块，通过声明块来指定要为元素设置的样式

声明块由一个一个的声明组成

声明是一个名值对结构

一个样式名对应一个样式值， 名和值之间以 : 连接， 以 ; 结尾

3.常用选择器

（1）id选择器：

作用:根据元素的id属性值选中一个元素

语法: #id属性值{ }

（2）类选择器：

作用:根据元素的cLass 屠性值选中一组元素

语法: . class 属性值

例子：.abc{ font-size: 20px; }

class 是一个标签的属性，它和id类似，不同的是class可以重复使用

可以通过class,属性来为元素分组

可以同时为一个元素指定多个class属性

（3）通配选择器：

作用:选中页面中的所有元素

语法:\*

例子：

\*{ color: red; }

p \* { }p和\*之间要有空格

4.复合选择器

（1）交集选择器:

作用:选中同时复合多个条件的元素

语法:选择器1选择器2选择器3选择器n{ }

注意点:

交集选择器中如果有元素选择，必须使用元素选择器开头

（2）并集选择器:

作用:同时选择多个选择器对应的元素

语法:选择器1,选择器2，选择器3，选择器n{ }

#b1, .p1,h1, span, div.red{ }

5.关系选择器

（1）子元素选择器

作用:选中指定父元素的指定子元素

语法: 父元素>子元素

（2）后代元素选择器:

作用:选中指定元素内的指定后代元素

语法:祖先 后代(中间是空格)

（3）兄弟选择器

选择下一个兄弟

语法:前一个 + 下一个

选择下边所有的兄弟

语法:兄~弟

如：P ~ span{

color: Ored; }

6.属性选择器

[属性名]选择含有指定属性的元素

[属性名=属性值]选择含有指定属性和属性值的元素

[属性名^=属性值]选择属性值以指定值开头的元素

[属性名$= 属性值]选择属性值 以指定值结尾的元素

[属性名\*=属性值]选择属性值中含有某值的元素的元素

如：p[title$=abc]{ }

7.伪类选择器

伪类(不存在的类，特殊的类)

伪类用来描述一个元素的特殊状态

比如:第一个子元素、被点击的元素、鼠标移入的元素...

伪类一般情况下都是使用:开头

:first-child第一个子元素，这些都是限定词，针对：前面的元素

:last-child 最后一个子元素

:nth-child()选中第n个子元素

特殊值:

n 第n个n的范围0到正无穷

2n或even表示选中偶数位的元素

2n+1或odd表示选中奇数位的元素

以上这些伪类都是根据所有的子元素进行排序

:first-of-type

:last-of-type

:nth-of- type()

这几个伪类的功能和上述的类似，不通点是他们是在同类型元素中进行排序

:not()否定伪类

将符合条件的元素从选择器中去除

如：uL>li:nth-child(2n+1){ /\*指的是li是ul的2n+1个孩子时变红色\*/

coLor: red; }

注意：li和后面的：之间不要有空格！

a元素的伪类

:link 用来表示没访间过的娃接 (正常的链接)

:visited用来表示访问过的链接 由于隐私的原因，所以visited这个伪类只能修改链接的颜色

:hover 用来表示鼠标移入的状态

:active用来表示鼠标点击

8.样式的继承

样式的继承，我们为一个元素设置的样式同时也会应用到它的后代元素上

继承是发生在祖先后后代之间的

继承的设计是为了方便我们的开发，

利用继承我们可以将一些通 用的样式统一设 置到共同的祖先元素上，

这样只需设置一次即可让所有的元素都具有该样式

注意:并不是所有的样式都会被继承:

比如背景相关的，布局相关等的这些样式都不会被继承。

9.选择器的权重

样式的冲突

当我们通过不同的选择器，选中相同的元素，并且为相同的样式设置不同的值时，此时就发生J样式的冲突。

发生样式冲突时，应用哪个样式由选择器的权重(优先级)决定

选择器的权重

内联样式 1000

id选择器 100

类和伪类选择器 10

元素选择器 1

通配选择器 0

继承的样式 没有优先级

比较优先级时，需要将所有的选择器的优先级迸行相加计算，最后优先级越高，则越优先显示(分组选择器是单独计算的)，

选择器的累加不会超过其最大的数量级，类选择器在高也不会超过id选择器

如果优先级计算后相同，此时则优先使用靠下的梓式

可以在某一个样式的后边添加! important，则此时该样式会获取到最高的优先级，甚至超过内联样式，

注意:在开发中这个玩意一定要慎用!

10.长度单位:

像素

屏幕(显示器)实际上是由一个一个的小点点构成的

不同屏幕的像素大小是不同的，像素越小的屏幕显示的效果越清晰

-所以同样的200px在不同的设 备下显示效果不一样

百分比

也可以将属性值设置为相对于其父元素属性的百分比

设置百分比可以使子元素跟随父元素的改变而改变

em

em是相对于元素的字体大小来计算的

1em = lfont-size

em会根据字体大小的改变而改变

rem

rem是相对于根元素的字体大小来计算

11.颜色:

在CSS中可以直接使用颜色名来设置各种颜色

比如: red、 orange、 yellow、 blue、 green

但是在css中直接使用颜色名是非常的不方便

RGB值:

RGB通过三种颜色的不同浓度来调配出不同的颜色

Rred,Ggreen，Bblue

每一种颜色的范围在0-255(0%-100%)之间

语法: RGB(红色， 绿色，蓝色)

RGBA:

就是在rgb的基础上增加了一个a表示不透明度

需要四个值，前三个和rgb -一样，第四个表示不透明度

1表示完全不透明0表示完全透明.5半透明

十六进制的RGB值:

语法: #红色绿色蓝色

颜色浓度通过00-ff

如果颜色两位两位重复可以进行简写

#aabbcc --> #abc

HSL值HSLA值

H色相(0 - 360)

S饱和度，颜色的浓度0% - 100%

L亮度，颜色的亮度0%- 100%

第三章 layout

1.文档流(normal fLow)

网页是一个多层的结构，一层接着一层

通过CSS可以分别为每一层来设置祥式

作为用户来进只能看到最顶上一层

这些层中，最底下的一层称为文档流，文档流是网页的基社

我们所创建的元就都是在文档说中进行排列

对于我们来元素主要有两个状态

在文档流中

不在文档流中(脱离文档流)

元者在文档流中有什么特点:

块元素

块元素会在页面中独占一行(自上向下垂直排列)

默认宽度是父元亲的全部(会把父元病掉满)

默认高度是被内容撑开(子元素)

行内元素

行内元素不会独占页面的一行，只占自身的大小

行内元责在页面中左向右水平排列，如果一行之中不能容纳下所有的行内元素

则元素会换间第二行继续自左向右排列(书写习惯一致)

行内元素的默认宽度和高度都是被内容撑开

2.盒模型、盒子模型、框模型(box model)

CSS将页面中的所有元亲都设置为了一个矩形的盒子

将元素设置为矩形的盒子后，对页面的布局就变成将不同的盒子摆放到不同的位置

每一个盒子都由一下/几个部分组成:

内容区(content )

内边距(padding )

边框(border)

外边距(margin)

边框

边框的宽度border-width

3.边框

border-width: 10px;

默认值，一般都是3个像素

border-width可以用来措定四个方向的边框的宽庹

值的情况

四个值:上右下左

三个值:上左右下

两个值:上下左右

一个值:上下左右

除了border-width还有一组border-xxx -width

xxx可以是top right bottom Left

用来单独指定某一条边的宽度

边框的颜色border-color

border-color.用来指定边框的颜色，同样可以分别指定四个边的边框

规则和border-width一样

border-color也可以省略不写，如果省略则自动使用color的值

边框的样式border-style

solid 表示实线

dotted点优虚线

dashed虚线

double双线

border-style的就以值是none表示没有边框

4.内边距  
内边距（padding）

内容区和边框之间的距离是内边距

一共有四个方向的内边距：

padding-top

padding-right

padding-bottom

padding-left

内边距的设置会影响到盒子的大小

背景颜色会延伸到内边距上

一共盒子的可见框的大小，由内容区 内边距 和 边框共同决定，

所以在计算盒子大小时，需要将这三个区域加到一起计算

5.外边距

外边距（margin）

外边距不会影响盒子可见框的大小

但是外边距会影响盒子的位置

一共有四个方向的外边距：

margin-top

上外边距，设置一个正值，元素会向下移动

margin-right

默认情况下设置margin-right不会产生任何效果

margin-bottom

下外边距，设置一个正值，其下边的元素会向下移动

margin-left

左外边距，设置一个正值，元素会向右移动

margin也可以设置负值，如果是负值则元素会向相反的方向移动

元素在页面中是按照自左向右的顺序排列的，

所以默认情况下如果我们设置的左和上外边距则会移动元素自身

而设置下和右外边距会移动其他元素

margin的简写属性

margin 可以同时设置四个方向的外边距 ，用法和padding一样

margin会影响到盒子实际占用空间

6.元素的水平方向的布局：

元素在其父元素中水平方向的位置由以下几个属性共同决定“

margin-left

border-left

padding-left

width

padding-right

border-right

margin-right

一个元素在其父元素中，水平布局必须要满足以下的等式

margin-left+border-left+padding-left+width+padding-right+border-right+margin-right = 其父元素内容区的宽度 （必须满足）

以上等式必须满足，如果相加结果使等式不成立，则称为过度约束，则等式会自动调整

调整的情况：（可以变成负值）

如果这七个值中没有为 auto 的情况，则浏览器会自动调整margin-right值以使等式满足

这七个值中有三个值和设置为auto

width

margin-left

maring-right

如果某个值为auto，则会自动调整为auto的那个值以使等式成立

如果将一个宽度和一个外边距设置为auto，则宽度会调整到最大，设置为auto的外边距会自动为0

如果将三个值都设置为auto，则外边距都是0，宽度最大

如果将两个外边距设置为auto，宽度固定值，则会将外边距设置为相同的值，居中

所以我们经常利用这个特点来使一个元素在其父元素中水平居中

示例：

width:xxxpx;

margin:0 auto;

7.垂直布局

默认情况下父元素的高度被内容撑开

子元素是在父元素的内容区中排列的，

如果子元素的大小超过了父元素，则子元素会从父元素中溢出

使用 overflow 属性来设置父元素如何处理溢出的子元素

可选值：

visible，默认值 子元素会从父元素中溢出，在父元素外部的位置显示

hidden 溢出内容将会被裁剪不会显示

scroll 生成两个滚动条，通过滚动条来查看完整的内容

auto 根据需要生成滚动条

8.垂直外边距的重叠(折叠)

相邻的垂直方向外边距会发生重叠现象

兄弟元素

兄弟元素间的相邻垂直外边距会取两者之间的较大值(两者都是正值)

特殊情况:

如果相邻的外边距一正一负，则取两者的和

如果相邻的外边距都是负值，则取两者中绝对值较大的

兄弟元素之间的外边距的重叠，对于开发是有利的，所以我们不需要进行处理

父子元素

父子元素间相邻外边距，子元素的会传递给父元素(上外边距)

父子外边距的折叠会影响到页面的布局，必须要进行处理

处理方式：

1. 不用外边距，用内边距（此时要调整父元素高度）
2. 给父元素加一个边框，来隔开两个外边距

9.行内元素的盒模型

行内元素不支持设置宽度和高度

行内元素可以设置padding,但是垂直方向padding不会影响页面的布局

行内元素可以设置border,垂直方向的border不会影响页面的布局

行内元素可以设置margin,垂直方向的margin不会影响布局

display用来设置元素显示的类型

可选值:

inline将元素设置为行内元素

block将元素设置为块元素

inline-block 将元素设置为行内块元素

行内块，既可以设置宽度和高度又不会独占一行

table将元素设置为一个表格

none元素不在页面中显示

visibility用来设置元素的显示状态

可选值:

visible 默认值，元素在页面中正常显示

hidden元素在页面中隐藏不显示，但是依然占据页面的位置

10.默认样式:

通常情况，浏览器都会为元素设置一些默认样式

默认样式的存在会影响到页面的布局，

通常情况下编写网页时必须要去除浏览器的默认样式(PC 端的页面)

重置样式表：专门用来对浏览器的样式进行重置的

reset.css 直接去除了浏览器的默认样式

normalize.css 对默认样式进行了统一

要让一个文字在父元素中垂直居中，只需将父元素的line-height设置为一个和父元素height一样的值

line-height: 25px;

11.盒子的尺寸

默认情况下，盒子可见框的大小由内容区、内边距和边框共同决定

box-sizing 用来设置盒子尺寸的计算方式（设置width和height的作用）

可选值：

content-box 默认值，宽度和高度用来设置内容区的大小

border-box 宽度和高度用来设置整个盒子可见框的大小

width 和 height 指的是内容区 和 内边距 和 边框的总大小

12.阴影，轮廓和圆角

box-shadow 用来设置元素的阴影效果，阴影不会影响页面布局

第一个值 水平偏移量 设置阴影的水平位置 正值向右移动 负值向左移动

第二个值 垂直偏移量 设置阴影的水平位置 正值向下移动 负值向上移动

第三个值 阴影的模糊半径

第四个值 阴影的颜色

如：box-shadow: 0px 0px 50px rgba(0, 0, 0, .3) ;

outline 用来设置元素的轮廓线，用法和border一模一样

如： outline: 10px red solid;

轮廓和边框不同的点，就是轮廓不会影响到可见框的大小

border-radius: 用来设置圆角 圆角设置的圆的半径大小

border-radius 可以分别指定四个角的圆角

四个值 左上 右上 右下 左下

三个值 左上 右上/左下 右下

两个个值 左上/右下 右上/左下

也可以单独制定一个角

border-top-left-radius:

border-top-right-radius

border-bottom-left-radius:

border-bottom-right-radius:

此时如果后面带两个值则为椭圆形圆角

将元素设置为一个圆形 ：border-radius: 50%;

第四章 float

1.浮动：

通过浮动可以使一个元素向其父元素的左侧或右侧移动，使用 float 属性来设置于元素的浮动可选值：

none 默认值 ，元素不浮动

left 元素向左浮动

right 元素向右浮动

注意，元素设置浮动以后，水平布局的等式便不需要强制成立

元素设置浮动以后，会完全从文档流中脱离，不再占用文档流的位置，

所以元素下边的还在文档流中的元素会自动向上移动

2.浮动的特点：

1、浮动元素会完全脱离文档流，不再占据文档流中的位置

2、设置浮动以后元素会向父元素的左侧或右侧移动，

3、浮动元素默认不会从父元素中移出

4、浮动元素向左或向右移动时，不会超过它前边的其他浮动元素

5、如果浮动元素的上边是一个没有浮动的块元素，则浮动元素无法上移

6、浮动元素不会超过它上边的浮动的兄弟元素，最多最多就是和它一样高

7.浮动元素不会盖住文字，文字会自动环绕在浮动元素的周围，所以我们可以利用浮动来设置文字环绕图片的效果

简单总结：

浮动目前来讲它的主要作用就是让页面中的元素可以水平排列，

通过浮动可以制作一些水平方向的布局

3.脱离文档流的特点：

块元素：

1、块元素不在独占页面的一行

2、脱离文档流以后，块元素的宽度和高度默认都被内容撑开

行内元素：

1、行内元素脱离文档流以后会变成块元素，特点和块元素一样

2、脱离文档流以后，不需要再区分块和行内了

4.高度塌陷问题：

BFC(Block Formatting Context) 块级格式化环境

BFC是一个CSS中的一个隐含的属性，可以为一个元素开启BFC

开启BFC该元素会变成一个独立的布局区域

元素开启BFC后的特点：

1.开启BFC的元素不会被浮动元素所覆盖

2.开启BFC的元素子元素和父元素外边距不会重叠

3.开启BFC的元素可以包含浮动的子元素

可以通过一些特殊方式来开启元素的BFC：

1、设置元素的浮动（不推荐）

2、将元素设置为行内块元素（不推荐）

3、将元素的overflow设置为一个非visible的值

常用的方式 为父元素设置 overflow:hidden 开启其BFC 以使其可以包含浮动元素而不塌陷

如果我们不希望某个元素因为其他元素浮动的影响而改变位置，

可以通过clear属性来清除浮动元素对当前元素所产生的影响

clear：

作用：清除浮动元素对当前元素所产生的影响

可选值：

left 清除左侧浮动元素对当前元素的影响

right 清除右侧浮动元素对当前元素的影响

both 清除两侧中最大影响的那侧

原理：

设置清除浮动以后，浏览器会自动为元素添加一个上外边距，以使其位置不受其他元素的影响

高度塌陷最终解决方案：在style里面加个

.clearfix::before, .clearfix::after{

content: '';

display: table;

clear: both;

}

//这既可以解决高度塌陷（后面加table）也可以解决上边距重叠（前面加table隔开）

再把父元素加一个class=clearfix就行了

第五章 position

1.定位的简介

定位是一种更加高级的布局手段，但一定多用浮动少用定位

通过定位可以将元素摆放到页面的任意位置

使用position属性来设置定位

可选值：

static 默认值，元素是静止的没有开启定位

relative 开启元素的相对定位

absolute 开启元素的绝对定位

fixed 开启元素的固定定位

sticky 开启元素的粘滞定位

2.相对定位：

当元素的position属性值设置为relative时则开启了元素的相对定位

相对定位的特点：

1.元素开启相对定位以后，如果不设置偏移量元素不会发生任何的变化

2.相对定位是参照于元素在文档流中的位置进行定位的

3.相对定位会提升元素的层级

4.相对定位不会使元素脱离文档流

5.相对定位不会改变元素的性质块还是块，行内还是行内

偏移量（offset）

当元素开启了定位以后，可以通过偏移量来设置元素的位置

top定位元素和定位位置上边的距离

bottom定位元素和定位位置下边的距离

定位元素垂直方向的位置由top和bottom两个属性来控制

通常情况下我们只会使用其中一

top值越大，定位元素越向下移动

bottom值越大，定位元素越向上移动

left 定位元素和定位位置的左侧距离

right 定位元素和定位位置的右侧距离

定位元素水平方向的位置由left和right两个属性控制

通常情况下只会使用一个

left越大元素越靠右

right越大元素越靠左

3.绝对定位

当元素的position属性值设置为absolute时，则开启了元素的绝对定位

绝对定位的特点：

1.开启绝对定位后，如果不设置偏移量元素的位置不会发生变化

2.开启绝对定位后，元素会从文档流中脱离

3.绝对定位会改变元素的性质，行内变成块，块的宽高被内容撑开

4.绝对定位会使元素提升一个层级

5.绝对定位元素是相对于其包含块进行定位的

包含块( containing block )

正常情况下：

包含块就是离当前元素最近的祖先块元素

<div> <div></div> </div>

<div><span><em>hello</em></span></div>

绝对定位的包含块:

包含块就是离它最近的开启了定位的祖先元素，

如果所有的祖先元素都没有开启定位则根元素就是它的包含块

html（根元素、初始包含块）

4.固定定位：

将元素的position属性设置为fixed则开启了元素的固定定位

固定定位也是一种绝对定位，所以固定定位的大部分特点都和绝对定位一样

唯一不同的是固定定位永远参照于浏览器的视口进行定位

固定定位的元素不会随网页的滚动条滚动

5.粘滞定位

当元素的position属性设置为sticky时则开启了元素的粘滞定位

粘滞定位和相对定位的特点基本一致，

不同的是粘滞定位可以在元素到达某个位置时将其固定

6.绝对定位元素的布局

水平布局

left + margin-left + border-left + padding-left + width + padding-right + border-right + margin-right + right = 包含块的内容区的宽度

当我们开启了绝对定位后:

水平方向的布局等式就需要添加left 和 right 两个值

此时规则和之前一样只是多添加了两个值：

当发生过度约束：

如果9个值中没有 auto 则自动调整right值以使等式满足

如果有auto，则自动调整auto的值以使等式满足

可设置auto的值

margin width left right

因为left 和 right的值默认是auto，所以如果不指定left和right

则等式不满足时，会自动调整这两个值

垂直方向布局的等式的也必须要满足

top + margin-top/bottom + padding-top/bottom + border-top/bottom + height = 包含块的高度

7.元素的层级

对于开启了定位元素，可以通过z-index属性来指定元素的层级

z-index需要一个整数作为参数，值越大元素的层级越高

元素的层级越高越优先显示

如果元素的层级一样，则优先显示靠下的元素

祖先的元素的层级再高也不会盖住后代元素