大鹏一日同风起，扶摇直上九万里，

假令风歇时下来，犹能簸却沧溟水。

世人见我恒殊调，闻余大言皆冷笑。  
宣父犹能畏后生，丈夫未可轻年少。

1.line-height和height组合到一块儿，还有一种效果，就是如果把它们的值设置的一样了，文字就会在垂直方向居中

2.用line-height只对文字起作用 对于图片时失效

3.padding内边距

4.快捷键link+top

5.flex-wrap:wrap会折行

6.在块级元素设定 width 值 的情况下，设置其 margin: 0 auto 可以在父级元素中达到水平居中

7、块元素和行内元素

块元素：h1-h6、p、div(主要用来对页面进行布局)

行内元素：span（专门用来选中文字，为文字来设置样式）、a、img、iframe

块元素主要用来页面布局，内联元素用来选中文本设置样式

一般情况只使用块元素去包含内联元素，而不会用内联元素去包含一个块元素

a元素可以包含任意元素，除了他本身

p元素不可以包含任何块元素

8、常用选择器(并集选择器)：

选择器元素

id选择器

类选择器,可以为一个元素设置多个class属性值，多个值之间使用空格隔开

<p class="p1 hello"></p>

选择器分组：通过选择器分组可以同时选中多个选择器对应的元素，语法：选择器1，选择器2，选择器N{}

通配选择器：\*

复合选择器（交集选择器）：可以选中同时满足多个选择器的元素

语法：选择器1选择器2选择器N{}

9、选择器：

元素之间的关系：

父元素：直接包含子元素的元素

子元素：直接被父元素包含的元素

祖先元素：直接或间接包含后代元素的元素，父元素也是祖先元素

后代元素：直接或间接被祖先元素包含的元素，字元素也是后代元素

兄弟元素：拥有相同父元素的元素

后代元素选择器：选中指定元素的指定后代元素，语法：祖先元素 后代元素{}

子元素选择器（IE6及以下浏览器不支持）：作用：选中指定父元素的指定子元素

语法：父元素>子元素

10、伪类选择器：表示元素的一种的特殊的状态

a:link：普通链接，没访问过的链接

a:visited:访问过的链接（浏览器通过历史记录来判断一个链接是否访问过，由于涉及用户的隐私问题，只能设置字体颜色）

a:hover：鼠标悬浮

a:active:超链接被点击的状态,不松手

四个选择器的优先级是一样的，伪类的顺序：a:link>a:visited>a:hover>a:active

hover和active也可以给其他元素设置,IE6中不支持超链接以外的设置

:focus获取焦点

::selection:为p标签中选中的内容使用样式

注意：这个::selection伪类在火狐中需要采取另一种方式编写（p::-moz-selection）

----------------------------------------------------------------

11、伪元素：表示元素中的一些特殊的位置

:first-letter:第一个字符设置特殊样式

:first-line:表示第一行设置样式

:before:表示元素最前面的部分，一般都需要结合content这个样式一起使用

p:before{content:"加入文字"}

:after:表示元素最后面的部分，一般都需要结合content这个样式一起使用

1. 属性选择器：可以根据元素中的属性或属性值来选取指定元素

Title属性：这个属性可以给任何标签指定，当鼠标移入到元素上时，元素中的title属性的值将会作为提示文字显示

语法：

[属性名]选取含有指定属性的元素

[属性名=“属性值”]选取会有指定属性值的元素

[属性名^=“属性值”]选取属性值以指定内容开头的元素

[属性名$=“属性值”]选取属性值以指定内容结尾的元素

[属性名\*=“属性值”]选取属性值以包含指定内容的元素

1. 子元素的选择器

:first-child:可以选中第一个子元素

:last-child:可以选中最后一个子元素

:nth-child():可以选中任意位置的子元素,可以指定一个参数，even表示偶数位置的子元素，odd表示奇数位置的子元素

:nth-child(*n*)选择器匹配属于其父元素的第 N 个子元素，不论元素的类型。

以上三个是在所有的子元素中排列

:first-of-type、:last-of-type、:nth-of-type是在当前类型的子元素中排列

[:nth-of-type()](https://www.w3school.com.cn/cssref/selector_nth-of-type.asp" \o "CSS :nth-of-type() 选择器) 该选择器选取父元素的第 N 个指定类型的子元素。

1. 兄弟元素选择器：

后一个兄弟元素选择器：可以选中一个元素后紧跟着的指定兄弟元素

语法：前一个+后一个{}

选中后边所有的兄弟元素

语法：前一个~后边所有{}

1. 否定伪类：可以从已选中的元素中删除某些元素

语法：not(选择器)，例如p:not(.hello){}

1. 样式的继承

像儿子可以继承父亲的遗产一样，在css中，祖先元素上的样式会被后代元素继承

利用继承，可以将一些基本的样式设置给祖先元素，这样所有的后代元素将会自动继承这些样式

但是并不是所有样式都被继承，比如背景相关的都不会被继承，默认值是透明的，边框相关的样式，定位相关的样式

1. 选择器的优先级

当使用不同的选择器，选中同一个元素时并且设置相同的样式时，样式产生冲突，最终到底采用哪个，根据选择器的优先级决定

优先级规则：

内联样式：优先级1000

Id选择器：优先级100

类和伪类：优先级10

元素选择器：优先级1

通配\*：优先级0

继承的样式：没有优先级

当选择器中包含多种选择器时，需要将多种选择器的优先级相加然后在比较

但是注意，选择器优先级不会超过他的最大的数量级，如果选择器的优先级一样，则选择靠后的样式

并集选择器的优先级是单独计算，不会合并在一起计算：div,p,#p1,.hello{}

可以在样式的最后，添加一个！Important,则此时该样式将会获得一个最高的优先级，将会优先于所有的样式显示甚至超过内联样式，但是在开发中尽量避免使用！Important

--------------------------------------------------------------------------------------------------------------------------

1. 文本标签：

<em>斜体：表示语气上的强调

<strong>粗体：表示强调的内容

<i>斜体，没有任何语义

<b>加粗，没有任何语义

H5规范中规定，对于不需要着重的内容而是单纯斜体加粗就用<i>和<b>

<small>标签中的内容会比他的父元素中的文字小一些，在h5中使用来表示一些细则一类的内容，比如：合同中的小字，网站的版权声明都可以放到small

<big>没有语义，淘汰

<cite>:网页中所有的加书名号的内容都可以使用这个标签，表示参考的内容

比如：书名，歌名，话剧名，电影名。。。

<q>标签表示一个短的引用（行内引用，行内元素），q标签引用的内容浏览器会默认加上引号

<blockquote>标签表示一个长引用（块级引用，块元素）

使用<sup>标签来设置一个上标

使用<sub>表示一个下标

<del>表示一个删除的内容，会自动添加删除线

<ins>表示一个插入的内容，其中的内容会自动添加下滑线

需要页面中直接编写一些代码：

<pre>是一个预格式标签，会将代码中的格式保存，不会忽略多个空格

<code>专门用来表示代码，只是表示是代码，不会保存格式

一般结合使用pre和code来表示一段代码

1. 列表：无序列表

<ul type=””>

<li></li>

<li></li>

<li></li>

</ul>

通过type属性可以改变无序列表的项目符号：

可选值：

Disc，默认值，实心的圆点

Square,实心的方块

Circle,空心的圆圈

默认的项目符号我们一般都不使用，都是ul{list-style:none},如果需要设置项目符号，就可以采用为li设置背景图片

Ul和li都是块元素

有序列表

<ol type=””>

<li></li>

<li></li>

<li></li>

</ol>

通过type属性可以改变无序列表的项目符号：

可选值：

1. 默认值，数字

A/a大小写字母作为序号

I/i采用大写罗马数字

Ol是块元素

列表之间都是可以相互嵌套

定义列表

<dl>

<dt></dt>

<dd></dd>

</dl>

dt:被定义的内容

dd:对定义内容的描述

Dl,ul,ol之间可以相互嵌套

1. 长度单位

像素px:在网页中使用最多的一个单位，一个像素相当于我们屏幕中的一个小点，我们屏幕实际上就是由这些像素点构成的，这些像素点，不能直接看见的

不同显示器一个像素的大小也不相同，显示效果越好越清晰，像素就越小，反之像素越大

百分比%：浏览器将会根据其父元素的样式来计算该值，

使用百分比好处是，当父元素的属性值发生变化是，子元素也会按照比例发生改变，

在我们创建一个自适应的页面时，经常使用百分比作为单位

em:和百分比类似，它是相对于当前元素的字体大小来计算的

1em=1font-size,一个em相当于一个字体大小

使用em,当字体大小发生改变时，em也会随之改变

当设置字体相关的样式时，经常会使用em

颜色单位：在css中可以直接使用颜色的单词来表示不同的颜色，也可以通过RGB值来表示不同的颜色，

颜色浓度需要一个0-255之间的值，255表示最大，0表示没有。随便一个截图工具下面会有rgb值，可以看到

浓度也可以采用一个百分数来设置，需要以0%-100%之间的数字，使用百分数最终也会转换为0-255之间的数

也可以使用十六进制的rgb值来表示颜色，原理和上面rgb原理一样，只不过使用十六进制数来代替，使用三组两位的十六进制数组来表示一个颜色

每组表示一个颜色，第一组表示红色的浓度，范围00-ff

第二组表示绿色的浓度，范围00-ff

第三组表示蓝色的浓度，范围00-ff

语法：#红色绿色蓝色

十六进制：0 1 2 3 4 5 6 7 8 9 a b c d e f

00-ff:00表示没有，表示rgb中0，ff表示最大，表示rgb中255，例如红色#ff0000，也可以写成#f00

1. 字体样式：

文字的颜色：color

文字的大小：font-size,浏览器中一般默认的文字大小都是16px,他设置的并不是文字本身的大小，在页面中，每个文字都是处在一个看不见的框中，我们设置的font-size实际上设置格的高度，并不是字体的大小，一般情况下文字都要比这个格要小一些，也有时会比格大，根据字体的不同，显示效果不同

文字的字体：font-family，当采用某种字体时，如果浏览器支持则使用该字体，不支持，则使用默认字体

该样式可以指定多个字体，多个字体之间使用，分开，当采用多个字体时，浏览器会优先使用前边的字体，如果前边没有在尝试下一个

浏览器使用的字体默认就是计算机中的字体，如果计算机中有，则使用，如果没有就不用，在开发中字体太奇怪，用的太少，尽量不用，有可能客户电脑没有，就不能达到想要的效果

文字的斜体：font-style

可选值：

Normal,默认值，文字正常显示

Italic,文字会以斜体显示

Oblique,文字以倾斜的效果显示

大部分浏览器都不会对倾斜和斜体做区分，设置Italic和Oblique他们效果一样，一般我们只会使用Italic

文字的粗细：font-weight

可选值：

Normal，默认值，文字正常显示

Bold，文字加粗显示

在css中提供一个样式叫font，使用该样式可以同时设置字体相关的所有样式，可以将字体样式的值，统一写再font样式中，不同的值中间用空格隔开

使用font设置字体样式时，斜体 加粗 小大字母，没有顺序要求，甚至可写可不写，如果不写则使用默认值，但是要求文字的大小和字体必须写，而且字体（font-family）必须是最后一个样式，大小（font-size）必须是倒数第二个样式

实际上使用简写属性也会有一个比较好的性能  
21、字体分类：

在网页中将字体分为5大类（不要求记住）：

Serif（衬线字体）

Sans-serif（非衬线字体）

Monospace（等宽字体）

Cursive（草书字体）

fantasy （虚幻字体）

可以将字体设置为这些大的分类，当设置为大的分类以后，浏览器会自动选择指定的字体并应用样式。一般会将字体的大分类，指定为font-family中的最后一个字体

1. 行间距=行高 - 字体大小

在css中并没有为我们提供一个直接设置行间距的方式，我们只能通过设置行高来间接的设置行间距，行高越大行间距越大，使用line-height来设置行高

行高类似于上学单线本，单线本是一行一行，线与线之间的距离就是行高

网页中的文字实际上也是写在一个看不见的线中的，而文字会默认在行高中垂直居中显示

对于单行文本来说，可以将行高设置为何父元素的高度一致，这样可以是单行文本在父元素中垂直居中

在font中也可以指定行高，在字体大小后可以添加/行高（30px/50px），来指定行高，该值是可选的，如果不指定则会使用默认值，如果有font样式，line-heigh在font之后设置

1. 文本样式

Text-transform可以设置文本的大小写

可选值：

None，默认值，该怎么显示怎么显示，不做任何处理

Capitalize 单词的首字母大写,通过空格来识别单词

Uppercase 所有的字母都大写

Lowercase 所有的字母都小写

Text-decoration可以设置文本的装饰

可选值：

None，默认值，不添加任何修饰，正常显示

Underline为文本添加下划线

Overline为文本添加下划线

Line-through为文本添加删除线

超链接会默认添加下划线，去除下划线，需要设置为none

Letter-spacing可以指定字符间距

Word-spacing可以设置单词之间的距离，实际上就是设置词与词之间空格的大小

Text-align用于设置文本对齐方式

可选值：

Left默认值，文本靠左对齐

Right文本靠右对齐

Center文本居中对齐

Justify两端对齐，通过调整文本之间的空格大小，来达到一个两端对齐的目的

Text-indent可以设置首行缩进，用em作为单位

当给他指定一个正值时，会自动向右侧缩进指定的像素

如果为它指定一个负值，则会向左移动指定的像素，通过这种方式可以将一些不想显示的文字隐藏起来

1. 盒子模型

内容区（content）

内边距：内容和盒子的距离（padding）

一共有四个方向的内边距，可以通过

Padding-top

Padding-right

Padding-bottom

Padding-left

内边距会影响盒子的可见框的大小，元素的背景会延伸到内边距，

盒子的大小由内容区，内边距和边框共同决定

盒子可见框的宽度=border-left-width + padding-left + width + padding-right + border-right-wight

盒子可见框的高度=border-top-width + padding-top + width + padding-bottom + border-bottom-wight

使用padding可以同时设置四个边框的样式，规则和border-width一样

边框（border），要为元素设置边框必须指定三个样式，缺一不可

Border-width：宽度，

Border-width指定四个值，则四个值会分别设置给上 右 下 左

Border-width指定三个值，则三个值会分别设置给上 左右 下

Border-width指定两个值，则两个值会分别设置给上下 左右

Border-width指定一个值，则四边都是这个值

除了Border-width，css中还提供了四个Border-xxx-width

Xxx可能是top/right/bottom/left，专门用来设置指定边的宽度

Border-color：颜色，和宽度一样，也提供四个方向的颜色，分别指定颜色

Border-style：样式，也可以提供四个边的样式

可选值：

None无边框

Solid实线

Dashed虚线

Dotted点状边框

Double双线

设置边框，大部分浏览器中，边框的宽度和颜色都是有默认值，而边框的样式默认值是none

border简写样式，通过它可以同时设置四个边框的样式，宽度，颜色，没有顺序要求，一指定就是四个边，不能分别指定

外边距（margin）：指是当前盒子与其他盒子之间的距离，他不会影响可见框的大小，而是会影响到盒子的位置

盒子有四个方向的外边距：

Margin-top

Margin-right

Margin-bottom

Margin-left

由于页面中的元素都是靠左靠上摆放的，所以注意当我们设置上和左外边距时，会导致盒子自身的位置发生改变，而如果是设置右和下外边距会改变其他盒子的位置

外边距可以是指定为一个负值，如果设置的是负值，则元素会向反方向移动

Margin还可以设置为auto（自动）,auto一般只设置给水平方向的margin，如果只指定左外边距或右外边距的margin为auto则会将外边距设置为最大值，垂直方向外边距如果设置为auto，对外边距默认就是0

如果将left和right同时设置为auto，则会将两侧的外边距设置为相同的值，就可以使元素自动在父元素中居中，所以经常在左右外边距设置为auto，以使子元素在父元素中水平居中

Margin可以同时设置四个方向的的外边距，规则和padding一样

Margin:0 auto，水平居中

1. 垂直外边距的重叠
2. 在网页中垂直方向的相邻外边距会发生外边距的重叠，所谓的外边距重叠指兄弟元素之间的相邻外边距会取最大值而不是取和，比如：一个盒子的 margin-bottom: 100px;，另一个盒子margin-top: 100px;中间相差就是100px,如果margin-bottom: 200px，margin-top: 100px;中间会相差200px,这就是取最大值
3. 如果父子元素的垂直外边距相邻了，则子元素的外边距会设置给父元素
4. 浏览器默认样式

浏览器在页面中没有样式时，也可以让页面有一个比较好的显示效果，

所以为很多的元素都设置了一些默认的margin和padding，而这些默认样式，正常情况下不需要使用，所以在编写样式之前需要将浏览器中默认的margin和padding样式去掉

\*{margin:0;padding:0}

1. 内联元素的盒模型

内联元素不能设置width和height

内联元素可以设置水平方向的内边距。可以设置垂直方向内边距，但是不会影响页面布局

内联元素可以设置水平内边框，但是垂直边框不会影响页面的布局，水平会影响

内联元素可以设置水平方向的外边距，水平方向的外边距不会重叠，而是求和，内联元素不支持垂直外边距

1. Display：可以修改元素的类型

可选值：

Inline:可以将一个元素作为内联元素显示

Block:可以将一个元素设置块元素显示

Inline-block:将一个元素转换为行内块元素，可以使一个元素既有块元素又有行内元素的特点，既可以设置宽高，又不会独占一行

None:不显示元素，并且元素不会再页面中继续占有位置

Visibility:用来设置元素的隐藏和显示的状态

可选值：

visible:默认值，元素默认在页面中显示

Hidden:元素会隐藏不显示，隐藏的元素虽然不会再页面中显示，但是它的位置会依然保持

1. Overflow

子元素默认是存在于父元素的内容区中，理论上讲子元素的最大等于元素内容区大小，如果子元素的大小超过子元素的内容区，则超过的大小会在父元素以外的位置显示，超过父元素的内容，我们称为溢出的内容

父元素默认是将溢出内容，在父元素外显示，通过Overflow可以设置父元素如何处置溢出内容

可选值：

Visible,默认值，不会对溢出内容做处理，元素会在父元素以外的位置显示

Hidden,溢出的内容，会被修剪，不会显示

Scroll,会为父元素添加滚动条，通过拖动滚动条来查看完整内容，该属性不论内容是否溢出，都会添加水平和垂直双方向的滚动条

Auto,会根据需求自动添加滚动条，需要水平就添加水平，需要垂直就添加垂直，都不需要就都不添加

1. 文档流

文档流处在网页的最底层，它表示的是一个页面中的位置，我们所创建的元素默认都处在文档流中

元素在文档流中的特点：

块元素

1. 在文档流中独占一行，块元素会自上向下排列
2. 块元素在文档流中默认宽度是父元素的100%
3. 当元素宽度的值为auto（就是不设置高和宽）时，此时指定的内边距不会影响可见框的大小，而是会自动修改宽度，以适应内边距
4. 块元素在文档流中高度默认被内容撑开

内联元素

1. 在文档流中只占自身的大小，会默认从左向右排列，如果一行中不足以容纳所有的内联元素，会换到下一行，继续自左向右
2. 在文档流中内联元素的高度和宽度默认都被内容撑开
3. 浮动

如果希望块元素咋页面中水平排列，可以使块元素脱离文档流，使用float来使元素浮动，从而脱离文档流

可选值：

None默认值，元素默认在文档流中排列

Left元素会立即脱离文档流，向页面左侧浮动

right元素会立即脱离文档流，向页面右侧浮动

1、当为一个元素设置浮动以后，元素会立即脱离文档流，元素脱离文档流以后，它下边的元素会立即向上移动，元素浮动以后，会尽量向页面的左上或右上漂浮，直到遇到父元素的边框或者其他浮动元素

2、如果浮动元素上边是一个没有浮动的块元素，则浮动元素不会超过块元素

3、浮动的元素不会超过她上边的兄弟元素，最多最多一边齐

4、浮动的元素不会盖住文字，文字会自动环绕在浮动元素的周围，可以通过浮动来设置文字环绕图片的效果

5、当元素设置浮动以后，会完全脱离文档流，块元素脱离文档流以后，高度和宽度都被内容撑开

6、内联元素设置浮动，脱离文档流后，会变成块元素

31、塌陷

在文档流中，父元素的高度默认是被子元素撑开的，也就是子元素多高，父元素就多高，但是当为子元素设置浮动时，子元素会完全脱离文档流，此时将会导致子元素无法撑起父元素的高度，导致父元素的高度塌陷，由于父元素的高度塌陷了，则父元素下的所有元素都会向上移动，这样会导致页面布局很乱

解决1：根据W3C的标准，在页面中元素都有一个隐含的属性叫做Block Formating Context,简称BFC，该属性可以设置打开或者关闭，默认是关闭的，当开启元素的BFC以后，元素将会具有如下特性：

1. 父元素的垂直外边距不会和子元素重叠
2. 开启BFC的元素不会被浮动元素所覆盖
3. 开启BFC的元素可以包含浮动的子元素

如何开启元素的BFC

1. 设置元素浮动:使用这种方式开启，虽然可以撑开父元素，但是会导致父元素的宽度丢失，而且使用这种方式会导致下边的元素上移，不能解决问题
2. 设置元素绝对定位
3. 设置元素为inline-block：可以解决问题，会导致宽度丢失，不推荐这种方式
4. 将元素的overflow设置为一个非visible的值

推荐方式：将overflow设置为hidden事副作用最小的开启BFC的方式（overflow：hidden/auto），但是在IE6及以下浏览器中不支持BFC,所以使用这种方式不能兼容IE6，在IE6中虽然没有BFC，但是有另一个隐含的属性hasLayout，该属性作用和BFC类似，所在IE6浏览器可以通过开启hasLayout来解决

开启方式：直接将元素的zoom设置为1,zoom表示放大的意思，后边跟着一个数值，写几就将元素放大几倍，zoom:1表示不放大元素，但是通过该样式可以开启

解决二：可以直接在高度塌陷的父元素的最后，添加一个空白的div，由于这个div并没有浮动，所以他是可以撑开父元素的高度的，然后在对其进行清除浮动，这样可以通过这个空白的div来撑开父元素的高度

<div class="box1">

        <div class="box2"></div>

        <div class="clear"></div>

</div>

.clear{clear:both}

使用这种方式虽然可以解决问题，但是会在页面中添加多余的结构

解决三：可以通过after伪类向元素的最后添加一个空白的块元素，然后对其清除浮动，这样做和添加一个div的原理一样，可以达到一个相同的效果，而且不会再页面中添加多余的div,就是我们最推荐使用的方式，几乎没有副作用，但是在IE6中不支持after伪类，所以在IE6中还需要使用haslayout来处理

<div class="box1 clearfix">

        <div class="box2"></div>

    </div>

.clearfix:after{content: "";display: block;clear:both}

.clearfix{zoom:1}

1. 清除浮动

有时希望清除其他元素浮动对当前元素产生的影响，这时可以使用clear来完成功能，clear可以用来清除其他浮动元素对当前元素的影响

可选值：

None:默认值，不清除浮动

Left:清除左侧浮动

Right:清除右侧浮动

Both:清除两侧浮动元素的影响（清除对他影响最大的元素的浮动）

清除浮动以后，会回到其他元素浮动之前的位置

1. 定位

就是将指定的元素摆放到页面的任意位置，通过定位可以任意摆放元素。

通过position属性设置元素的定位

可选值：

static:元素没有开启定位

Relative：开始元素的相对定位

1. 当开启时，不设置偏移量时，元素不会发生任何变化
2. 可以通过left、right、top、bottom四个属性来设置元素的偏移量：left:元素相对于其定位位置的左侧偏移量

Right:元素相对于其定位位置的右侧偏移量

Top:元素相对于其定位位置的上边偏移量

Bottom:元素相对于其定位位置的下边偏移量

通常偏移量只需要使用两个，就可以对一个元素进行定位了，一般会选择水平方向的一个偏移量和垂直方向的偏移量为一个元素进行定位

1. 相对定位是相对于元素在文档流中原来的位置进行定位
2. 相对定位的元素不会脱离文档流
3. 相对定位会使元素提升一个层级（定位元素会盖住文档流元素）
4. 相对定位不会改变元素的性质，块还是块，内联还是内联

Absolute:开始元素的绝对定位

1. 开启绝对定位，会使元素脱离文档流
2. 开启之后，如果不设置偏移量，则元素的位置不会发生变化
3. 绝对定位是相对于离他最近的开启的定位的祖先元素进行定位的（一般情况，开启子元素的绝对定位都会同时开启父元素的相对定位）
4. 如果所有的祖先元素没有开启定位，则会相对于浏览器窗口进行定位
5. 绝对定位会使元素提升一个层级（定位元素会盖住文档流元素）
6. 绝对定位会改变元素的性质：内联元素变成块元素，块元素的宽度和高度默认都被内容撑开

fixed：开始元素的固定定位（也是绝对定位的一种）

固定定位也是一种绝对定位，他的大部分特点和绝对定位一样

不同的是：1、固定定位永远都会相对于浏览器窗口进行定位

2、固定定位会固定在浏览器窗口的某个位置，不会随滚动条滚动

IE6不支持固定定位

如果定位元素的层级是一样，则下边的元素会往上边的

1、通过z-index属性可以用来设置元素的层级，可以为z-index指定一个正整数作为值，该值将会作为当前元素的层级，层级越高，越优先显示

2、对于没有开启定位的元素不能使用z-index

3、父元素的层级再高，也不会盖住子元素

33、opacity

设置元素的透明背景，它需要一个0-1之间的值

0表示完全透明

1表示完全不透明

0.5表示半透明

这个属性在IE8及以下的浏览器中不支持，所以需要使用如下属性代替：

alpha(opacity=透明度)，透明度需要一个0-100之间的值

0表示完全透明

100表示完全不透明

50表示半透明

这种方式支持IE6,但是这种效果在IE Tester无法测试

Filter:alpha(opacity=50)

1. 背景

Background-attachment用来设置背景图片是否随页面一起滚动

可选值：

scroll,默认值，背景图片随窗口滚动

Fixed,背景图片会固定在某一位置，不随页面滚动

不随窗口滚动的图片，我们一般都是设置给body,而不设置给其他元素（如果设置给其他元素，会相对于浏览器定位，而且元素会随浏览器滚动，当元素看不见时，图片也看不见了）

当背景图片的Background-attachment设置为fixed时，背景图片的定位（background-position）永远相对于浏览器窗口

1. 图片整合技术

由于加载图片需要一定的时间，所以在加载和显示过程会有一段时间，背景图片无法显示，导致出现闪烁的情况，为了解决这一问题，可以将三个图片整合为一张图片，这样可以同时将三张图片一起加载，就不会出现闪烁的问题了，然后再通过background-position来切换要显示的图片的位置，这就叫图片整合技术

优点：

1. 将多个图片整合为一张图片里，浏览器只需要发送一次请求，可以同时加载多个图片，提高访问效率，提高用户体验
2. 将多个图片整合为一张图片，减少了图片的总大小，提高请求的速度，增加了用户体验
3. 表格

Tr行

Td列

Th可以表示表头中的内容，他的用法和td一样，不同的是它会有一些默认效果（加粗、居中）

Colspan:横向合并单元格

Rowspan:纵向合并单元格

Table和td边距之间默认有一个距离，通过border-spacing属性可以设置这个距离

Border-collapse:collapse可以设置表格的边框合并，如果设置了边框合并，则border-spacing自动失效

长表格：需要将表格分为三个部分，表头、表格的主体、表格底部

在html中为我们提供了三个标签：

Thead表头

Tbody表格主体

Tfoot表格底部

这三个标签的作用，就来区分表格的不同部分，他们都是table的子标签，都需要直接写到table中，tr需要写在这些标签当中

Thead中的内容，永远显示在表格的头部

Tfoot中的内容，永远显示在表格的底部

Tbody中的内容，永远显示在表格的中间

如果表格中没有写tbody,浏览器会自动在表格中添加tbody,并且将所有的tr都放到tbody中，所以注意tr并不是table的子元素，则是tbody的子元素，通过table>tr无法选中行，需要通过tbody>tr

1. 表单

使用form标签创建一个表单，form标签中必须指定一个action属性，该属性指向的是一个服务器的地址，当我们提交表单时将会提交到action属性对应的地址

如果希望表单项中的数据会提交到服务器中，还必须给表单项指定一个name属性，name表示提交内容的名字

用户填写的信息会附在url地址的后边以查询字符串的形式发送的服务器，url地址？查询字符串

1. 框架集

框架集和内联框架的作用类似，都是用于在一个页面中引入其他的外部的页面，框架集可以随时引入多个页面，而内联框架只能引入一个，在h5标准中，推荐使用框架集，而不是使用内联框架

使用frameset来创建一个框架集，注意frameset不能和body出现在同一个页面中，所以要使用框架集，页面中就不可以使用body标签

属性：rows,指定框架集中的所有的框架，一行一行的排列

Cols,指定框架集中的所有的页面，一列一列的排列

这两个属性frameset必须选择一个，并且需要再属性中指定每一部分所占大小

Frameset中也可以在嵌套frameset

Frameset和iframe一样，他里面的内容都不会被搜素引擎所检索，所以如果搜索引擎检索到的页面是一个框架页的话，他是不能去判断里面的内容的，使用框架集则意味着页面中不能有自己的内容，只能引入其他的页面，而我们每单独加载一个页面，浏览器都需要重新发送一次请求，引入几个页面就需要发送几次请求，用户体验比较差

在frameset中使用frame子标签来指定要引入的页面，引入几个页面就写几个frame

<frameset>

        <frame src="导航条.html"> </frame>

</frameset>

1. ie6png的修复

在IE6中对图片格式png24支持度不高，

如果使用的图片格式是png24，则会导致透明效果无法正常显示

解决方法：

1.可以使用png8来代替png24，即可解决问题，

但是使用png8代替png24以后，图片的清晰图会有所下降

2.使用JavaScript来解决该问题，需要向页面中引入一个外部的JavaScript文件

然后在写一下简单的JS代码，来处理该问题

<!-- 在body标签的最后引入外部的JS文件 -->

<scripttype="text/javascript src="js/DD\_belatedPNG\_0.0.8a-min.js"></script>

<!--再创建一个新的script标签，并且编写一些js代码 -->

<script type="text/javascript">

DD\_belatedPNG.fix("\*");

</script>

Javascrip

1. 控制浏览器弹出一个警告框：alert(“”)
2. 让计算机在页面中输出一个内容：document.write()可以向body中输出一个内容
3. 向控制台输出一个内容：console.log(“”)
4. js位置

①可以将js代码编写到标签的onclick属性中，当我们点击按钮时，js代码才会执行。②可以将js代码写在超链接的href属性中，这样当点击超链接时，会执行js代码。③可以将js代码编写到script中。可以将js代码编写到外部js文件中，然后通过script标签引入，script一旦用于引入外部文件了，就不能编写代码，即使编写了浏览器也会忽略，如果需要则可以在创建一个新的srcipt标签用于编写内部代码

1. Js基本语法

多行注释/\*\*/

单行注释//

Js中严格区分大小写

Js中每一条语句以分号；结尾，如果不写分号，浏览器会自动添加，但是会消耗一些系统资源，而且有些时候，浏览器会加错分号，所以在开发中分号必须写

Js中会忽略多个空格和换行，所以我们可以利用空格和换行对代码进行格式化

1. 字面量和变量

字面量：都是一些不可改变的值，比如：1 2 3 4。。，字面量都是可以直接使用，但是我们一般都不会直接使用

变量：可以用来保存字面量，而且变量的值是可以任意改变的，变量更加方便我们使用，所以在开发中都是通过变量去保存一个字面量，而很少直接使用字面量

声明变量，在js中使用var关键字来声明一个变量

1. 标识符

在js中所有的可以由我们自主命名的可以称为是标识符，例如：变量名，函数名，属性名

命名一个标识符时需要遵守如下的规则：

1. 标识符中可以含有字母、数字、\_、$
2. 标识符不能以数字开头
3. 标识符不能是JS中的关键字或保留字
4. 标识符一般都采用驼峰命名法（首字母小写，每个单词的开头字母大写，其余小写），例如：helloWorld

Js保存标识符时实际上是采用的Unicode编码，所以理论上讲，所有的utf-8中含有的内容都可以作为标识符

1. 数据类型

指的是字面量的类型

在js中一共有六种数据类型

String 字符串

在js中字符串需要使用引号引起来

使用双引号或单引号都可以，但是不要混着用

引号不能嵌套引号

在字符串中我们可以使用\作为转义字符，当表示一些特殊符号时可以使用\进行转义

\n表示换行 \t制表符（空格） [\\表示\](\\\\表示\\)

Number 数值

在js中所有的数值都是Number类型，包括整数和浮点数（小数）

可以使用一个运算符typeof，来检查一个变量的类型，语法：typeof 变量

检查字符串时返回string

检查数值时返回number

Js中可以表示的数字的最大值：Number.MAX\_VALUE

如果使用Number表示的数字超过了最大值，则会返回一个Infinity，Infinity表示正无穷，-Infinity表示负无穷，使用typeof检查Infinity也会返回Number

Number.MIN\_VALUE大于0的最小值，5e-324

Js中整数的运算基本可以保证精确

如果使用js进行浮点运算，可能得到一个不精确的结果，所以千万不要使用js进行对精确度要求比较高的运算

NaN是一个特殊的数字，表示Not A Number，使用typeof检查一个NaN也会返回number

Boolean 布尔值

布尔值只有两个，主要用来做逻辑判断：

True,表示真

Flase，表示假

使用typeof检查一个布尔值时，会返回boolean

Null 空值

Undefined未定义

Object对象

其中String Number Boolean Null Undefined属于基本数据类型

Object属于引用数据类型