# flex布局复习

#### 兼容性

flex布局在pc端兼容性差。主要是IE。

不过在移动端非常受欢迎。布局非常简单灵活，不用考虑浮动、清浮动的问题。

任一容器都可以指定flex布局。

当父元素设定为flex布局后，成为**flex容器**（flex container）。

子元素（flex item）的float、clear、vertical-align都将失效。

## 1、flex容器属性

#### flex-direction

flex-direction属性决定主轴的方向（即项目的排列方向)，默认为row。

有column|column-reverse|row|row-reverse



#### flex-wrap

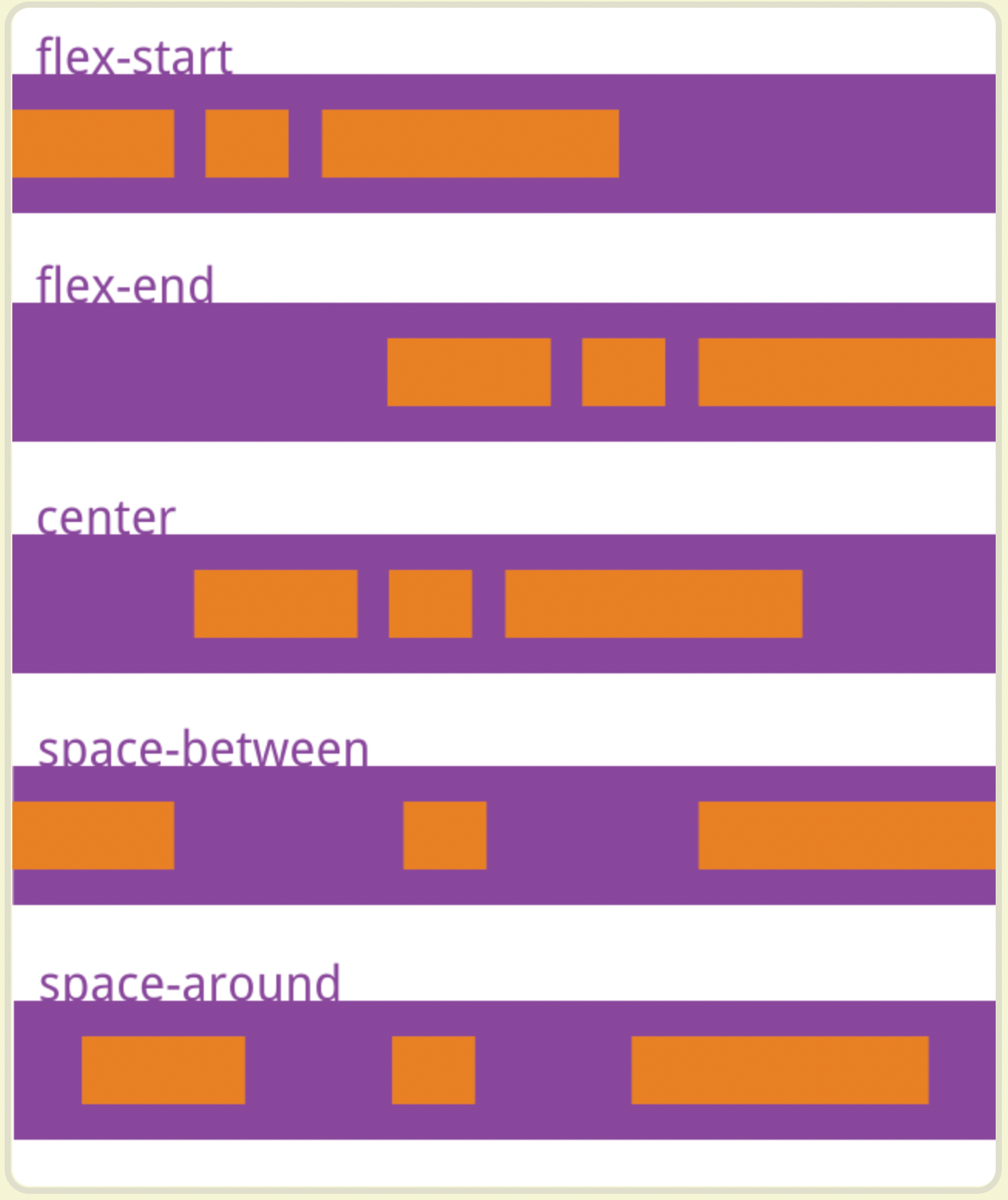
flex-wrap属性定义，如果一条轴线排不下，如何换行。默认nowrap。

#### flex-flow(没什么用，只是简写)

flex-flow属性是flex-direction属性和flex-wrap属性的简写形式，默认值为row nowrap

#### justify-content

justify-content属性定义了项目在主轴上的对齐方式。



space-around是平均分配剩余空间

space-between是先两边贴边对齐，再平均分配剩余空间

#### align-items

align-items属性定义项目在交叉轴上如何对齐。

与justify-content作用于主轴类似，align-items作用于侧轴

* 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: 项目的第一行文字的基线对齐。
* stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。

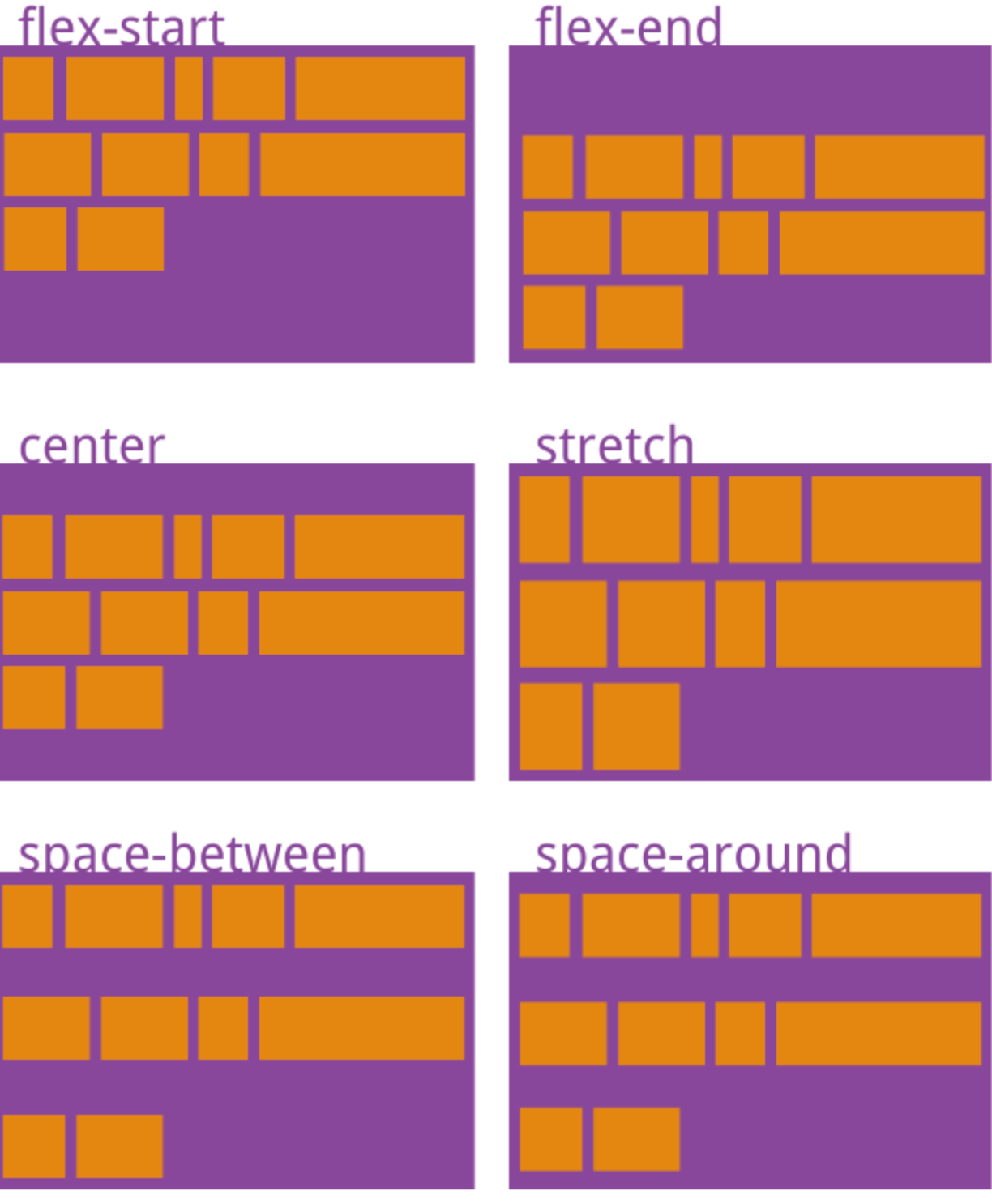
使用flex实现水平垂直居中

<style>  
 .father {  
 height: 500px;  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 background-color: #ccc;  
 }  
  
 .box {  
 width: 100px;  
 height: 100px;  
 background-color: aqua;  
 }  
</style>  
<body>  
 <div class="father">  
 <div class="box"></div>  
 </div>  
</body>

#### align-content

Align-content属性定义了多根轴线的对齐方式。即出现了换行。

如果项目只有一根轴线，该属性不起作用。

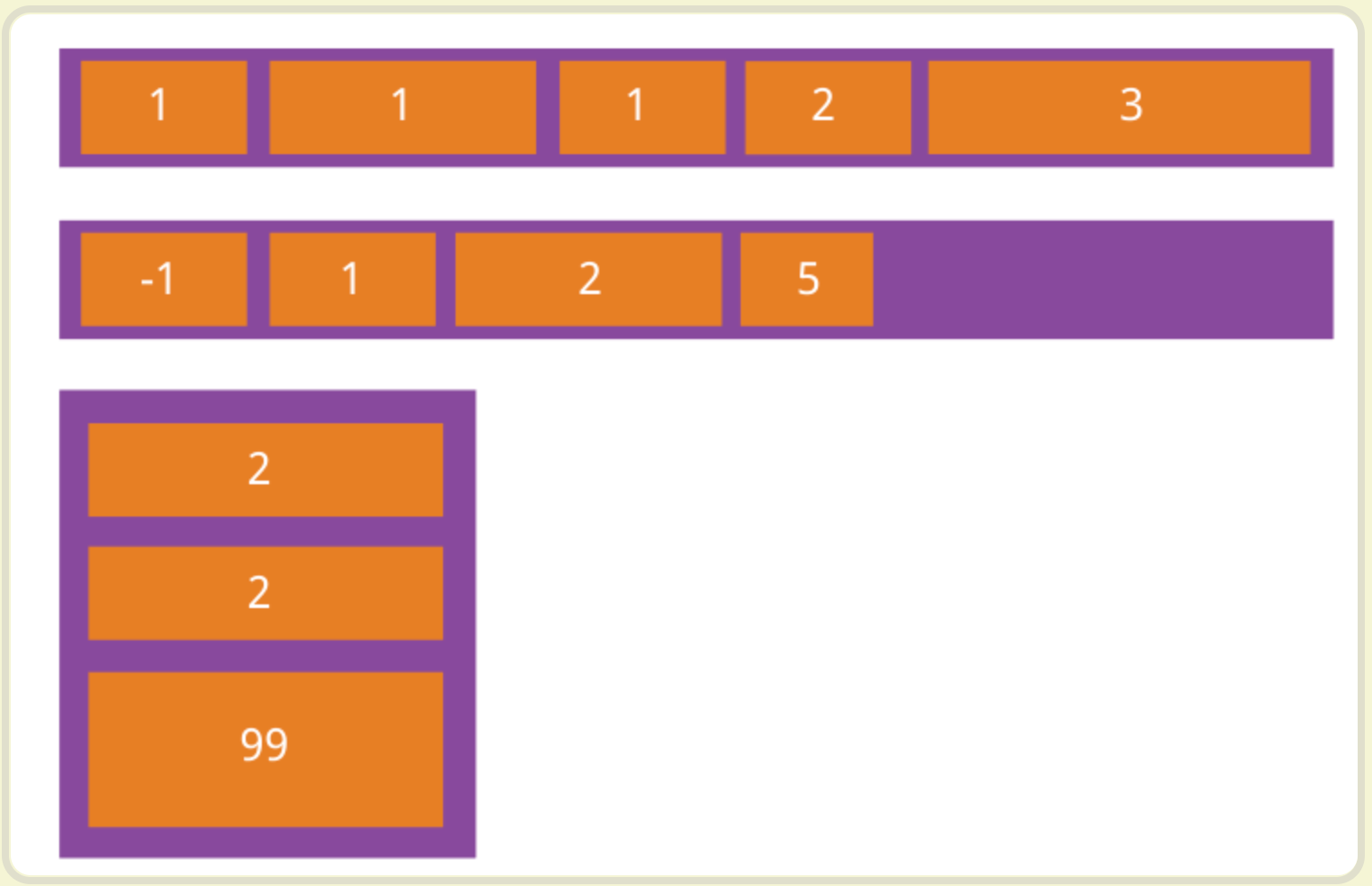


## 下面是子项目的属性

#### order

默认是0，越小越靠前。

.item {  
 order: <integer>;  
}



#### flex-grow

flex-grow属性定义项目的放大比例，用于分配剩余空间，默认为0。

#### flex-shrink

属性定义了项目的缩小比例，在空间不足时的缩小比例。

默认为1。

若为0，则即使空间不足也不缩小。

#### 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）。默认为auto，基本上不用。

#### flex

是flex-grow,flex-shrink,flex-basis的缩写。

建议优先使用这个属性，而不是单独写三个分离的属性，因为浏览器会推算相关值。

#### align-self

属性允许单个项目在交叉轴上与其他项目不一样的对齐方式，覆盖align-items属性。