宽高自适应、

1. 宽度自适应

语法：width：100%；块元素默认宽度100%。

1. 高度自适应

语法：height：auto； 或者不设置高度。

1. 最小、最大高度 、宽度
2. 最小高度

语法：min-height：数值+单位；

兼容性：在IE6下不能识别min-height属性，解决方案如下：

**方案一：min-height：100px； \_height:100px;**

**添加下划线的css属性只有IE6识别，其他浏览器不识别。**

方案二：min-height：100px； height：auto！important； height：100px；

添加！important关键词的css属性值除了IE6不识别 其他浏览器都识别。

1. 最大高度

语法：max-height：数值+单位；

1. 最小宽度

语法：min-width：数值+单位；

注：块元素设置min-width属性时，需要先转换为内联块元素。

1. 最大宽度

语法：max-width：数值+单位；

1. 高度塌陷问题（常见的几种清浮动方法）

描述：父元素高度自适应，子元素设置float后，导致父级元素高度为0，称为高度塌陷问题。

1. 给父级设置高度

缺点：违背了高度自适应原则。

1. 给父级添加overflow：hidden；

原理：子元素设置浮动后脱离了默认文档流，在父元素中不占据空间，父元素在计算高度时，浮动的子元素不参与父元素的高度计算，导致父元素高度为0.当给父级设置了overflow:hidden;就触发了BFC（block formatting context 中文译为块级格式化上下文），此时闭合了子元素浮动，那么父元素在计算高度时子元素参与了父元素的高度计算，父元素高度不再为0。

缺点:溢出再父元素之外的部分会被隐藏。

1. 在浮动元素下方添加空div，并设置如下样式：

Div{clear:both; height:0; overflow:hidden;}

注：设置height 为0 和overflower：hidden；是因为在IE6不识别较小高度的容器，所以通过overflow：hidden；把超过高度以外的部分隐藏掉，保证浏览器的兼容性

缺点：在页面中添加若干个无意义的div容易造成代码冗余。

1. 万能清除浮动法

父元素:after{content:””; display:block; clear:both; height:0; overflow:hidden; visibility:hidden;}

优点：官方推荐清楚浮动的方法。

扩展：display：none；和visibility：hidden；的区别

Display：none；元素隐藏不可见，位置不保留。

Visibility:hidden; 元素隐藏不可见，位置保留。

扩展：

E:first-letter 匹配段落文本中第一个字符。

E:first-line 匹配段落文本中的第一行

E::selection 改变文本选中时效果（只能改变文字颜色和背景色。）

1. div和span
2. div

语法：<div>< /div>

作用：划分板块。

1. span

语法：<span></span>

注：1 span没有固定的表现形式，只有对他设置样式时，才会产生视觉上的变化

2 当我们需要把一部分文字单独设置 就要用span

1. 伪元素
2. 在父元素内所有内容之前添加一个伪元素

语法：E:before{

Content：“文字内容”；| content:url(图片路径)

Color： ;

….

}

注：我们经常使用**：** 或者**：：**来区别伪类和伪元素，两种写法呈现效果是一样的

无论内容是否为空 ，都不能省略content

伪元素是内联元素  
 只有：after可以用来清浮动

1. 行类元素、块元素水平居中

1.内联元素水平居中

如果被设置元素是行内元素，居中时通过给父级添加text-align：center；来实现如果 。

如果需要文本水平居中，给文本所在元素添加text-align：centrer

2.块元素水平居中

定宽 块状

margin：0 auto；

3.不定块元素水平居中

方案一：

父元素{text-align：center；}

子元素{display：inline-block；}

方案二、

父元素{display：table；margin：0 auto；}

注：display：table；将元素转换为表格形式。（table是内容撑开宽度，但是也是块属性的元素）

1. 屏幕窗口居中及子元素在父元素中水平垂直居中
2. 已知大小元素在屏幕窗口居中

元素{

width：200px；

height ：100px；

margin-left：-100px；

margin-top：-50px；

position：fixed；

left：50%；

top：50%；

}

1. 未知元素在屏幕窗口 水平垂直居中

元素{

position：fixed；

left：0；

right:0;

top:0;

bottom:0;

margin:auto;

}

1. 已知大小子元素在父元素中水平垂直居中

父元素设置相对定位：relative；

子元素{

width：20px；

height：10px；

position：absolute；

left：50%；

right：50%；

margin-left：负值父级宽度一半；

margin-top：负值父级高度一半；

}

1. 未知大小的子元素在父元素中水平垂直居中。

父元素设置相对定位：relative；

子元素{  
 position：absolute；

left：0；

right:0;

top:0;

bottom:0;

margin:auto;}

扩展：

父元素{

display：table-cell；

text-align：center；

vertical-align:middle;

}

注：display：table-cell;是将元素转换为表格单元格形式（等价于td）

1. 垂直对齐方式。

语法：vertical-align：baseline基线 底线（默认值）|top|middle（中部对齐）|bottom；

备注：1。应用在表格中实现单元格内容的垂直对齐方式。

2.应用在内联元素上，实现内联元素在一条线上对齐

3.vertical-align只对内联元素有效。