一.复杂选择器

3.伪类选择器

⑥伪元素选择器

|  |
| --- |
| :first-letter或者::first-letter  给首字符定义样式  :first-line或者::first-line  给首行定义样式  ::selection 必须是两个冒号  匹配用户选取的内容的样式  注意，只能设置文本颜色和背景颜色 |

4.内容生成

|  |
| --- |
| 使用css命令，添加html元素。  :before 或者::before  在元素的内容区域的最前面添加一个伪元素  默认类似于span，添加的内容使用content  可以设置各种样式  注意：content只能添加文本或者url  :after 或者 ::after  在元素的内容区域的最后面添加一个伪元素 |
| 内容生成，可以解决的问题  1.外边距溢出  给父元素添加内容生成  #parent:before{  content:"";  display:table;  }  2.使用内容生产解决高度坍塌！  #parent:after{  content:"";  display:block;  clear:both;  } |

二.弹性布局

|  |
| --- |
| 是一种布局方式  主要解决某个元素中所有子元素的布局方式  为布局提供了很大的灵活性 |

1.弹性布局的专有名词

|  |
| --- |
| 容器：设置了display:flex的元素，这是个父元素  要发生弹性布局的元素们，他们的父元素  项目：要发生弹性布局的元素们  就是容器的子元素  主轴：项目们排列的方向，叫做主轴  如果项目们横向排列，主轴就是x轴  如果项目们纵向排列，主轴就是y轴  项目们在主轴上排列顺序，称为主轴的起点和终点  交叉轴:永远与主轴方向垂直的一条轴，叫做交叉轴  项目们在交叉轴上的排列顺序，称为交叉轴的起点和终点 |

2.语法

|  |
| --- |
| 将元素设置为弹性容器之后，容器的所有子元素都变成弹性项目，  这些项目都允许按照弹性布局的方式排列  属性：display:  取值：1.flex 将块级元素设置为弹性容器  2.inline-flex 将行内元素设置为弹性容器  注意：元素设置为弹性容器后，text-align vertical-align失效  项目的float、clear失效 |

3.容器的属性

①主轴的方向

|  |
| --- |
| flex-direction:  取值 1.row 默认值， 主轴为x，主轴起点左侧  2.row-reverse 主轴为x,主轴起点右侧  3.column 主轴为y，主轴起点上部  4.column-reverse 主轴为y，主轴起点为下部11：01~11：16休息 |

②设置项目换行

|  |
| --- |
| flex-wrap:  取值 nowrap 默认值  wrap  wrap-reverse |

③主轴方向和换行的缩写

|  |
| --- |
| 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 主轴终点对齐  3.center 主轴中心对齐  4.space-between 两端对齐，两端无空白距离  5.space-around 每个间距大小相等 |

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.stretch 项目如果不定义高度，项目充满整个交叉轴 |

4.项目的属性，设置在某一个项目上，不影响其他项目

①项目排列的顺序

|  |
| --- |
| order：无单位整数  值越小，越靠近主轴起点，默认值0 |

②flex-grow

|  |
| --- |
| 如果主轴剩余空间够大，项目将按比例放大(比例不准确)  flex-grow：整数 默认值 0 |

③flex-shrink

|  |
| --- |
| 主轴空间不够时，项目将按比例缩小(比例不准确)  flex-shrink: 整数，取值越大，缩的越小，默认值为1 |

超纲的知识点

|  |
| --- |
| flex:grow shrink basis; |

④某一个项目在交叉轴上的对齐方式

|  |
| --- |
| align-self  取值：1.flex-start  2.flex-end  3.center  4.stretch  5.auto 使用容器设置的align-items的值 |

三.转换

|  |
| --- |
| 改变元素在页面中的位置，大小，角度，以及形状  2D转换，只有x轴和y轴的转换  3D转换，添加了z轴，模拟3d |

1.转换的属性

|  |
| --- |
| transform：  取值：一个或者多个转换函数，多个转换函数之间用空格分开  转换原点，对旋转产生影响  transform-origin:  取值：2个值，把原点放在x轴和y轴上  3个值，添加了z轴  以px为单位的数字  %  关键字 left/center/right top/center/bottom  总结：学习转换，就是学习不同的转换函数 |

2.2D转换

①位移，改变元素的位置

|  |
| --- |
| transform:translate()  取值：translate(x)等同于translateX(x)  指定元素在x轴上的位移  + → - ←  translateY(y)  指定元素在y轴上的位移  + ↓ - ↑  translate(x,y) 同时设置x，y轴的位移 |

②旋转，改变元素的角度

|  |
| --- |
| transform:rotate(ndeg)  n:+ 顺时针， - 逆时针  注意：  1.旋转原点会影响旋转效果  2.旋转连同坐标轴一起旋转，会影响旋转之后的位移方向 |

③缩放，改变元素的大小

|  |
| --- |
| transform:scale(val)  取值 1.取一个值  val>1 x轴和y轴同时放大  0<val<1 x和y轴同时缩小  -1<val<0 x和y轴同时缩小，翻转  val<-1 x轴和y轴同时放大,翻转  2.取两个值，分别设置x轴和y轴的缩放  3.scaleX(val）  4.scaleY(val) |

④倾斜

|  |
| --- |
| transform:skewX(ndeg)等同于skew(ndeg)    让元素向着x轴发生倾斜，实际上改变的是y轴的角度  n:+ y轴逆时针， n: - y轴顺时针  skewY(ndeg)    让元素向着y轴发生倾斜，实际上改变的是x轴的角度  n:+ x顺时针 - x逆时针  skew(xdeg,ydeg) |