总结一下

前面先写一些基本，主要的是后面的flex布局

Display的属性值有 inline, block,inline-block,none,flex,grid

下面详细介绍

Display就是告诉浏览器应该如何计算和显示这个元素。

1. Block，块级元素，独占一行，元素后面自带换行符。支持宽高。不设置宽度的时候，和父级一样大。
2. Inline，行内元素，不支持宽高，内容撑开宽高，不独占一行，会多个元素排在一行，直到放不下，才会自动换行
3. Inline-block 融合了inline和block的特征，不设置宽高的时候，内容撑开宽高，不会独占一行，后面没有换行符，简单的说就是，既支持宽高，又不会独占一行。
4. None,为对象设置该属性之后，对象就消失了，不同于visibility的hidden属性，none是不会给对象留物理空间的。也就是说该对象不占位置。
5. Table, 将该元素设置为块级的表格来显示，类似table标签，前后带有换行符。
6. Grid 网格布局，后面慢慢填，先写flex布局；
7. Flex，css3新出现的一种布局方式，下面详解。

学习一个新东西，先要了解一下他能解决什么问题，

Flex容器内的项目可以自动的填满容器的空间，主要是来适应各种类型的显示设备和屏幕大小。应用方面flex布局比较适合一些小组件或者小规模的布局，在大小调整，伸展收缩方面有优势。

Flex布局，也叫弹性布局。很灵活，基本兼容所有浏览器。

每个容器都可以用，行内元素也可以，Webkit内核的浏览器，必须加上-webkit前缀。

下面说语法，需要注意的一点是，使用flex布局之后，子元素的float,clear, vertical-align等属性将失效。

使用很简单，直接给display的属性值设置为flex就可以了。

有flex属性的元素称为容器，容器的子元素自动的成为容器的成员。

一般，容器默认的有两条轴，水平轴和垂直轴、水平轴为主轴。

Flex有六个属性，后面的布局要靠这六个属性。

增加一点，

Flex布局的计算顺序是，

1. 先计算每个子元素（项目）的大小，
2. 确定之后计算剩下多少空白，
3. 将这些空白分配给子元素
4. Flex-direction 决定元素排列的方向，是从上到下，还是从下到上，还是从左到右，从右到左。其实也是设置主轴方向。

有四个值可选

Row:默认值，主轴为水平方向，元素从左到右排列

Row-reverse:主轴为水平方向，元素从右到左排列

Column:主轴为垂直方向，从上往下

Column:主轴为纯挚方向，从下往上。

（reverse:相反，反转，逆转）借用阮一峰大神的图



1. Flex-wrap 决定子元素如何换行，（warp:包 包裹）

有三个值可选

Nowrap.默认值，不换行，元素再多也不换行，一直挤到元素变形也不换行



Wrap:换行，一行挤不下之后自动换到第二行，主轴方向不变。



Warp-reverse:换行，一行挤不下之后第二行自动到第一行上面，主轴方向不变



1. Flex-flow;这个属性是flex-direction和flex-wrap联合在一起的简写

默认值是 row ||nowrap

注意，这里只接受一个值，可以写flex-direction的属性也可以写flex-wrap的属性。



1. Justify-content 定义了子元素在主轴上的对其方式 (justify:证明，辩解)

有5个值

Flex-start,左对齐 （起点对齐）

Flex-end 右对齐（终点对齐）

Center居中

Space-between:两边对齐，元素之间的间隔都相等

Space-around.每个元素两侧的的间隔相等，



这里重点区分一下align-items和align-content,

主要是align-items可以适用于多行和单行的

Align-content只适用于多行的，单行是无效的

1. Align-items:规定项目在交叉轴上如何对齐，这里我理解为主轴之外的另一条轴，主轴是水平轴的话，交叉轴为竖直轴，这里要注意，如果容器不设置宽高的话，这个属性是不生效的

可以选5个值

Flex-start:交叉轴的起点对齐

Flex-end:交叉轴的终点对齐

Center交叉轴的中点对齐

Baseline 以元素内的第一行文字的基线对齐

Stretch 默认值，如果元素没有设置高度，或者设置为auto，元素会默认铺满容器的整个高度

1. Align-content定义多根轴线的对齐方式。如果项目只有一根轴线，该属性不起作用

flex-start：与交叉轴的起点对齐。

flex-end：与交叉轴的终点对齐。

center：与交叉轴的中点对齐。

space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。

space-around：每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。

stretch（默认值）：轴线占满整个交叉轴。



下面写一下容器内的子元素的一些属性

order

flex-grow

flex-shrink

flex-basis

flex

align-self

1. order，定义了子元素的排列顺序，这个属性接收一个数值，数值越小，排列的位置越靠前。默认值是0



1. flex-grow定义了元素是否放大，接收一个数值，数值越大放大的越多

默认值为0.，就是说就算有空间也不放大，如果一个是1，一个是2，那么2所占的位置比1大一半。

另一种理解是主轴方向上空白区域，子元素分配的比例。

就是所有子元素的宽度加起来都比主轴的宽度小的话，肯定会有一部分空白部分，flex-grow就是决定空白部分的大小是按照什么比例分配给这些子元素的。

如果每个元素的flex-grow都相同，那么空白区域的大小元素们平均分配。

如果说 有一个是2，剩下两个是1，那么空白部分的分配也是2：1：1，



1. flex-shrink:定义了元素是否缩小，默认值是1，如果空间不足，将缩小，0表示不缩小。

如果所有元素的flex-shrink值相同，空间不足时，所有元素等比例缩小。

如果一个项目的flex-shrink属性为0，其他项目都为1，则空间不足时，前者不缩小。

另一层理解，如果主轴方向的子元素的宽度的和超过了主轴的元素，这些子元素肯定要缩小，这个属性就是决定子元素按照什么比例来缩小的。

1. flex-basis设置该项目占据多大空间，浏览器根据这个属性，计算主轴是否有多余的空间。默认值auto，即原来的大小，它可以用其他的单位，如px

如果我们设定了值，那么他就会覆盖掉该该元素的宽度。

1. flex:是前三个属性，flex-grow flex-shrink flex-basis的集合，注意，这个属性里面的参数不用逗号隔开，要用空格隔开 默认值是 0 1 auto；

强烈建议用这个属性， 因为他会很好的帮我们分配子元素的比例。

我们来详细的分析一下这个属性，

如果三个子元素设置为flex：1，flex：2，flex：3的话，那么这三个元素就会按照

1：2：3的比例来分配大小，但是当我们想用上面的三个属性来这样分配的时候就发现很困难。

如果一个属性一个属性自己慢慢计算的话，它是按照分配空白空间的方法来计算的，

举个例子，就是 1：1：2的比例的时候，元素本身是20px，假如空白空间还有40像素，那么空白空间按照这个比例分配完之后情况就变成了，30px，30px，40px，很显然不是我们想象的1：1：2

1. align-self 这个属性可以让单个元素有和其他元素不一样的对其方式，覆盖align-items的值，它决定的是主轴交叉轴方向的排列方式

默认值是auto,其他值和align-auto的值相同

align-self: auto | flex-start | flex-end | center | baseline | stretch;

1. 123arg