CSS flex属性深入理解

> 链接：https://www.zhangxinxu.com/wordpress/2019/12/css-flex-deep/

CSS flex属性属性还是很难理解的，但是flex布局要想玩得溜溜溜，这一关必须得过，来来来，一起看看究竟是什么意思，如何更容易理解与记忆。

一、flex属性是一种简写

首先`flex`属性是`flex-grow`，`flex-shrink`和`flex-basis`的缩写。

等下，已经晕了！

`flex-grow`是谁？`flex-shrink`是哪位？`flex-basis`又是何人？

稍安稍安，这3个CSS属性我们后面会详细展开，这里就先三言两句稍微提一下他们的作用。

这flex布局就好像是有钱人家分家产。

故事是这样的，范闲和林婉儿生了5个孩子，分别叫做范张，范鑫，范旭，范帅和范哥。

要是只有一个孩子，那好说，家产100%继承。但是现在5个孩子，要是不提早定好家产分配规则，万一哪天提前翘辫子，那分家产时候扯皮事情就多了，说不定还会兄弟反目，甚至闹出人命。

而这个`flex`属性的作用就是制定了每个人该如何分配到家产的规则。

- `flex-basis`就是分配固定的家产数量。
- `flex-grow`就是家产剩余家产仍有富余的时候该如何分配。
- `flex-shrink`就是家产剩余家产不足的时候该如何分配。

具体分配细节这里先不展开，当务之急，我们先捋一捋`flex`属性的语法。语法是表面的，我们由表及里慢慢深入。

二、捋捋flex属性的语法

语法

```
flex: none | auto | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]
```

上面这种CSS语法被称为格式化语法，大部分CSS属性值都比较简单，格式化语法看不出有什么优势。但是如果CSS属性值比较的复杂，规则较多，例如CSS渐变或者这里的`flex`属性，则格式化语法就比较重要了，可以知道一些平时没有注意到的语法上的细节，也能一看就知道语法规则。

有人会反驳，你这是在扯犊子吧，“一看就知道语法规则”，我看上面各种交错的字符，根本就不知道说的什么意思，还不如直接给我展示几个例子实际呢！

这种体验类似使用Markdown语法写文章，在不了解Markdown语法的时候，还是觉得可视化操作来得实用。但是如果Markdown语法很熟悉，对吧，是不是“真香”就来了？

CSS语法规则也是一样的，觉得看起来吃力不方便是因为不理解规则。

所以，要想CSS学得专业且深入，CSS语法规则一定要很熟悉。

回到这里，我们一点点对语法进行解构，顺便带大家了解下CSS语法中的一些特殊符号的含义。

语法解构

CSS语法中的特殊符号的含义绝大多数就是正则表达式中的含义，例如单管道符`|`，方括号`[]`，问号`?`，个数范围花括号`{}`等。具体说明：

首先是单管道符`|`。表示排他。也就是这个符号前后的属性值都是支持的，且不能同时出现。因此，下面这些语法都是支持的：

```
flex: auto;
flex: none;

flex: [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]
```

接下来是`[ ... ]`这一部分。其中方括号`[]`表示范围。也就是支持的属性值在这个范围内。我们先把方括号`[]`内其他特殊字符去除，可以得到下面的语法：

```
flex: auto;
flex: none;

flex: [ <'flex-grow'> <'flex-shrink'> <'flex-basis'> ]
```

这就是说，flex属性值支持空格分隔的3个值，因此，下面的语法都是支持的。

```
flex: auto;
flex: none;
/* 3个值 */
flex: 1 1 100px;
```

然后我们再看方括号 `[]` 内的其他字符，例如问号 `?`，表示0个或1个。也就是 `flex-shrink` 属性可有可无。因此，flex属性值也可以是2个值。因此，下面的语法都是支持的。

```
flex: auto;
flex: none;
/* 2个值 */
flex: 1 100px;
/* 3个值 */
flex: 1 1 100px;
```

然后我们再看双管道符 `||`，是或者的意思。表示前后可以分开独立合法使用。也就是 `flex: flex-grow flex-shrink?` 和 `flex-basis` 都是合法的。于是我们又多了2种合法的写法：

```
flex: auto;
flex: none;
/* 1个值，flex-grow */
flex: 1;
/* 1个值，flex-basis */
flex: 100px;
/* 2个值，flex-grow和flex-basis */
flex: 1 100px;
/* 2个值，flex-grow和flex-shrink */
flex: 1 1;
/* 3个值 */
flex: 1 1 100px;
```

//zxx: 评论有人反馈，IE和Chrome的默认值不一样，因此，建议实际开发都写全3个值。

文字表述

- **1个值**

  如果flex的属性值只有一个值，则：如果是数值，例如 `flex: 1`，则这个 `1` 表示 `flex-grow`，~~此时 `flex-shrink` 和 `flex-basis` 都使用默认值，分别是 `1` 和 `auto`~~。**更正为：**此时 `flex-shrink` 和 `flex-basis` 的值分别是 `1` 和 `0%`，注意，这里的 `flex-basis` 的值是 `0%`，而不是默认值 `auto`。如果是长度值，例如 `flex: 100px`，则这个 `100px` 显然指 `flex-basis`，因为3个缩写CSS属性中只有 `flex-basis` 的属性值是长度值。~~此时 `flex-grow` 和 `flex-shrink` 都使用默认值，分别是 `0` 和 `1`。~~**更正为：**此时 `flex-grow` 和 `flex-shrink` 都是 `1`，注意，这里的 `flex-grow` 的值是 `1`，而不是默认值 `0`。

- **2个值**

  如果flex的属性值有两个值，则第1个值一定指 `flex-grow`，第2个值根据值的类型不同表示不同的CSS属性，具体规则如下：如果第2个值是数值，例如 `flex: 1 2`，则这个 `2` 表示 `flex-shrink`，~~此时 `flex-basis` 使用默认值 `auto`~~。**更正为：**此时 `flex-basis` 计算值是 `0%`，并非默认值 `auto`。如果第2个值是长度值，例如 `flex: 1 100px`，则这个 `100px` 指 `flex-basis`，此时 `flex-shrink` 使用默认值 `0`。

- **3个值**

  如果 `flex` 的属性值有3个值，则这长度值表示 `flex-basis`，其余2个数值分别表示 `flex-grow` 和 `flex-shrink`。下面两行CSS语句的语法都是合法的，且含义也是一样的：`/* 下面两行CSS语句含义是一样的 */ flex: 1 2 50%;  flex: 50% 1 2;`

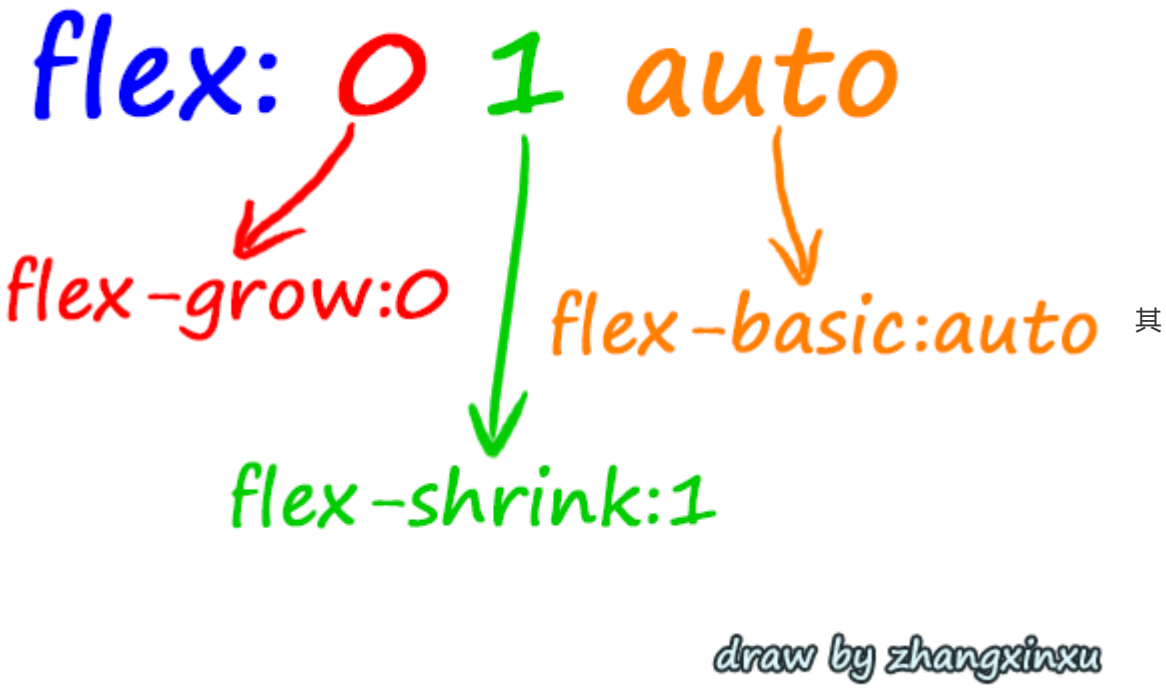
三、flex属性值深入

下面到了最后一步，对flex属性值进行深入的理解与学习。

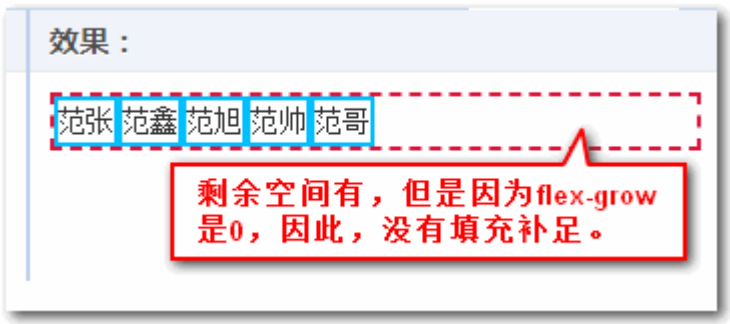
关键字属性值

- **initial**

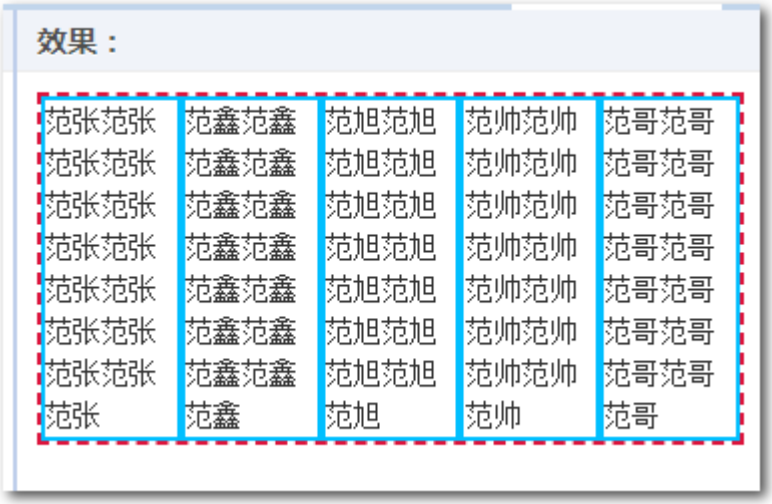
  初始值。`flex:initial`等同于设置`"flex: 0 1 auto"`。可以理解为flex属性的默认值。任意打开一个页面，我们打开控制台执行下面的代码：`window.getComputedStyle(document.body).flex; // 结果就是"flex: 0 1 auto"`也就是`flex`属性默认值是`"0 1 auto"`，等同于`initial`关键字的计算值。该默认值图形示意如下：



其行为表现文字描述为：*不会增长变大占据flex容器中额外的剩余空间（flex-grow:0），会收缩变小以适合容器（flex-shrink:1），尺寸根据自身宽高属性进行调整（flex-basis:auto）*。**案例**于是如果我们只给容器设置`display:flex`，同时子项元素内容都很少，就会有下图所示的效果（用我最爱的深天蓝色高亮下轮廓）：



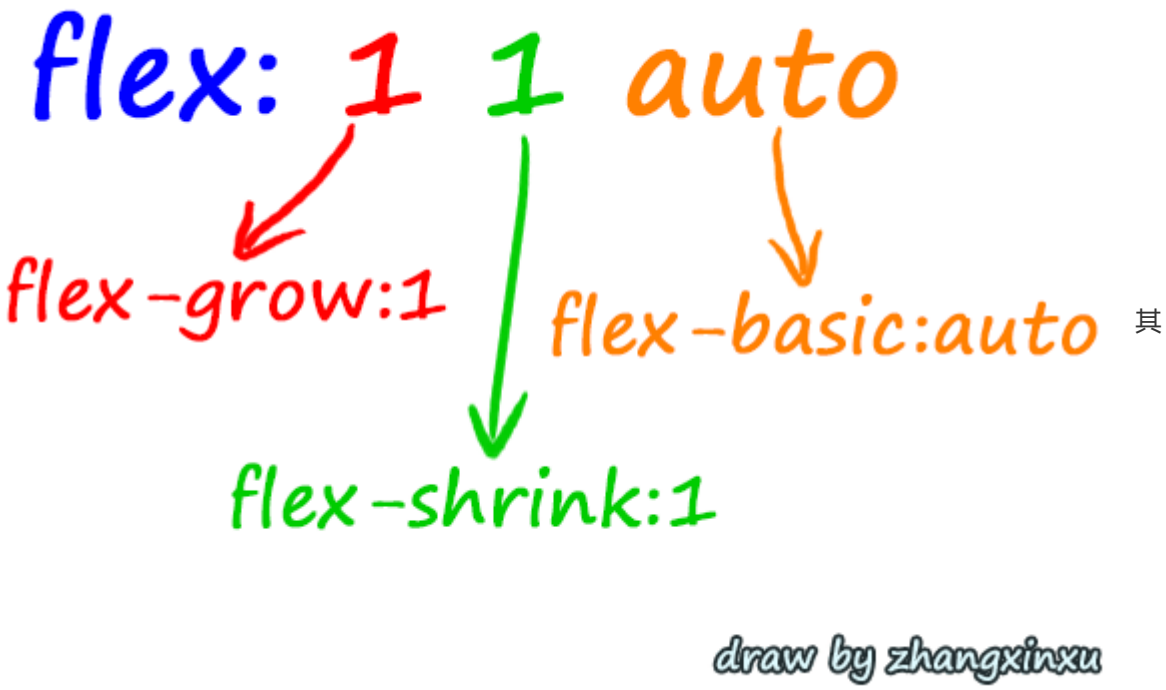
如果子项内容很多，由于`flex-shrink:1`，因此，会缩小，表现效果就是文字换行。如下：



关键CSS代码和HTML代码如下：`.container { display: flex; }``<div class="container"> <item>范张</item> <item>范鑫</item> <item>范旭</item> <item>范帅</item> <item>范哥</item> </div>`您可以狠狠地点击这里：CSS flex:initial布局效果demo

- **auto**

  `flex:auto`等同于设置`"flex: 1 1 auto"`。图示如下：



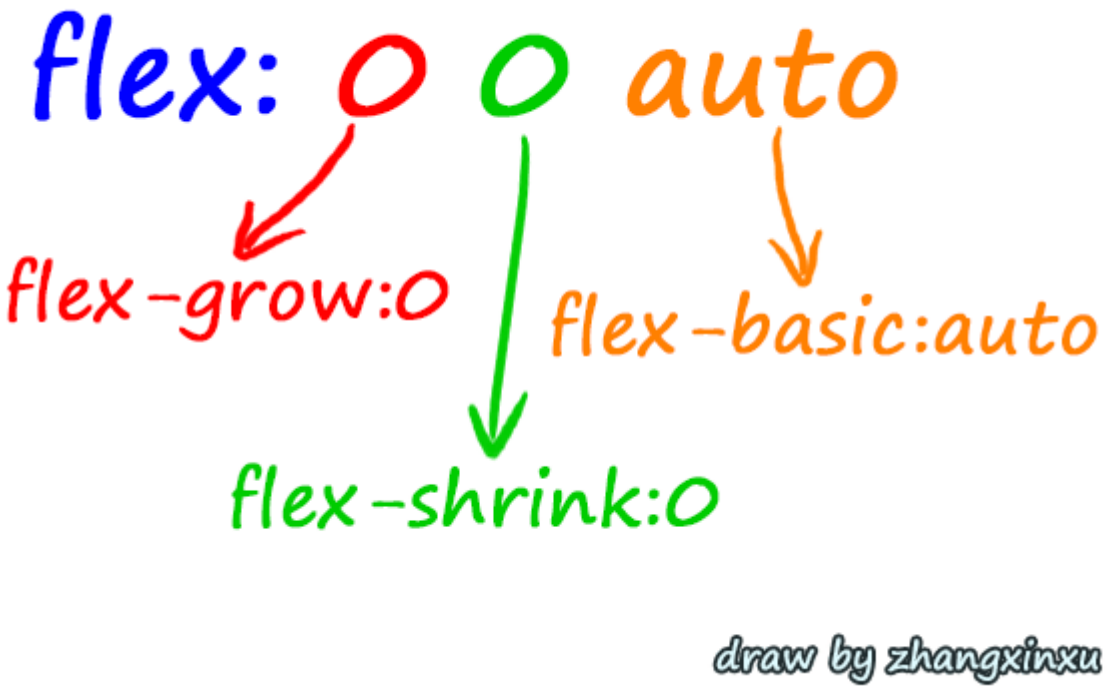
  其行为表现文字描述为：*子项会增长变大占据flex容器中额外的剩余空间（flex-grow:1），会收缩变小以适合容器（flex-shrink:1），尺寸根据自身宽高属性进行调整（flex-basis:auto）。***案例**HTML不变，子项元素的CSS增加`flex:auto`的设置：`.container { display: flex; } .container item { flex: auto; }`结果就是下面这样，子项宽度变大填满了剩余空间，由于每一个子项元素的`flex-grow`的值都是`1`，因此，5等分

填充。 

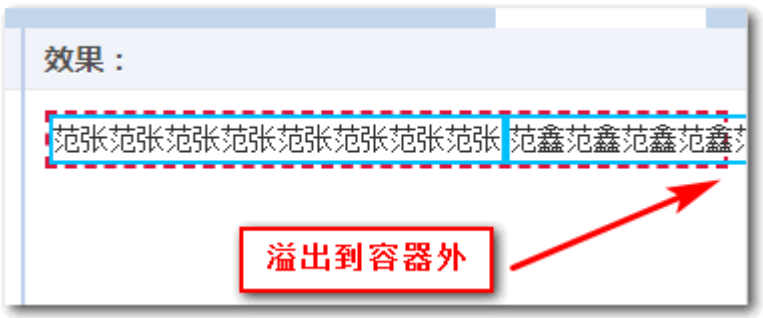
您可以狠狠地点击这里：CSS flex:auto布局效果demo

- **none**

  `flex:none`等同于设置`"flex: 0 0 auto"`。图示如下：



其行为表现文字描述为：*子项会不会增长变大占据flex容器中额外的剩余空间（flex-grow:0），也不会收缩变小以适合容器（flex-shrink:0），尺寸根据自身宽高属性进行调整（flex-basis:auto）*。**案例**HTML不变，每个子项内容很多，同时子项元素的CSS增加`flex:none`的设置：`.container { display: flex; }` `.container item { flex: none; }`结果就是下面这样，子项宽度超过了容器的尺寸，由于`flex-shrink`的值都是`0`，因此，不会收缩变小导致子项的内容撑爆了容器。



您可以狠狠地点击这里：CSS flex:none布局效果demo

flex财产分配三剑客

最后再说说一开始提到的`flex-grow`，`flex-shrink`和`flex-basis`。

一定要牢记这3个属性默认值，不然遇到只有1~2个参数时候，根本不知道什么意思。

- **flex-grow**

  `flex-grow`指定了容器剩余空间多余时候的分配规则，默认值是`0`，多余空间不分配。

- **flex-shrink**

  `flex-shrink`指定了容器剩余空间不足时候的分配规则，默认值是`1`，空间不足要分配。

- **flex-basis**

  `flex-basis`则是指定了固定的分配数量，默认值是`auto`。如会忽略设置的同时设置`width`或者`height`属性。`flex-basis`包含大量的细节知识，这个可以专门开一篇文章深入探讨。

首先解释下，为什么会有容器剩余空间多余和不足的情况出现？

我们还是要分配财产的例子去说明。

范闲的财产分配遗嘱是自己50岁时候制定的。由于范张，范鑫和范旭都已经成家立业，自己在外独立生活，因此，给这三人分配了固定数目的财产，每人100万；而范帅和范哥尚未成年，和范闲还住在一起，所以，遗嘱就是剩下的财产两人评分，按照50岁时候的资产，也是人均100万的样子。

但是世事难料，谁知没过几年，家道中落，范闲总资产已经不足300万，此时，扣除答应范张，范鑫和范旭的300万，已经没有都与家产了，范帅和范哥就要喝西北风了，这就是容器剩余空间不足的情况。

为了应对各种状况出现，因此，财产分配规则制定的时候，一定要明确好基本财产数量`flex-basis`，财产有多余时候的分配规则`flex-grow`，以及财产不足时候的分配规则`flex-shrink`。

案例

请实现：

范张，范鑫和范旭每人100万固定家产，范帅和范哥则20万保底家产。如果范闲归西那天家产还有富余，范帅和范哥按照3:2比例分配；如果没有剩余财产，则范张，范鑫和范旭三位兄长按照2:1:1的比例给两人匀20万保底家产。

HTML结构如下：

```
<div class="container">
    <item clas="zhang">范张</item>
    <item clas="xin">范鑫</item>
    <item clas="xu">范旭</item>
    <item clas="shuai">范帅</item>
    <item clas="ge">范哥</item>
</div>
```

大家可以想想CSS代码该怎么写……

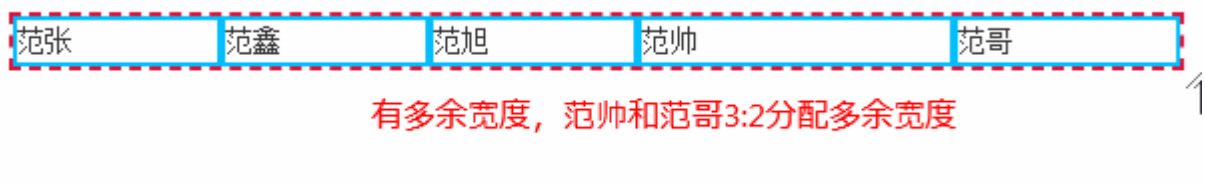
拿出纸和笔，自己写写看……

假设你自己已经写好了，可以对比看一下是不是下面我写的一样：

```
.container {
    /* 范闲：来，家产分配开始了~ */
```

```
    display: flex;
}
.zhang {
    /* 老大不会争夺多余财产，但是会在财产不足时候分出老二老三分出的2倍的财产，这是作为老大应有的姿态 */
    flex: 0 2 100px;
}
.xin,
.xu {
    /* 老二和老三不会争夺多余财产，但是会在财产不足时候分出部分财产，照应老四和老幺
    这里也可以直接写成：flex: 100px*/
    flex: 0 1 100px;
}
.shuai {
    /* 老四会争夺多余财产，且会在财产不足时候享用哥哥们分出的财产，确保能够活下去，感谢三位哥哥的照顾
*/
    flex: 3 0 20px;
}
.ge {
    /* 老五会争夺多余财产，不过比例比哥哥少一点，且会在财产不足时候享用哥哥们分出的财产，感谢哥哥们的照
顾 */
    flex: 2 0 20px;
}
```

最终的效果参见下面的GIF录屏+实时标注说明：



极端情况下，几个哥哥会财产小到几乎饿死，而依然保证两位弟弟有20万的保底财产，多么感人的兄弟情啊！

如果你现在已经理解上面这个兄弟5个分配财产的案例，则说明你对flex的学习和理解已经更上一层楼了，可以顺利毕业了。

眼见为实，您可以狠狠地点击这里：CSS flex属性综合应用布局demo

四、结语碎碎念

有些人在使用`flex`实现一些效果的时候，会配合`min-width`/`max-width`属性，让规律子项的布局更加智能，实际上，是多余的，多此一举，虽然最终实现的效果看上去不错啊，实际上，只需要一个`flex`属性就可以搞定了，这个打算在下一篇文章，flex-basis属性深入的时候介绍。