浮动

【配合 test-7 使用】

关于标准文档流

在 Java 中我们已经初步了解过流的概念，大概就是如水流一般连续的数据；web页面的制作，是个“流”，必须从上而下，像“织毛衣”。

文档流分级

标准文档流等级森严。标签分为 **两种等级**：

- **行内元素** 什么是行内元素呢？简单来说就是不会自动换行，仅占据行内一部分空间的元素，可以和其它行内元素挤在同一层
- **块级元素** 一个元素独占一行

行内元素和块级元素的区别： （非常重要）

行内元素：

- 与其他行内元素并排；

- 不能设置宽、高。默认的宽度，就是文字的宽度。

块级元素：

- 霸占一行，不能与其他任何元素并列；
- 能接受宽、高。如果不设置宽度，那么宽度将默认变为父亲的100%。

行内元素和块级元素的分类：

在以前的HTML知识中，我们已经将标签分过类，当时分为了：文本级、容器级。

从HTML的角度来讲，标签分为：

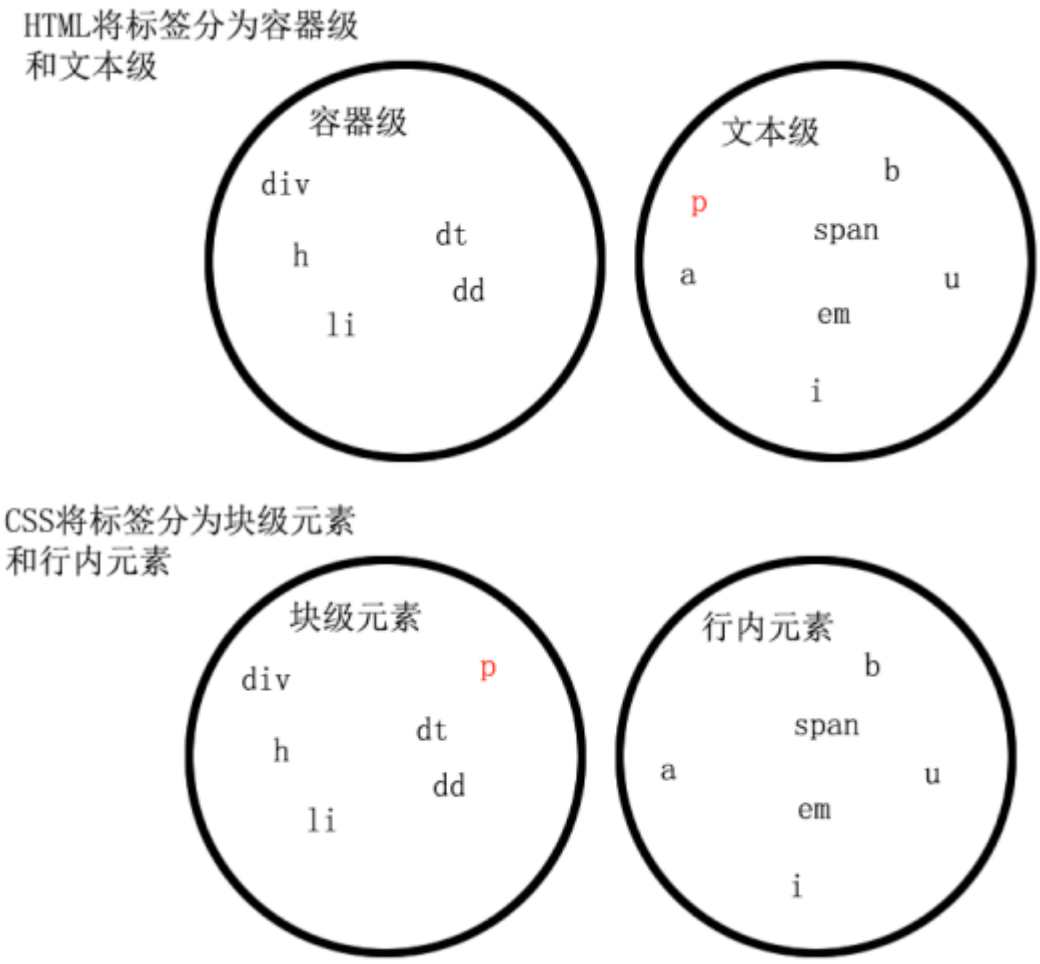
- 文本级标签：p、span、a、b、i、u、em。
- 容器级标签：div、h系列、li、dt、dd。

> PS：为甚么说p是文本级标签呢？因为p里面只能放文字&图片&表单元素，p里面不能放h和ul，p里面也不能放p。

现在，从CSS的角度讲，CSS的分类和上面的很像，就p不一样：

- 行内元素：除了p之外，所有的文本级标签，都是行内元素。p是个文本级，但是是个块级元素。
- 块级元素：所有的容器级标签都是块级元素，还有p标签。

我们把上面的分类画一个图，即可一目了然：



三大特性

1. 空白折叠现象：

   无论多少个空格、换行、tab，都会折叠为一个空格。标签内或者外都是这样

2. 高矮不齐，底边对齐：

   想象成站军姿，反正同一行的都站在同一块地板上

3. 自动换行，一行写不满，换行写

行内元素和块级元素的相互转换

庆贺吧，display 的属性终于可以排上用场了

我们可以通过 `display`属性将块级元素和行内元素进行相互转换。display即“显示模式”。

inline 是行内，**block** 是块级，改变后相应种族特性也会改变

所以说。。。

貌似也没什么大用

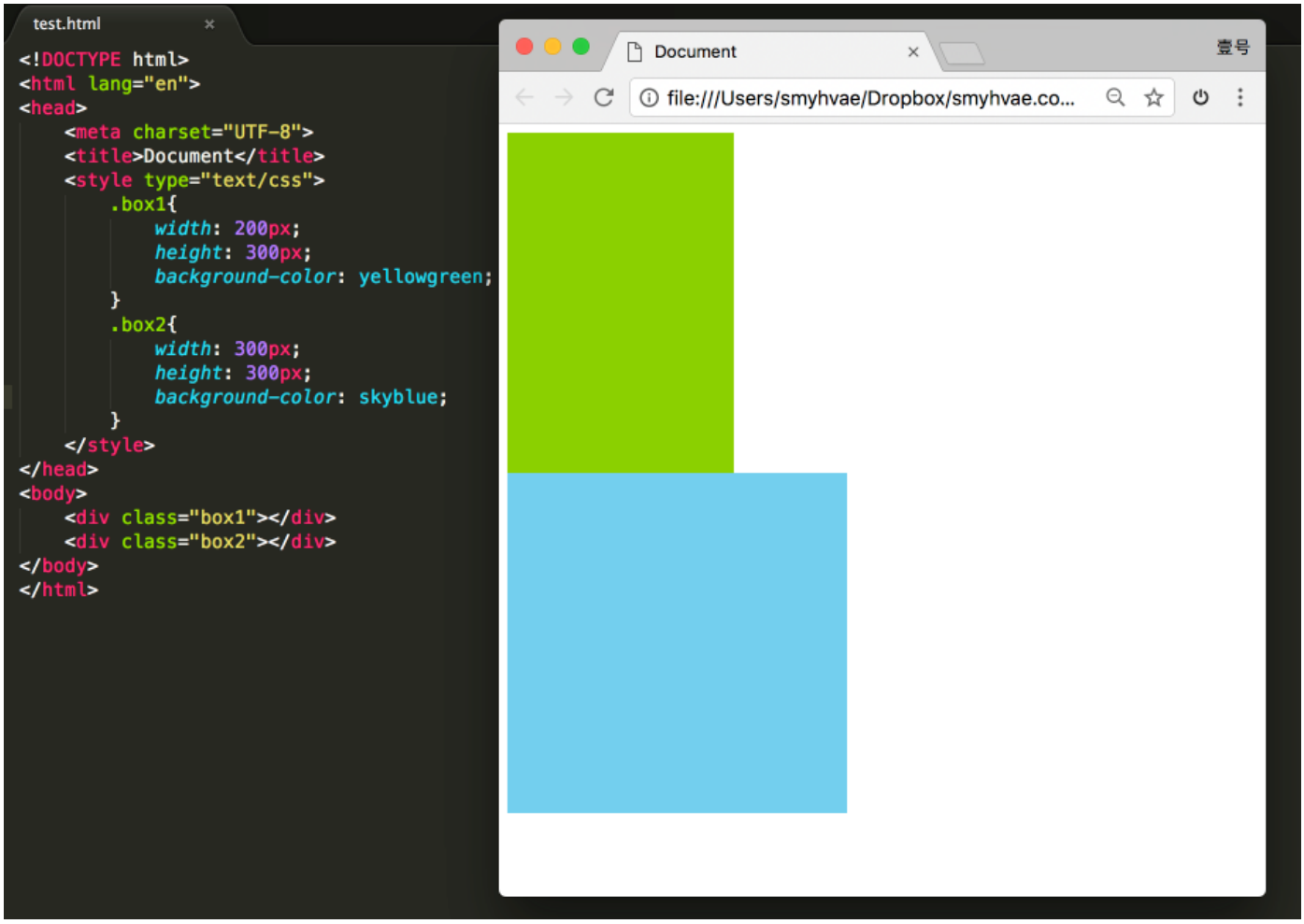
而且限制也不是一般的多，因此我们要想做出更多更自由的效果，还得另寻他法。

浮动的性质

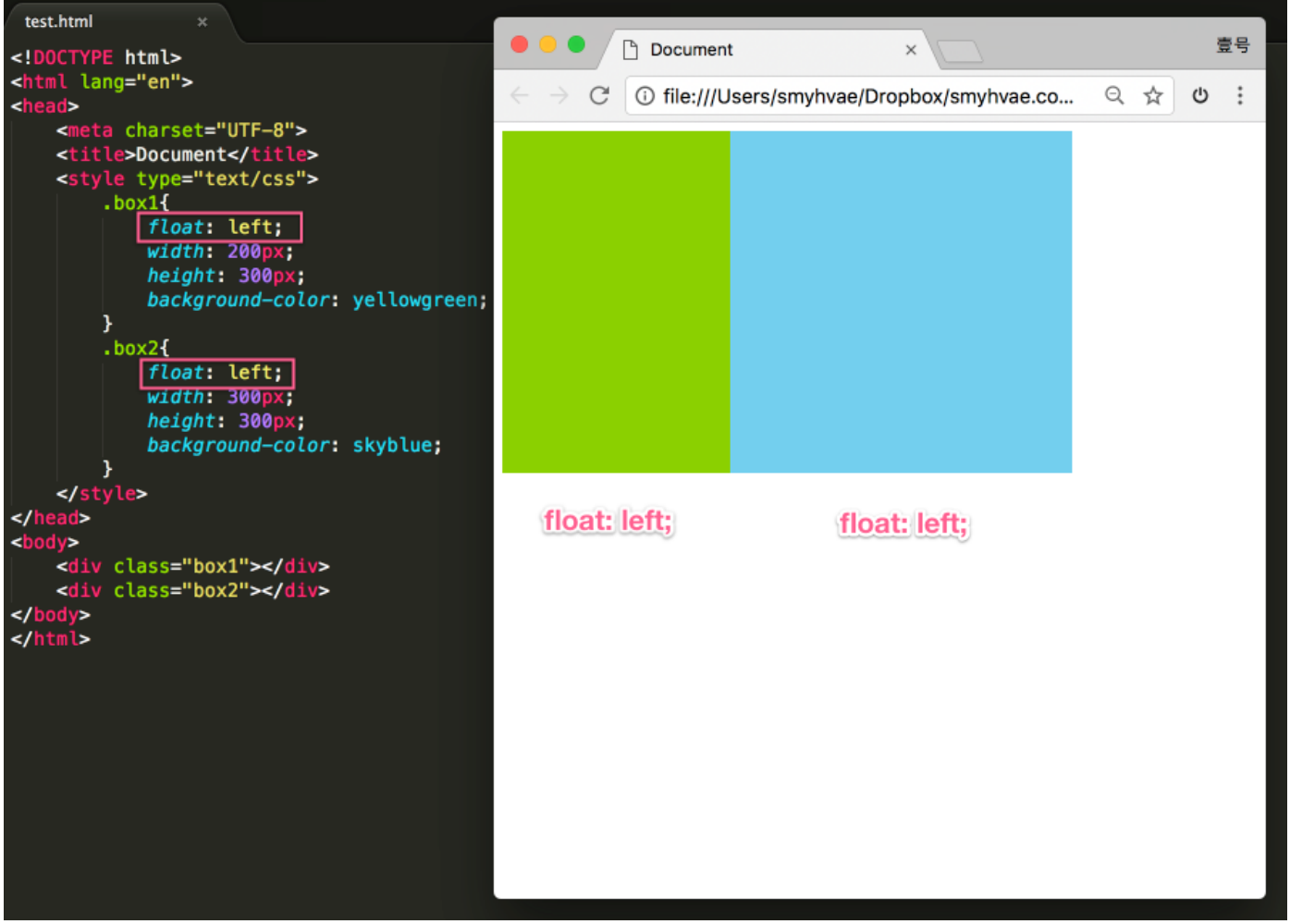
不然今天学浮动干嘛啊，就是解决这个问题的呗。

一般来说浮动在 css 布局中使用非常多

现在有两个div，分别设置宽高。我们知道，它们的效果如下：



此时，如果给这两个div增加一个浮动属性，比如 `float: left;`，效果如下：

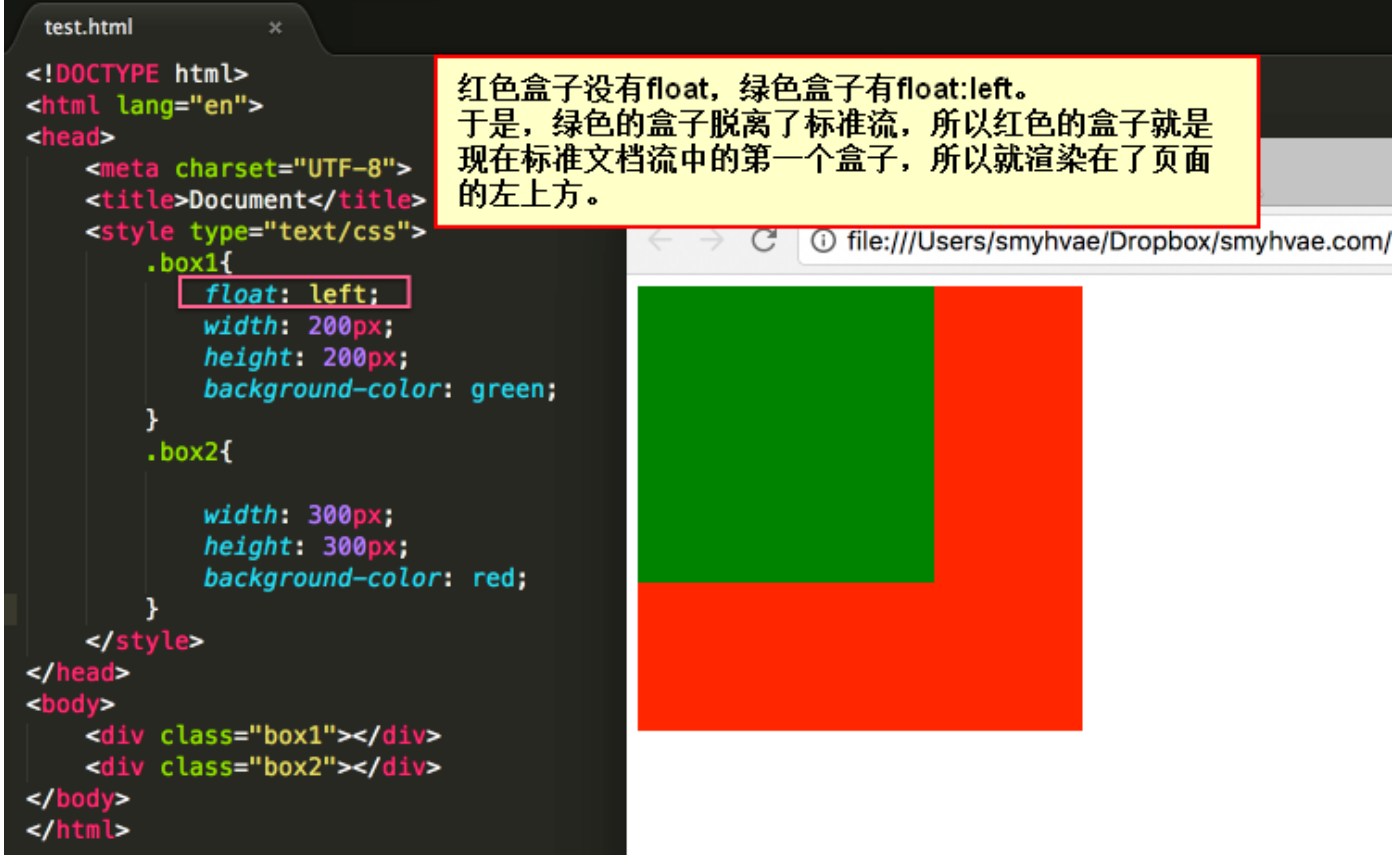


这就达到了浮动的效果。此时，两个元素并排了，并且两个元素都能够设置宽度、高度了（这在上一段的标准流中，不能实现）。

仔细观察一下对应的代码文件，渲染后到底长什么样。试着删掉 float 属性试试看？

浮动四大性质

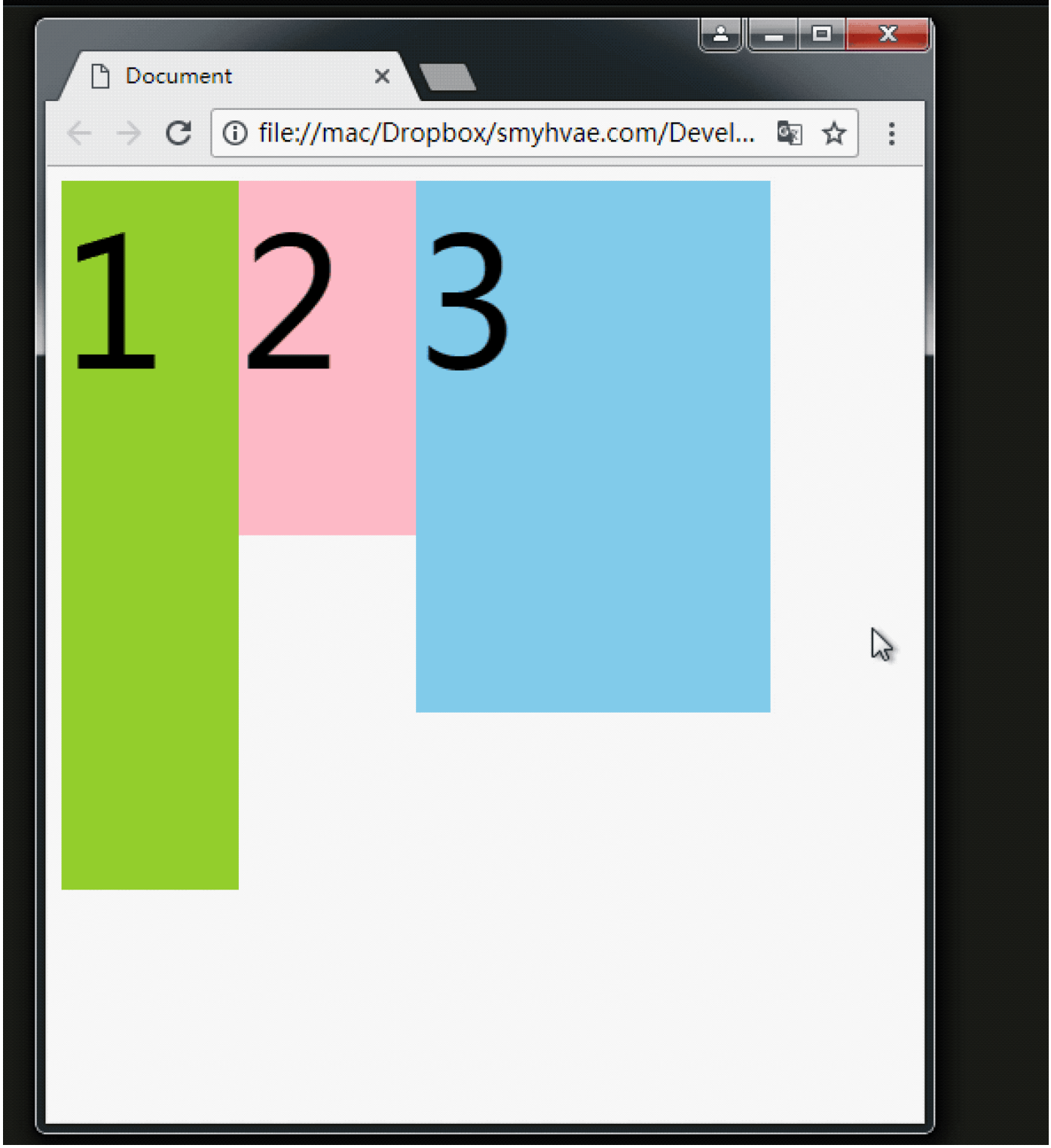
性质**1**：浮动的元素脱标



相信在项目文件中删除 float 也会出现这种状况，可以将 float 后的元素与标准流看作两条平行线了，互不干扰。

同时，元素也变得可以自由安排宽高了，皆大欢喜。

性质2：浮动的元素互相贴靠



要贴靠的方向有 left 和right两种，没有上下一说

性质3：浮动的元素有“字围”效果

自己渲染出来看一下：**div挡住了p，但不会挡住p中的文字** ，形成“文字环绕图片”效果。

性质4：收缩

收缩：一个浮动的元素，如果没有设置width，那么将自动收缩为内容的宽度（这点非常像行内元素）。

上图中，div本身是块级元素，如果不设置width，它会单独霸占整行；但是，设置div浮动后，它会收缩

浮动的清除

这里所说的清除浮动，指的是清除浮动与浮动之间的影响。

具体有什么影响呢？我们知道在一般标准文档流中，容器是可以被自己子类或者内容撑起来的，然而由于我们使用了浮动，结果就是这种性质消失了。

1、加高法

简单粗暴有效，直接给盒子添加高度，但是既麻烦又不方便适应页面变化。

2、`clear:both;`法

clear 指的是盒子两边的浮动清除，`clear:both;` 即两边都不允许有浮动元素。

雀食清除了，就是 margin 也一并被清除了。

3、外墙法 or 内墙法

外墙法就是在元素中间加装一个带有 `clear:both;` 的元素——即墙

然后这玩意也有弊端——就是仍然没有让盒子拥有高度

改用内墙法，将墙放在元素内，可以给它所在的家撑出高度

4、overflow:hidden;

这是个偏方，说不出什么原理

加上就能让其被子元素撑出高度