

# day14

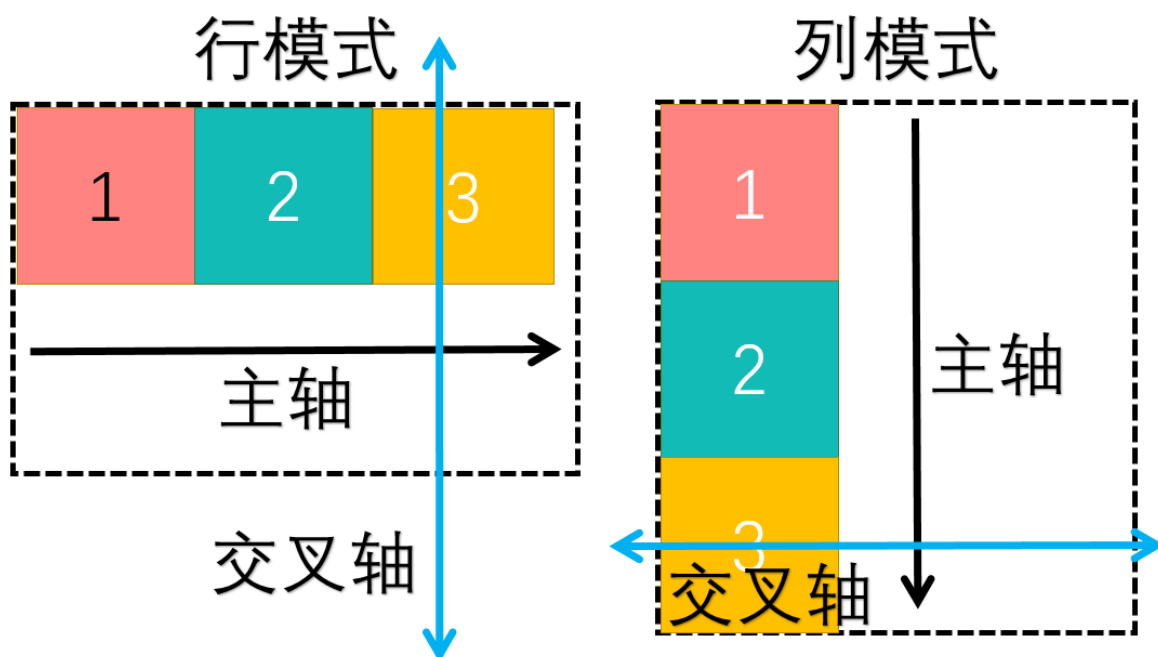
## flex布局

### 3.主轴的排列方向【9:10】

#### (1) 行模式和列模式

行模式就是子元素（项目）在容器中以行（x轴）方向排列的模式

列模式就是子元素（项目）在容器中以列（y轴）方向排列的模式



#### (2) 主轴和交叉轴

主轴和交叉轴见上图

当使用flex的时候，首先想到的应该是两个轴，主轴和交叉轴

主轴由 `flex-direction` 定义，另一条垂直于它的轴就叫做交叉轴。

### 4.主轴排序

主轴方向可以取四个值，注意这四个值都是要在容器中写的

```
flex-direction: row; /*默认行模式：左（开始）-右（结束）*/
flex-direction: row-reverse; /*行模式：右（结束）-左（开始）*/
flex-direction: column; /*列模式：上（开始）-下（结束）*/
flex-direction: column-reverse; /*列模式：下（结束）-上（开始）*/
```

## 5.换行模式

`flex-wrap` 属性来控制flex子项（项目）单行显示换行还是不换行，默认情况不换行。即使容器无法承载所有项目，他们会按照等比例压缩，强制在主轴方向显示。

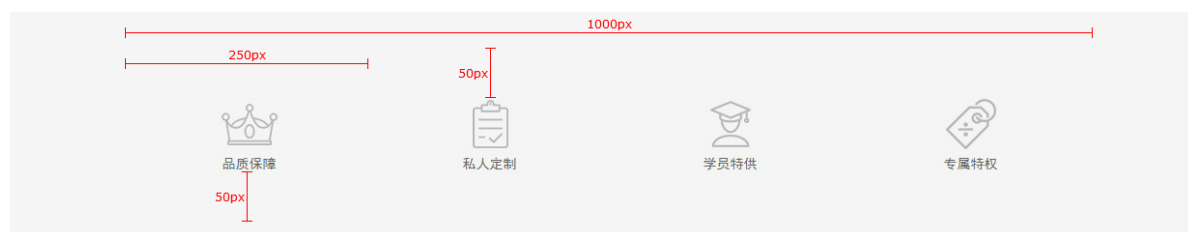
```
/* 默认不换行 */
flex-wrap: nowrap;
/* 换行 */
flex-wrap: wrap;
```

## 6.简写属性

`flex-flow` 是将主轴排序和是否换行两个属性组合写在一起。第一个值主轴排序，第二个值是是否换行。

```
flex-flow: row nowrap;
```

## 练习【10:40】



结构

```
<div class="center">
  <div class="item">
    <img src="" alt="">
    <p></p>
  </div>
</div>
```

所有代码

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style type="text/css">
      * {
        margin: 0;
        padding: 0;
      }
      .center {
        width: 1000px;
        margin: 0 auto;
        /* 作为flex容器 */
        display: flex;
      }
    </style>
  </head>
  <body>
    <div class="center">
      <div class="item">
        <img src="" alt="">
        <p></p>
      </div>
      <div class="item">
        <img src="" alt="">
        <p></p>
      </div>
      <div class="item">
        <img src="" alt="">
        <p></p>
      </div>
      <div class="item">
        <img src="" alt="">
        <p></p>
      </div>
    </div>
  </body>
</html>
```

```

        .item {
            width: 250px;
            padding: 50px 0;
            text-align: center;
            background-color: yellow;
        }
    </style>
    <title></title>
</head>
<body>
    <div class="center">
        <div class="item">
            
            <p>品质保障</p>
        </div>
        <div class="item">
            
            <p>品质保障</p>
        </div>
        <div class="item">
            
            <p>品质保障</p>
        </div>
        <div class="item">
            
            <p>品质保障</p>
        </div>
    </div>
</body>
</html>

```

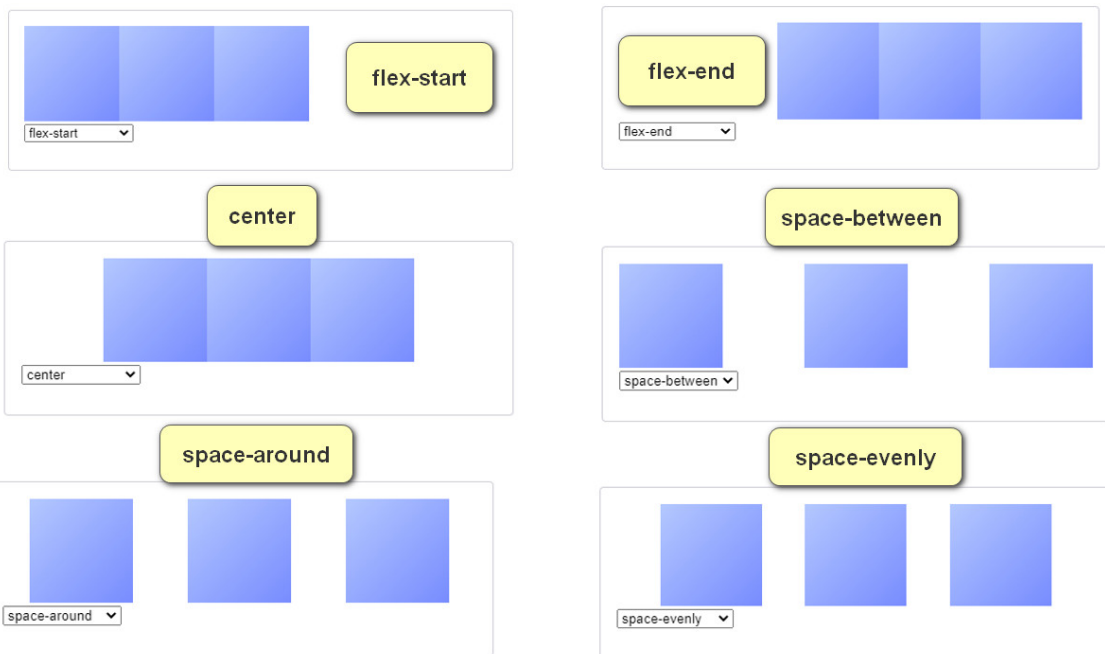
## 7.主轴上的对齐方式【11:19】

`justify-content` 定义了项目在主轴方向上的对齐方式，需要注意的是，要区别行模式和列模式，当然主要可以以行模式为例。

```

/* 默认起点对齐 */
justify-content: flex-start;
/* 终点对齐 */
justify-content: flex-end;
/* 居中对齐 */
justify-content: center;
/* 两端对齐 */
justify-content: space-between;
/* 周围分布相同空间 */
justify-content: space-around;
/* 均匀空间 */
justify-content: space-evenly;

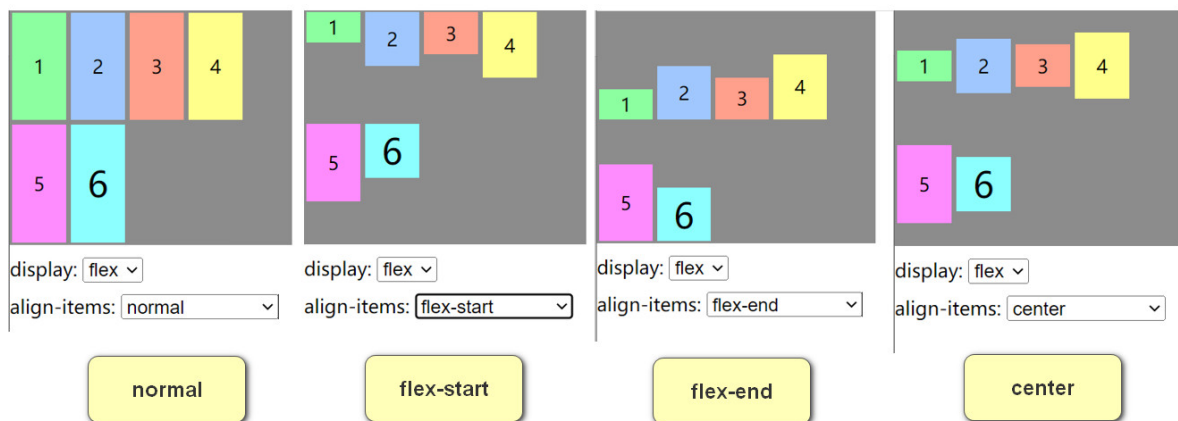
```



## 8.交叉轴对齐方式【11:54】

`align-items` 指的是交叉轴的对齐方式。如果没有指定交叉轴对齐方式默认是 `normal` 项目无高度，默认撑满容器高度。

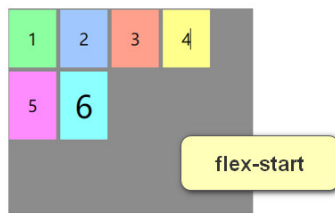
```
/* 交叉轴起点*/
align-items: flex-start;
/* 交叉轴终点 */
align-items: flex-end;
/* 交叉轴居中 */
align-items: center;
```



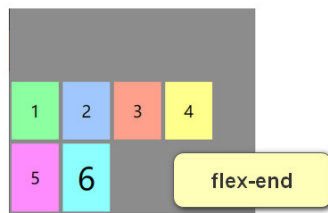
## 9.多轴线对齐【14:00】

`align-content` 属性设置浏览器如何沿着弹性布局的横纵项，多轴线分布的空间方式。

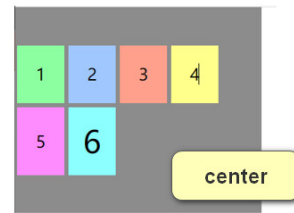
```
/* 多轴线对齐 */
/* 多轴线，交叉轴起点 */
align-content: flex-start;
/* 多轴线，交叉轴终点对齐 */
align-content: flex-end;
/* 多轴线，居中 */
align-content: center;
/* 多轴线，两端对齐 */
align-content: space-between;
/* 多轴线，行间距相等 */
align-content: space-around;
/* 多轴线，均分间距 */
align-content: space-evenly;
```



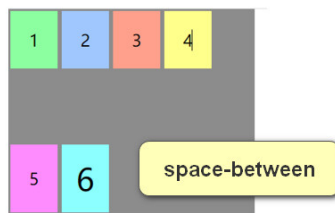
display: flex  
align-content: flex-start



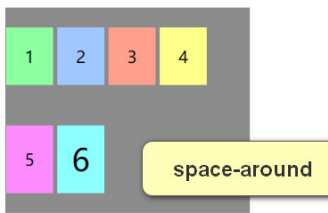
display: flex  
align-content: flex-end



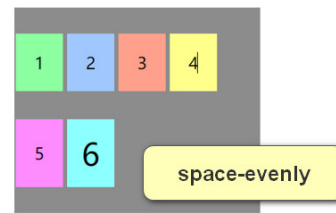
display: flex  
align-content: center



display: flex  
align-content: space-between



display: flex  
align-content: space-around



display: flex  
align-content: space-evenly

## 练习

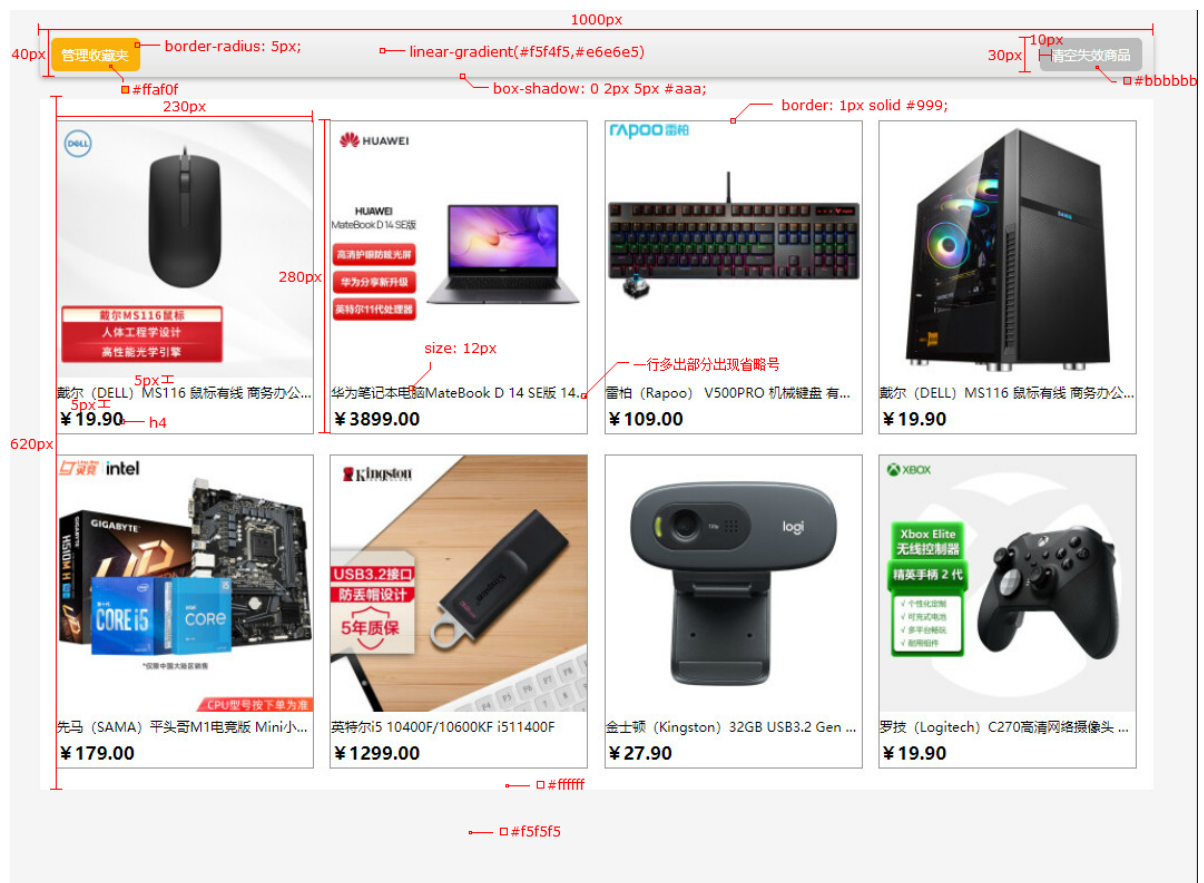
### 补充知识

`border-radius: 5px;` 圆角，平时尽量不写百分比的值

值越大，弧度越大

当弧度为高度的一半时，就变成了了胶囊型

如果需要圆形，首先元素要是正方形，再写弧度 `border-radius: 50%;`



## 10.项目属性【15:52】

### (1) 项目的排序顺序

`order` 项目在布局时的顺序，数值越小，排列越靠前。默认值为0，可以为负数，数值没有任何单位。

```
/* 排序规则 默认值0 */
order: -1;
```

### (2) 项目的增长规则

`flex-grow` 规定了项目在flex容器中分配剩余空间的比例。负值无效，默认值0，数值不需要单位。

- `flex-grow` 给某项目，前提容器中主轴方向是有剩余空间的
- 项目本身是有自己的尺寸的，不放大时使用的是自身的长度尺寸。如：宽度，设置了增长规则之后，原本尺寸失效了，使用增长规则的系数
- 剩余空间的比重分配，是所有项目的 `flex-grow` 系数相加，之后再按照比例分配

```
flex-grow: 1;
分配比例
.item:nth-child(2){
flex-grow: 1;/*1/7*/
}
.item:nth-child(3){
flex-grow: 6;/*6/7*/
}
```

### (3) 项目的收缩规则

`flex-shrink` 指定了flex项目的收缩规则，默认值1.不允许负值，没有任何单位。

- 收缩前提，容器中无剩余空间，项目才会被压缩
- 设置了收缩比例之后，原尺寸会在压缩时失效，而使用收缩系数
- 收缩数值越大，压缩比例就越大
- 压缩空间的分配比重，缩小的尺寸是所有项目缩小系数之和，再按比例分配
- 强制不缩小 `flex-shrink:0`

```
flex-shrink: 1; /*默认*/
flex-shrink: 2; /*缩小系数*/
flex-shrink: 0; /*坚决不缩小*/
```

### (4) 项目的初始化尺寸

`flex-basis` 指定了flex项目元素在主轴方向上的初始化大小（尺寸）。不允许负值，默认 `auto`。如果指定了该属性，则原本尺寸，将失效。

```
flex-basis: 200px;
```

### (5) 项目的flex简写💕

`flex` 属性指的是，增长规则，缩小规则，初始化尺寸的集合写法。

- 默认 `flex:0 1 auto`
- `flex:1` 指增长规则的系数 💕
- `flex: 0 0 200px` 指项目在容器中不放大，也不缩小，初始化尺寸200px

## 总结【17:30】

### 用于容器的属性

- `display: flex` 指定flex布局
- `flex-direction` 主轴的排序（主轴方向），默认值 `row`
- `flex-wrap: nowrap` 项目在容器中是否换行显示，常用值 `wrap` 换行
- `justify-content` 主轴的对齐方式
- `align-items` 交叉轴的对齐方式（一般用于单轴线）
- `align-content` 多轴线对齐方式

### 用于项目的属性

- `order: 0` 项目的排序
- `flex-grow:0` 放大规则，默认0，不放大
- `flex-shrink: 1` 收缩规则，默认1，缩小。不缩小用0
- `flex-basis: 200px` 指定主轴上的长度尺寸，不允许负值，默认auto，优先级比主轴上的尺寸高
- `flex` 集合写法，放大，缩小，初始尺寸，默认 `0 1 auto`。 `flex:1` 指放大， `flex:0 0` 不放大不缩小

# 作业

1. 学子商城的以下界面（尽量使用flex）

[http://101.96.128.94:9999/my\\_collect.html](http://101.96.128.94:9999/my_collect.html)

2. 账号上有个作业（也是flex的）

3. 没事儿看看浮动布局的视频

amber-web