day13

一.复杂选择器

3.伪类选择器

|  |
| --- |
| :link :visited :hover :active :focus |

①目标伪类

|  |
| --- |
| 在锚点被激活时，让锚点元素应用的样式  :target{} |

②结构伪类

|  |
| --- |
| 1.选择器:first-child{}  匹配的元素是属于其父元素的第一个子元素(找大哥)  同时，这个大哥还需要符合选择器的要求  2.选择器:last-child{}  匹配的元素是是属于其父元素的最后一个子元素(找小弟)  同时，这个小弟还需要符合选择器的要求  3.选择器:nth-child(n) n从1开始 |

练习

|  |
| --- |
| 02\_ex  创建4\*4的表格，400px\*400px  通过结构伪类选择器设置如下样式  第一行背景为#0ff  最后一行背景为#ff0  第三行第二列背景为#f0f |

③:empty

|  |
| --- |
| 匹配内部没有任何元素标签，包括文本，空格，回车，都不能有 |

④:only-child

|  |
| --- |
| 匹配属于其父元素的唯一子元素 |

⑤否定伪类

|  |
| --- |
| :not(selector) |

4.伪元素选择器

①匹配元素的首行首字符

|  |
| --- |
| :first-letter 或者 ::first-letter |

②匹配元素的首行

|  |
| --- |
| :first-line 或者 ::first-line  当首行与首字符发生了冲突，应用首字符的样式 |

③匹配用户选择的文本

|  |
| --- |
| 这里必须是双：：  ::selection{}  这里的样式，只能修改文本颜色和背景颜色 |

④伪元素选择器，内容生成

|  |
| --- |
| 生成的是一个元素  可以设置这个元素的显示方式，使用content:设置这个元素文本内容  以及所有可以使用的样式  content中只能给字符串和url(图片)  :before 或者 ::before 代表的是内容区域中，最靠前的部分  div::before{  content:"蒙奇.D.路飞说:";  display:block;  color:#f00;  background:#0ff;  width:100px;height:100px;  }  :after 或者 ::after 匹配到某元素内容区域最后的位置 |
| 伪元素添加内容，可以解决的问题  1.外边距溢出    2.解决高度坍塌 |

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

1.什么是弹性布局

|  |
| --- |
| 弹性布局，是一种布局方式。  主要解决的是某个元素中子元素的布局方式  让页面布局更加的灵活 |

2.弹性布局专有的名词解释

|  |
| --- |
| 1.容器  要发生弹性布局的子元素，他们的父元素，成为容器  就是设置display:flex的那个元素  2.项目  要发生弹性布局的子元素们，叫项目  就是设置了display:flex的元素的子元素们  3.主轴  项目们排列方向的一根轴，称之为主轴  如果项目们是按照水平排列，那么主轴就是x轴  如果项目们按照垂直排列，那么主轴就是y轴  项目们在主轴排列的顺序，称为主轴起点和主轴的终点  4.交叉轴  与主轴垂直的一根轴，叫做交叉轴  项目们在交叉轴的排列顺序，称为交叉轴的起点和终点 |

3.语法

|  |
| --- |
| 将元素设置为弹性容器之后，他所有的子元素，都会变为弹性项目  display:flex; 将块级元素设置为容器  inline-flex 将行内元素设置为容器  注意：  元素设置为flex容器后，容器的text-align，vertical-align失效  项目float，clear失效 |

4.容器的属性

①设置主轴的方向

|  |
| --- |
| flex-direction  取值 row 默认值，主轴是x轴，主轴起点在左端  row-reverse 主轴是x轴，主轴起点在右端  column 主轴是y轴，主轴起点在顶部  column-reverse 主轴是y轴，主轴起点在底部 |

②设置项目的换行

|  |
| --- |
| flex-wrap:  取值: 默认值 nowrap 空间不够时，不换行，项目会自动缩小  wrap 空间不够时，项目不缩小，换行  wrap-reverse 项目换行，并反转 |

③上面两个属性的简写

|  |
| --- |
| flex-flow:direction wrap; |

④定义项目在主轴上的对齐方式

|  |
| --- |
| justify-content  取值 1.flex-start 默认值，主轴起点对齐  2.flex-end 主轴终点  3.center 主轴中心  4.space-around 每个外边距相同，两端有空白  5.space-between 两端对齐，两端无空白 |

5.项目们在交叉轴上的对齐方式

|  |
| --- |
| align-items:  取值  1.flex-start 默认值，交叉轴起点对齐  2.flex-end 交叉轴终点对齐  3.center 交叉轴中间对齐  4.baseline 基线，同flex-start类似  5.stretch 项目不写高，充满容器整个高度 |

5.项目属性

|  |
| --- |
| 设置在项目中的属性，只会影响当前项目  不影响其他项目 |

①项目顺序

|  |
| --- |
| order 定义项目排列的顺序，值越小。越靠近起点。默认值为0  取值：无单位数字 |

②flex-grow

|  |
| --- |
| 如果容器有足够大的剩余空间，项目将按比例放大(比例不准确)  取值，无单位数字 默认值 0 |

③flex-shrink

|  |
| --- |
| 如果容器空间不够，项目将按比例缩小(比例不准确)  取值，无单位数字 默认值 1 取值越大，缩小的越快 |

④align-self

|  |
| --- |
| 定义某一个项目在交叉轴上的对齐方式，不影响其他项目  取值  1.flex-start交叉轴起点对齐  2.flex-end 交叉轴终点对齐  3.center 交叉轴中间对齐  4.baseline 基线，同flex-start类似  5.stretch 项目不写高，充满容器整个高度  6.auto 使用align-items定义的值 |

三.css hack

|  |
| --- |
| 由于不同的浏览器对css的解析认知是不同的，会导致同一份css在不同的浏览器生成的页面效果不同  我们要针对不同的浏览器写不同的css代码  这种写不同css代码的过程，就叫css hack  tmooc有相关的视频  -webkit-  -ms-  -moz-  -o- |

四.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  2D转换，只在x轴和y轴上发生转换  3D转换，增加了z轴的转换效果 |

2.转换属性

|  |
| --- |
| transform:  取值：1.none 默认值，无任何转换效果  2.转换函数 transform-function  转换函数 transform-function:位移，旋转，放大，倾斜，3D旋转 |

3.转换原点

|  |
| --- |
| transform-origin:  转换原点会影响元素旋转时的效果  取值：1.以px为单位的数字  2.%  3.关键字 left/center/right top/center/bottom |

4.2D转换

①位移

|  |
| --- |
| transform:translate();  取值 1.translate(x) 和 translateX(x)  x轴位移 + 往右 - 往左  2.translateY(y)  y轴位移 + 往下 - 往上  3.translate(x,y) 同时设置x轴和y轴的位移 |

②缩放

③旋转

④倾斜

作业：

1.要求webstrom可以正常使用

设置问题，参考百度

注册问题，见视频

2.弹性布局，完成学子商城1L

