

flex 布局的应用

介绍

Flex 是 Flexible Box 的缩写，意为"弹性布局"，用来为盒状模型提供最大的灵活性（空间分布和对齐）。设为 Flex 布局以后，子元素的 float、clear 和 vertical-align 属性将失效。Flex 是一种一维布局，一个 flexbox 一次只能处理一个维度上的元素布局，一行或者一列。

flexbox 的轴线

flexbox 有两根轴线：主轴和交叉轴。主轴由 flex-direction 定义，交叉轴垂直于主轴

flex-direction 可以取 4 个值：flex-direction: row | row-reverse | column | column-reverse

- row | row-reverse 主轴沿 inline 方向延伸，交叉轴沿着列向下
- column | column-reverse 主轴沿上下（block）方向，交叉轴水平方向

flex 容器的初始值

所有 flex 元素都有如下属性

- 元素排列为一行
- 元素从主轴的起始线开始
- 元素不会在主维度的方向拉伸，但是可以缩小
- 元素被拉伸来填充交叉轴的大小
- flex-basis 属性为 auto
- flex-wrap 属性为 nowrap

实现换行效果

- flex-wrap 实现多行效果
 - nowrap 不换行
 - wrap 换行
 - wrap-reverse 第一行在下方

属性简写

flex-flow是 flex-direction 和 flex-wrap 的缩写

flex 元素的属性

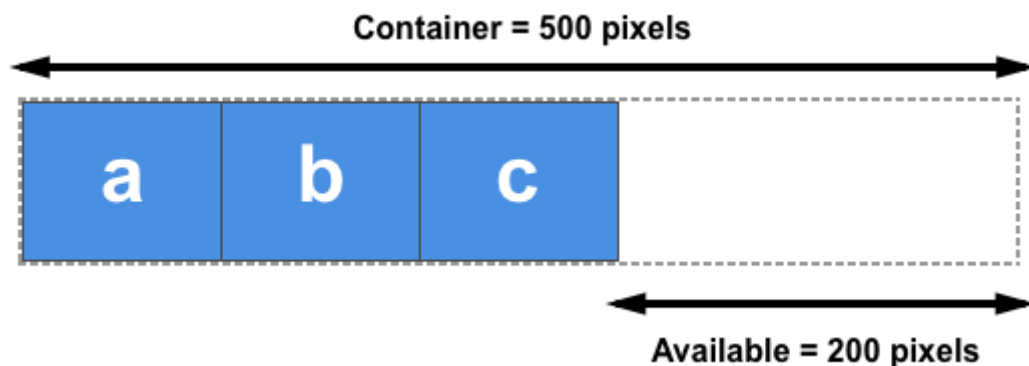
为了更好地控制 flex 元素，有三个属性可以作用于它们：

- **flex-basis** 定义了空间分配发生之前初始化 flex 子元素的尺寸，属性默认值 auto；flex 子元素未伸张和收缩之前，它的大小是多少。

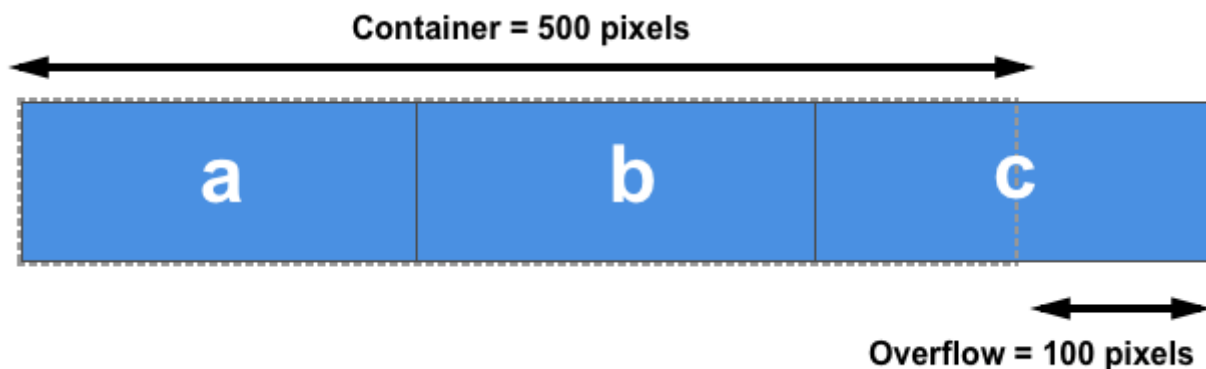
如果 flex-basis 设置为 auto，浏览器会先检查 flex 子元素的主尺寸是否设置了 flex 子元素的初始值。比如说你已经给你的 flex 子元素设置了 150px 的宽，则 150px 就是这个 flex 子元素的 flex-basis；如果没有设置，则 auto 会解析为其内容的大小。如果你想 flexbox 完全忽略 flex 子元素的尺寸就设置 flex-basis 为 0。

在介绍下面两个属性前先看两个概念 positive free space 正向自由空间和 negative free space 反向自由空间：

- 正向自由空间



- 反向自由空间



接下来就是如何分配正负自由空间了

- **flex-grow** 默认值 0，若被赋值为一个正整数，flex 元素会以 flex-basis 为基础，沿主轴方向增长尺寸，并占据可用空间。flex-grow 按比例分配增长空间。

等比例增长

不同比例增长

想让开始时尺寸不同的元素内容大小相等，可以将 `flex-basis` 设置为 0（完全忽略 flex 子元素的尺寸）
`flex-grow` 为 1（等比例分配）

- `flex-shrink` 属性指定了 flex 缩小值, 它确定在分配 negative free space 时, flex 子元素相对于 flex 容器中其余 flex 子元素收缩的程度。默认值 1。用于减少盒子空间使盒子适应容器而不溢出(为了避免 border 干扰去掉边框)

不同的 `flex-shrink` 值

flex 预定的简写值

- `flex: initial` 的扩展为 0 1 auto
- `flex: auto` 的扩展为 1 1 auto
- `flex: none` 的扩展为 0 0 auto
- `flex: <positive-number>` 的扩展为 `<positive-number> 1 0`

flex 子元素没有 positive free space 就不会增长，没有 negative free space 就不会缩小。

元素间的对齐和空间分配

Flexbox 的一个关键特性是能够设置 flex 元素沿主轴方向和交叉轴方向的对齐方式，以及它们之间的空间分配。

- `align-items` 属性可以使元素在交叉轴方向对齐。
 - stretch
 - flex-start
 - flex-end
 - center
- `justify-content` 属性用来使元素在主轴方向上对齐
 - stretch
 - flex-start
 - flex-end
 - center
 - space-around
 - space-between