八.文本格式化

1.字体属性

|  |
| --- |
| ①字号  ②系列  ③加粗  ④斜体  ⑤小型大写字母  ⑥简写 font: |

2.文本属性

①字体颜色

|  |
| --- |
| color:合法的颜色值 |

②文本对齐方式

|  |
| --- |
| text-align: left/center/right/justify(两端对齐)  text-align:center; 让内部的孩子水平居中(文本，行内，行内块)  margin:0 auto; 让元素本身水平据中 |

③文本的行高

|  |
| --- |
| 定义一行文本的高度  特性：如果行高大于字体本身字号大小，那么该行文字，将在指定行高的中间显示  line-height: 取值：1.px为单位的数字，  一般给行高设定成容器的高度，  让文字在容器垂直居中显示  2.无单位的数字，行高是字号的倍数  注意：行高定义文本垂直居中，不建议在多行文本下使用 |

④文本线条修饰

|  |
| --- |
| text-decoration:  overline  underline  line-through  none 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; |

九.表格的属性

1.表格的常用属性

①table

|  |
| --- |
| 常用属性基本都可以使用  尺寸，边框，背景，字体，文本，内外边距都可以用  注意，table的边框，只设置最外层边框 |

②td/th

|  |
| --- |
| 常用属性基本都可以使用  尺寸，边框，背景，字体，文本，内边距都可以用  外边距无效 |

③table的特性

|  |
| --- |
| table的尺寸，当内容少的时候，执行table设置的尺寸  当内容多的时候，靠内容撑开  table的渲染方式：先读到内存中，然后一次性渲染 |

PS：附加知识点，html对一个单词的判定

|  |
| --- |
| 当数字/字母，连着写，没有空格，html认为他是一个完整的单词，不换行。中间如果有空格，html判定，他是多个单词，一行显示不下就换行。中文被判定为一个文字是一个单词 |

2.table的特有属性

①table

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.边框的合并  border-collapse:  separate 默认值，分离状态  collapse 合并  2.边框边距  border-spacing:20px 20px;  取一个值，同时设置x，y方向的间距  取两个值，分别设置x,y方向的间距  前提：表格必须是边框分离状态border-collapse:separate;  3.标题位置  caption-side:top/bottom  4.设置表格的显示规则  table-layout:  1.auto 默认值，自动布局表格  列的尺寸，实际由内容决定  内容大，列就大，内容小，列按照设置尺寸  2.fixed 固定表格布局，列的尺寸永远按照设置的尺寸为准   |  |  | | --- | --- | | 自动布局 | 固定布局 | | 单元格大小会自动适应内容 | 单元格大小以设置为准 | | 表格复杂的时候，加载速度慢 | 任何情况下，加载速度都很快 | | 比较灵活 | 不够灵活 | | 适用于不确定每列大小，并且不复杂的表格 | 适用于，确定每列大小的表格 | |

②tr/td/th特有属性

|  |
| --- |
| vertical-align: 设置内容的垂直对齐方式  取值 top/middle/bottom |

十.定位

1.什么定位

改变元素在页面中的位置

2.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（默认文档流定位）

|  |
| --- |
| 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绘制  3.块级元素按照从上往下的顺序，逐行排列，每个元素独占一行  4.行内/行内块，多个元素在一行中，从左往右显示 |

4.浮动定位

|  |
| --- |
| 让块级元素横向显示  float: none  left  right  元素浮动：1.脱离文档流(页面不占空间，后续元素上前补位)  2.在当前行，向左/右对齐，如果之前有其他浮动元素，停靠其他浮动元素的后面 |
| 浮动元素的特点  1.元素一旦浮动，不占页面空间，后续元素上前补位  2.浮动元素停靠在父元素的左/右边缘,或者其他已浮动元素的后面  3.父元素横向显示不下所有浮动元素，会把显示不下的部分挤下去  4.块级元素横向显示，靠浮动解决 |

5.浮动元素引发的特殊效果

①浮动元素占位问题

|  |
| --- |
| 当父元素横向显示不下所有元素时，显示不下的元素会换行。  但是,已浮动会根据自己的浮动方向占据位置。导致被挤下去的元素会绕开被占据的位置显示。 |

②.浮动元素，如果没有定义宽度，最后的宽度是靠内容撑开

③元素一旦浮动之后，都会变成块级(设置宽高，垂直外边距有效)

④文本，行内元素，行内块，不会被浮动元素压在下面，而是巧妙的绕开，环绕着浮动元素显示

6.清除浮动

|  |
| --- |
| 清除之前浮动元素带来的影响  就是本元素不上前补位  clear:left  right  both |

7.高度坍塌

|  |
| --- |
| 发生高度坍塌的原因  父元素不设置高度，高度靠内容撑起  所有子元素都浮动，父元素认为页面上没内容  高度就为0；  解决高度坍塌  1.给父元素设置高度，很多时候，事先不知道明确的高度  2.父元素也浮动，不过会给父元素的兄弟元素带来影响  3. overflow:hidden;  4.父元素内，追加一个空的div，只设置clear:both |

|  |
| --- |
| 1.找元素  从外往里，从左往右，从上往下  2.写样式  ①宽高，大体位置  ②背景，边框  ③文本  ④微调 |