day13

CSS核心

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  兄弟选择器,只能往后找,不能往前找  ①相邻兄弟选择器  选择器1+选择器2{}  获取紧紧挨在选择器1元素后面的选择器2的元素  ②通用兄弟选择器  选择器1~选择器2{}  获取选择器1后面所有符合选择器2的弟弟元素  多用于一组元素,设置除了老大以外,其它元素的公共样式 |

2.属性选择器

|  |
| --- |
| 1.[attr]{} attr表示属性名称  匹配页面中所有带有attr属性的元素[class] {color:#f00;}  2.[attr1][attr2]....{}  匹配页面中同时带有attr1,attr2,.....属性的元素[class] [title] { }  3.elem[attr1][attr2]..{}  匹配页面中同时带有attr1,attr2,.....属性的elem元素  span[title]{color:#f00;}  4.elem[attr1=v1][attr2=v2]....{}  匹配页面中同时带有attr1并且值为v1,attr2并且值为v2,.....属性的elem元素  p[title="千里眼顺风耳"]{color:#f00;}  5.属性值的模糊查询  [attr^=value]{} 匹配attr的值以value开头的元素  [attr$=value]{} 匹配attr的值以value结尾的元素  [attr\*=value]{} 匹配attr的值中有value的元素  [attr~=value]{} 匹配attr的值中有value这个独立的单词的元素(value前后有空格) |

3.伪类选择器

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

①目标伪类

|  |
| --- |
| 让被激活的锚点,应用的样式  :target{} |

②结构伪类

|  |
| --- |
| selector:first-child{}  找的是selector的父元素的第一个儿子  而且第一个儿子还得符合selector  selector:last-child{}  找的是selector的父元素的最后一个儿子  而且第一个儿子还得符合selector  selector:nth-child(n){} n从1开始 |

练习:创建4\*4表格 400px\*400px

通过结构伪类,完成下列样式

第一行背景色#0ff;

最后一行背景色#ff0

第三行第二列背景色#f0f

③:empty

|  |
| --- |
| 匹配内部没有任何元素的标签  内部不许有空格,文字,元素,回车 |

④:only-child

|  |
| --- |
| 匹配属于其父元素唯一子元素  唯一:仅限于元素,可以添加文本,空格,回车 |

⑤否定伪类

|  |
| --- |
| :not(selector) 符合selector的都不要 |

4.伪元素选择器

|  |
| --- |
| :first-letter 或者 : :first-letter  匹配第一个字符  :first-line 或者 ::first-line  匹配第一行文本(如果与首字母冲突,听首字母的样式)  ::selection  匹配用户鼠标选中的文本  必须是双::,首字母无效,只能设置背景色和字体颜色 |

5.伪元素选择器----内容生成

|  |
| --- |
| 使用css代码,动态的添加html元素  ::before 或者 :before  是在当前元素的内容区域的最前方(还是在内容区中),添加一个假的元素  这个元素默认是行内元素  使用content添加内容 (content只能设置文本和图片)  可以设置其他任意样式  可以设置display改变元素的显示方式  ::after或者 : after  是在当前元素的内容区域的最后面(还是在内容区中),添加一个假的元素  伪元素内容生成,可以解决的问题  1.外边距溢出  #parent:before{  content:"";  display:table;  }  2.解决高度坍塌  #parent:after{  content:"";  display:block;  clear:both;  } |

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

1.什么是弹性布局

|  |
| --- |
| 就是一种布局方式,  主要解决某个元素的子元素的布局方式(横向排列,或者纵向排列)  为布局提供很大的灵活性 |

2.弹性布局的相关概念

|  |
| --- |
| 容器:  要发生弹性布局的子元素们的父元素,称之为容器  也就是设置了display:flex的元素.  这个元素本身不是弹性布局,他的儿子们是弹性布局  项目:  要做弹性布局的子元素们,称之为项目  就是设置了display:flex的元素的子元素们  主轴:(4根)  项目们的排列的方向的一根轴,称之为主轴  如果项目们横向排列,x轴就是主轴  如果项目们纵向排列,y轴就是主轴  项目们在主轴上的排列顺序,就是主轴的起点和终点  交叉轴(2根)  与垂直相交的一根轴,叫做交叉轴  项目们在交叉轴上的对齐方式,就是交叉轴的起点和终点 |

3.语法

|  |
| --- |
| 在项目中,设置display:flex 将块级元素设置为弹性容器  inline-flex 将行内元素设置弹性容器  特点:容器的text-align vertical-align 失效  项目的float clear失效 |

4.容器的属性

①主轴方向

|  |
| --- |
| flex-direction:  row 默认值,主轴是x轴,主轴起点在左边  row-reverse 主轴是x轴,主轴起点在右边  column 主轴是y轴,主轴起点在顶端  column-reverse 主轴是y轴,主轴起点在底部 |

②设置项目是否换行

|  |
| --- |
| flex-wrap:  nowrap 默认值,不换行  wrap 换行  wrap-reverse 换行,并反转 |

③主轴方向和换行的缩写

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

④项目在主轴上的对齐方式

|  |
| --- |
| justify-content:  flex-start 默认值,主轴起点对齐  flex-end 主轴终点对齐  center 主轴中间对齐  space-between 两端对齐,两端无空白  space-around 每个间距大小相同,两端有空白 |

⑤项目在交叉轴的对齐方式

|  |
| --- |
| align-items:  flex-start 默认值,在交叉轴起点对齐  flex-end 在交叉轴终点  center 在交叉轴中间对齐  stretch 前提 项目不定义高度,设置stretch,项目充满整个交叉轴 |

练习,使用弹性完成品质保障

5.项目的属性,只能设置在某一个项目上,不影响其他项目的效果

①项目排列顺序

|  |
| --- |
| order:无单位整数;  定义项目排列顺序,值越小,越靠近主轴起点,默认值为0 |

②flex-grow

|  |
| --- |
| 定义项目的放大比例  如果容器有足够大的剩余空间,项目将变大  默认值 0 不放大,取值越大,项目放大的越快 |

③flex-shrink

|  |
| --- |
| 如果容器空间不足,设置项目该如何缩小  默认值 1. 设置为0不缩小,取值越大,项目缩小越快 |

④align-self

|  |
| --- |
| 单独设置每一个项目交叉轴的对齐  flex-start 默认值,在交叉轴起点对齐  flex-end 在交叉轴终点  center 在交叉轴中间对齐  stretch 前提 项目不定义高度,设置stretch,项目充满整个交叉轴  auto 使用给容器定义的align-items的值 |

三.CSS hack

|  |
| --- |
| 由于不同浏览器,对css的解析认知不同,会导致同一份代码,在不同浏览器下,页面显示效果不同  1.开发人员要针对不同的浏览器写不同的样式.这个行为就叫写css hack  -o-  -webkit-  -ms-  -moz-  2.css reset |

作业

