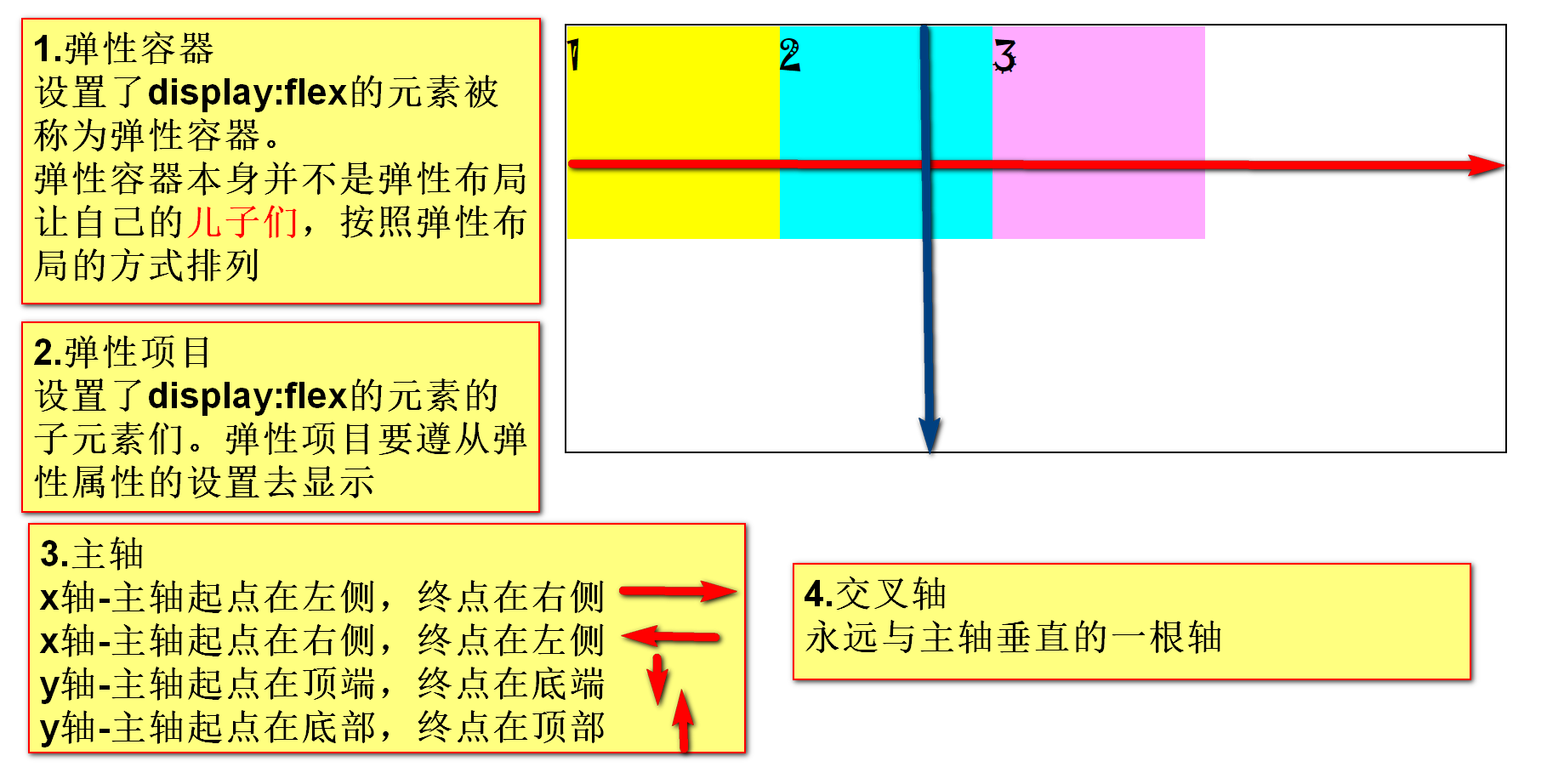
day14

二.弹性布局

|  |
| --- |
| 是一种应用最广泛，最流行的布局方式 |

1.弹性布局的相关概念



3.语法

|  |
| --- |
| display:flex; 让块级和行内元素变为容器  inline-flex; 让行内元素变为容器  当外部容器为弹性容器，他的子元素们都是弹性项目  项目在容器中所有的对齐方式，都用主轴和交叉轴控制  容器的text-align失效  项目的float失效 |

4.容器的样式属性

|  |
| --- |
| 控制所有项目  ①主轴方向  flex-direction:  row 默认值，x轴，起点在左侧  row-reverse x轴，起点在右侧  column y轴，起点在顶部  column-reverse; y轴，起点在底部  ②项目在主轴方向上的换行  flex-wrap: nowrap; 默认值，主轴方向空间不足时，项目压缩，不换行  wrap; 主轴方向空间不足时,项目不压缩，自动换行  ③上面两个属性的简写  flex-flow:direction wrap;  ④设置项目在主轴上的对齐方式  justify-content:  flex-start 默认值，主轴起点对齐  center 主轴中间对齐  flex-end 主轴终点对齐  space-around 每个项目左右间距相同  space-between; 主轴两端对齐  ⑤设置项目在交叉轴上的对齐方式  align-items: flex-start 默认值，交叉轴起点对齐  center 交叉轴中间对齐  flex-end 交叉轴终点对齐  baseline 以项目中的文本基线对齐  stretch 如果项目在交叉轴方向上没有设置尺寸  让项目在交叉轴方向填满 |

5.项目的样式属性

|  |
| --- |
| 控制当前项目 |

①控制某一个项目交叉轴上的对齐方式

|  |
| --- |
| align-self: flex-start 默认值，交叉轴起点对齐  center 交叉轴中间对齐  flex-end 交叉轴终点对齐  baseline 以项目中的文本基线对齐  stretch 如果项目在交叉轴方向上没有设置尺寸  让项目在交叉轴方向填满  auto 使用父容器设置algin-items的值 |

②设置项目在主轴上排列顺序

|  |
| --- |
| order：不带单位的数字  数值越小，离主轴起始点越近 |

③设置项目自动增长

|  |
| --- |
| 当主轴空间有剩余，设置项目是否自动增长，如何增长  flex-grow:0; 默认值 ，不增长  把每个项目flex-grow:的取值相加，得到n  把主轴剩余空间/n，为一份  每个项目在主轴上的最终尺寸为，项目原始尺寸+(主轴剩余空间/n)\* flex-grow的值 |

④设置项目自动缩小

|  |
| --- |
| 当主轴空间不足，设置项目是否自动缩小，如何缩小  flex-shrink:1； 默认值，自动缩小  0 不缩小  把容器主轴方向不足的空间，按照flex-shrink的取值进行处理  每个项目按照自己的flex-shrink分配要缩水的空间 |

⑤flex-basis

|  |
| --- |
| 设置项目在主轴上的基础占地尺寸  如果正常设置主轴上的尺寸，由于主轴方向会发生变化，我们设置宽或者高  使用flex-basis可以不管主轴方向，直接设置项目在主轴上的占地尺寸  flex-basis的优先级比主轴方向上的width、height高  注意：flex-basis不允许直接设置在img标签上。可以在img外层添加一个div  把flex-basis设置在div上  flex-basis:px为单位的数字  % |

⑥flex-grow,flex-shrink,flex-basis的简写

|  |
| --- |
| flex:grow shrink basis;  flex:0 0 50%; |

三.转换

|  |
| --- |
| 使用css改变元素在页面中的位置，大小，角度，形状  2D转换  3D转换 |

1.转换属性

|  |
| --- |
| transform:转换函数1 转换函数2 转换函数3......;  2D转换---4个转换函数（3个必须会，1个了解）  3D转换---1个转换函数 |

2. 2D转换

①位移，修改元素在页面上的位置

|  |
| --- |
| translate(x); 设置x轴的偏移属性  translatex(x) 设置x轴的偏移属性  translatey(y) 设置y轴的偏移属性  translate(x,y) 设置x轴和y轴 |

②缩放，修改元素在页面上的大小

|  |
| --- |
| scale(v1)  1.取一个值： 同时设置x,y轴的尺寸倍数  v1>1 放大  0<v1<1 缩小  -1<v1<0 缩小并反转  v1<-1 放大并反转  2.scale(x,y)  3.scalex(x)  4.scaley(y) |

③旋转，改变元素的角度

|  |
| --- |
| rotate(ndeg)  n:+ 顺时针，- 逆时针  注意：1.转换原点默认为元素中心，transform-origin: x y;  取值：px为单位的数字  %  关键字 x:left/center/right y:top/center/bottom  2.旋转是连同坐标轴一起旋转的，旋转之后再位移，是在新的坐标轴方向进行位移 |

④倾斜，改变元素的形状

|  |
| --- |
| transform:skewx(ndeg)等同于skew(ndeg)  skewy(ndeg)  skew(xdeg,ydeg) |

3.3D转换

|  |
| --- |
| 3D转换都是模拟的  所有3D效果必须设置透视距离，透视距离必须设置在3D转换的元素的父元素上 |
| 3D旋转 transform:rotate3d(x,y,z,ndeg);  x,y,z的取值 0：不参与旋转  >0 参与旋转  rotate3d(1,0,0,ndeg) 只有x轴参与旋转，围绕着x转，烤羊腿，老式爆米花机  rotate3d(0,1,0,ndeg) 只有y轴参与旋转，围绕着y转，旋转门，土耳其烤肉  rotate3d(0,0,1,ndeg) 只有z轴参与旋转，围绕着z转，煎饼，电风扇 |

四.过渡

|  |
| --- |
| 使用伪类激活，两个css值之间的变化过程 |

1.设置参与过渡的样式属性

|  |
| --- |
| transition-property:属性1 属性2 属性3....;  all 所有支持过渡的样式属性，都参与过渡  支持过渡的样式属性  1.颜色属性  2.大多数以px为单位的数值  3.阴影  4.转换 |

2.设置过渡持续时长

|  |
| --- |
| transition-duration: 5s; |

3. 设置时间曲线函数

|  |
| --- |
| transition-timing-function:  ease; 默认值 慢速开始，加速，减速，慢速结束  linear 匀速  ease-in 慢慢开始，一直加速  ease-out 快速开始，一直减速  ease-in-out 慢速开始，疯狂加速，疯狂减速，慢速结束 |

4.设置过渡的延迟时间

|  |
| --- |
| transition-delay: |

5.过渡简写方式

|  |
| --- |
| transition: property duration timing-function delay;  最简方式  transition:duration; |

6.过渡代码的编写位置

|  |
| --- |
| 写在原本样式中，过渡效果，有去有回  写在激活的伪类中，过渡效果，有去无回 |

练习

|  |
| --- |
| 1.翻滚吧，亮亮  2.亮亮的吻。呕~ |

五.动画

|  |
| --- |
| 动画是过渡的升级版  过渡的使用时机---使用伪类激活，两个css值之间的变化过程  动画的使用时机---使用伪类/刷新页面激活，多个css之间的变化过程 |

1.使用动画的步骤

|  |
| --- |
| ①创建动画----类似于创建方法  关键帧----------每一帧-----一种样式，以及这种样式出现的时间点  @keyframes 动画名称{  0% | from{样式}  .....  100% | to{样式}  }  ②调用动画  /\* 1.设置调用动画的名称 \*/  animation-name: jump;  /\* 2.设置动画的时间 \*/  animation-duration: 2s;  /\* 3.设置时间曲线函数 \*/  animation-timing-function: ease;  /\* 4.设置动画的延迟时间 \*/  animation-delay: 1s; |

2.动画的专有样式

①设置动画的执行次数

|  |
| --- |
| animation-iteration-count: infinite; 无限  具体数字 |

②设置动画的执行顺序

|  |
| --- |
| animation-direction:  normal； 默认值 0%-->100%  reverse; 方向运行 100%--->0%  alternate 第一遍0%-->100%  第二遍100%-->0%  第三遍0%-->100%  ....... |

③动画的简写方式

|  |
| --- |
| animation:name duration timing-function delay count direction;  最简方式 animation:name duration; |

④动画的填充模式

|  |
| --- |
| 在执行动画的延迟时间中，和动画执行完毕之后，元素的css效果是如何显示  animation-fill-mode:  backwards; 在动画的延迟时间内，填充动画的第一帧  forwards；动画结束后，填充动画的最后一帧  both 前后都要 |

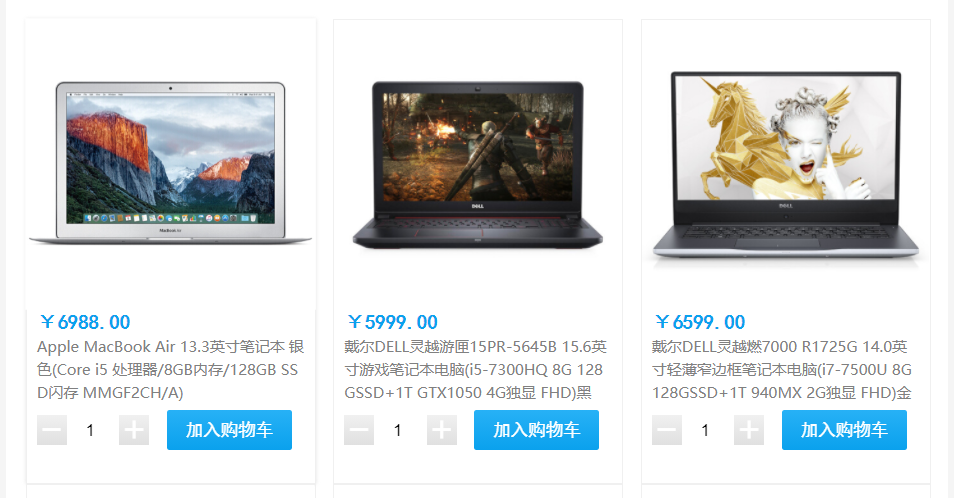
3.animate.css的使用

作业

1.使用弹性，使用过渡完成学子1L



2.使用弹性完成商品列表页部分列表



3.动画练习，完成动画时钟

