day11

八.文本格式化(重点\*\*\*\*)

1.字体属性

①文本颜色

|  |
| --- |
| color |

②文本的对齐

|  |
| --- |
| text-align: left/center/right/justify  注意:  text-align:center和margin:0 auto;的区别  margin:0 auto; 让块级元素自己水平居中显示  text-align:center;让元素内部的孩子(非块级)水平居中 |

③行高

|  |
| --- |
| 定义一行文本的高度line-height:  行高的特点:当行高大于文本字号时,文本将在行高的垂直中间显示  文本想相对于一个容器垂直居中,把行高高度设置容器的高度  取值:1.以px为单位的数字  2.无单位的数字(整数小数都可以) 行高值为字号的整倍数  在文本有多行的情况下不建议使用行高控制垂直居中 |

④文本线条的修饰

|  |
| --- |
| text-decoration: overline 上划线  underline 下划线  line-through 删除线  none 去除所有线条  text-decoration:none;去掉a标签的下划线 |

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 水平方向偏移量  v-shadow 垂直方向偏移量  blur 模糊距离  color; 阴影颜色 |

练习

|  |
| --- |
| 01\_ex  定义div,内容随意,有中文  设置字体系列  设置文本大小  加粗并倾斜  所有小写字符转为小型大写字母  两端对齐  垂直居中  文本有下划线 首行缩进60px 文本颜色 文本 阴影 9:55~10:10休息 |

九.表格的相关样式

1.表格常用属性

|  |
| --- |
| table  尺寸,边框,背景,字体,文本,内外边距  table设置边框,只有最外一圈的边框,内部没用  td/th  尺寸,边框,背景,字体,文本,内边距,外边距失效  vertical-align:top/middle/bottom |

2.表格特殊表现方式

|  |
| --- |
| td中的数据长度少:  table设置的尺寸值大,td设置的尺寸值小,显示以table设置的为准  table设置的尺寸值小,td设置的尺寸值大,table宽度以table设置为准  table高度,以td设置为准  td中的数据长度多:  table尺寸,以数据的长度为准  总结:table是一种特殊的表现方式,数据少,尺寸大,听尺寸的  数据多,尺寸小,听数据的 |

3.表格特有的属性

①边框合并

|  |
| --- |
| border-collapse:默认值 separate/collapse; |

②边框边距

|  |
| --- |
| border-spacing: 前提,需要边框是分离状态border-collapse: separate  取值, px为单位的数字  一个值,水平和垂直距离同时设置  两个值,第一个值水平,第二个是垂直 |

③标题位置

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

④设置表格显示规则

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| table-layout: auto默认值 自动布局  列的尺寸,实际上由内容决定的,  内容比尺寸大,按内容显示  内容比尺寸小,按尺寸显示  表格数据预先读取进内存,再一次性渲染的到页面  fixed 固定布局,列的尺寸,就以设置的设置的尺寸为准   |  |  | | --- | --- | | table自动布局的特点 | table的固定布局特点 | | 单元格大小自动适应内容 | 单元格大小,取决于设置值 | | 表格复杂时,加载速度慢 | 任何情况下,加载都很快 | | 表格比较灵活 | 固定表格不灵活 | | 适用于不确定每列大小,并且不复杂的表格 | 适用于确定每列尺寸的表格 | |

三.定位(重要\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

1.什么是定位

|  |
| --- |
| 设置页面中元素的位置 |

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.每个元素在页面中都占自己的空间  2.每个元素都是从父元素的左上角开始显示  3.块级元素单独成行,从上往下逐个排列  4.行内元素和行内块,与其他行内元素和行内块共用一行  从左往右往右逐个显示,一行放不下自动换行 |

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,  停靠父元素的左边,或者挨着其它已浮动元素  2.right 右浮动,让元素浮动后,  停靠父元素的右边,或者挨着其它已浮动元素  3.none 默认缺省值,不做任何浮动,就是默认文档流  浮动的特点  1.元素一旦浮动,脱离文档流(不占页面空间,后面元素上前补位)  2.浮动元素停靠父元素的左边或者右边,或者是其它已浮动元素的边缘  3.父元素横向显示不下所有浮动的子元素时,显示不下的元素会自动换行  4.浮动元素主要解决,多个块级元素在同一行显示的问题 |

5.浮动引发特殊情况

①元素一旦浮动,如果元素没有定义宽度,那么元素浮动之后的宽度将以内容为准

②元素一旦浮动,就会变成块级元素(允许设置尺寸,垂直外边距有效)

③浮动元素存在占位问题

|  |
| --- |
| 浮动元素会在浮动的方向上占位,导致被挤下去的元素不能占用之前浮动元素浮动方向上位置 |

④天生不被浮动元素压着

|  |
| --- |
| 行内元素,行内块,文本,不会被浮动元素压在下面  而是环绕着浮动元素显示 |

脱离文档,意味着什么

|  |
| --- |
| 1.元素不在文档流占位置  2.后续元素上前补位  3.如果没有设置宽,宽度以内容为准  4.元素就会变成块级(设置宽高有效,上下外边距有效) |

6.清除浮动(清除之前浮动元素带来的影响)

|  |
| --- |
| clear:left  right  both  清除之前浮动元素,给我带来的影响(不被浮动元素压着) |

7.高度坍塌

|  |
| --- |
| 由于父元素没有定义高度,内部所有子元素都脱离文档流,导致父元素认为在文档流中没有子元素的高度了,所以父元素就失去了高度  解决方案:1.给父元素设置高度.弊端:很多时候,不能确定父元素高度  2.父元素也浮动.弊端:父元素浮动会影响父元素的同级元素  3.overflow:hidden/auto  4.在父元素中,追加一个空的块级元素,设置clear:both;16:15~16:30休息 |

写样式的思路

|  |
| --- |
| 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边框,背景  ③文本所有样式  ④微调 |

