

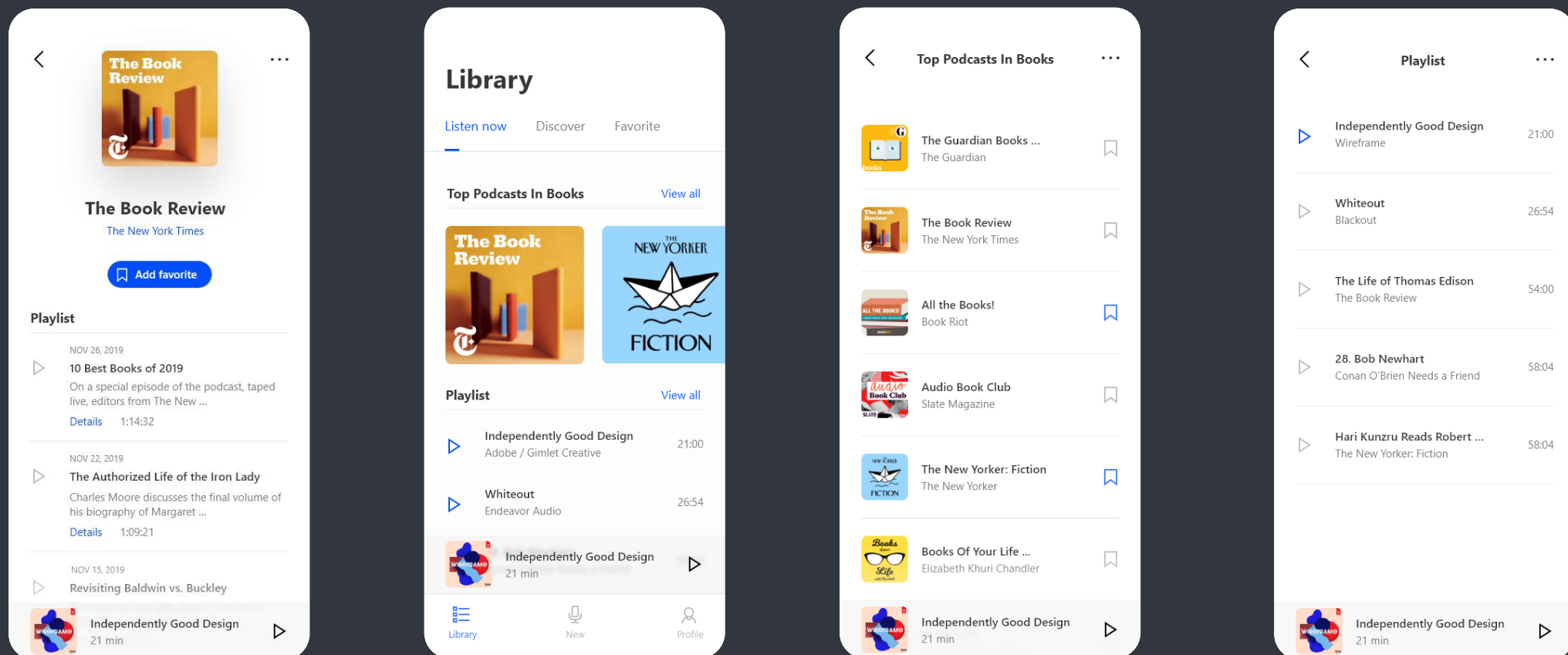
css极速入门 —— flex布局

主讲人与课程设计： 耕耕

css布局是什么

简单的说，布局就是你怎么规划html元素，通过css去控制元素的排列方式

一个典型的例子：



布局方式 —— 常用三种

1. 传统布局方式

利用position属性 + display属性 + float属性布局，兼容性最好，但是效率低，麻烦！

2. flex布局

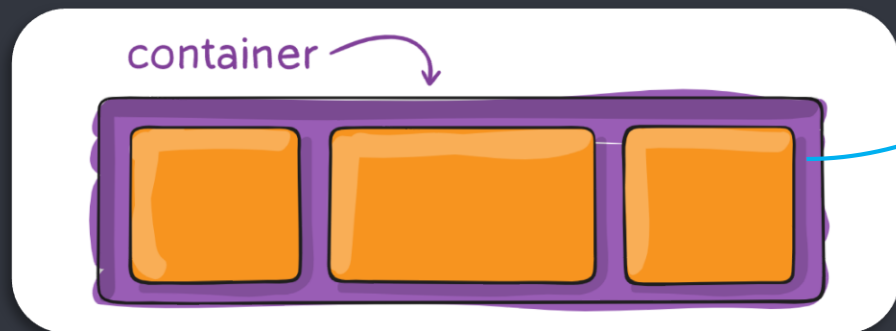
有自己的一套属性，效率高，学习成本低，兼容性强！

3. grid布局

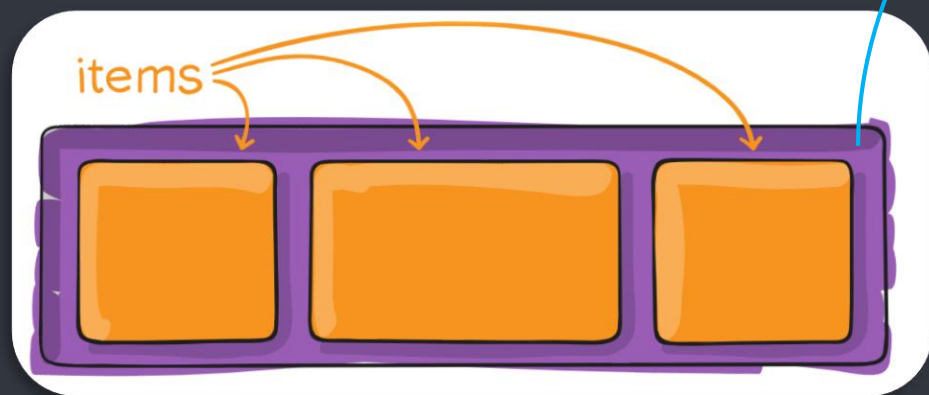
网格布局（Grid）是最强大的 CSS 布局方案。但是知识点较多，学习成本相对困难些，目前的兼容性不如flex好！

基本概念

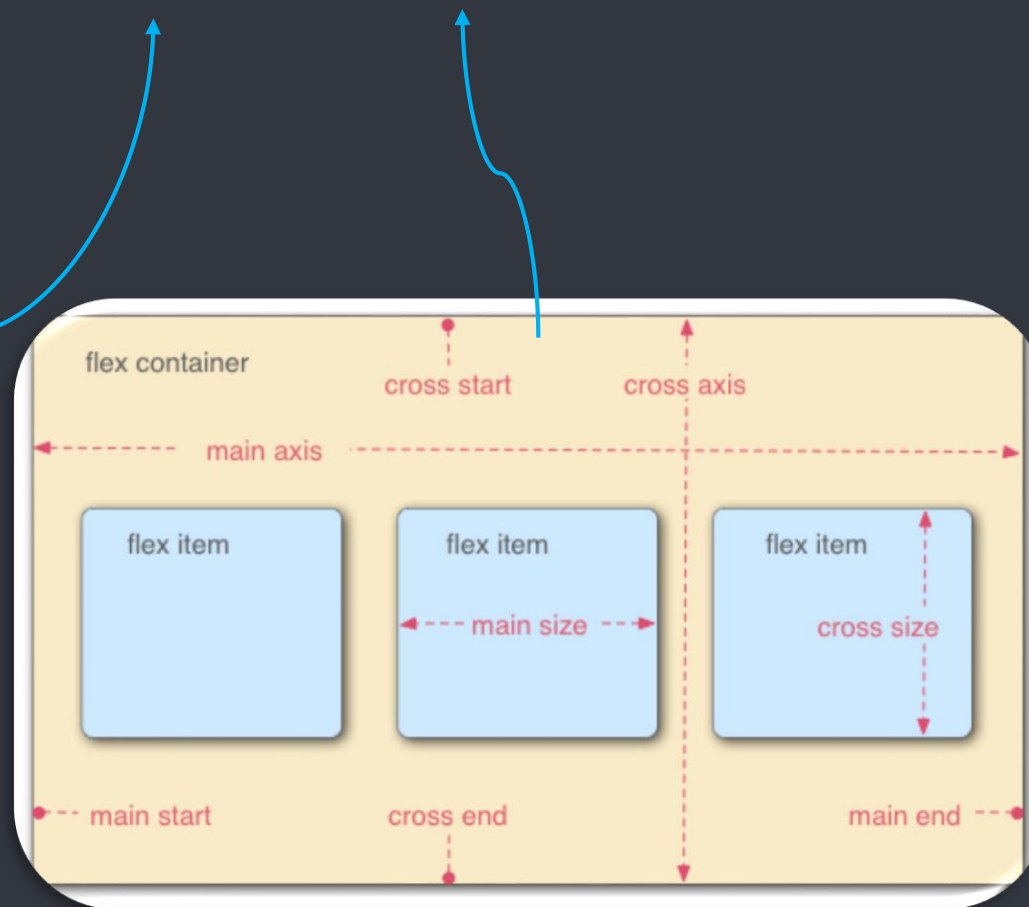
1. 容器 —— 有容器属性



2. 项目 —— 有项目属性



盒子模型当中的盒子 (除去margin)



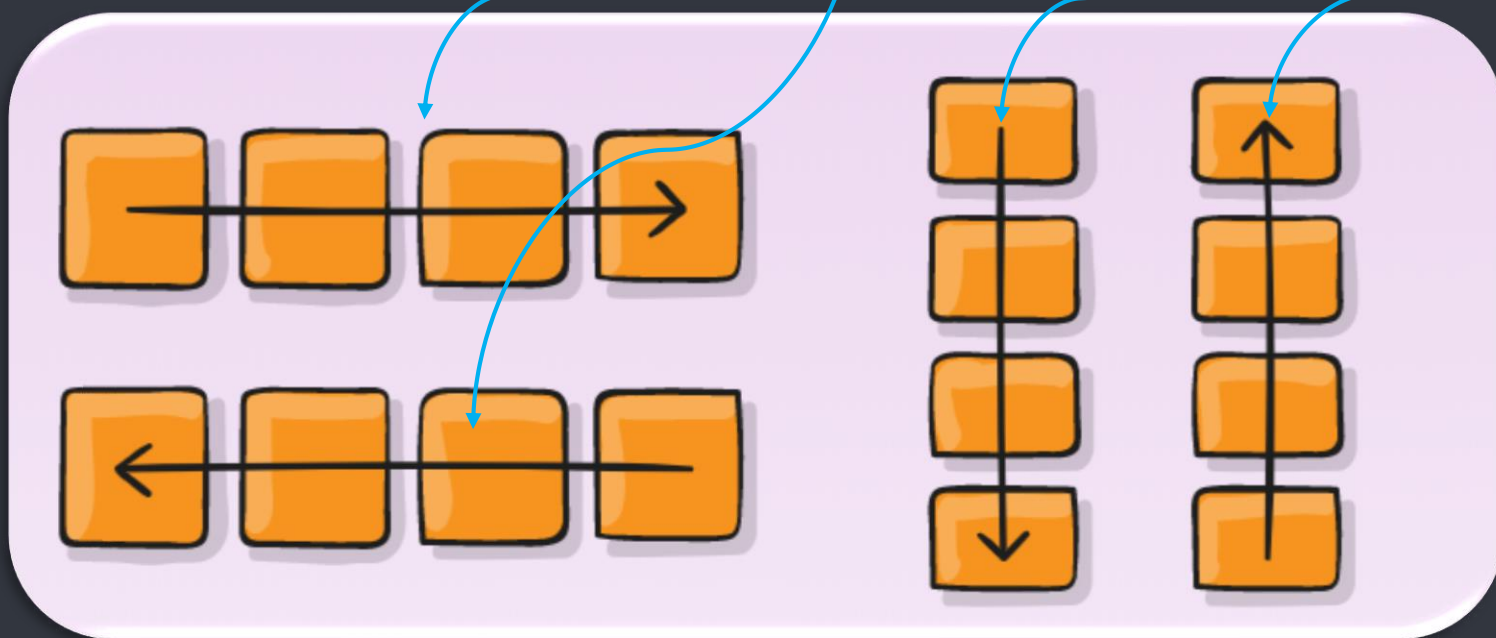
容器属性

1. flex-direction
2. flex-wrap
3. **flex-flow** (1和2的简写形式)
4. justify-content
5. align-items
6. align-content

容器属性

1. flex-direction —— 容器内子元素的排列方向

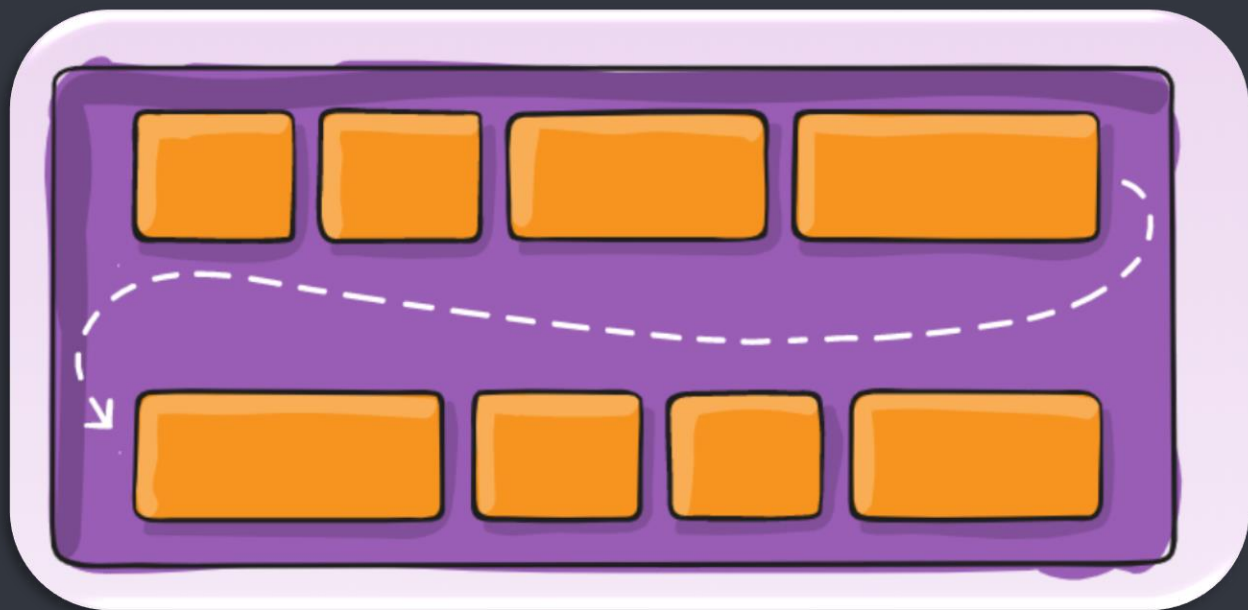
```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```



容器属性

2. flex-wrap —— 元素放不下了是否换行

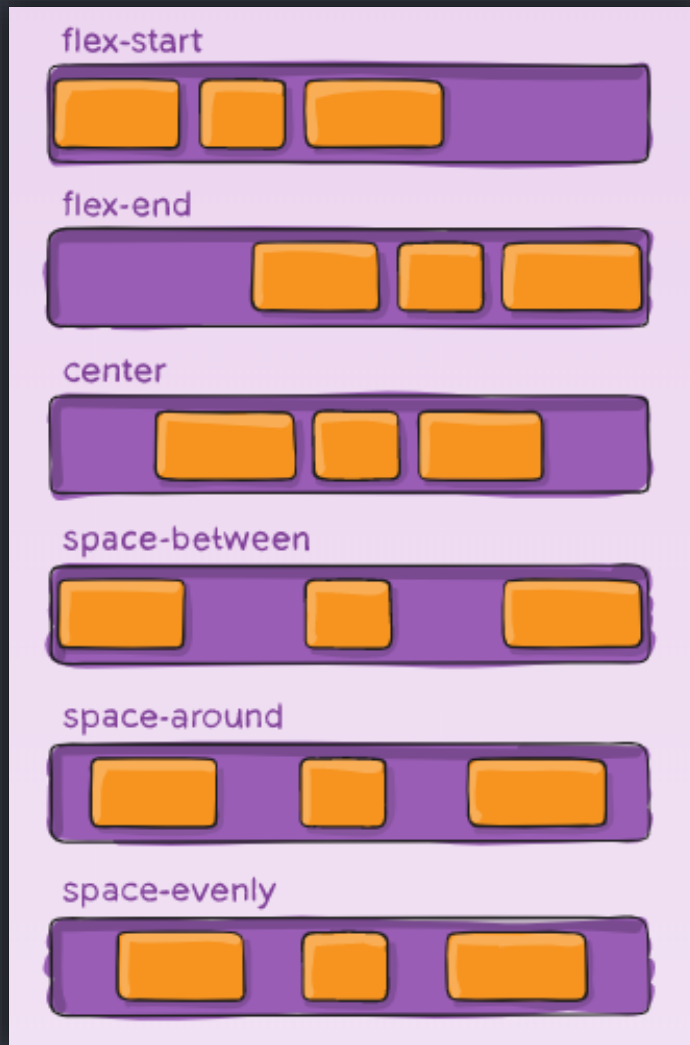
```
.container {  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```



容器属性

3. justify-content —— 属性定义了项目在**主轴上**的对齐方式

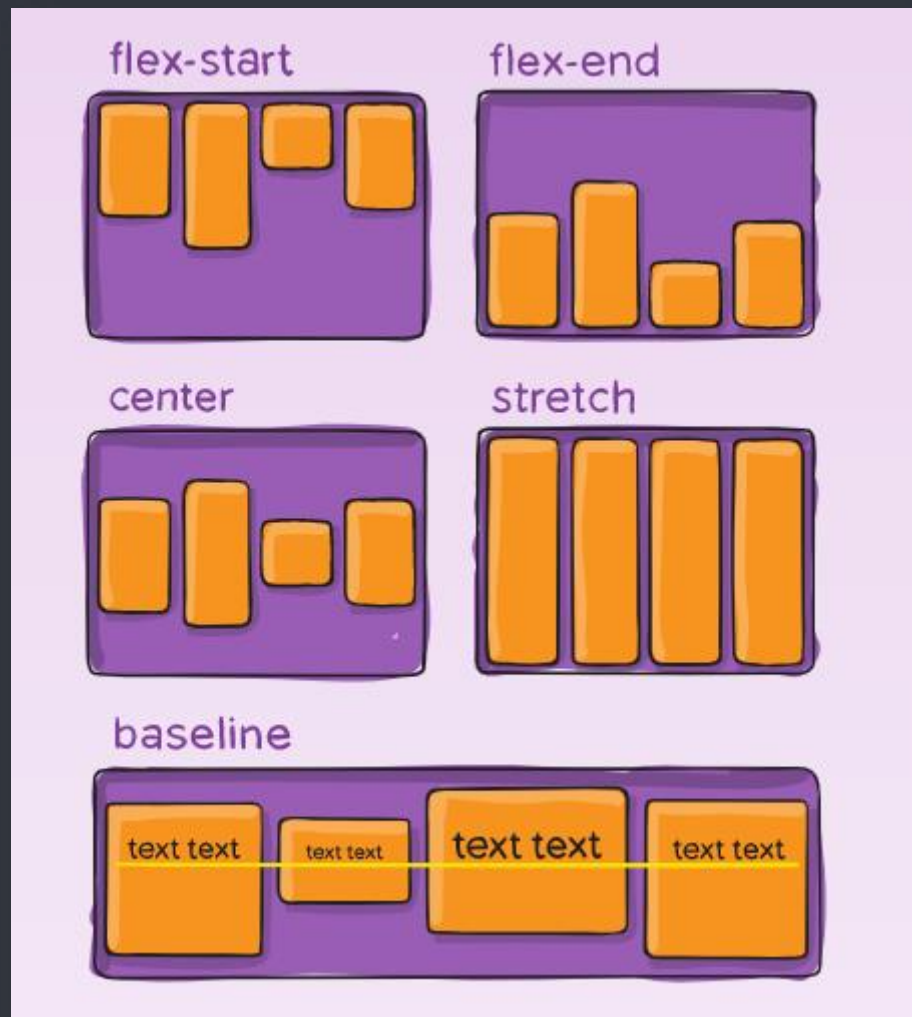
```
.container {  
  justify-content:  
    flex-start |  
    flex-end |  
    center |  
    space-between |  
    space-around |  
    space-evenly |  
    ...  
  ;  
}
```



容器属性

4. align-items —— 属性定义项目在**交叉轴**上如何对齐

```
.container {  
  align-items:  
    stretch |  
    flex-start |  
    flex-end |  
    center |  
    baseline |  
    ...;  
}
```



容器属性

5. align-content —— 属性定义了**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。

```
.container {  
  align-content:  
    flex-start |  
    flex-end |  
    center |  
    space-between |  
    space-around |  
    Stretch |  
    ...;  
}
```

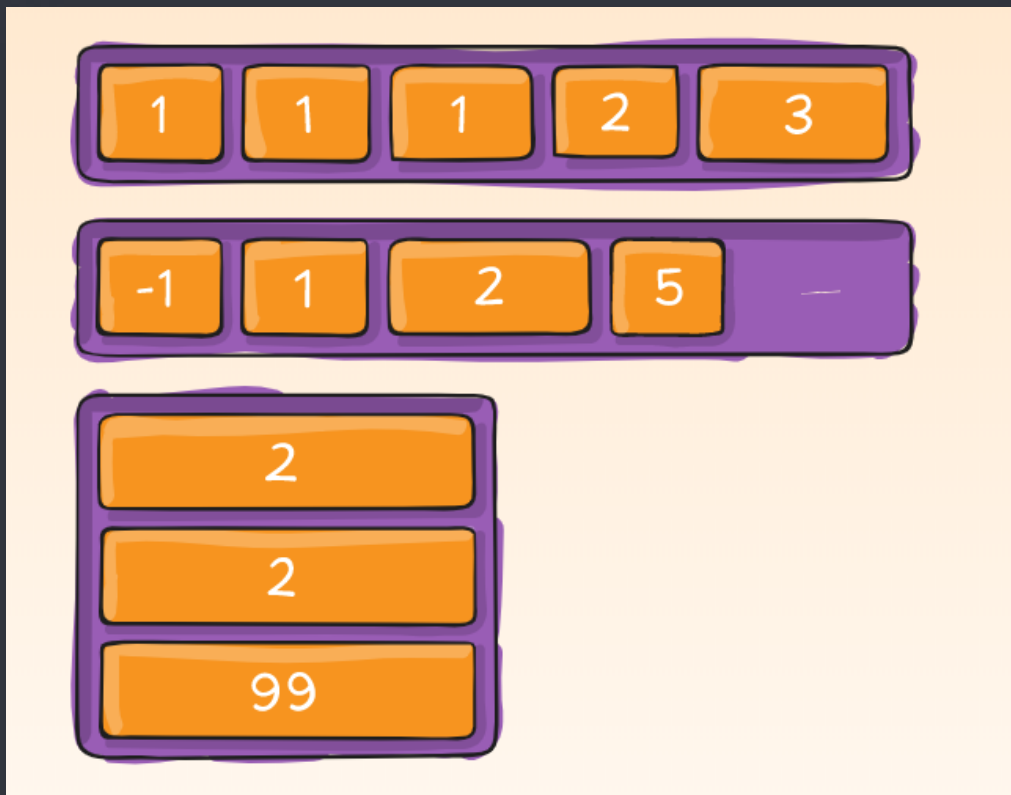


项目属性

1. order
2. flex-grow
3. flex-shrink
4. flex-basis
5. **flex (2, 3和4的简写形式)**
6. align-self

项目属性

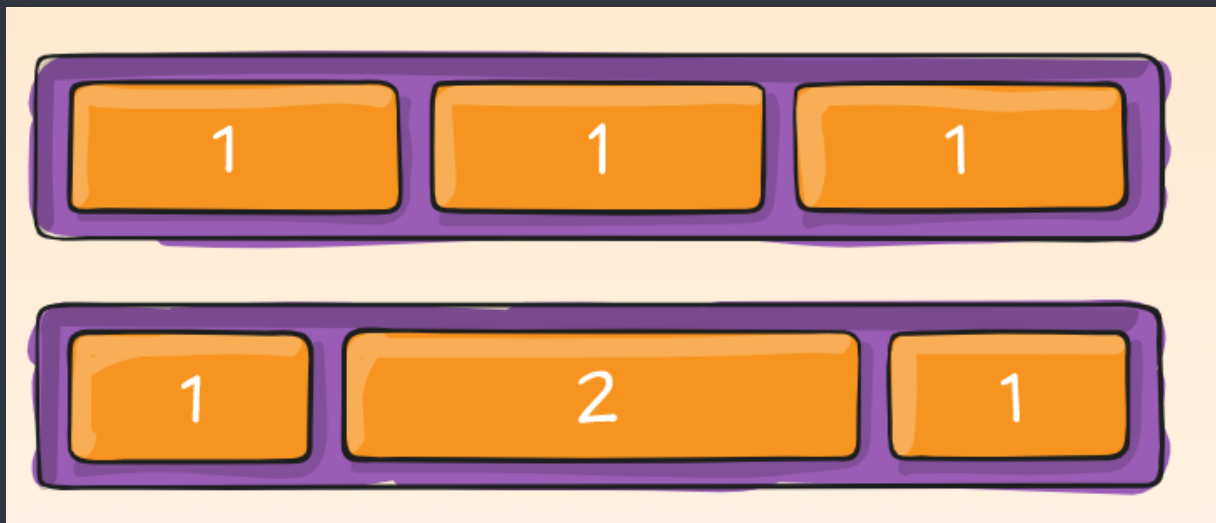
1. order —— 属性定义**项目的排列顺序**。数值**越小**，排列**越靠前**，默认为0



```
.item {  
    order: 100;  
}
```

项目属性

2. flex-grow —— 属性定义项目的放大比例(剩余空间按比例分配)。默认为0，即如果存在剩余空间，也不放大。



```
.item {  
    flex-grow : 1;  
}
```

项目属性

3. flex-shrink ——属性定义了**项目的缩小比例**，默认为1，即如果空间不足，该项目将缩小。



```
.item {  
    flex-shrink : 2;  
}
```

如果所有项目的flex-shrink属性都为1(默认)，当空间不足时，都将等比例缩小。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。**负值对该属性无效。**

项目属性

4. flex-basis ——属性定义了 在分配多余空间之前，项目占据的主轴空间（main size）

和width有一定的关系，大部分情况和width表现一致，如果两者同时使用flex-basis优先级会高一些

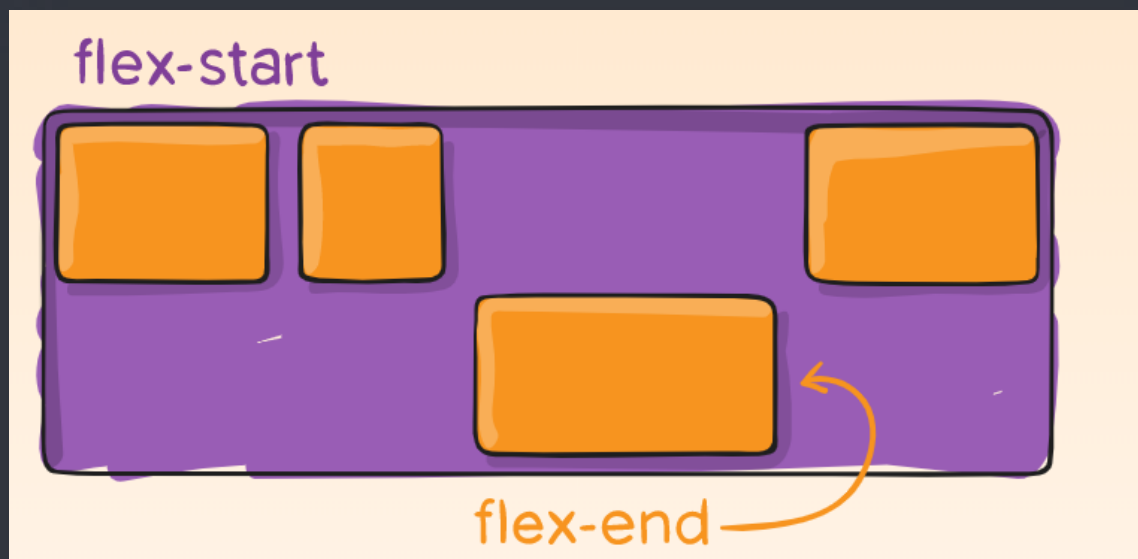
注意：

1. 单独使用width，正常使用
2. 如果单独使用flex-basis，超过会自动变大，不会限制，相当于flex-basis是最小宽度
3. 如果同时使用，**实际宽度**为限制flex-basis和width两者的最大者，并且会限制，内容超出直接溢出（不使用break-all）；
4. 如果有max-width和min-width，flex-basis受到这两个属性的限制， $\text{flex-basis} = \max\text{-width}$ ， $\text{flex-basis} = \min\text{-width}$

如果flex-basis没有设置， $\text{flex-basis} = \text{width}$ ；如果width也没有设置， $\text{flex-basis} = \text{content的宽度}$

项目属性

5. align-self —— 属性允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性，如果没有父元素，则等同于stretch。



实战演练

