```
1.什么是物理像素 ⑤ 答: 屏幕分辨率
                                              2.什么是开发像素 🕤 答: css像素px 就是开发像素
                                              3.开发像素在PC端和屏幕分辨率比例是一致吗? 
    答: 一致
                                 1.移动端相关知识
                                              4.开发像素在移动端和屏幕分辨率比例一致吗? ◎ 答:不一定一致,因为物理像素比可能不一样
                                              5.什么是物理像素比? ⑤ 答: 1个物理像素=多个开发像素
                                                              答:未来图片在移动端,因为物理像素比的存在,图片可能会放大而失真,影响用户
                                              6,物理像素比的影响 😑
                                                              体验,所以设计素材人员,在设计图片时,以多倍的方式设置
                                                  1.传统盒子(content-box) ⑤ 盒子大小:内容+padding+border ⑥ 缺点:盒子要是保持原来的大小,需要自己计算width 和 height
                                 2.盒子模型 box-sizing
                                                 2.CSS3盒子 (border-box) 🥏 盒子大小: 内容 (内容+padding+border) 💿 优点: 盒子的内容会自动适应模型的变化
              移动端 (流式) 布局
                                                 1.header 头部标签
                                                 2.footer 底部标签
                                                 3.section 区块标签
                                 3.扩展 (H5新增标签)
                                                 4.article 文本内容标签
                                                 5.aside 侧边标签
                                                 6.nav 导航标签
                                                 注意: 以上标签均可用div代替, 但是div没有语义
                                                                                             <meta name="viewport" content="width=device-width, user-
                                                       1.视口配置-理想视口(记得要写,写在字符编码下方即可)。
                                 4.京东移动端案例: (注意事项)
                                                                                             scalable=no,initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
                                                       2.初始化样式 🖯 normalize.css
                                                           1.row水平
                                    1.设置主轴方向 (flex-direction)
                                                          2.column垂直
                                                          3.row-reverse 反向/column-reverse 反向
                                                                     1.flex-start 从头开始
                                                                     2.flex-end 从尾部开始
                                    2.设置主轴上子元素的排列方式 (justify-content)
                                                                    3.center 居中
                                                                     4.space-around 平分间距
                                                                    5.space-between 两端对齐中间自动填充
                                                            1.nowrap 不换行 (默认)
                                    3.设置子元素是否换行 (flex-wrap)
                                                            2.wrap 换行
                                                                         1.flex-start 从上到下
              移动端flex (弹性) 布局
                                                                         2.flex-end 从下到上
移动端
                                    4.设置侧轴上的子元素排列方式(单行) (aligin-items)
                                                                         3.center 居中
                                                                         4.stretch 拉伸 (不能设置高度)
                                                                           1.flex-start (默认值在侧轴的头部开始)
                                                                          2.flex -end (在侧轴的尾部开始排列)
                                                                           3.center (在侧轴中间显示)
                                    5.设置侧轴上的子元素的排列方式(多行) (align-content)
                                                                           4.space-around (子项在侧轴平分剩余部分)
                                                                           5.space-between 子项在侧轴先分布在两头,在平分剩余空间
                                                                           6.stretch 设置子项元素高度平分父元素高度
                                             1.flex 属性定义子项目==分配容器的剩余空间==,用flex来表示占多少==份数==。
                                    6.flex属性
                                            3.order 属性定义项目的排列顺序/数值越小,排列越靠前,默认为0。
                                       1.em 

基于父亲的字体的大小
                                1.单位
                                       2.rem · 基于Html 字体的大小 (绝对唯一控制)
                                          作用: 查询屏幕, 相应屏幕变化
              移动端 (rem布局)
                                2.媒体查询 🥑 档位划分 例: @media screen and (min-width)
                                          等比:在各个挡位下,比例一直不变,变化的是rem后面的值
                                3.less 🐵 是一种css的扩展语言,辅助css计算使用
                                4.flexible.js 

引入此js 可以实现等比变化,不会一抽一抽的
                                                   w<768 超小屏幕 (xs:extra small)
                                                   768<= w <992 小屏设备
                                         1.档位划分
                                                   992<= w <1200 中等屏幕
                                                   1200<=w 大宽屏设备 (lg: large)
                                                   1.栅格系统 🕤 1.在各个档位下,控制子元素布局不同;将版心宽度均分为 12份
                                                           1.盒子有左右15padding
              响应式布局 (PC, 手机, pad)
                                         2.bootstrap
                                                   2.特点 🥑 2.盒子 col-lg: 有左右15padding
                                                          3.去除15padding: 加div.row (左右负 -15px margin 值)
                                                   3.作用:左右偏移,居中
                                                1.向下: 涉及向下布局, 需要手动写媒体查询
                                         3.备注
                                                                          例如: hidden-xs 只是xs时隐藏
                                                2.只要涉及隐藏,就要用到响应式工具
                                                                          例如: visible-xs 知识xs时显示
```