八.文本格式化

2.文本属性

①文本颜色

|  |
| --- |
| color:合法的颜色值 |

②文本对齐方式

|  |
| --- |
| text-align:  取值 left/center/right/justify  注意 1.一个元素写了text-align，控制的是内部的文本  2.这个元素自己想居中对齐，margin:0 auto;  3.text-align对是行内元素的孩子，都生效  4.孩子如果是块级元素，只会继承text-align。  而块级元素不会居中 |

③行高

|  |
| --- |
| line-height  取值 px为单位的数字  特性：如果行高的数值，大于字体的大小  那么该行文本将在指定的行高内部，成垂直居中的方式显示  行高，一般会设置成与容器高度相同的值，确保文字在容器垂直居中显示  如果文字发生折行，不建议使用行高做垂直居中 |

④文本的线条修饰

|  |
| --- |
| text-decoration:  取值 1.none 默认值，无线条  2.underline 下划线  3.overline 上划线  4.line-through 删除线  项目中使用最多的,a标签去除下划线 textdecoration:none; |

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  v-shadow 垂直偏移  blur 模糊距离  color 阴影颜色 |

练习

|  |
| --- |
| 01\_ex 定义一个div，内容随意，要求有中文，有英文，有数字  1.设置3个字体系列  2.设置文字大小  3.加粗并斜体  4.小型大写字母  5.两端对齐  6.有下划线  7.首行缩进50px  8.文本垂直居中  9.文本有阴影 |

九.表格

1.表格的常用样式

①table的样式

|  |
| --- |
| 尺寸，边框，背景，字体，文本，内外边距。  设置table的边框，只设置最外边一圈边框 |

②td/th

|  |
| --- |
| 尺寸，边框，背景，文字，文本，内边距都会起作用  外边距无效  vertical-align:设置单元格内部文本的垂直对齐方式  取值：top/middle/bottom  总结：vertical-align对tr,td,th有效，对table无效 |

练习

|  |
| --- |
| 02\_ex  创建4\*4的表格，宽高400px,内容随意  1.设置每个单元格宽高100px  2.设置表格和单元格的边框为1px solid #000  3.给每个单元格添加上外边距30px，无效 |

2.table特殊的 显示方式

|  |
| --- |
| 显示方式：行内元素，块级元素，行内块，table  table是一种特殊的表现方式  单元格内容较多，尺寸较小，单元格大小以内容为准  单元格内容较少，尺寸较大，单元格大小以尺寸为准  table在浏览器的渲染方式:先加载到浏览器内存中，在一次性画在页面上 |

3.表格的特有属性

①边框的合并和分离

|  |
| --- |
| border-collapse:  取值： separate 默认值，边框分离状态  collapse 边框合并状态 |

②边框的边距

|  |
| --- |
| border-spacing:x y;  取值，以px为单位的数字  前提，边框必须是分离状态，才有效border-collapse: separate； |

③表格的标题 <caption></caption>

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

④设置表格的显示规则

|  |
| --- |
| table-layout:  取值 auto 默认值，自动表格布局。  td的大小以内容和尺寸较大为准  fixed 固定表格布局  td的大小就以尺寸为准 |

表格自动布局和固定布局的区别

|  |  |
| --- | --- |
| 自动布局 | 固定布局 |
| 单元格的大小会自动适应内容 | 单元格取决于设置的值 |
| 表格复杂时，加载速度较慢 | 任何情况下，都是逐行渲染，渲染速度快 |
| 自动布局比较灵活 | 固定表格布局不够灵活 |
| 适用于不确定每列大小，并且不复杂的表格 | 适用于明确知道每列大小的表格 |

十.定位（重要\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*）

1.什么定位

|  |
| --- |
| 定义元素在页面中位置 |

2.定位的分类

|  |
| --- |
| 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.每个元素在页面都有自己的空间  2.每个元素都是从父元素的左上角开始渲染  3.行内元素和行内块在同一行显示，从左往右，一行放不下，折行  4.块级元素独占一行，从上往下排列 |

4.浮动定位

|  |
| --- |
| 让块级元素横向显示  float:  取值：1.left 左浮动，让元素浮动后停靠在父元素的左边，  或者其他已浮动元素的后面  2.right 左浮动，让元素浮动后停靠在父元素的右边，  或者其他已浮动元素的后面  3.none 默认值，不浮动 |
| 浮动的特点  1.元素一旦浮动，该元素脱离文档流  (不占页面空间,后面元素上前补位)  2.浮动元素，会在当前行，靠左/右，停靠在父元素的边缘，  或者是其他已浮动元素的边缘  3.父元素横向显示不下所有的浮动元素时，显示不下的部分会自动换行，  默认去距离当前行最近的位置  4.浮动，解决多个块级元素横向在一行显示的问题 |

5.浮动定位引发的特殊情况

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，最后显示不下会换行。  默认去离当前行最近的一行。  但是，已浮动元素会根据自己的浮动方向占据位置，导致被挤下去的元素，会躲开被占据的位置，在更下方显示 |

②元素一旦浮动，如果元素没有定义宽度，那么浮动之后，元素的宽度靠内容撑开

③元素一旦脱离文档流，会变成块级元素，尺寸，垂直外边距都生效

④文本，行内元素，行内块元素，是不会被任何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

脱离文档流，意味着什么

|  |
| --- |
| 1.不占页面空间 2.后续元素上前补位  3.变成块级元素 |

6.清除浮动效果(清除浮动)

|  |
| --- |
| 前面的浮动元素脱离文档，需要我上前补位  设置了清除浮动之后，我就不上前补位  clear:  取值：1.none;不清除浮动  2.right 清除之前右浮动元素给我的影响  3.left 清除之前左浮动元素给我带来的影响  4.both 清除左右元素给我带来的影响 |

7.高度坍塌

|  |
| --- |
| 父元素如果不设置高度，高度默认是内部内容撑起来的  如果内部所有元素都浮动了，那么父元素认为内部就没有内容了  所以高度为0  解决方案：  1.给父元素设置高度。弊端，很多时候，不知道确切的高度  2.父元素也浮动。弊端，父元素的兄弟元素，会受到影响  正确的解决方案：  在最后一个子元素的后面追加一个空的块级元素  给这个元素设置clear：both;  那么这个元素停留在文档流中，父元素的高可以找到他  这个元素不会被前面的浮动元素覆盖(clear:both)  这个元素不写宽高，没有内容，那么默认宽度是父级的100%  高度为0，不影响父级高度 |

css的编写步骤

|  |
| --- |
| 从上往下，从左往右，从外往里  1.宽高，大体位置  2.边框，背景  3.文字相关  4.微调 |

作业：

1.完成表格



2.完成专属定制



3.提高题

