[八.文本格式化（重点\*\*\*\*\*\*\*\*） 2](#_Toc11310178)

[1.字体属性 2](#_Toc11310179)

[2.文本属性 3](#_Toc11310180)

[九.表格 4](#_Toc11310181)

[1.表格常用的样式属性 4](#_Toc11310182)

[2.表格特有的样式属性 5](#_Toc11310183)

[十.定位（重点\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*） 6](#_Toc11310184)

[1.什么是定位 6](#_Toc11310185)

[2.定位分类 6](#_Toc11310186)

[3.普通流定位（默认文档流） 6](#_Toc11310187)

[4.浮动定位 7](#_Toc11310188)

[5.浮动引发的特殊情况 7](#_Toc11310189)

[6.清除浮动 8](#_Toc11310190)

[7.高度坍塌 9](#_Toc11310191)

day11

# 八.文本格式化（重点\*\*\*\*\*\*\*\*）

CSS编写思路

|  |
| --- |
| 1.从上往下写，从左往右写  2.宽高，大体位置  3.边框，背景  4.文字所有样式  5.微调 |

## 1.字体属性

①font-size

②font-family

③font-weight

④字体的样式

|  |
| --- |
| font-style  取值：1.italic 斜体  2.normal 正常 |

⑤小型大写

|  |
| --- |
| font-variant:small-caps; |

⑥字体属性的简写方式

|  |
| --- |
| font:italic small-caps bold 40px chiller;  style variant weight size family  最精简的方式：  font: size family; |

## 2.文本属性

①字体颜色

|  |
| --- |
| color:合法颜色值; |

②文本对齐方式

|  |
| --- |
| text-align:  取值：left/center/right /justify两端对齐  疑点解惑：  1.text-align:center;和margin:0 auto;的区别  text-align:center;控制这个元素内部内容的对齐方式  margin:0 auto;让元素本身水平居中  2.使用text-align,可以对齐的元素有哪些：  行内元素可以对齐  文本可以对齐  行内块元素可以对齐  块级元素不能对齐 |

③行高（纯文本的垂直对齐，不能用vertical-align）

|  |
| --- |
| 定义一行数据的高度  特性：如果行高高度大于字体的大小  该行文本将在指定的行高内，呈垂直居中的方式显示  取值：1.以px为单位的数字，一般，行高的值与容器的高度值相同，让文本在容器中垂直居中显示。但是，如果文本有多航，不建议使用行高  2.无单位的数字（整数小数都可以），行高的值为字号的倍数 |

④文本的线条修饰

|  |
| --- |
| text-decoration  取值：1.overline上划线  2. underline下划线  3.line-through删除线  4.none 去掉所有线条(a标签去掉下划线) |

⑤首行缩进

|  |
| --- |
| text-indent:以px为单位的数字; |

⑥文本阴影

|  |
| --- |
| text-shadow:h-shadow v-shadow blur color;  h-shadow水平偏移量  v-shadow 垂直偏移量  blur 模糊距离  color 阴影颜色 |

# 九.表格

## 1.表格常用的样式属性

①table，之前学习的属性，基本都可以用

|  |
| --- |
| 尺寸，边框，背景，字体，文本，内外边距  table设置边框，只设置外部最大的边框，对单元格边框无效 |

②td/th的样式属性

|  |
| --- |
| 尺寸，边框，背景，字体，文字，内边距有效  外边距margin无效  vertical-align设置单元格数据的垂直对齐方式  取值：top/middle/bottom |

## 2.表格特有的样式属性

①边框合并

|  |
| --- |
| border-collapse:  取值：1.separate 默认值，边框分离  2.collapse 边框合并 |

②边框边距

|  |
| --- |
| border-spacing:  取值：1.取一个值，同时设置水平和垂直边框边距  2.取两个值，第一个设置水平，第二个设置垂直  注意：如果边框是合并状态，此属性无效  表格边框必须保证是border-collapse:separate;状态 |

③表格标题位置

|  |
| --- |
| caption-side:  取值：top/bottom; |

④设置表格的显示规则

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| table-layout:  取值：1.auto默认值，自动表格布局  列的尺寸实际上由内容决定，内容比尺寸大，按内容显示，内容比尺寸小，按尺寸显示  2.fixed固定的表格布局  列的尺寸，以设置的宽高为准，不会根据内容多少变化   |  |  | | --- | --- | | 自动布局，默认 | 固定布局 | | 单元格的大小会自动适应内容 | 单元格的大小取决于设置的宽高 | | 表格复杂时，加载速度较慢 | 任何情况下，都会加速加载表格 | | 自动布局比较灵活（优点） | 固定表格布局不够灵活 | | 适用于不确定每列大小的，并且不复杂的表格 | 适用于明确知道每列大小的表格 | |

# 十.定位（重点\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*）

## 1.什么是定位

|  |
| --- |
| 改变元素在页面中的位置 |

## 2.定位分类

|  |
| --- |
| 1.普通流定位  2.浮动定位  3.相对定位  4.绝对定位  5.固定定位 |

## 3.普通流定位（默认文档流）

|  |
| --- |
| 1.每个元素在页面上都有自己的空间  2.每个元素都是从父元素的左上角开始显示  3.块级元素按照从上往下的方式逐个排列，每个元素单独成行  4.行内元素多个元素在一行中显示，从左往右 |

## 4.浮动定位

|  |
| --- |
| 让块级元素横向排列  float:  取值：left左浮动，让元素浮动后靠在父元素的左边  或者紧挨着左侧已有浮动元素  right右浮动，让元素浮动后靠在父元素的右边  或者紧挨着右侧已有浮动元素  none 默认值，没有浮动 |
| 浮动的特点  1.元素一旦浮动，脱离文档流（不占页面空间，后面未浮动元素上前补位）  2.浮动元素会停靠在父元素左边/右边，或者其他已浮动元素的左边或者右边  3.父级元素横向显示不下所有的浮动元素，显示不下的元素会自动换行  4.浮动元素解决了多个块级元素在一行同时显示的问题 |

## 5.浮动引发的特殊情况

①浮动元素的占位问题

|  |
| --- |
| 当父元素一行显示不下所有浮动元素时，最后显示不下的元素会换行。但是，已浮动元素会根据自己的浮动方向占据位置，导致被挤下去的浮动元素，需要在更下面的位置显示。 |

②浮动元素的宽度

|  |
| --- |
| 元素一旦浮动，如果元素没有定义宽度，那么浮动元素的宽度以内容为准 |

③元素一旦浮动，会变成块级元素

|  |
| --- |
| 可以设置尺寸，可以设置上下外边距 |

④文本，行内元素，行内块，遇到浮动元素的特殊情况

|  |
| --- |
| 它们如果不浮动，是不会被浮动元素压在下面的  而是，巧妙的避开，环绕着浮动元素显示 |

## 6.清除浮动

|  |
| --- |
| 清除浮动元素给我带来的影响  影响：之前元素浮动，我去补位  清除了这个影响：我不去补位  clear  取值：1.left清除之前左浮动的影响  2.right清除之前右浮动的影响  3.none不清除  4.both清除之前左右浮动的影响 |

## 7.高度坍塌

|  |
| --- |
| 块级元素的高，如果不设置，默认靠内容撑开  如果没有设置高的块级元素，内部的所有子元素都浮动（都脱离文档流）  这个块级元素认为自己内部没有元素了，所以就没有高了  这个现象就叫做高度坍塌 |
| 高度坍塌的解决方案  1.父元素设置高度，弊端：不是每次都能明确知道具体高度  2.父元素设置浮动，弊端：会影响父元素后续元素  3.在父元素中追加一个空的，没有宽高的div,设置clear:both; |

练习：



作业：

1.完成下图

