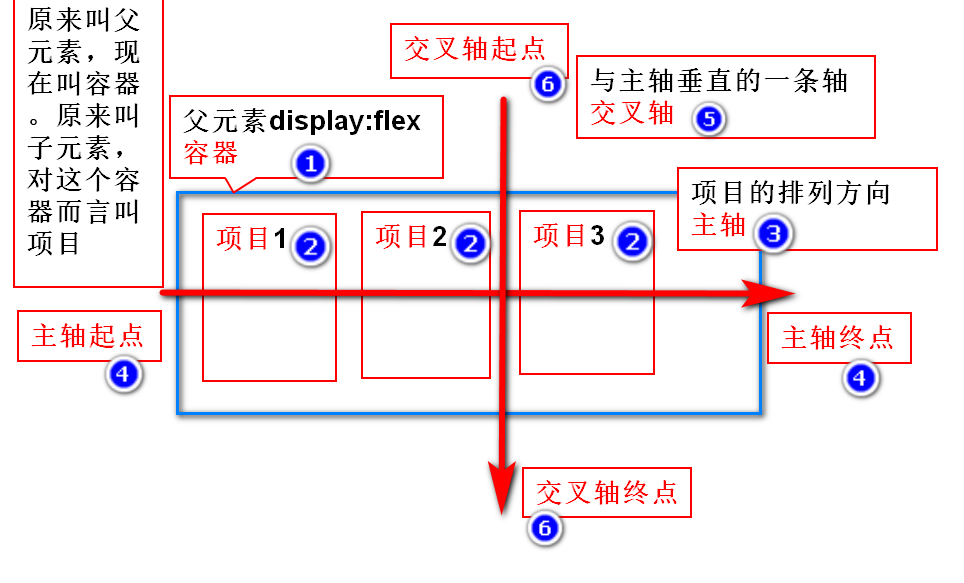
day13

**一.弹性布局 (重点\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)**

1.什么是弹性布局

|  |
| --- |
| 弹性布局就是一种布局方式  主要解决某元素中的子元素的布局方式，为布局提供最大的灵活性 |

2.弹性布局的相关概念和专业术语



|  |
| --- |
| 1.容器  要布局的子元素的父元素称之为容器，容器中写display:flex  2.项目  要布局的子元素称之为项目  3.主轴  项目们排列的方向，称之为主轴（水平和垂直）  如果项目们是按照横向排列，x轴就是主轴  如果项目们是按照纵向排列，y轴就是主轴  4.交叉轴  与主轴垂直相交的方向轴叫做交叉轴 |

语法

|  |
| --- |
| 将元素变为弹性容器，他所有的子元素将变成弹性项目，按照弹性布局的方式去排列显示  display  取值：flex, 将块级元素变为容器  inline-flex,将行内元素变为容器 |
| 元素设置为flex容器之后，子元素一些样式属性会失效  float/clear/vertical-align失效  子元素允许修改尺寸(项目是行内元素也可以修改尺寸)  容器的对齐方式text-align失效 |

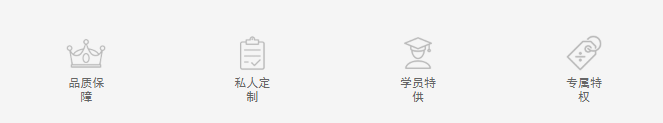
3.弹性容器的样式属性

|  |
| --- |
| 1.flex-direction 主轴方向  取值  1.row 默认值，主轴是x轴，主轴起点在左端  2.row-reverse, 主轴是x轴，主轴起点在右端  3.column 主轴是y轴，主轴起点在顶端  4.column-reverse主轴是y轴，主轴起点在底部 |
| 2.flex-wrap  当一个主轴排列不下所有项目时,项目的显示方式  取值  1.nowrap 默认值，空间不够，不换行，项目会自动压缩  2.wrap 空间不够，就换行，项目不压缩  3.wrap-reverse 换行，反转 |
| 3.flex-flow  是flex-direction和flex-wrap的缩写  取值 direction wrap;  ex:flex-flow:row wrap; |
| 4.justify-content  定义项目在主轴上的对齐方式  取值  1. space-between 两端对齐  2.space-around 每个间距相同  3.flex-start 默认值 在主轴起点对齐  4.flex-end 在主轴的终点对齐  5.center 在主轴上居中对齐 |
| 5.align-items (容器高度大于项目高度)  项目们在交叉轴上的对齐方式  取值：  1.flex-start 交叉轴起点对齐  2.flex-end 交叉轴终点对齐  3.center 交叉轴居中对齐  4.baseline 交叉轴基线对齐  5.stretch 如果项目未设置高度，在交叉轴上充满容器 |

4.项目的属性

|  |
| --- |
| 是单独设置给一个项目的，不影响容器和其他项目 |
| 1.order  取值为无单位的整数，定义项目的排列顺序，值越小，离起点越近，默认值为0； |
| 2.flex-grow  定义项目的放大比例  如果容器有足够大的剩余空间，项目将按照比例放大  取值为无单位整数，  默认值为0.不放大。取值越大，占据的剩余空间越多 |
| 3.flex-shrink  定义项目的缩小比例，空间不足时，项目该如何缩小  取值为无单位整数  默认值为1，空间不足，等比缩小  0，不缩小  取值越大，占据的空间越小 |
| 4.align-self  控制当前项目在交叉轴上对齐方式，与其他项目无关  1.flex-start 交叉轴起点对齐  2.flex-end 交叉轴终点对齐  3.center 交叉轴居中对齐  4.baseline 交叉轴基线对齐  5.stretch 如果项目未设置高度，在交叉轴上充满容器  6.auto 继承容器的align-items的效果 |

练习，使用弹性布局，完成下图



CSS hack

|  |
| --- |
| 由于不同的浏览器，比如IE6 IE7 firefox  对css的解析认识不同，会导致生成的页面效果不同，无法得到我们预期的效果  这个时候我们需要针对不同浏览器去写不同的css  让代码同时兼容不同的浏览器。  这个针对不同浏览器写不同css代码的过程，叫做CSS hack  课上不讲解，tmooc有对应的视频 |

二.转换（重点\*\*\*\*\*\*\*\*）

1.什么是转换

|  |
| --- |
| 改变元素在页面中的 位置 大小 角度 形状等的一种方式  2D转换，在x轴和y轴上，发生转换效果  3D转换，添加了z轴 |

2.转换属性

|  |
| --- |
| transform:  取值 transform-function  none  如果有多个转换函数，用空格隔开 |
| 转换原点  transform-origin  取值：px为单位的数字/%/关键字  2个值，表示原点在x轴和y轴上的位置  3个值，x,y,z轴 |

3.2D转换函数

|  |
| --- |
| 1.位移，改变元素在页面中的位置  transform:translate(x)指定元素在x轴上的位移距离  + 元素往右  - 元素往左  transform:translate(x，y)指定元素在x和y轴上的位移距离  x取值同上  y取值 + 元素往下 - 元素往上  transform:translateX(x)指定元素在x轴上的位移距离  transform:translateY(y)指定元素在y轴上的位移距离 |
| 2.缩放  改变元素在页面中的尺寸  transform:scale(n)  取值，无单位的数字  n>1 根据转换原点放大  0< n <1 根据转换原点缩小  -1< n <0 翻转缩小 （水平和垂直方向都翻转了180度）  n<-1 翻转放大  transform:scale(x,y) x和y缩放尺寸  transform:scaleX(x) 单独设置x的缩放  transform:scaleY(y) 单独设置y的缩放 |
| 3.旋转  改变元素在页面中的角度  transform:rotate(ndeg);  n为正 顺时针  n为负 逆时针  注意：  转换原点会影响最后的旋转效果、  旋转是连同坐标轴一起旋转的，会影响旋转后的位移效果 |

练习

|  |
| --- |
| d1先旋转45度，然后位移200px  d2先位移200px，再旋转45度 |

|  |
| --- |
| 4.倾斜  改变元素在页面中的形状  transform:skew(x),等同于skewX(x)  让元素向着x轴发生倾斜，实际上改变的y轴的角度  + 逆时针  - 顺时针  transform:skewY(y)  让元素向着y轴发生倾斜，实际上改变的时候x轴的角度  + 顺时针  - 逆时针  transform:skew(x,y) |

练习

|  |
| --- |
| 06\_ex创建200\*200的div  设置背景颜色，鼠标悬停时，该元素向右偏移200px;  向下偏移200px；旋转135deg，向着x轴倾斜45度，放大1.5倍 |

4.3D转换

|  |
| --- |
| 浏览器不支持3D的位移，z轴看不到，只能模拟  1.透视距离  模拟人的眼睛到3D转换元素之间的距离  perspective  该属性要加载转换元素的父元素上 |
| 2.3D旋转  transform:rotate3D(0,0,1,20deg);  取值  1.rotateX(xdeg)  以x轴为中心轴，旋转元素的角度(老式爆米花机，烤羊腿)  2.rotateY(ydeg)  以y轴为中心轴，旋转元素的角度(旋转门)  3.rotateZ(zdeg)  以z轴为中心轴，旋转元素的角度(风车，摩天轮)  4. rotate3D(x,y,z,20deg);  x,y,z取值大于0的数字时，表示该轴参与旋转  取值为0，表示不参与旋转 |

三.过渡

|  |
| --- |
| CSS属性值，在一段时间内容平缓的变化 |
| transition-property:background;  border-radius;  all;  能够使用过渡效果的属性  1.颜色  2.取值为数字的属性  3.转换  4.阴影  5.渐变  6.visibility |
| 指定过渡持续时间  transition-duration:1s/1000ms |
| 简写方式  transition:all 1s;  transition:1s;----不要写在hover中，写在元素本身的样式里 |

作业：

1.把转换的脑图自己总结

2.预习一下boot

3.完成学子首页，（尽量使用弹性布局）



4.

