day12

十一.显示

1.显示方式

|  |
| --- |
| 决定元素在网页中的表现形式(块级，行内，行内块，table)  语法：  display:  取值：  1.display:block； 让元素以块级的方式显示  2.display:inline； 让元素以行内的方式显示  3.display:inline-block； 让元素以行内块的方式显示  4.display:table； 让元素以table的方式显示  5.display:none； 让元素隐藏，脱离文档流，不占位置 |

练习

|  |
| --- |
| 01\_ex；创建5个div，使用display设置5个点不同显示方式  这个5div有统一的宽高，统一的背景 |

2.显示效果

|  |
| --- |
| 显示/隐藏  visibility  取值：1.visible 默认值，可见的  2.hidden 隐藏  问题：visibility:hidden和display:none的区别  display:none脱离文档流，后续元素上前补位  visibility:hidden，不脱离文档流，虽然看不到，但是还占位置 |

3.透明度

|  |
| --- |
| opacity:  取值 0~1 1是不透明，0是全透明  问题：opacity和rgba的区别  opacity作用于元素，只要是颜色相关的属性都会改变透明度  rgba只会修改当前颜色的透明度 |

4.垂直对齐方式

|  |
| --- |
| vertical-align 一般只有两个地方使用  table中使用 取值 top/middle/bottom  设置table中的内容的 对齐方式  img中使用  设置图片与图片前后文字的垂直对齐方式  取值 top/middle/bottom baseline 基线 默认值  一般项目中通常会将所有的图片与文字垂直对齐方式，  更改为非基线对齐 |

5.光标

|  |
| --- |
| cursor:  默认值 default  小手 pointer  十字 crosshair  文本 text  等待 wait  帮助 help |

十二.列表的样式

1.列表标识项的样式

|  |
| --- |
| list-style-type  取值 1.disc 默认值  2.none 去掉标识项  3.circle  4.square |

2.列表项设置为图片

|  |
| --- |
| list-style-image:url(图片路径) |

3.列表项的位置

|  |
| --- |
| ul默认自带上下外边距(chrome解析16px)  自带左内边距(chrome解析40px)  list-style-position  可以设置列表项在li里，或者设置列表在内边距里  取值 outside 默认值 在内边距里，在li外  inside 在li里 |

4.简写方式

|  |
| --- |
| list-style:type/url position;  最简方式，项目中使用最多的方式  list-style:none; |

十三.定位---相对，绝对，固定定位

|  |
| --- |
| position:  取值：1.static 默认，静态(默认文档流)  2.relative 相对定位  3.absolute 绝对定位  4.fixed 固定定位  当一个元素设置了position属性，并且取值为  relative/absolute/fixed其中一种时  这个元素被称为 已定位元素  已定位元素解锁了4个偏移属性  top： + ↓ -↑  right +← -→  bottom +↑ -↓  left +→ -← |

1.相对定位

|  |
| --- |
| position:relative; 配合偏移属性实现定位  相对定位，没有脱离文档流  如果相对定位元素，不写偏移属性，效果与没写定位是一样的，不影响任何布局，只是把这个元素变成了已定位元素  相对定位，相对自己原来的位置偏移某个距离  使用场合：  1.元素本身，位置微调(类似margin)  2.一般作为绝对定位的祖先元素 |

2.绝对定位

|  |
| --- |
| position：absolute; 配合偏移量使用  绝对定位，脱离文档流  绝对定位元素，如果祖先级没有已定位元素，那么就相对于body左上角，执行偏移量  绝对定位元素，会相对于离自己最近的，祖先级，已定位元素的左上角，进行偏移  绝对定位，由于脱离文档流，产生如下效果  1.页面不占据空间，后续元素上前补位  2.绝对定位的元素，会变成块级  3.没有写宽度元素，发生绝对定位以后，宽度靠内容撑开 |

3.固定定位

|  |
| --- |
| position:fixed; 配合偏移属性使用  将元素固定在页面上某个位置，不会随着滚动条滚动，发生变化  一直固定在页面的可视区域  特点：脱离文档流，位置始终相对body初始化 |

4.堆叠顺序

|  |
| --- |
| 注意：  1.默认的堆叠顺序，html元素后写的堆叠顺序高  2.浮动和定位的堆叠顺序不是同一个体系，尽量避免在一起处理  3.手动调整堆叠顺序  z-index:整数  4.堆叠顺序对父子关系无效，儿子永远在爹的上面  5.只有已定位元素，可以设置堆叠顺序 |

**CSS3 CORE**

一.复杂选择器

1.兄弟选择器

|  |
| --- |
| 兄弟元素，具备相同父元素的平级元素之间称为兄弟元素  兄弟选择器，只能找弟弟，不能找哥哥，只能往后找，不能往前找 |

①相邻兄弟选择器

|  |
| --- |
| 选择器 + 选择器{}  获取紧紧挨在某元素后面的兄弟元素  p+.c1{color:#f00;} |

②通用兄弟选择器

|  |
| --- |
| 选择器 ~ 选择器{}  获取元素后面所有符合条件的兄弟  #p1~.c1{color:#f00;} |

练习

|  |
| --- |
| h3-静夜思  div-床前明月光  div-疑是地上霜  div-举头望明月  div-低头思故乡  标题与正文之间30px  每行正文之间20px |

2.属性选择器

|  |
| --- |
| 允许通过元素所附带的属性，及其值来匹配页面元素，很精准  attr表示任意属性  [attr]{} 匹配有attr这个属性的元素  [attr1][attr2]{} 匹配同时拥有attr1和attr2属性的元素  [attr=value]{} 匹配拥有attr并且值为value的元素  elem[attr=value]{} 匹配拥有attr并且值为value的elem元素  elem[attr1][attr2]{}  模糊属性值的查询  [attr^=value] 匹配attr值value开头的元素  [attr$=value] 匹配attr值value结尾的元素  [attr\*=value] 匹配attr值中含有value的元素  [attr~=value] 匹配attr值中含有value这个单词的元素 |

作业：

1.学子商城1L



2.复习所有复杂选择器